

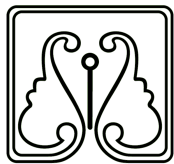


کتابچه مقالات

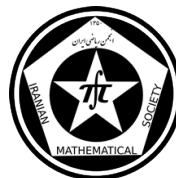
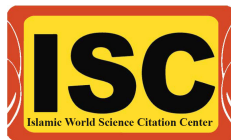
نهمین سمینار آنالیز عددی و کاربردهای آن

۱۹-۲۱ اردیبهشت ۱۴۰۱





University of Guilan



The 9th Seminar on Numerical Analysis and its Applications

9-11 May 2022, University of Guilan, Rasht, Iran

Website: <http://snaa9.guilan.ac.ir/Home/Index>

Address: [Faculty of Mathematical Sciences, University of Guilan](#), Namjoo Blvd, Rasht, Iran

Postal box: 41335-19141

Postal code: 41938-33697

Fax: +981333333509

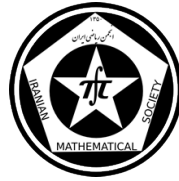
Phone: +981333343630-5

Mail: 9thnumanalseminar@gmail.com

By: Dr. Mohsen Masoudi



دانشگاه گیلان



نهمین سمینار

آنالیز عددی و کاربردهای آن

Numerical Analysis

۱۹-۲۱ اردیبهشت ماه ۱۴۰۱ رشت، ایران

The 9th Seminar on

and its Applications

دانشگاه گیلان

وبسایت: <http://snaa9.guilan.ac.ir/Home/Index>

آدرس: رشت- بلوار نامجو- دانشکده علوم ریاضی

صندوق پستی: ۴۱۳۳۵-۱۹۱۴۱

کد پستی: ۴۱۹۳۸-۳۳۶۹۷

دورنگار: ۰۱۳-۳۳۳۳۳۵۰۹

تلفنخانه: ۰۱۳-۳۳۳۴۳۶۳۰-۵

ایمیل: 9thnumanalseminar@gmail.com

طراحی و تدوین: دکتر محسن مسعودی

کمیته علمی

- دکتر جعفر بی آزار
- دکتر میرکمال میرنیا
- دکتر اسماعیل بابلیان
- دکتر فائزه توتونیان
- دکتر هاشم صابری نجفی
- دکتر جلیل رشیدی نیا
- دکتر نصیر تقی زاده
- دکتر سعید عباس بندی
- دکتر داود خجسته سالکویه
- دکتر مازیار صلاحی
- دکتر محمد کیانپور
- دکتر مصطفی شمسی
- دکتر مهرداد لکستانی
- دکتر امجد علی پناه
- دکتر حسین جعفری
- دکتر عباس سعادت‌مندی
- دکتر داود میرزایی
- دکتر محمود هادی زاده
- دکتر دکتر غلامرضا حجتی
- دکتر بهنام هاشمی
- دکتر فاطمه پنجه علی بیک
- دکتر پیام مختاری
- دکتر سعید کریمی

کمیته اجرایی

- دکتر محمدرضا یاقوتی
- دکتر حسین امینی خواه
- دکتر فرشید مهر دوست
- دکتر اسماعیل عزیزپور
- دکتر منصور هاشمی
- دکتر علی جمالیان
- دکتر صادق اسکندری
- دکتر رضا ابراهیمی آتانی
- دکتر علی مرتضی پور



گروه پشتیبان

- دکتر میخک پلکویی
- نسیم مداح شریعتی
- ارسلان سجادی
- حامد ابراهیمی



سخنرانی‌های کلیدی

- Prof. Michele Benzi

Solving linear systems of the form $(A + \gamma UU^T)x = b$

- Prof. Masoud Hajarani

Recent developments in tensor computations and multilinear algebra



کارگاه های آموزشی

• دکتر علی مس فروش

معرفی و آشنایی با نرم افزار FENICS

• دکتر علی فروش باستانی

معرفی مدل های مبتنی بر معادلات دیفرانسیل کسری تصادفی در ریاضیات مالی

Numerical Analysis

عنوان کارگاه: معرفی و آشنایی با نرم افزار FEniCS

چکیده: برای حل معادلات دیفرانسیل با مشتقات جزئی به روش عناصر متناهی نرم افزارهای گوناگونی وجود دارد. استفاده از برخی از نرم افزارها مانند متلب پیچیدگیها و محدودیتهای خاصی دارد. نرم افزار FEniCS، نرم افزار اپن سورس است که توسط تیم بزرگی شامل ریاضیدانان، مهندسين، دانشجویان و ... برای پلاتفرمهای Windows، Linux و Mac تهیه شده است. این نرم افزار برای حل مسائل مقدار مرزی از شکل وردشی مسأله استفاده می کند و با کدنویسی کوتاهی که به زبان پایتون انجام می شود، جواب مسأله را هم به شکل عددی و هم به صورت نمودار بدست می آورد که می توان آن را با استفاده نرم افزار Paraview تحلیل کرد. در این کارگاه به معرفی مختصر این نرم افزار پرداخته می شود و تلاش بر این است تا شرکت کنندگان در کارگاه با چگونگی نصب، نوشتن برنامه و استفاده از Solver های موجود در این نرم افزار آشنا شوند.

عنوان کارگاه: معرفی مدل های مبتنی بر معادلات دیفرانسیل کسری تصادفی در ریاضیات مالی

چکیده: در این کارگاه، پس از ذکر مقدمه ای کوتاه بر تاریخچه حرکت براونی کسری و معرفی این فرآیند به عنوان تعمیمی از حرکت براونی استاندارد، به مروری بر خواص اساسی این فرآیند از جمله خاصیت خود متشابهی و نمودهای وابسته پرداخته و با تاکید بر «وابستگی دوربرد» به عنوان کلیدی ترین مفهوم کاربردی در دنیای واقعی، به معرفی چند پدیده در ریاضیات مالی و مهندسی مالی از جمله مدل تلاطم ناهموار (Rough Volatility) که به طور طبیعی به این فرآیند منتهی می شوند، خواهیم پرداخت. در ادامه چند مساله عددی مطرح در این زمینه بخصوص در زمینه تولید مسیرهای این فرآیند را مورد مطالعه و بررسی قرار داده و به چند مساله تحقیقاتی نیز اشاره خواهد شد.

English Articles

1	SYMMETRICAL WENO- η SCHEMES FOR HYPERBOLIC CONSERVATION LAWS (Rooholah Abedian)	12
2	SEPTIC B-SPLINE QUASI-INTERPOLATION WENO SCHEMES FOR HYPERBOLIC CONSERVATION LAWS (Rooholah Abedian)	16
3	T_M -EIGENVALUE OF ODD-ORDER TENSORS (Mehri Pakmanesh, Hamireza Afshin)	20
4	PERTURBED EXTENDED SECOND DERIVATIVE BACKWARD DIFFERENTIATION FORMULA (Tahere Majidi, Ali Abdi, Gholamrza Hojjati)	24
5	RESTRICTING THE SOLUTION SET OF THE INTERVAL LINEAR SYSTEMS WITH MULTIPLE RIGHT-HAND SIDES (Marzieh Dehghani-Madiseh)	28
6	PRICING OF EUROPEAN OPTION USING THREE TYPES OF B-SPLINE FUNCTIONS (Farshid Nourian, Mehrdad Lakestani, Sedigheh Sabermahani, Yadollah Ordokhani)	32
7	APPROXIMATE SOLUTION OF THE LOCAL TIME M-FRACTIONAL VOLTERRA INTEGRAL EQUATIONS OF THE SECOND KIND (Mousa Ilie, Ali Khoshkenar)	36
8	A FAST NUMERICAL METHOD FOR FRACTIONAL ORDINARY DIFFERENTIAL EQUATIONS (Roghayeh Katani)	41
9	AN INTERPOLATION-BASED METHOD FOR THE NUMERICAL SOLUTION OF MULTI-POINT BOUNDARY VALUE PROBLEMS (Nasibeh Karamollahi, Mohammad Heydari, Ghasem Barid Loghmani)	45
10	NUMERICAL ANALYSIS APPLICATION IN VIRTUAL TEACHING WITH EJS SOFTWARE (Maedeh GholamAzad, Amir Pishkoo)	49
11	A NEW NUMERICAL ALGORITHM FOR THE FRACTIONAL MODEL OF HUMAN LIVER WITH NEW MODIFIED PARAMETERS (Hamed Ebrahimi, Jafar Biazar)	52
12	AN EFFICIENT METHOD FOR SOLVING THE SYSTEM OF GENERALIZED ABEL INTEGRAL EQUATIONS IN THREE UNKNOWNNS (Sayed Arsalan Sajjadi, Hashem Saberi Najafi, Hossein Aminikhah)	56
13	SOME BOUNDARY VALUE PROBLEMS ON THE HEISENBERG LIE GROUPS (Abdolrahman Razani, Farzaneh Safari)	62
14	NUMERICAL METHOD FOR DISTRIBUTED-ORDER TIME-FRACTIONAL SUB-DIFFUSION EQUATIONS (Tahereh Eftekhari, Jalil Rashidinia)	67

15	RBF-FD METHOD FOR A FRACTIONAL INVERSE PROBLEM (Fatemeh Taghipour, Ahmad Shirzadi)	70
16	EFFICIENT ALGORITHMS TO SOLVE NONLINEAR VOLTERRA INTEGRAL EQUATIONS OF THE FIRST KIND (Roya Montazeri, Jafar Biazar)	74
17	EASY JAVA SIMULATIONS SOFTWARE FOR NUMERICAL MODELING (Amir Pishkoo, Maedeh GholamAzad)	78
18	A REDUCED-ORDER DIFFERENCE POTENTIAL ALGORITHM BASED ON POD METHOD FOR PARABOLIC INTERFACE PROBLEM (Mahboubeh Tavakoli Tameh, Fatemeh Shakeri)	82
19	FEEDBACK SOLUTION FOR HIGH-NONLINEAR HAMILTON-JACOBI-BELLMAN EQUATION BY A PSEUDOSPECTRAL DOMAIN DECOMPOSITION TECHNIQUE (Mehdi Ghasemi, Homa Shirmardi, Mohammad Maleki)	86
20	A NUMERICAL METHOD FOR SOLVING MULTI-VARIABLE ORDER FRACTIONAL INTEGRO- DIFFERENTIAL EQUATIONS USING THE KRALL-LAGUERRE POLYNOMIALS (Hadis Azin, Ali Habibirad)	90
21	SPARSE TWO-GREEDY SUBSPACE KACZMARZ METHOD WITH OBLIQE PROJECTION FOR COMPRESSED SENSING (Farshid Abdollahi, Fatemeh Pirayesh Dehkordi)	94
22	AN <i>hp</i> -VERSION COLLOCATION METHOD FOR WEAKLY SINGULAR INTEGRAL EQUATIONS (Khadijeh Nedaiaasl, Raziye Dehbozorgi)	98
23	A NUMERICAL METHOD BASED ON DAUBECHIES WAVELET TO SOLVE A CLASS OF FDEs (Nasim Madah Shariati)	106
24	MODIFIED OSTROWSKI METHOD FOR SOLVING NONLINEAR EQUATIONS AND ITS DYNAMIC (Vali Torkashvand, Manochehr Kazemi, Elham Hashemizadeh)	111
25	ON THE VARIABLE PARAMETER UZAWA METHOD FOR DOUBLE SADDLE POINT SYSTEMS (Mohammad Mahdi Izadkhah)	117
26	THE ERROR ANALYSIS OF AN EFFICIENT NUMERICAL METHOD FOR DISTRIBUTED SCHRÖDINGER EQUATION (Fatemeh Hashemi, Esmail Hesameddini, Ameneh Taleei)	121
27	A ROBUST SPECTRAL SCHEME FOR NON-LINEAR DYNAMICAL MODEL OF COVID-19 DIS- EASE (Amin Faghih, Payam Mokhtary)	125
28	A NEW PRECONDITIONER FOR THE SOR METHOD TO SOLVE MULTI-LINEAR SYSTEMS (Afsaneh Hasanpour, Maryam Mojarrah)	129
29	ACCURATELY APPROXIMATING FLAT FUNCTIONS BY A NEW TRANSCENDENTAL RBF (Maryam Mohammadi, Mohammad Heidari)	133

30	EXTENDED MINIMAL RESIDUAL BICONJUGATE GRADIENT STABILIZED METHOD FOR GENERALIZED COUPLED SYLVESTER TENSOR EQUATIONS (Mohammad Mahdi Izadkhah)	138
31	AN EFFICIENT LUCAS WAVELET GALERKIN METHOD FOR SOLVING TIME-DELAY OPTIMAL CONTROL PROBLEMS (Sedigheh Sabermahani, Yadollah Ordokhani, Farshid Nourian, Mehrdad Lakestani)	142
32	NUMERICAL SOLUTION OF MIXED FRACTIONAL HESTON PARTIAL DIFFERENTIAL EQUATION (Fereshteh Goldoust)	146
33	STABLE GAUSSIAN ELIMINATION ALGORITHM FOR ILL-CONDITIONED BANDED TOEPLITZ (Nasser Akhoundi)	151
34	PERTURBED SIMPSON-TYPE INEQUALITY VIA h -CONVEX FUNCTIONS (Ali Barani, Naser Abbasi)	155
35	THE IRREDUCIBLE REPRESENTATION OF GENERAL LINEAR METHODS (Rana Akbari, Gholamreza Hojjati, Ali Abdi)	159
36	A MODIFICATION OF GRÜN WALD-LETNIKOV DERIVATIVE IN IMAGE PROCESSING (Hoda Jalalinejad, Ali Tavakoli)	163
37	A SPECTRAL COLLOCATION METHOD FOR SOLVING THE NONLINEAR WEAKLY SINGULAR FIPDE (Mehran Taghipour, Hossein Aminikhah)	167
38	NUMERICAL SOLUTION OF NON-LINEAR REACTION-DIFFUSION EQUATIONS USING MESHLESS RADIAL POINT HERMITE INTERPOLATION METHOD (Masoud Pendar, Kamal Shanazari)	172
39	A NOVEL SHIFTED JACOBI OPERATIONAL MATRIX METHOD: AN APPLICATION FOR SOLVING NONLINEAR MULTI-TERM FRACTIONAL VARIABLE-ORDER DIFFERENTIAL EQUATIONS (Hamidreza Khodabandehlo, Elyas Shivanian)	176
40	A DIFFERENCE SCHEME FOR FOURTH-ORDER FRACTIONAL PARTIAL INTEGRO-DIFFERENTIAL EQUATION (Mehran Taghipour, Hossein Aminikhah)	180
41	A MESHLESS METHOD FOR SOLVING THE FRACTIONAL SCHNAKENBERG MODEL USED TO BIOLOGICAL PATTERN FORMATION (Marzieh Raei)	184
42	THE DIRECT MESHLESS LOCAL PETROVE GALERKIN METHOD FOR SOLVING EQUATIONS WITH DISTRIBUTED ORDER DERIVATIVES (Ali Habibirad, Hadis Azin, Esmail Hesameddini)	188
43	EXPONENTIAL BASIS FUNCTIONS WITH THREE SHAPE PARAMETERS (Jamshid Saeidian, Bahareh Nouri)	193
44	ANALYSIS OF COVID-19 PANDEMIC IN IRAN AND GUILAN BASED ON SIRU MODEL (Hossein Taheri, Nasrin Eghbali, Masoumeh Pourabd)	197

45	COMPUTATIONAL ASPECTS OF STOCHASTIC PARTIAL DIFFERENTIAL EQUATIONS USING FINITE ELEMENT METHOD (Mahdiah Arezoomandan, Ali Reza Soheili)	201
----	---	-----



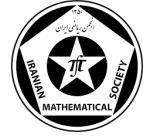
- ۱ بررسی حالت تطبیقی روش های استیفنس گون با حافظه برای حل معادلات غیرخطی با بیشترین شاخص کارایی ممکن
(محمدجواد لاله چینی) ۲۰۵
- ۲ یک روش تکراری برای حل مسئله بهینه سازی با محدودیت PDE
(حمید میرچی، داود خجسته سالکویه) ۲۱۰
- ۳ روش Topsis در انتخاب و رت بندی سرویس های ابری
(امیرحسین شاه بخش رضوی، مهدی زعفرانی) ۲۱۵
- ۴ انتخاب ویژگی به کمک روش ترکیبی تحلیل مولفه اصلی و بهره ای اطلاعات
(نگین علی خانی) ۲۲۰
- ۵ یک روش عددی برای حل معادله اول پینلوف با استفاده از پایه های موجکی شودر
(مجید عرفانیان، حامد زیدآبادی) ۲۲۴
- ۶ حل عددی معادله فیشر با استفاده از شبکه عصبی عمیق لژاندر
(مریم بابائی، علیرضا افضل آقائی نائینی، کورش پرند) ۲۲۸
- ۷ کاربرد توابع متعامد لژاندر در تحلیل مولفه های اصلی جهت افزایش دقت روش های یادگیری ماشین
(زهرا بهروزه، علیرضا افضل آقائی نائینی، کورش پرند، مسعود حجاریان) ۲۳۲
- ۸ ساخت ماتریس متقارن غیر بدیهی برای طیف معین
(سمیه زنگوئی زاده، مسعود سبک خیز) ۲۳۷
- ۹ تجزیه و تحلیل روش هم محلی در فضای چند جمله ایهای تکه ای پیوسته برای معادلات انتگرال ولترای غیر استاندارد
باهسته منفرد ضعیف
(افسانه حمزه لو، سعید پیش بین) ۲۴۱
- ۱۰ روش خطی سازی متوالی برای حل مسئله انتقال گرما روی پرک های نیم کره ای
(صفیه باقری) ۲۴۵
- ۱۱ حل عددی معادله جریان جفری-هامل با استفاده از روش تفاضل متناهی تکراری
(زهرا سادات یزدانی، محمد حیدری، اعظم سادات هاشمی) ۲۴۹
- ۱۲ ارائه رویکردی مبتنی بر ماشین بردار پشتیبان ترکیبی برای کشف کلاهبرداری در سرویس های پرداخت مالی
(مجید ابتیاع، معراج ابتیاع) ۲۵۳
- ۱۳ مروری بر حل مسأله تخمین پارامتر معادله دیفرانسیل جزئی واکنش-انتشار
(یاسمن رزاقی، خدیجه ندایی اصل، سلمان خدایی فر) ۲۵۷
- ۱۴ حل عددی دستگاه معادلات دیفرانسیل حرکت پرتابگر با رویکرد فازی
(سعید سرآبادان، محمد مقدس، علی آخوندزاده) ۲۶۱
- ۱۵ روش هم مکانی فیوناتچی برای تقریب جواب دستگاه معادلات دیفرانسیل جزئی کسری دو بعدی
(شاهد مشهودی، اسمعیل بابلان) ۲۶۷
- ۱۶ یک روش مبتنی بر هسته های بازگشتی در حل نامساوی های تغییراتی پدید آمده از اختیارات های معامله آمریکایی
(مجتبی مرادی پور، ناصر عباسی) ۲۷۲
- ۱۷ روش ماتریس عملگر ژاکوبی انتقال یافته نوین برای معادلات دیفرانسیل تأخیری با مشتق از مرتبه کسری
(حمیدرضا خداپنده لو، الیاس شیوانیان) ۲۷۶

ابزارهای نوظهور در بازارهای مالی (مدل بندی و شبیه سازی اوراق قرضه وقایع فاجعه آمیز)	۱۸
(ساغر حیدری، فرزانه میر)	۲۸۲





University of Guilan



Symmetrical WENO- η schemes for hyperbolic conservation laws

Roohollah Abedian

Abstract. The aim of this work is to prepare a symmetrical WENO- η (SWENO- η) scheme in the framework of the finite volume for hyperbolic conservation laws (HCL). The SWENO- η scheme is a convex combination of a fifth degree polynomial with two third degree polynomials. This fifth-order SWENO- η scheme uses the same stencil as the traditional WENO proposed by Jiang and Shu (WENO-JS), could get less absolute truncation errors, and obtain the same accuracy order in a smooth region. The SWENO- η scheme has advantages over the WENO-JS scheme in its simplicity and easy extension to higher dimensions.

1. Introduction

Consider the 1D hyperbolic conservation laws

$$u_t + f(u)_x = 0. \quad (1.1)$$

An initial condition $u_0(x) = u(x, 0)$ is given along with appropriate boundary conditions. Despite the smoothness of $u_0(x)$, the solution to (1.1) may develop a discontinuity within a finite time. High order numerical approximations of the developed discontinuity suffer from the Gibbs phenomenon yielding spurious oscillations near the discontinuity. Nowadays, the classical WENO by Jiang and Shu [2] is the basis of other methods [1, 3], since it is one of the most powerful numerical methods that can successfully deal with the Gibbs oscillations. In this paper, a simple type of the symmetrical ENO/WENO methods is presented. ENO schemes for solving Eq. (1.1) are derived in a semi-discrete form. Thus, first a uniform spatial grid where the cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ has width Δx is considered. Assuming $\bar{u}_j \equiv \frac{1}{\Delta x} \int_{I_j} u(x, t) dx$, the semi-discretization formula is derived as

$$\frac{d\bar{u}_j(t)}{dt} = -\frac{1}{\Delta x} (f(u(x_{j+\frac{1}{2}}, t)) - f(u(x_{j-\frac{1}{2}}, t))). \quad (1.2)$$

For computing flux $f(u)$ at point $x_{j+\frac{1}{2}}$, we require a reconstruction polynomial, which is described in the next section. The numerical flux $\hat{f}_{j+\frac{1}{2}} = f(u(x_{j+\frac{1}{2}}, t))$ is computed by $\hat{f}_{j+\frac{1}{2}} = h(u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+)$ such that, h is the Lax-Friedrichs monotone flux. Details on how to form SWENO- η are provided in section 2. The numerical results of the new scheme are presented in section 3.

Keywords: Symmetrical WENO- η scheme, finite volume framework, conservation laws.

AMS Mathematical Subject Classification [2010]: 65M08, 35L65.

2. Symmetrical WENO- η schemes

Now, we explain in brief how symmetrical WENO- η method is derived to solve Eq. (1.1).

step 1. By considering the big stencil $S = \{I_{j-2}, \dots, I_{j+2}\}$, the fifth degree reconstruction polynomial p_1 can be easily obtained by

$$p_1(x) = \sum_{i=1}^5 U[x_{j-\frac{5}{2}}, \dots, x_{j-\frac{5}{2}+i}] \sum_{m=0}^{i-1} \prod_{l=0, l \neq m}^{i-1} (x - x_{j-\frac{5}{2}+l}) \\ + U[x_{j-\frac{5}{2}}, \dots, x_{j+\frac{5}{2}}, x_j] \sum_{m=0}^5 \prod_{l=0, l \neq m}^5 (x - x_{j-\frac{5}{2}+l}), \quad (2.1)$$

where $U[\cdot]$ is a divided difference of the function $U(x) = \int_{-\infty}^x u(\xi, t) d\xi$. Choose another two smaller stencils: $S_2 = \{I_{j-2}, I_{j-1}, I_j\}$ and $S_0 = \{I_j, I_{j+1}, I_{j+2}\}$. It is easy to get the two third degree polynomials as

$$p_r(x) = \sum_{i=1}^3 U[x_{j-r-\frac{1}{2}}, \dots, x_{j-r+\frac{1}{2}+i}] \sum_{m=0}^{i-1} \prod_{l=0, l \neq m}^{i-1} (x - x_{j-r+\frac{1}{2}+l}) \\ + U[x_{j-r-\frac{1}{2}}, \dots, x_{j-r+\frac{5}{2}}, x_j] \sum_{m=0}^3 \prod_{l=0, l \neq m}^3 (x - x_{j-r+\frac{1}{2}+l}), \quad r = 0, 2. \quad (2.2)$$

The divided difference $U[x_{j-r+\frac{5}{2}}, x_j]$ of Eqs. (2.1) and (2.2) is calculated by

$$U[x_{j-r+\frac{5}{2}}, x_j] = \frac{1}{\mathcal{A}} \left(\int_{-\infty}^{x_j} u(\xi, t) d\xi - \int_{-\infty}^{x_{j-r+\frac{5}{2}}} u(\xi, t) d\xi \right) = \frac{1}{\mathcal{A}} \int_{x_{j-r+\frac{5}{2}}}^{x_j} \sum_j L_j(x) \chi_j(x) dx, \quad (2.3)$$

where $\mathcal{A} = x_j - x_{j-r+\frac{5}{2}}$ and $\chi_j(x)$ is the characteristic function of the cell I_j . To complete Eq. (2.3), a polynomial is needed that retains the information in the cell I_j , therefore, similar to the Nessyahu and Tadmor (NT) scheme, the polynomial $L_j(x) = \bar{u}_j + (x - x_j) \frac{1}{\Delta x} u'_j$ is applied, where the numerical derivative u'_j is obtained by the uniformly non-oscillatory (UNO) limiter. By placing $L_j(x)$ in Eq. (2.3), the following equation is explicitly obtained

$$U[x_{j-r+\frac{5}{2}}, x_j] = \frac{1}{5-2r} (\bar{u}_j - (r^2 - r - 2)\bar{u}_{j+1} + (r^2 - 3r + 2)\bar{u}_{j+2} + \frac{1}{4}u'_j), \quad r = 0, 2. \quad (2.4)$$

step 2. Compute the *smoothness indicators* β_r , which measure how smooth the polynomials $p_r(x)$ are in the target cell I_j . The polynomials are smoother in I_j , if these smoothness indicators be smaller. The smoothness indicator associated with each stencil is computed as follows $\beta_r = \xi |L_{1,r}u| + |L_{2,r}u|$, $r = 0, 1, 2$, where the operators $L_{n,r}u$ are the generalized undivided differences. The number $\xi = 0.1$ is a parameter which is to balance the tradeoff between the accuracy around the smooth regions and the discontinuous regions.

step 3. Calculate the non-linear weights based on the linear weights d_r and the smoothness indicators. The main idea is here to construct a global smooth measurement, we define a variable η as $\eta = |L_{2,0}u + L_{2,2}u - 2L_{2,1}u|^2$. We define the new non-linear weights as

$$w_r = \frac{\alpha_r}{\sum_s \alpha_s}, \quad \text{such that } \alpha_r = d_r \left(1 + \frac{\eta}{\beta_r + \Delta x^2} \right), \quad s, r \in \{0, 1, 2\}. \quad (2.5)$$

step 4. The final approximations at the boundaries of each cell are given by

$$u_{j \pm \frac{1}{2}}^\mp = w_0 p_0(x_{j \pm \frac{1}{2}}) + w_1 \left(\frac{1}{d_1} p_1(x_{j \pm \frac{1}{2}}) - \frac{d_0}{d_1^{\frac{1}{2}}} p_0(x_{j \pm \frac{1}{2}}) - \frac{d_2}{d_1} p_2(x_{j \pm \frac{1}{2}}) \right) + w_2 p_2(x_{j \pm \frac{1}{2}}). \quad (2.6)$$

For linear weights any convex combination can be considered. Accordingly, we make the choice: $d_0 = d_2 = \frac{1}{8}, d_1 = \frac{6}{8}$.

step 5. The semi-discrete scheme (1.2) is discretized in time by the Runge-Kutta method, such as a fourth-order one [2].

A sufficient condition for the weights w_r to have the fifth-order accuracy is given as $w_r - d_r = \mathcal{O}(\Delta x^4)$ [2]. The next proposition verifies that the non-linear weights (2.5) fulfill the sufficient condition even near the presence of the critical points.

Proposition 2.1. *The non-linear weights w_r satisfy the relation $|w_r - d_r| \leq \Delta x^4$, even at critical points, where d_r with $r = 0, 1, 2$ are the linear weights with $d_0 + d_1 + d_2 = 1$.*

Proof. The Taylor's expansion of β_r with $r = 0, 1, 2$ and η are

$$\begin{aligned}\beta_2 &= \xi |\Delta x u_{j+\frac{1}{2}}^{(1)} - \frac{23}{24} \Delta x^3 u_{j+\frac{1}{2}}^{(3)}| + |\Delta x^2 u_{j+\frac{1}{2}}^{(2)} - \frac{3}{2} \Delta x^3 u_{j+\frac{1}{2}}^{(3)}| + \mathcal{O}(\Delta x^4), \\ \beta_1 &= \xi |\Delta x u_{j+\frac{1}{2}}^{(1)} + \frac{1}{24} \Delta x^3 u_{j+\frac{1}{2}}^{(3)}| + |\Delta x^2 u_{j+\frac{1}{2}}^{(2)} + \frac{1}{24} \Delta x^4 u_{j+\frac{1}{2}}^{(4)}| + \mathcal{O}(\Delta x^5), \\ \beta_0 &= \xi |\Delta x u_{j+\frac{1}{2}}^{(1)} + \frac{1}{24} \Delta x^3 u_{j+\frac{1}{2}}^{(3)}| + |\Delta x^2 u_{j+\frac{1}{2}}^{(2)} + \frac{1}{2} \Delta x^3 u_{j+\frac{1}{2}}^{(3)}| + \mathcal{O}(\Delta x^4), \\ \eta &= |-\Delta x^3 u_{j+\frac{1}{2}}^{(3)} + \mathcal{O}(\Delta x^4)|^2 = \Delta x^6 (A + \mathcal{O}(\Delta x^2)).\end{aligned}\tag{2.7}$$

If $u'_{j+\frac{1}{2}} = 0$ and $u''_{j+\frac{1}{2}} \neq 0$, therefore $\beta_r + \Delta x^2 = C \Delta x^2 (1 + \mathcal{O}(\Delta x))$, for some constant $C > 0$. Therefore, we obtain $\alpha_r = d_r (1 + \frac{\eta}{\beta_r + \Delta x^2}) = d_r (1 + \mathcal{O}(\Delta x^4))$. The given condition $d_0 + d_1 + d_2 = 1$ and with a straightforward algebraic operation, it can finally be concluded that $w_r = \frac{\alpha_r}{\sum_s \alpha_s} = d_r + \mathcal{O}(\Delta x^4)$, which leads to the conclusion. \square

3. Numerical results

The numerical results generated by SWENO- η are compared with the classical WENO-JS [2]. Accordingly, we solve $u_t + (\frac{u^2}{2})_x = 0$, known as the inviscid Burger's equation, with the initial condition $u(x, 0) = 1 + 0.5 \sin(\pi x)$ and periodic boundary condition. When $t = 0.12$ the solution is still smooth, and the errors and numerical orders of accuracy by SWENO- η and WENO-JS are shown in Fig. 1. We can see that both schemes achieve their designed order of accuracy, and SWENO- η produces less truncation errors. Fig. 1 shows that SWENO- η scheme needs less CPU time than WENO-JS. Now, we consider the initial condition $u(x, 0) = 1$ for $|x| < 1/3$ and $u(x, 0) = 0$ elsewhere. We end this problem with the solution on 80 cells in Fig. 1. SWENO- η is sharper than WENO-JS on the expansion wave and the shock.

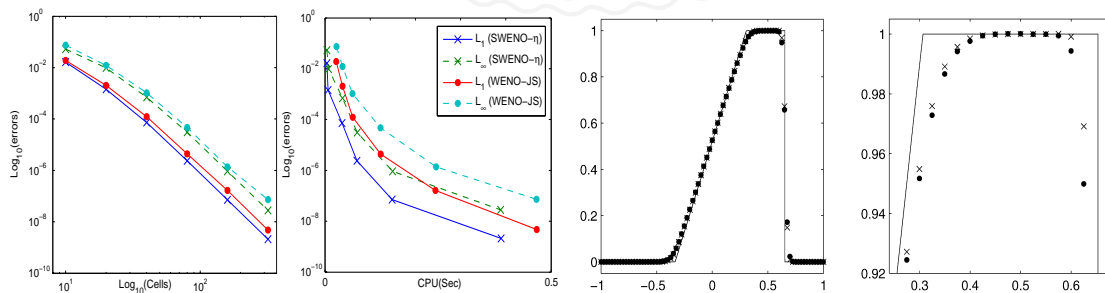


Figure 1: Left: Computing time and error. Right: solution at $t = 3.5/\pi^2$.

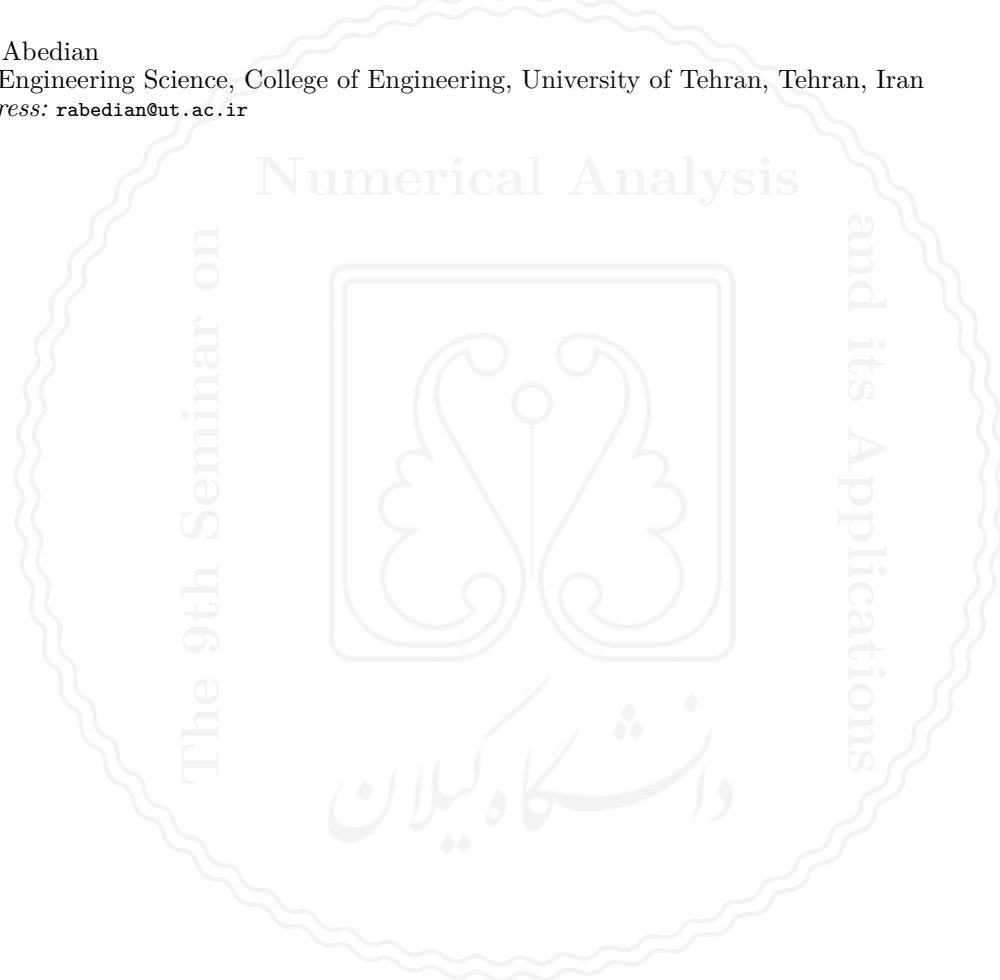
References

- [1] R. Abedian, A finite difference Hermite RBF-WENO scheme for hyperbolic conservation laws, *International Journal for Numerical Methods in Fluids*, (2022) 1–25, doi: 10.1002/fld.5067.
- [2] C.-W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, *SIAM Review*, 51 (2009) 82–126.
- [3] J. Zhu, J. Qiu, A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws, *Journal of Computational Physics*, 318 (2016) 110–121.

Rooholah Abedian

School of Engineering Science, College of Engineering, University of Tehran, Tehran, Iran

email address: rabedian@ut.ac.ir





University of Guilan



Septic B-spline quasi-interpolation WENO schemes for hyperbolic conservation laws

Roohollah Abedian

Abstract. In this work, a septic B-spline quasi-interpolation (SBSQI) based numerical scheme for hyperbolic conservation laws (HCL) is proposed. To ensure the non-oscillatory profile of the solution, an adaptive SBSQI (ASBSQI) scheme for HCL is considered. The ASBSQI method maintains higher order accuracy in the smooth regions using SBSQI approximation and in the non-smooth regions, the traditional WENO-JS method is used to preserve a non-oscillatory profile.

1. Introduction

For approximate solution of the 1D non-linear hyperbolic conservation law

$$u_t + f(u)_x = 0, \quad (1.1)$$

with the initial condition $u(x, 0) = u_0(x)$, high-resolution schemes are considered. The solution of Eq. (1.1) may admit regions of sharp transition. Shocks may develop in the solution within a finite time. Many methods such as spectral or finite difference methods fail to capture the solution accurately in the regions where the solution has large variations or shocks. B-splines are employed extensively in developing numerical schemes for partial differential equations, mainly because of their better approximation properties as compared to polynomials. Consider the interval $I = [a, b]$ to be partitioned into m subintervals $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, $j = 1, 2, \dots, m$ of equal length $\Delta x = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$. Let $\sum_m = \{x_j : j = 1, 2, \dots, m\}$ denotes the set of partition points also known as set of *knots* and $S^d(I, \sum_m)$ denotes the $(m + d)$ -dimensional spline space of degree d over the partition \sum_m . The set $\{B_j^d : j = 1, \dots, m + d\}$ forms the basis of spline space $S^d(I, \sum_m)$, where B_j^d denotes the B-spline of degree d . The septic B-spline quasi-interpolation is defined as $Q_d f = \sum_{j=1}^{m+d} \mu_j(f) B_j^d$ where coefficient $\mu_j(f)$ is the linear combination of discrete values of f (see [2] for more details).

2. Adaptive SBSQI WENO scheme

This section briefly describes how to design ASBSQI-WENO scheme to solve Eq. (1.1).

step 1. By considering the big stencil $S = \{I_{j-2}, \dots, I_{j+2}\}$, the fourth degree reconstruction polynomial p can be easily obtained by considering the following condition

$$\frac{1}{\Delta x} \int_{I_i} p(\eta) d\eta = \bar{u}_i, \quad i = j - 2, \dots, j + 2. \quad (2.1)$$

Keywords: B-spline, finite difference methods, conservation laws.

AMS Mathematical Subject Classification [2010]: 65M06, 35L65.

step 2. Identify the extreme points of $p(x)$. Since the degree of $p'(x)$ is at most three, therefore, the real zero points of $p'(x)$ can be explicitly solved and one is the extreme point of $p(x)$ if it is not a doubled zero point of $p'(x)$.

step 3. Now if the extreme points of the reconstruction polynomial $p(x)$ are outside the big stencil S or there is no extreme point at all,

$$u_t + \sum_{j=1}^{m+5} \mu_j(f)(B_j^5(x))' = 0, \quad x \in [a, b]. \quad (2.2)$$

Using the notations $f_k = f(u_k)$ and $u_k = u(x_k, t)$ for $x_k \in \sum_m$, the above equation reduces to a system of ODEs

$$\frac{du_j(t)}{dt} + \frac{1}{\Delta x} \sum_{i=j-5}^{j+5} b_i f_i = 0, \quad (2.3)$$

where the coefficients b_k can be found from [2] and the procedure jumps to step 5.

step 4. Now if there is one or more extreme points in the big stencil S , the traditional WENO-JS process proposed by Jiang and Shu [1] is applied as follows. The approximation of Eq. (1.1) leads to system of ODEs by applying the method of lines, where the finite difference approximation is replaced to the spatial derivative and yields a semi-discrete scheme

$$\frac{du_j(t)}{dt} + \frac{1}{\Delta x} (\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}) = 0. \quad (2.4)$$

Here, $\hat{f}_{j\pm\frac{1}{2}}$ are called numerical fluxes. By defining a function $h(x)$ implicitly as

$$f(u) = \frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\eta) d\eta, \quad (2.5)$$

we have $f(u)|_{x=x_j} = \frac{1}{\Delta x} (h(x_{j+\frac{1}{2}}) - h(x_{j-\frac{1}{2}}))$, where $h(x_{j\pm\frac{1}{2}})$ is a approximation to the numerical flux $\hat{f}_{j\pm\frac{1}{2}}$. To ensure the numerical stability and to avoid entropy violating solutions, the flux $f(u)$ is splitted into two parts f^+ and f^- , thus $f(u) = f^+(u) + f^-(u)$ where $\frac{df^+(u)}{du} \geq 0$ and $\frac{df^-(u)}{du} \leq 0$. The numerical fluxes $\hat{f}_{j\pm\frac{1}{2}}$ is obtained by Eq. (2.5) which are positive and negative parts of $f(u)$ respectively and with this we have $\hat{f}_{j+\frac{1}{2}} = \hat{f}_{j+\frac{1}{2}}^+ + \hat{f}_{j+\frac{1}{2}}^-$. The negative part of the split flux, is symmetric to the positive part with respect to $x_{j+\frac{1}{2}}$, therefore we will only describe how $\hat{f}_{j+\frac{1}{2}}^+$ is approximated. From here onwards, the '+' sign in the superscript is dropped for simplicity. To construct $\hat{f}_{j+\frac{1}{2}}$, the traditional fifth-order WENO scheme employs the big stencil S which is subdivided into three candidate sub-stencils $S_k = \{I_{j+k}, I_{j+k-1}, I_{j+k-2}\}$ with $k = 0, 1, 2$. Let $\hat{f}_{j+\frac{1}{2}}^k = \sum_{q=0}^2 c_{k,q} f_{j+k+q-2}$ be the second-degree polynomial constructed on S_k to approximate $h(x_{j+\frac{1}{2}})$ where the coefficients $c_{k,q}$ are the Lagrange's interpolation coefficients depending on the shifting parameter k . The flux values on each sub-stencils can be seen in [1]. The convex combination of the flux functions define the approximation to the value of $h(x_{j+\frac{1}{2}})$ which is $\hat{f}_{j+\frac{1}{2}} = \sum_{k=0}^2 w_k \hat{f}_{j+\frac{1}{2}}^k$, where w_k are the non-linear weights. The non-linear weights w_k are defined as

$$w_k = \frac{\alpha_k}{\sum_{q=0}^2 \alpha_q}, \quad \alpha_k = \frac{d_k}{(\epsilon + \beta_k)^2}, \quad (2.6)$$

where $0 < \epsilon \ll 1$ is considered to prevent the denominator becoming zero and the coefficients d_k are known as the ideal weights because they generate the upstream central fifth-order scheme for the five-point stencil S . The values of ideal weights are given by $\{d_0, d_1, d_2\} = \{0.1, 0.6, 0.3\}$. Also, β_k is a smoothness indicator which measures the smoothness of a solution over a particular stencil. The suggested smoothness indicators β_k of Jiang and Shu are given by

$$\beta_k = \sum_{q=1}^2 \Delta x^{2q-1} \int_{I_j} \left(\frac{d^q \hat{f}^k}{dx^q} \right)^2. \quad (2.7)$$

step 5. The semi-discrete scheme (2.4) is discretized in time by the Runge-Kutta method, such as a fourth-order one [1].

3. Numerical results

In this section, the numerical results obtained by ASBSQI-WENO are compared with WENO-JS [1]. Therefore, Eq. (1.1) with $\mathbf{u} = (\rho, \rho u, E)$ and $f(\mathbf{u}) = (\rho u, \rho u^2 + p, u(E + p))$, known as Euler equations, with two different initial conditions is considered. Here, $\rho, u, p = 0.4(E - \frac{1}{2}\rho u^2)$ and E are the density, velocity, pressure and total energy, respectively. The initial conditions are (α) $(\rho, u, p) = (1 + 0.99 \sin(x), 1, 1)$ for $x \in [0, 2\pi]$; (β) $(\rho, u, p) = (0.445, 0.698, 3.528)$ for $x \in [0, 0.5)$ and $(\rho, u, p) = (0.5, 0, 3.571)$ for $x \in [0.5, 1]$. For case α , periodic boundary conditions are applied and the exact solution of ρ is $\rho(x, t) = 1 + 0.99 \sin(x - t)$. In case β , known as Lax problem, transmissive boundary conditions are applied. The results of ASBSQI-WENO and WENO-JS for density are shown in Table 1. Fig. 1 shows the numerical results of both schemes.

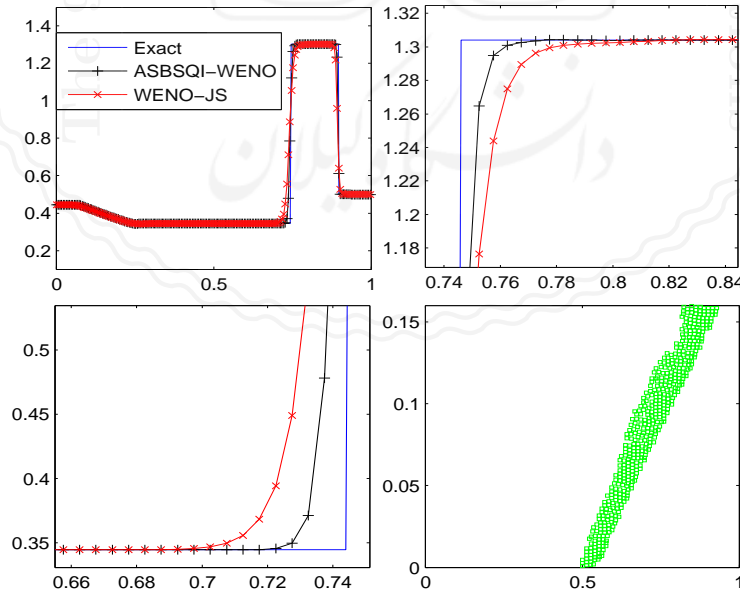


Figure 1: The Lax problem with $N = 200$. Top(left): the density. Top(right): the zoomed region of the density. Bottom(left): the zoomed region of the density. Bottom(right): the points where the WENO reconstruction procedure is used in the SWENO scheme “□”.

Table 1: L_1 and L_∞ errors and the order of convergence for case α .

N	WENO-JS				ASBSQI-WENO			
	L_1 -error	L_1 -order	L_∞ -error	L_∞ -order	L_1 -error	L_1 -order	L_∞ -error	L_∞ -order
40	8.64(-06)	-	1.79(-05)	-	5.08(-06)	-	1.04(-05)	-
80	2.87(-07)	4.91	5.74(-07)	4.96	1.07(-07)	5.57	5.36(-07)	4.28
160	8.89(-09)	5.01	1.87(-08)	4.94	6.00(-09)	4.16	1.57(-08)	5.09
320	2.90(-10)	4.94	5.57(-10)	5.07	2.53(-10)	4.57	5.44(-10)	4.85
640	8.68(-12)	5.06	1.74(-11)	5.00	8.25(-12)	4.94	1.53(-11)	5.15
1280	2.81(-13)	4.95	4.85(-13)	5.16	2.52(-13)	5.03	4.30(-13)	5.15

References

- [1] G.S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126 (1996) 202–228.
- [2] Sablonnière, Paul, Univariate spline quasi-interpolants and applications to numerical analysis. *Rendiconti del Seminario Matematico Università e Politecnico di Torino*, 63 (2005) 211–222.

Rooholah Abedian

School of Engineering Science, College of Engineering, University of Tehran, Tehran, Iran

email address: rabedian@ut.ac.ir



University of Guilan



T_M -eigenvalue of odd-order tensors

Mehri Pakmanesh, Hamid Reza Afshin

Abstract. In this paper, we propose a definition for eigenvalues of odd-order tensors based on some operators. Also, we define the Schur form and the Jordan canonical form of such tensors, and discuss commuting families of tensors. Furthermore, we prove some eigenvalue inequalities for Hermitian tensors. Finally, we introduce characteristic polynomials of odd-order tensors.

1. Introduction

In 2010, Misha E. Kilmer introduced a notion of product for third-order tensors, and used it to define an SVD decomposition and an approximation of the CP decomposition of such tensors. The product has been widely used in many areas, including computer vision, image processing, signal processing, data completion and denoising, low-rank tensor recovery, and robust tensor PCA. Due to the importance of the T -product, some researchers have focused on third-order F -square tensors and their properties [1]. Liu [1] defined T -eigenvalues based on the T -product of third-order F -square tensors in 2021. Qi, Miao, and Wei in [2] presented the definition of generalized tensor function according to the tensor singular value decomposition (T-SVD) based on the tensor T -product. In [3], T -similarity, T -Jordan canonical forms, and T -eigenvalues of third-order F -square tensors were introduced.

Our idea is to propose a product for all odd-ordered tensors. This is in fact a generalization of the product introduced by Kilmer. Using this product, one can define a similar T-SVD decomposition, which in turn can be utilized to approximate the CP decomposition of higher-order tensors

In this paper, we introduce T_M -eigenvalues of odd-order tensors, and discuss their properties. Moreover, we study commuting families of tensors, extend Weyl's theorem and Cauchy's interlacing theorem from matrices to tensors, and introduce Schur and Jordan canonical forms for odd-order tensors. Finally, we define T_M -characteristic polynomials.

2. The T_M -product

An odd-order tensor $\mathcal{A} = (a_{i_1, i_2, \dots, i_m, j_1, j_2, \dots, j_m, d})$ is a multi-array of entries $a_{i_1, i_2, \dots, i_m, j_1, j_2, \dots, j_m, d} \in \mathbb{C}$, where $i_j = 1, \dots, n_j$ for $j = 1, \dots, m$, $j_k = 1, \dots, q_k$ for $k = 1, \dots, m$, and $d = 1, \dots, p$. Sometimes, we simply call $n_1 \times n_2 \times \dots \times n_m \times q_1 \times q_2 \times \dots \times q_m \times p$ the size of \mathcal{A} . We denote the set of all odd-order tensors by $\mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times q_1 \times q_2 \times \dots \times q_m \times p}$.

Definition 2.1. For $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times q_1 \times q_2 \times \dots \times q_m \times p}$, we let

$$\mathcal{A}^{(k)} = \mathcal{A}(\dots; \dots; k) \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times q_1 \times q_2 \times \dots \times q_m}.$$

Keywords: T_M -product, T_M -eigenvalue, T_M -Schur form, T_M -Jordan canonical form, Odd-order tensor.

AMS Mathematical Subject Classification [2010]: 15A48, 15A69, 65F10.

By transforming the indices, a tensor can be represented by a matrix which is called the matricization of the tensor. For a tensor $A^{(k)} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times q_1 \times q_2 \times \dots \times q_m}$, we introduce the matricization $A_M^{(k)}$ which is an $(n_1 n_2 \dots n_m) \times (q_1 q_2 \dots q_m)$ matrix defined by

$$A_M^{(k)}(d, c) = a_{i_1, i_2, \dots, i_m, j_1, j_2, \dots, j_m, k},$$

where

$$d = 1 + \sum_{l=1}^m (i_l - 1) \prod_{l'=1}^{l-1} n_{l'}, \quad c = 1 + \sum_{l=1}^m (j_l - 1) \prod_{l'=1}^{l-1} q_{l'}.$$

The operators bcirc_M , unfold_M and fold_M are defined by

$$\text{bcirc}_M(\mathcal{A}) := \begin{bmatrix} A_M^{(1)} & A_M^{(n)} & A_M^{(n-1)} & \dots & A_M^{(2)} \\ A_M^{(2)} & A_M^{(1)} & A_M^{(n)} & \dots & A_M^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_M^{(n)} & A_M^{(n-1)} & \dots & A_M^{(2)} & A_M^{(1)} \end{bmatrix},$$

$$\text{unfold}_M(\mathcal{A}) := \begin{bmatrix} A_M^{(1)} \\ A_M^{(2)} \\ \vdots \\ A_M^{(n)} \end{bmatrix},$$

and $\text{fold}_M(\text{unfold}_M(\mathcal{A})) := \mathcal{A}$, which means that fold_M is the inverse operator of unfold_M . It is easy to check that these operators are bijective. Therefore, we also consider bcirc_M^{-1} as the inverse operator of bcirc_M , so that $\text{bcirc}_M^{-1}(\text{bcirc}_M(\mathcal{A})) = \mathcal{A}$.

Definition 2.2. If $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times q_1 \times q_2 \times \dots \times q_m \times p}$ and $\mathcal{B} \in \mathbb{C}^{q_1 \times q_2 \times \dots \times q_m \times p_1 \times p_2 \times \dots \times p_m \times p}$, then the T_M -product of \mathcal{A} and \mathcal{B} , denoted by $\mathcal{A} * \mathcal{B}$, is an $n_1 \times n_2 \times \dots \times n_m \times p_1 \times p_2 \times \dots \times p_m \times p$ tensor defined by

$$\mathcal{A} * \mathcal{B} = \text{fold}_M(\text{bcirc}_M(\mathcal{A})\text{unfold}_M(\mathcal{B})).$$

Definition 2.3. Let \mathcal{A} be an $n_1 \times n_2 \times \dots \times n_m \times q_1 \times q_2 \times \dots \times q_m \times p$ tensor. The tensor \mathcal{A}^T is a $q_1 \times q_2 \times \dots \times q_m \times n_1 \times n_2 \times \dots \times n_m \times p$ tensor, obtained as follows. Transpose each $A_M^{(k)}$, and then commute $A_M^{(2)}$ with $A_M^{(p)}$, $A_M^{(3)}$ with $A_M^{(p-1)}$, and continue until the end. Also, the conjugate transpose \mathcal{A}^* is obtained using the following procedure. Conjugate transpose each $A_M^{(k)}$, and then commute $A_M^{(2)}$ with $A_M^{(p)}$, $A_M^{(3)}$ with $A_M^{(p-1)}$, and continue until the end.

Lemma 2.4. If $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times q_1 \times q_2 \times \dots \times q_m \times p}$ and $\mathcal{B} \in \mathbb{C}^{q_1 \times q_2 \times \dots \times q_m \times p_1 \times p_2 \times \dots \times p_m \times p}$, then the following hold.

- (i) $\text{unfold}_M(\mathcal{A}) = \text{bcirc}_M(\mathcal{A})E_1^{pq_1 q_2 \dots q_m \times q_1 q_2 \dots q_m}$.
- (ii) $\text{bcirc}_M(\text{fold}_M(\text{bcirc}_M(\mathcal{A})E_1^{pq_1 q_2 \dots q_m \times q_1 q_2 \dots q_m})) = \text{bcirc}_M(\mathcal{A})$.
- (iii) The operator bcirc_M is a linear operator, that is,

$$\text{bcirc}_M(\alpha\mathcal{A} + \beta\mathcal{B}) = \alpha\text{bcirc}_M(\mathcal{A}) + \beta\text{bcirc}_M(\mathcal{B}),$$

where \mathcal{A} and \mathcal{B} are of the same size, and $\alpha, \beta \in \mathbb{C}$.

$$(iv) \text{bcirc}_M(\mathcal{A} * \mathcal{B}) = \text{bcirc}_M(\mathcal{A})\text{bcirc}_M(\mathcal{B}).$$

$$(v) \text{bcirc}_M(\mathcal{A}^T) = (\text{bcirc}_M(\mathcal{A}))^T \text{ and } \text{bcirc}_M(\mathcal{A}^*) = (\text{bcirc}_M(\mathcal{A}))^*.$$

Definition 2.5. Suppose that $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times n_1 \times n_2 \times \dots \times n_m \times p}$, $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times 1 \times 1 \times \dots \times 1 \times p}$ and $\mathcal{X} \neq 0$. If

$$\mathcal{A} * \mathcal{X} = \lambda \mathcal{X}, \quad (2.1)$$

for some $\lambda \in \mathbb{C}$, then λ is called a T_M -eigenvalue of \mathcal{A} , and \mathcal{X} is said to be a T_M -eigenvector of \mathcal{A} .

Lemma 2.6. The tensor eigenvalue problem in (2.1) is equivalent to the matrix eigenvalue system

$$\text{bcirc}_M(\mathcal{A})\text{unfold}_M(\mathcal{X}) = \lambda\text{unfold}_M(\mathcal{X}). \quad (2.2)$$

According to Lemma 2.6, it is possible to calculate the T_M -eigenvalues and T_M -eigenvectors of \mathcal{A} by solving the eigenvalue problem of matrix $\text{bcirc}_M(\mathcal{A})$.

Now, we define symmetric and Hermitian tensors.

Definition 2.7. A tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times n_1 \times n_2 \times \dots \times n_m \times p}$ is called symmetric if $\mathcal{A}^T = \mathcal{A}$, and it is called Hermitian if $\mathcal{A}^* = \mathcal{A}$.

Since T_M -eigenvalues of a Hermitian tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times n_1 \times n_2 \times \dots \times n_m \times p}$ are real, we adopt the convention that they are always arranged in the algebraically nondecreasing order, that is,

$$\lambda_{\min} = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n_1 n_2 \dots n_m p - 1} \leq \lambda_{n_1 n_2 \dots n_m p} = \lambda_{\max}. \quad (2.3)$$

In the following, we extended Weyl theorem for tensors.

Theorem 2.8. Let $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times n_1 \times n_2 \times \dots \times n_m \times p}$ be Hermitian tensors, and let the respective T_M -eigenvalues of \mathcal{A}, \mathcal{B} and $\mathcal{A} + \mathcal{B}$ be $\{\lambda_i(\mathcal{A})\}_{i=1}^{n_1 n_2 \dots n_m p}$, $\{\lambda_i(\mathcal{B})\}_{i=1}^{n_1 n_2 \dots n_m p}$ and $\{\lambda_i(\mathcal{A} + \mathcal{B})\}_{i=1}^{n_1 n_2 \dots n_m p}$, each algebraically ordered as in (2.3). Then

$$\lambda_i(\mathcal{A} + \mathcal{B}) \leq \lambda_{i+j}(\mathcal{A}) + \lambda_{n_1 n_2 \dots n_m p - j}(\mathcal{B}), \quad j = 0, 1, \dots, n_1 n_2 \dots n_m p - i, \quad (2.4)$$

for each $i = 1, 2, \dots, n_1 n_2 \dots n_m p$, with equality for some pair i, j if and only if there exists a nonzero tensor

$$\mathcal{X} = \text{fold}_M(F_p^* e_k \otimes x) \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_m \times 1 \times 1 \times \dots \times 1 \times p}, \quad (2.5)$$

such that

$$\mathcal{A} * \mathcal{X} = \lambda_{i+j}(\mathcal{A})\mathcal{X}, \quad \mathcal{B} * \mathcal{X} = \lambda_{n_1 n_2 \dots n_m p - j}(\mathcal{B})\mathcal{X}, \quad (\mathcal{A} + \mathcal{B}) * \mathcal{X} = \lambda_i(\mathcal{A} + \mathcal{B}) * \mathcal{X},$$

where $x \in \mathbb{C}^{n_1 n_2 \dots n_m}$, F_p is the $p \times p$ discrete Fourier matrix, e_k is the k^{th} column of the $p \times p$ identity matrix, and $1 \leq k \leq p$. Also,

$$\lambda_{i-j+1}(\mathcal{A}) + \lambda_j(\mathcal{B}) \leq \lambda_i(\mathcal{A} + \mathcal{B}), \quad j = 1, 2, \dots, i, \quad (2.6)$$

for each $i = 1, 2, \dots, n_1 n_2 \dots n_m p$, with equality for some pair i, j if and only if there exists a nonzero tensor \mathcal{X} such that

$$\mathcal{A} * \mathcal{X} = \lambda_{i-j+1}(\mathcal{A}) * \mathcal{X}, \quad \mathcal{B} * \mathcal{X} = \lambda_j(\mathcal{B})\mathcal{X}, \quad (\mathcal{A} + \mathcal{B}) * \mathcal{X} = \lambda_i(\mathcal{A} + \mathcal{B})\mathcal{X},$$

where \mathcal{X} is defined as in (2.5). If \mathcal{A} and \mathcal{B} have no common T_M -eigenvectors, then the inequalities in (2.4) and (2.6) are strict.

References

- [1] W.H. Liu, X.Q. Jin, A study on T-eigenvalues of third-order tensors, *Linear Algebra and its Applications*, 612 (2021) 357–374.
- [2] Y. Miao, Q. Liqu, W. Yimin, Generalized tensor function via the tensor singular value decomposition based on the T-product, *Linear Algebra and its Applications*, 590 (2020) 258–303.
- [3] Y. Miao, Q. Liqu, W. Yimin, T-Jordan canonical form and T-Drazin inverse based on the T-product, *Communications on Applied Mathematics and Computation*, 32 (2021) 201–220.

Mehri Pakmanesh

Department of Mathematics, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran

email address: mehri.pakmanesh@stu.vru.ac.ir

Hamid Reza Afshin

Department of Mathematics, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran

email address: afshin@vru.ac.ir



University of Guilan



Perturbed extended second derivative backward differentiation formula

Tahere Majidi, Ali Abdi, Gholamreza Hojjati

Abstract. The purpose of this paper is to study methods with super-future point technique from second derivative general linear methods point of view, which makes it possible to have satisfactory stability properties. We present one new general class of methods by perturbing the abscissa vector of the modified extended second derivative backward differentiation formula, that can be considered as modified second derivative linear multistep methods. Some numerical experiments confirm the efficiency and robustness of the proposed method in solving stiff problems.

1. Introduction

Consider the stiff initial value problem

$$\begin{cases} y'(t) = f(y(t)), & t \in [t_0, T], \\ y(t_0) = y_0, \end{cases} \quad (1.1)$$

where $f : \mathbb{R}^m \rightarrow \mathbb{R}^m$ and m is the dimension of the system. One of the most popular directions to construct methods with a good accuracy and desirable stability properties is using of super-future point technique based on backward differentiation formula (BDF) and higher derivatives of the solution [3]. Using this idea leads into extended second derivative BDF methods (ESDBDFs) [4]. Second derivative general linear methods (SGLMs) as a unifying framework for the traditional numerical methods using second derivative of the solution [1, 2]. The k -step ESDBDFs have the general form [4]

$$\sum_{j=0}^k \hat{\alpha}_j y_{n+j} = h \hat{\beta}_k f_{n+k} + h^2 (\hat{\gamma}_k g_{n+k} - \hat{\gamma}_{k+1} g_{n+k+1}). \quad (1.2)$$

The predictor was defined by second derivative BDF (SDBDF)

$$y_{n+k} + \sum_{j=0}^{k-1} \alpha_j y_{n+j} = h \beta_k f_{n+k} + h^2 \gamma_k g_{n+k}. \quad (1.3)$$

The algorithm based on ESDMM approach goes as follows:

- (1) Compute \bar{y}_{n+k} as the solution of the k -step SDBDF

$$\bar{y}_{n+k} + \sum_{j=0}^{k-1} \alpha_j y_{n+j} = h \beta_k \bar{f}_{n+k} + h^2 \gamma_k \bar{g}_{n+k}, \quad (1.4)$$

Keywords: Extended second derivative multistep methods, Second derivative methods, Second derivative general linear methods, Stiff problems, A^- and $A(\alpha)$ -stability.

AMS Mathematical Subject Classification [2010]: 65L05.

(2) Compute \bar{y}_{n+k+1} as the solution of the k -step SDBDF

$$\bar{y}_{n+k+1} + \alpha_{k-1}\bar{y}_{n+k} + \sum_{j=0}^{k-2} \alpha_j y_{n+j+1} = h\beta_k \bar{f}_{n+k+1} + h^2 \gamma_k \bar{g}_{n+k+1}, \quad (1.5)$$

(3) Compute y_{n+k} as the solution of the k -step ESDBDF

$$y_{n+k} + \sum_{j=0}^{k-1} \hat{\alpha}_j y_{n+j} = h\hat{\beta}_k f_{n+k} + h^2(\hat{\gamma}_k g_{n+k} - \hat{\gamma}_{k+1} \bar{g}_{n+k+1}). \quad (1.6)$$

The stage (3) can be replaced by a modified ESDBDF (MESDBDF) as follows:

(3*) Compute y_{n+k} from (1.2) as the solution of

$$y_{n+k} + \sum_{j=0}^{k-1} \hat{\alpha}_j y_{n+j} = h(\hat{\beta}_k - \beta_k) \bar{f}_{n+k} + h\beta_k f_{n+k} + h^2(\hat{\gamma}_k - \gamma_k) \bar{g}_{n+k} - h^2 \hat{\gamma}_{k+1} \bar{g}_{n+k+1} + h^2 \gamma_k g_{n+k}. \quad (1.7)$$

2. A review on the SGLMs

SGLMs for the numerical solution of (1.1) are defined by

$$\begin{aligned} Y^{[n]} &= h(A \otimes I_m) f(Y^{[n]}) + h^2(\bar{A} \otimes I_m) g(Y^{[n]}) + (U \otimes I_m) y^{[n-1]}, \\ y^{[n]} &= h(B \otimes I_m) f(Y^{[n]}) + h^2(\bar{B} \otimes I_m) g(Y^{[n]}) + (V \otimes I_m) y^{[n-1]}, \end{aligned} \quad (2.1)$$

$n = 1, 2, \dots, N$, where $Nh = T - t_0$ and h is the stepsize, \otimes the Kronecker product of two matrices and I_m stands for the identity matrix of dimension m . Here, the vector $Y^{[n]} = [Y_i^{[n]}]_{i=1}^s$ denotes approximations of the stage order q to the vector $y(t_{n-1} + ch) = [y(t_{n-1} + c_i h)]_{i=1}^s$ where $c = [c_1 \ c_2 \ \dots \ c_s]^T$ is the abscissa vector. Also $y^{[n-1]} = [y_i^{[n-1]}]_{i=1}^r$ and $y^{[n]} = [y_i^{[n]}]_{i=1}^r$ are the input and output vectors at the step number n , respectively. For more details on SGLMs, see [1, 2].

2.1. MESDBDF as SGLMs

In order to represent MESDBDF as SGLMs, it is necessary to replace (1.4) into (1.5). Then MESDBDF scheme can be written as SGLM (2.1) with three stages, k inputs and the abscissae vector defined by

$$c = [k+1 \quad k+2 \quad k+1]^T. \quad (2.2)$$

3. Perturbed MESDBDF

In this section, we perturb the abscissa vector c of the MESDBDF methods that will be called modified second derivative linear multistep methods (MSLMMs). Therefore, the resulting coefficient vector takes the form

$$c = [k+1 + \xi_1, \quad k+1 + \xi_2, \quad k+1]^T. \quad (3.1)$$

Here, we investigate the stability properties of MSLMMs. The stability polynomial $p(w, z)$ of these methods can be taken the following form

$$p(w, z) = \frac{1}{(1 - \lambda z - \mu z^2)^3} \sum_{j=0}^k a_j(z) w^j, \quad (3.2)$$

where each $a_j(z)$, $j = 0, 1, \dots, k$, is a polynomial of degree at most six in z , whose coefficients depend on the two parameters ξ_1 and ξ_2 . We aim to find the optimum values for the free parameters ξ_1 and ξ_2 to maximize the angle α of $A(\alpha)$ -stability of the methods. By using the boundary locus technique [5], we constructed an objective function

$$f_n : (\xi_1, \xi_2) \rightarrow [0, \frac{\pi}{2}].$$

We minimize objective function f_n by using `fminsearch` command from MATLAB. The values of angle α of $A(\alpha)$ -stability of MSLMMs are reported and compared with MESDBDFs in Table 1.

Table 1: Angles α of $A(\alpha)$ -stability for MSLMMs and MESDBDFs for $k = 5, 6, 7, 8$.

k	ξ_1	ξ_2	MSLMM		MESDBDF	
			p	α	p	α
5	26	23	7	90°	7	89.86°
	19237	160				
6	795	2653	8	90°	8	88.49°
	89826	36923				
7	69	811	9	89.99°	9	85.43°
	5000	5000				
8	187	325	10	89.85°	10	81.81°
	2500	2082				

4. Numerical experiments

We consider the linear stiff system

$$\begin{cases} y_1' = -\alpha y_1 - \beta y_2 + (\alpha + \beta - 1)e^{-t}, & y_1(0) = 1, \\ y_2' = \beta y_1 - \alpha y_2 + (\alpha - \beta - 1)e^{-t}, & y_2(0) = 0, \end{cases} \quad (4.1)$$

with the exact solution $y_1(t) = y_2(t) = e^{-t}$. In our numerical experiments, we select $\alpha = 0.6$, $\beta = 25$. Numerical results in Table 2 show that MSLMMs are more accurate than MESDBDF.

Table 2: Computed error at the end of the interval of integration $[0, 60]$ for MESDBDF and MSLMM applied to problem 4.1.

$k = 7$	h	0.1	0.05	0.025
	MESDBDF	1.10×10^2	3.84×10^{-27}	1.76×10^{-23}
	MSLMM	5.12×10^{-35}	4.52×10^{-31}	5.97×10^{-31}

References

- [1] A. Abdi, M. Bras, G. Hojjati, On the construction of second derivative diagonally implicit multistage integration methods for ODEs, *Applied Numerical Mathematics*, 76 (2014) 1–18.
- [2] J.C. Butcher, G. Hojjati, Second derivative methods with RK stability, *Numerical Algorithms*, 40 (2005) 415–429.

- [3] J.R. Cash, Second derivative extended backward differentiation formulas for the numerical integration of stiff systems, *SIAM Journal on Numerical Analysis*, 18 (1981) 21–36.
- [4] G. Hojjati, M.Y.R. Ardabili, S.M. Hosseini, New second derivative multistep methods for stiff systems, *Applied Mathematical Modelling*, 30 (2006) 466–476.
- [5] J.D. Lambert, *Numerical methods for ordinary differential systems*, Wiley, New York, 1991.

Tahere Majidi

Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

email address: t.majidi@tabrizu.ac.ir

Ali Abdi

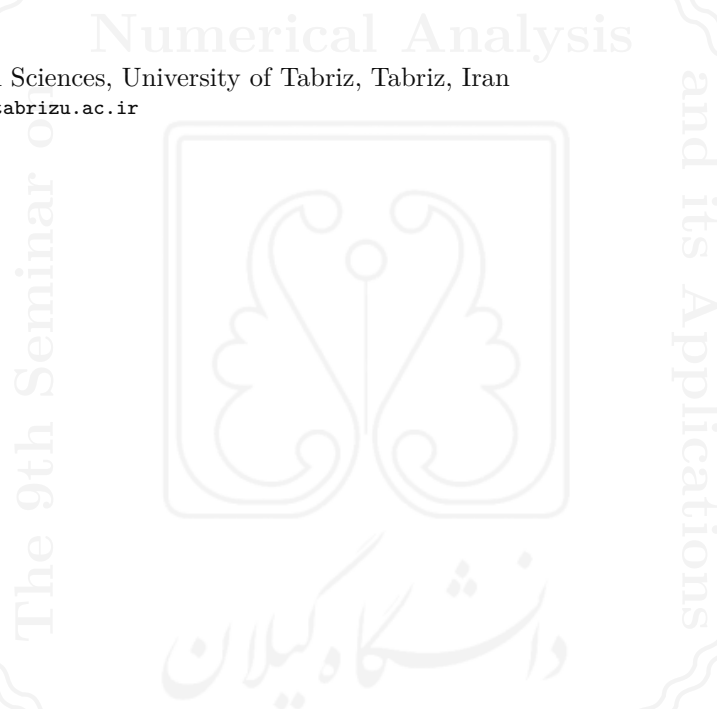
Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

email address: a_abdi@tabrizu.ac.ir

Gholamreza Hojjati

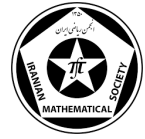
Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

email address: ghojjati@tabrizu.ac.ir





University of Guilan



Restricting the solution set of the interval linear systems with multiple right-hand sides

Marzieh Dehghani-Madiseh

Abstract. In this work, we study the interval linear systems with multiple right-hand sides $\mathbf{A}X = \mathbf{B}$ and consider it as an interval linear matrix equation. Using the Kronecker product, this interval matrix equation is converted to the interval linear system $(I \otimes \mathbf{A})\text{vec}(X) = \text{vec}(\mathbf{B})$. Solution set of the Kronecker form encloses the solution set of the main interval matrix equation $\mathbf{A}X = \mathbf{B}$. Thus we try to impose some conditions which imply regularity of $I \otimes \mathbf{A}$ and so boundedness the solution set of the main problem. The proposed conditions do not deal directly with the large interval matrix $I \otimes \mathbf{A}$ and so in computational point of view are efficient.

1. Introduction

Some class of problems such as structural mechanics, computation of the frequency response matrix and electromagnetic scattering naturally lead to solve several linear systems that have the same coefficient matrix but differ in their right-hand sides, i.e.,

$$A(x_1, x_2, \dots, x_n) = (b_1, b_2, \dots, b_n).$$

These systems can be written in a matrix form

$$AX = B, \tag{1.1}$$

therein A and B are m -by- m and m -by- n known matrices respectively, and X is the m -by- n unknown matrix. Practically, components of A and B are obtained from experience, but due to the measurement errors, theses components will accompany with some uncertainties which can be presented in an interval form and so we will have the interval matrix equation

$$\mathbf{A}X = \mathbf{B}, \tag{1.2}$$

in which \mathbf{A} and \mathbf{B} are interval matrices. The interval matrix equation (1.2) was studied by Hashemi and Dehghan [3]. They presented some analytic results and characterizations for AE-solution sets of this interval matrix equation. Dehghani-Madiseh and Dehghan [2] considered the parameterized version of the equation (1.1), i.e., the parametric matrix equation $A(p)X = B(p)$.

A first idea for dealing with the interval equation (1.2) is to convert it to the interval linear system

$$\mathbf{G}x = \mathbf{b}, \tag{1.3}$$

therein $\mathbf{G} = I \otimes \mathbf{A}$, $x = \text{vec}(X)$, $\mathbf{b} = \text{vec}(\mathbf{B})$ and I stands for the identity matrix of order n . The Kronecker product $E \otimes F$ of two matrices E and F is the block matrix whose (i, j) -th block is $e_{ij}F$. For $X = (x_{ij}) \in \mathbb{R}^{m \times n}$, the vector $\text{vec}(X) \in \mathbb{R}^{mn}$ is obtained by stacking the columns of X , i.e., $\text{vec}(X) = (x_{11}, \dots, x_{m1}, \dots, x_{1n}, \dots, x_{mn})^\top$.

Here using regularity concept of the interval matrices, we present some conditions under which the coefficient matrix $I \otimes \mathbf{A}$ is regular and besides the solution set of (1.2) is bounded. Advantage of our idea is that we do not directly deal with the large interval matrix $I \otimes \mathbf{A}$, since this matrix is of high dimension specially when n is large. We just work with the main matrix \mathbf{A} which considerably reduces computational costs.

Notations. In this note, bold face letters denote interval quantities and ordinary letters stand for real quantities. $\mathbb{IR} = \{\mathbf{x} = [\underline{\mathbf{x}}, \overline{\mathbf{x}}] : \underline{\mathbf{x}} \leq \overline{\mathbf{x}}, \underline{\mathbf{x}}, \overline{\mathbf{x}} \in \mathbb{R}\}$ is the set of real intervals and the set of m -by- n real interval matrices is denoted by $\mathbb{IR}^{m \times n}$. For the real interval $\mathbf{x} = [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ define midpoint $x^c := (\overline{\mathbf{x}} + \underline{\mathbf{x}})/2$ and radius $x^\Delta := (\overline{\mathbf{x}} - \underline{\mathbf{x}})/2$. The concepts of midpoint and radius for interval vectors and matrices are defined componentwise. For a real matrix A , $\rho(A)$ denotes its spectral radius.

2. Our result

In this section using the same convention in [3], we first introduce the concept of solution set for the interval matrix equation (1.2) and then using its Kronecker form we present some conditions under which this solution set is bounded.

Definition 2.1. The solution set of the interval matrix equation (1.2) is defined as

$$\Xi(\mathbf{A}, \mathbf{B}) = \{X \in \mathbb{R}^{m \times n} : (\exists A \in \mathbf{A})(\exists B \in \mathbf{B})(AX = B)\}. \quad (2.1)$$

Let $\Xi(\mathbf{G}, \mathbf{b})$ be the solution set of (1.3) and define \mathbb{S} as

$$\mathbb{S} = \{\text{vec}(X) : X \in \Xi(\mathbf{A}, \mathbf{B})\}, \quad (2.2)$$

then it is obvious that $\mathbb{S} \subseteq \Xi(\mathbf{G}, \mathbf{b})$. Therefore, by solving the interval linear system (1.3) using the existing methods, we can specify the columns of the interval matrix \mathbf{X} as an enclosure for $\Xi(\mathbf{A}, \mathbf{B})$. But the enclosure of $\Xi(\mathbf{A}, \mathbf{B})$ is achievable if it is a bounded set. Here, using the Kronecker form (1.3), we present some conditions for boundedness the solution set of the interval system (1.2).

Definition 2.2. The square interval matrix \mathbf{A} is regular if each $A \in \mathbf{A}$ is nonsingular.

Theorem 2.3. [1] Let $\mathbf{G} \in \mathbb{IR}^{n \times n}$. If $\rho(|(G^c)^{-1}|G^\Delta) < 1$, then \mathbf{G} is regular.

Lemma 2.4. If $A, C \in \mathbb{R}^{m \times m}$, $B, D \in \mathbb{R}^{n \times n}$ and $X \in \mathbb{R}^{m \times n}$, then we have

1. $(A \otimes B)(C \otimes D) = AC \otimes BD$,
 2. $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$,
 3. $\text{vec}(AX) = (I \otimes A)\text{vec}(X)$,
 4. $\lambda(A \otimes B) = \lambda(A)\lambda(B)$,
- therein $\lambda(A)$ and $\lambda(B)$ denote the eigenvalues of A and B , respectively.

Theorem 2.5. Consider the interval linear system of equations (1.3). If $\rho(|(A^c)^{-1}|A^\Delta) < 1$ then \mathbf{G} is regular and the solution set $\Xi(\mathbf{A}, \mathbf{B})$ of the interval matrix equation (1.2) is bounded.

Proof. Since $\mathbf{G} = I \otimes \mathbf{A}$, we have

$$G^c = (I \otimes \mathbf{A})^c = I \otimes A^c, \quad G^\Delta = (I \otimes \mathbf{A})^\Delta = I \otimes A^\Delta.$$

So using Lemma 2.4 we can write

$$\begin{aligned} |(G^c)^{-1}|G^\Delta &= (|(I \otimes A^c)^{-1}|)(I \otimes A^\Delta) \\ &= (|I \otimes (A^c)^{-1}|)(I \otimes A^\Delta) \\ &= (I \otimes |(A^c)^{-1}|)(I \otimes A^\Delta) \\ &= I \otimes (|(A^c)^{-1}|A^\Delta). \end{aligned}$$

Now again using Lemma 2.4 and the above relation, we have

$$\lambda(|(G^c)^{-1}|G^\Delta) = \lambda(I)\lambda(|(A^c)^{-1}|A^\Delta) = \lambda(|(A^c)^{-1}|A^\Delta).$$

Thus condition $\rho(|(A^c)^{-1}|A^\Delta) < 1$ yields

$$\rho(|(G^c)^{-1}|G^\Delta) < 1,$$

and according to Theorem 2.3 we conclude that the interval matrix \mathbf{G} is regular. Regularity of the coefficient matrix \mathbf{G} yields boundedness of the solution set $\Xi(\mathbf{G}, \mathbf{b})$ of the interval linear system of equations (1.3). So by the mentioned point, we conclude that the solution set $\Xi(\mathbf{A}, \mathbf{B})$ of the interval matrix equation (1.2) is bounded. \square

Now we present another condition which yields the solution set $\Xi(\mathbf{A}, \mathbf{B})$ of the interval matrix equation (1.2) is bounded. This condition also yields regularity of the interval matrix \mathbf{G} .

Theorem 2.6. *Consider the interval matrix equation (1.2). If the inequality $|\text{mid}(\mathbf{A})X| \leq \text{rad}(\mathbf{A})|X|$ has only the trivial solution $X = 0 \in \mathbb{R}^{m \times n}$, then the interval matrix \mathbf{G} in (1.3) is regular and the solution set $\Xi(\mathbf{A}, \mathbf{B})$ is bounded.*

Proof. If we define $x = \text{vec}(X)$, then using Lemma 2.4, we obtain the following equivalent forms for inequality $|\text{mid}(\mathbf{A})X| \leq \text{rad}(\mathbf{A})|X|$

$$\begin{aligned} |\text{mid}(\mathbf{A})X| &\leq \text{rad}(\mathbf{A})|X| \\ \iff |\text{vec}(\text{mid}(\mathbf{A})X)| &\leq \text{vec}(\text{rad}(\mathbf{A})|X|) \\ \iff |(I \otimes A^c)x| &\leq (I \otimes A^\Delta)|x| \\ \iff |(I \otimes \mathbf{A})^c x| &\leq (I \otimes \mathbf{A})^\Delta |x| \\ \iff |G^c x| &\leq G^\Delta |x|. \end{aligned}$$

So assumption of the theorem yields that the inequality $|G^c x| \leq G^\Delta |x|$ has only the trivial solution $x = 0 \in \mathbb{R}^{mn}$. Therefore by Theorem 4.1 of [4], we conclude that the interval matrix \mathbf{G} is regular and similar to what has been mentioned in the previous theorem, regularity of the coefficient matrix \mathbf{G} yields boundedness the solution set $\Xi(\mathbf{A}, \mathbf{B})$ of the interval matrix equation (1.2). \square

Acknowledgement

The author would like to thank the Shahid Chamran University of Ahvaz for financial support under the grant number SCU.MM1400.33518.

References

- [1] H. d. Beeck, Zur Problematik der Huellenbestimmung von Intervallgleichungssystemen. *in: Interval Mathematics (K. Nickel, ed.), Lecture Notes in Comput. Sci. 29*, Springer-Verlag, Berlin, (1975) 150–159.
- [2] M. Dehghani-Madiseh, M. Dehghan, Parametric AE-solution Sets to the Parametric Linear Systems With Multiple Right-Hand Sides and Parametric Matrix Equation $A(p)X = B(p)$. *Numerical Algorithm*, 73 (2016) 245–279.
- [3] B. Hashemi, M. Dehghan, Results Concerning Interval Linear Systems With Multiple Right-Hand Sides and the Interval Matrix Equation $AX = B$, *Journal of Computational and Applied Mathematics*, 235 (2011) 2969–2978.
- [4] J. Rohn, Forty Necessary and Sufficient Condition for Regularity of Interval Matrices: A Survey, *Electronic Journal of Linear Algebra*, 18 (2009) 500–512.

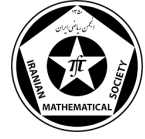
Marzieh Dehghani-Madiseh

Department of Mathematics, Faculty of Mathematical Sciences and Computer, Shahid Chamran University of Ahvaz, Ahvaz, Iran

email address: m.dehghani@scu.ac.ir



University of Guilan



Pricing of European option using three types of b-spline functions

Farshid Nourian, Mehrdad Lakestani, Sedigheh Sabermahani, Yadollah Ordokhani

Abstract. In this paper, we present a numerical method for pricing European options. This approximation method is based on the characteristic function and family of B-Spline function (including: Linear, Quadratic and Cubic B-Spline).

1. Introduction

One of the most important issues in quantitative finance is pricing options using numerical methods that include numerical solution of PDE, numerical integration and the Monte Carlo method. There are various techniques for numerical integration, such as the Cos method [1], the Wavelet method [3] and the SWIFT method [2].

2. option valuation

Here we consider this risk-neutral option valuation formula [2]

$$\begin{aligned} v(x, t) &= \exp(-r(T - t))E^Q[v(y, T) | x] \\ &= \exp(-r(T - t)) \int_R v(y, T)f(y | x)dy, \end{aligned}$$

where v denotes the option value, T is the maturity time, t is the initial date, E^Q is the expectation operator under the risk-neutral measure Q , x and y are state variables at time t and T , respectively, $f(y | x)$ is the probability density of y given x , and r is the deterministic risk-neutral interest rate. The density function f is unknown, while the characteristic function is available for different asset price dynamics, which is the Fourier transform of f .

The variables x and y are also defined as follows

$$x = \ln\left(\frac{S_t}{K}\right), \quad y = \ln\left(\frac{S_T}{K}\right),$$

with S_t the underlying price at time t and K the strike price. Also, the pay-off for European option is obtained from the following equation

$$v(y, T) = [\alpha \cdot K(\exp(y) - 1)]^+,$$

$$\alpha = \begin{cases} 1, & \text{call,} \\ -1, & \text{put.} \end{cases}$$

Keywords: Option pricing, B-Spline function, Characteristic function, Collocation method .

AMS Mathematical Subject Classification [2010]: 91B25, 41A15, 74G15 .

3. B-Spline and Characteristic function

Definition 3.1. The s th order of the B-Spline function is defined as follows [3]

$$N_s(x) = \int_{-\infty}^{+\infty} N_{s-1}(x-t)N_0(t)dt = \int_0^1 N_{s-1}(x-t)dt, \quad s \geq 1,$$

where,

$$N_0(x) = \begin{cases} 1, & x \in [0, 1), \\ 0, & o.w. \end{cases}$$

Definition 3.2. The characteristic function, $g_X(\omega)$ for $\omega \in R$ of the random variable X , is the Fourier-Stieltjes transform of the cumulative distribution function $F_X(x)$, i.e.,

$$g_X(\omega) := E[e^{i\omega X}] = \int_R e^{i\omega x} dF_X(x) = \int_R e^{i\omega x} f_X(x) dx. \quad (3.1)$$

4. Numerical approximation

According to the definition of the characteristic function (3.1) for a specific random variable with the density function f , we have

$$g(\omega) = \int_R e^{i\omega x} f(x) dx.$$

For fixed J , a function $f \in L^2[a, b]$ can be approximated using B-Spline functions as

$$f(x) \approx f_P(x) = \sum_k c_{J,k} \varphi_{J,k}(x) = C^T \Phi(x), \quad (4.1)$$

where $\varphi_{J,k}$ s are B-Spline bases, and C and Φ are vectors which their entries are $c_{J,k}$ s and $\varphi_{J,k}$ s, respectively.

Since $f(x)$ rapidly decays to zero as $x \rightarrow \pm\infty$, we truncate the infinite integration range to $[a, b] \subset R$, without losing significant accuracy,

$$g(\omega) = \int_R e^{i\omega x} f(x) dx \simeq \int_a^b e^{i\omega x} f(x) dx.$$

Using relation (4.1), we get

$$g(\omega) = C^T \int_R e^{i\omega x} \Phi(x) dx \simeq C^T \int_a^b e^{i\omega x} \Phi(x) dx.$$

Assume

$$\Psi(\omega) = \int_a^b e^{i\omega x} \Phi(x) dx,$$

so we get

$$g(\omega) = C^T \Psi(\omega). \quad (4.2)$$

Using a suitable collocation method we change the equation (4.2) to a system of Algebraic equation, which can be solve to find the vector C . So the unknown function f , can be found using relation (4.1).

5. Numerical Example

Assume that asset price dynamics follow the GBM(Geometric Brownian Motion) process, where the characteristic function is as [3]

$$g_{GBM}(\omega) = \exp(-i\omega x - iw(r - q - \frac{1}{2}\sigma^2)(T - t) - \frac{1}{2}\sigma^2\omega^2(T - t)).$$

Let

$$S_0 = 100, \quad r = 0.1, \quad q = 0, \quad T = 0.1, \quad \sigma = 0.25, \quad (5.1)$$

we solve the problem for two strike price $K = 80$ and $K = 120$.

Table 1 and 2 show the absolute errors for different values of j and different orders of B-Spline functions.

Table 1: The absolute errors for different values of j and different orders of B-Spline functions, with parameters as in (5.1); $K = 80$; reference val.= 20.799226309.

-	$j = 4$	$j = 5$
Linear B-Spline	1.1×10^{-3}	3.7×10^{-6}
Quadratic B-Spline	2.9×10^{-3}	8.2×10^{-7}
Cubic B-Spline	3.9×10^{-4}	3.09×10^{-7}

Table 2: The absolute errors for different values of j and different orders of B-Spline functions, with parameters as in (5.1); $K = 120$; reference val.= 0.044577814.

-	$j = 4$	$j = 5$
Linear B-Spline	6.7×10^{-4}	2.5×10^{-5}
Quadratic B-Spline	2.8×10^{-5}	2.4×10^{-6}
Cubic B-Spline	1.4×10^{-5}	2.5×10^{-7}

References

- [1] F. Fang, C. W. Oosterlee, A novel pricing method for European options based on fourier-cosine series expansions, *SIAM Journal on Scientific Computing*, 31 (2008) 826–848.
- [2] L. Ortiz-Gracia, C. W. Oosterlee, A highly efficient Shannon wavelet inverse fourier technique for pricing European options, *SIAM Journal on Scientific Computing*, 38 (2016) B118–B143.
- [3] L. Ortiz-Gracia, C. W. Oosterlee, Robust Pricing of European Options with Wavelets and the Characteristic Function, *SIAM Journal on Scientific Computing*, 35 (2013) B1055–B1084.

Farshid Nourian

Department of Applied Mathematics, Faculty of Mathematics, Statistics and Computer science, University of Tabriz, Tabriz, Iran

email address: f.nourian@tabrizu.ac.ir

Mehrdad Lakestani

Department of Applied Mathematics, Faculty of Mathematics, Statistics and Computer science, University of Tabriz, Tabriz, Iran

email address: lakestani@tabrizu.ac.ir



Sedigheh Sabermahani

Department of Mathematics, Faculty of Mathematical Sciences, Alzahra University, Tehran, Iran

email address: s.saber@alzahra.ac.ir

Yadollah Ordokhani

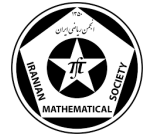
Department of Mathematics, Faculty of Mathematical Sciences, Alzahra University, Tehran, Iran

email address: ordokhani@alzahra.ac.ir





University of Guilan



Approximate solution of the local time M-fractional Volterra integral equations of the second kind

Mousa Ilie, Ali Khoshkenar

Abstract. In the current article, the local time M-fractional Volterra integral equations are presented and so the homotopy perturbation method is developed for solving time M-fractional Volterra integral equations of the second kind. Convergence of this technique is proved. The proposed method has been called M-fractional homotopy perturbation method (MFHPM). The results obtained demonstrate the efficiency of the proposed method for the local time M-fractional Volterra integral equations. Some numerical examples are presented to illustrate the proposed approach.

1. Introduction

Definition 1.1. Given a function $f : [a, \infty) \rightarrow \mathbb{R}$, $a \geq 0$. Then local M-fractional integral of f order α is defined by

$${}_M\mathcal{T}_a^{\alpha,\beta} f(t) = \Gamma(\beta + 1) \int_a^t \frac{f(x)}{x^{1-\alpha}} dx, \quad (1.1)$$

where the integral is the usual Riemann improper integral, and $\alpha \in (0, 1)$ and $\beta > 0$ [1]. Consider the local time M-fractional Volterra integral equations, as the following

$$x(t) = y(t) + \lambda {}_M\mathcal{T}_a^{\alpha,\beta} (K(t, s)x(s)), \quad \forall \alpha \in (0, 1), \beta > 0, \quad (1.2)$$

where y and K are known functions, λ and a are constant and x , is an unknown function. Applying the local time M-fractional integral definition on equation (1.2), results in

$$x(t) = y(t) + \lambda \int_a^t \frac{\Gamma(\beta + 1)K(t, s)x(s)}{s^{1-\alpha}} ds, \quad (1.3)$$

by changing the kernel as the following form

$$K^{\alpha,\beta}(t, s) = \frac{\Gamma(\beta + 1)K(t, s)}{s^{1-\alpha}}, \quad (1.4)$$

as the local time M-fractional Volterra kernel, and substituting (1.4) in (1.3), we derive

$$x(t) = y(t) + \lambda \int_a^t K^{\alpha,\beta}(t, s)x(s) ds. \quad (1.5)$$

Keywords: M-fractional integral; Time M-fractional Volterra integral equations; Homotopy perturbation method; Theorem of convergence; .

AMS Mathematical Subject Classification [2010]: 45D99-65R20-34A08.

According to equation (1.5), the operator form of MFVIEs (1.2), can be presented as follows

$$x = y + \lambda K^{\alpha, \beta} x, \quad \forall \alpha \in (0, 1), \beta > 0, \quad (1.6)$$

or

$$L_{\beta}^{\alpha} x = (I - \lambda K^{\alpha, \beta}) x = y. \quad (1.7)$$

It is obvious that the local M-fractional Volterra integral equations of the second kind (1.2), are changed into Volterra integral equations of the second kind, for $\alpha, \beta = 1$.

Definition 1.2. Let's consider $\lambda = \lambda_0$, $\alpha = \alpha_0$, $\beta = \beta_0$, and $(L_{\beta_0}^{\alpha_0})^{-1}$ as an \mathcal{L}^2 a operator, exists and satisfies

$$(L_{\beta_0}^{\alpha_0})^{-1} L_{\beta_0}^{\alpha_0} = L_{\beta_0}^{\alpha_0} (L_{\beta_0}^{\alpha_0})^{-1} = I, \quad (1.8)$$

then λ_0 is called a regular value of the local M-fractional operator K^{α_0, β_0} [2].

Theorem 1.3. If for a given $\alpha = \alpha_0$, $\beta = \beta_0$, and $\lambda = \lambda_0$, the operator $(L_{\beta_0}^{\alpha_0})^{-1}$ exists, then it is unique [2].

Proof. In a similar way to proof in reference [2, 3] proof is clear. \square

Theorem 1.4. If λ is a regular value of the local M-fractional fractional operator $K^{\alpha, \beta}$, with inverse the local M-fractional fractional operator $(L_{\beta}^{\alpha})^{-1}$, then for any L^2 function y , Eq. (1.6) has a unique \mathcal{L}^2 solution say x , satisfying see [4].

$$x = (L_{\beta}^{\alpha})^{-1} y. \quad (1.9)$$

Proof. In a similar way to proof in reference [2, 3], proof is clear. \square

2. Local M-fractional homotopy perturbation method (MFHPM)

We construct the local M-fractional homotopy perturbation (MFHPM) as follows,

$$(1 - p)[v(t, p) - y(t)] + p[v(t, p) - y(t) - \lambda_M \mathcal{T}_a^{\alpha, \beta}(K(t, s)v(s, p))] = 0, \quad \forall \alpha \in (0, 1), \beta > 0, \quad (2.1)$$

where $p \in [0, 1]$ is an embedding parameter. We assume that solution of (2.1) is as the following

$$v(t, p) = \sum_{n=0}^{\infty} v_n(t) p^n = v_0(t) + v_1(t)p + v_2(t)p^2 + v_3(t)p^3 + \dots \quad (2.2)$$

Substitution of (2.2) into Eq. (2.1), we drive

$$(1 - p) \left[\sum_{n=0}^{\infty} v_n(t) p^n - y(t) \right] + p \left[\sum_{n=0}^{\infty} v_n(t) p^n - y(t) - \lambda_M \mathcal{T}_a^{\alpha, \beta}(K(t, s) \sum_{n=0}^{\infty} v_n(s) p^n) \right] = 0. \quad (2.3)$$

Collecting terms of the same powers p in (2.3), we obtain

$$p^0 : v_0(t) = y(t), \quad (2.4)$$

$$p^1 : v_1(t) = \lambda_M \mathcal{T}_a^{\alpha, \beta}(k(t, s)y(s)) = (\lambda K^{\alpha, \beta} y)(t),$$

$$p^2 : v_2(t) = \lambda^2_M \mathcal{T}_a^{\alpha, \beta}(k(t, s)_M \mathcal{T}_a^{\alpha, \beta}(k(s, s_1)y(s_1))) = \left((\lambda K^{\alpha, \beta})^2 y \right)(t),$$

\vdots

Solving Eqs. (2.4) lead to, a solution of the local M-fractional Volterra integral equations (1.2) as the following

$$x(t) = \lim_{p \rightarrow 1} v(t, p) = v_0(t) + v_1(t) + v_2(t) + \cdots = y(t) + \sum_{n=1}^{\infty} ((\lambda K^{\alpha, \beta})^n y)(t). \quad (2.5)$$

The operator form of Eq. (2.5), that is the solution of the local M-fractional Volterra integral equation (1.2), can be as follows

$$x = y + \sum_{n=1}^{\infty} (\lambda K^{\alpha, \beta})^n y, \quad (2.6)$$

where Eq. (2.6), is called the local M-fractional series for the solution x , of MFVIEs (1.2). And also, the n -th approximate solution of MFVIE (1.2), can be as follows

$$x^n(t) = y(t) + \sum_{i=1}^n ((\lambda K^{\alpha, \beta})^i y)(t).$$

3. The convergence of method

Theorem 3.1. (sufficient condition of convergence) The local M-fractional series (2.3), for $(L_{\beta}^{\alpha})^{-1}$ converges strongly if $\|\lambda K^{\alpha, \beta}\| < 1$, for a given $\alpha \in (0, 1)$ and $\beta > 0$.

Proof. In a similar way to proof in reference [2, 3], proof is clear. \square

Lemma 3.2. $K^{\alpha, \beta}$, is a \mathcal{L}^2 local M-fractional Volterra operator for a given α, β and $b > a$, then

$$\left| \left(K^{\alpha, \beta} \right)^{n+1} (t, s) \right| \leq \frac{\|K^{\alpha, \beta}\|_E^{n+1}}{[(n-1)!]^{\frac{1}{2}}} K_1^{\alpha, \beta} (t) K_2^{\alpha, \beta} (s),$$

where $K_1^{\alpha, \beta} (t) = \left[\int_a^s |K^{\alpha, \beta} (t, s)|^2 ds \right]^{\frac{1}{2}}$, and $K_2^{\alpha, \beta} (s) = \left[\int_t^b |K^{\alpha, \beta} (t, s)|^2 dt \right]^{\frac{1}{2}}$.

Proof. For $\alpha, \beta = 1$, refer [4]. \square

Theorem 3.3. If $K^{\alpha, \beta}$, is a \mathcal{L}^2 local M-fractional Volterra operator for a given α, β the series (1.9), converges strongly for all λ , to the inverse the local M-fractional operator of $K^{\alpha, \beta}$.

Proof. In a similar way to proof in reference [2, 3], proof is clear. \square

Example 3.4. Consider the following the local M-fractional Volterra integral equation

$$x(t) = 2 + t^2 + {}_M \mathcal{T}_0^{\alpha, \beta} ((t-s)x(s)), \quad \forall \alpha \in (0, 1), \beta > 0. \quad (3.1)$$

Where for $\alpha = 1$ and $\beta = 1$, the exact solution of MFVIE (3.1), is as follows

$$x(t) = 4 \cosh(t) - 2.$$

By the proposed M-fractional HPM approach, we read

$$\begin{aligned} x(t) = & 2 + t^2 + \frac{t^{(\alpha+1)} (t^2 \alpha^2 + t^2 \alpha + 2 \alpha^2 + 10 \alpha + 12) \Gamma(\beta + 1)}{\alpha (\alpha + 1) (\alpha + 2) (\alpha + 3)} \\ & + \frac{t^{(2\alpha+2)} (2t^2 \alpha^4 + 5t^2 \alpha^3 + 4t^2 \alpha^2 + 4 \alpha^4 + t^2 \alpha + 34 \alpha^3 + 106 \alpha^2 + 144 \alpha + 72) (\Gamma(\beta + 1))^2}{2 \alpha (\alpha + 1)^2 (2 \alpha + 1) (\alpha + 2)^2 (\alpha + 3) (2 \alpha + 3)} + \cdots. \end{aligned}$$

$$\lim_{\alpha, \beta \rightarrow 1} x(t) = 2.000000000 + 2.000000000t^2 + 0.166666667t^4 + \dots$$

According to Taylor expansion of $x(t)$, clearly has seen that

$$\lim_{\alpha, \beta \rightarrow 1} x(t) = 4 \cosh(t) - 2.$$

This solution is the same exact solution of this MFVIE (3.1), for non-fractional case. In Figures 1, the seventh-order approximate solution of Local M-fractional Volterra integral equation for different Values α, β , and exact solution for $\alpha, \beta = 1$ are plotted.

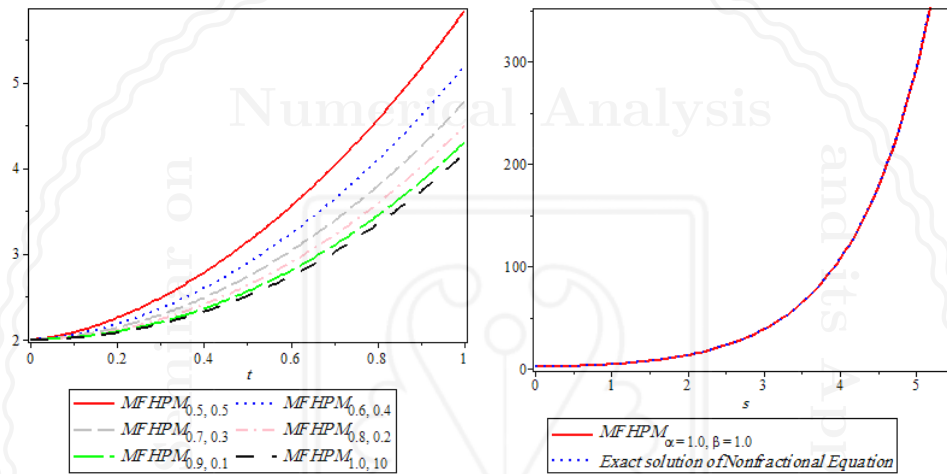


Figure 1: The 7th-order approximation of MFHPM for different Values α, β , and for $\alpha, \beta = 1$, versus exact solution of Non-fractional Volterra integral equation.

4. Conclusion

According to this study, the Volterra integral equation of the second kind is developed for the local M-fractional integral equations and homotopy perturbation method is presented for local M-fractional Volterra integral equations. The proposed method has been called the local M-fractional homotopy perturbation method. It was successfully utilized to find an exact or approximate solution of MFVIEs. Since for $\alpha, \beta = 1$, MFHPM method is changed into the Neumann method, thus not unexpected that local M-fractional HPM method has the same accuracy and efficiency as Neumann method for Volterra integral equations of the second kind [4]. The results have confirmed this fact. In this study, the norm $\| \cdot \|_2$ is used.

References

- [1] J.V.C. Sousa, E.C.D. Oliveira, A new truncated M-fractional derivative type unifying some fractional derivative types with classical properties, *International Journal of Analysis and Applications*, 16 (2018) 83-96.
- [2] L.M. Delves, J.L. Mohamed, *Computational methods for integral equations*, Cambridge University Press, 1985.

- [3] M. Ilie, J. Biazar, Z. Ayati, Neumann method for solving conformable fractional Volterra integral equations, *Computational Methods for Differential Equations*, 8 (2020) 54-68.
- [4] M. Mohammad, A. Trounev, Fractional nonlinear Volterra–Fredholm integral equations involving Atangana–Baleanu fractional derivative: framelet applications. *Advances in Difference Equations*, (2020) no. 618.

Mousa Ilie

Department of Mathematics, Rasht Branch, Islamic Azad University, Rasht, Iran
email address: ilie@iaurasht.ac.ir

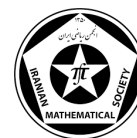
Ali Khoshkenar

Department of Mathematics, Rasht Branch, Islamic Azad University, Rasht, Iran
email address: mousailie52@gmail.com





University of Guilan



A fast numerical method for fractional ordinary differential equations

Roghayeh Katani

Abstract. In this work, we propose and analyze an efficient numerical method for solving Caputo fractional differential equations (FDE) with smooth solutions. Properties of the Caputo derivative allow us to reduce the FDE into a weakly singular Volterra integral equation and then a numerical scheme is developed for solving this Volterra type integral equation. The order of convergence of the numerical method is $O(h^5)$. A numerical example is given to show that the numerical results are consistent with the theoretical results.

1. Introduction

We consider a numerical method for solving the fractional differential equation

$$\begin{aligned} D_0^\alpha y(t) &= f(t, y(t)), & 0 < t < T, \\ y^{(k)}(0) &= y_0^{(k)}, & k = 0, 1, 2, \dots, [\alpha] - 1, \end{aligned} \quad (1.1)$$

where the $y_0^{(k)}$ may be arbitrary real numbers and $\alpha > 0$. Here D_0^α denotes the differential operator in the sense of Caputo denoted by

$$D_0^\alpha y(t) = \frac{1}{\Gamma(n - \alpha)} \int_0^t (t - u)^{n - \alpha - 1} y^{(n)}(u) du, \quad (1.2)$$

where $n = [\alpha]$ is the smallest integer α .

Existence and uniqueness of solution for (1.1) have been studied [1, 2, 7]. Numerical methods for solving fractional differential equations have been considered by many authors for example see [3–6]. In this paper by using properties of the Caputo derivative we reduce the FDE into a weakly singular Volterra integral equation. Once this is done, a number of numerical schemes developed for Volterra type integral equation can be applied to find numerical solution of FDEs. For this purpose the total time is divided into a set of small intervals, and between each interval the unknown function is approximated using quadrature rules and product integration. These approximations are substituted into the transformed Volterra type equation to obtain a set of algebraic system of equations. Solution of these systems provides the solution of the FDE.

Keywords: Fractional differential equations, Caputo differential operator, weakly singular integral equations.

AMS Mathematical Subject Classification [2010]: 65R20.

2. The numerical algorithm

It is well-known ([2]) that the initial value problem described by (1.1) is equivalent to the Volterra integral equation

$$y(t) = \sum_{k=0}^{\lceil \alpha \rceil - 1} y_0^{(k)} \frac{t^k}{k!} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-u)^{\alpha-1} f(u, y(u)) du, \quad 0 < t < T. \quad (2.1)$$

In order to explain the numerical method, divide the time T into N equal parts, and let $h = T/N$ be the time interval of each part. The time at the grid points are given as $t_j = jh$, $j = 0, 1, \dots, N$. For given real numbers c_j with $0 = c_0 < c_1 < \dots < c_4 = 1$, define the set $\Pi_n := \{t_{n,j}\}$ of mesh points by $t_{n,j} := t_n + c_j h$, $j = 0, 1, \dots, 4$, $n = 0, 1, \dots, N-1$ where $c_j = \frac{j}{4}$. For simplicity we will use the following notations $y(t_{n,j}) \approx y_{n,j}$ and $f(t_{n,j}, y(t_{n,j})) \approx F_{n,j}$. Discretization at each node points $t_{n,j}$ leads to

$$\begin{aligned} y(t_{n,j}) &= \sum_{k=0}^{\lceil \alpha \rceil - 1} y_0^{(k)} \frac{t_{n,j}^k}{k!} + \frac{1}{\Gamma(\alpha)} \int_0^{t_{n,j}} (t_{n,j} - u)^{\alpha-1} f(u, y(u)) du \\ &= \sum_{k=0}^{\lceil \alpha \rceil - 1} y_0^{(k)} \frac{t_{n,j}^k}{k!} + \frac{1}{\Gamma(\alpha)} \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} (t_{n,j} - u)^{\alpha-1} f(u, y(u)) du \\ &\quad + \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n,j}} (t_{n,j} - u)^{\alpha-1} f(u, y(u)) du. \end{aligned} \quad (2.2)$$

The first integral in above relation is approximated by using the two step Romberg quadrature rule, then we can write

$$\int_{t_i}^{t_{i+1}} (t_{n,j} - u)^{\alpha-1} f(u, y(u)) du \approx h \sum_{i'=0}^4 w_{i'} (t_{n,j} - t_{i,i'})^{\alpha-1} F_{i,i'}, \quad (2.3)$$

where $w_0 = w_4 = 7/90$, $w_2 = 2/15$, $w_1 = w_3 = 16/45$. The Romberg quadrature rule can not be used for the second integral and we use product integration method, then we have

$$\int_{t_n}^{t_{n,j}} (t_{n,j} - u)^{\alpha-1} f(u, y(u)) du \approx \sum_{i'=0}^4 F_{n,i'} \int_{t_n}^{t_{n,j}} (t_{n,j} - u)^{\alpha-1} l_{i'}(u) du, \quad (2.4)$$

where

$$l_{i'}(u) = \prod_{j'=0, j' \neq i'}^4 \frac{u - t_{n,j'}}{t_{n,i'} - t_{n,j'}}.$$

The right side integral in relation (2.4) can be calculated exactly, then substituting the approximations (2.3) and (2.4) in equation (2.2) yields

$$\begin{aligned} y_{n,j} - \sum_{k=0}^{\lceil \alpha \rceil - 1} y_0^{(k)} \frac{t_{n,j}^k}{k!} - \frac{1}{\Gamma(\alpha)} \sum_{i=0}^{n-1} h \sum_{i'=0}^4 w_{i'} (t_{n,j} - t_{i,i'})^{\alpha-1} F_{i,i'} \\ - \frac{1}{\Gamma(\alpha)} \sum_{i'=0}^4 F_{n,i'} \int_{t_n}^{t_{n,j}} (t_{n,j} - u)^{\alpha-1} l_{i'}(u) du = 0, \quad n = 0, 1, \dots, N-1, j = 1, \dots, 4. \end{aligned} \quad (2.5)$$

Equations (2.5) then gives rise to a system of 4 algebraic equations which for nonlinear cases, can be solved by using iterative methods such as Newton.

3. Convergence analysis

Theorem 3.1. Assume that $y(t_{n,j})$ is the exact solution of the fractional differential equation (1.1) and $y_{n,j}$ be the approximate solution obtained by the method of this paper, then for sufficiently small h , we have

$$\|y(t_{n,j}) - y_{n,j}\|_{\infty} \rightarrow 0,$$

and the order of convergence is $O(h^5)$ when the function f has at least six order continuous derivative.

Proof. The results can be obtained by using a generalized discrete Gronwall lemma. \square

4. Numerical results

In this section, in order to test experimentally the convergence of the proposed method we consider a test problem.

Example 4.1. Consider the initial value problem

$$D_0^{1/2}y(t) = t^2 - y(t) + t + \frac{2}{\sqrt{\pi}}\sqrt{t} + \frac{8}{3\sqrt{\pi}}t^{3/2}, \quad t \in [0, 1],$$

$$y(0) = 0,$$

with exact solution $y(t) = t^2 + t$.

Table 1 displays the absolute error and ratios of the errors for $h = 0.2$ and $h = 0.1$ at certain mesh points. We observe that the ratios are more than $2^5 = 32$ that indicating the theoretical convergence order of at least h^5 .

Table 1: Numerical results of example 4.1.

t	$N = 5$	$N = 10$	Ratio
0.1	1.4806e-06	4.2097e-8	35.17
0.2	4.1272e-06	8.9359e-8	47.79
0.3	3.9489e-06	9.3511e-08	42.22
0.4	4.3430e-06	1.2323e-07	35.24
0.5	6.2515e-06	1.5985e-07	39.11
0.6	1.7405e-05	5.1696e-07	33.66
0.7	2.3539e-05	6.3365e-07	37.15
0.8	5.8393e-05	1.6711e-06	34.94
0.9	6.3477e-05	2.0168e-06	31.47
1	6.5160e-05	1.9275e-06	33.81

References

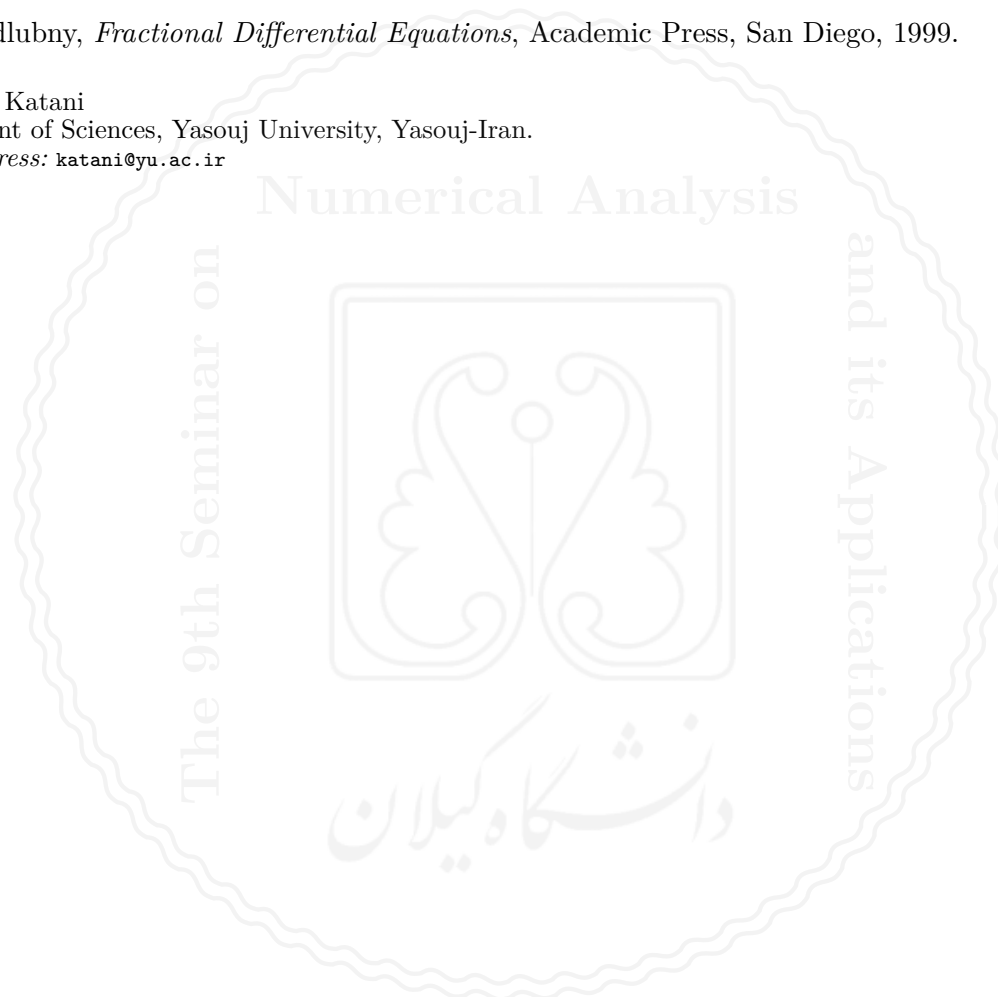
- [1] K. Diethelm, *The Analysis of Fractional Differential Equations, An Application-Oriented Using Differential Operators of Caputo Type*, Lecture Notes in Mathematics, Vol. 2004, Springer, 2010.
- [2] K. Diethelm, N. J. Ford, Analysis of fractional differential equations, *Journal of Mathematical Analysis and Applications*, 265 (2002) 229–248.
- [3] K. Diethelm, N. J. Ford, A.D. Freed, Detailed error analysis for a fractional Adams method, *Numerical Algorithms*, 36 (2004) 31–52.

- [4] K. Diethelm, N. J. Ford, A. D. Freed, A predictor-corrector approach for the numerical solution of fractional differential equations, *Nonlinear Dynamics*, 29 (2002) 3–22.
- [5] K. Diethelm, Y. Luchko, Numerical solution of linear multi-term initial value problems of fractional order, *Journal of Computational Analysis and Applications* 6 (2004) 243–263.
- [6] N. J. Ford, M. L. Morgado, M. Rebelo, A nonpolynomial collocation method for fractional terminal value problems, *Journal of Computational Analysis and Applications*, 275 (2015) 392–402.
- [7] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, 1999.

Roghayeh Katani

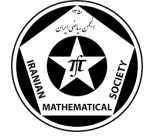
Department of Sciences, Yasouj University, Yasouj-Iran.

email address: katani@yu.ac.ir





University of Guilan



An interpolation-based method for the numerical solution of multi-point boundary value problems

Nasibeh Karamollahi, Mohammad Heydari, Ghasem Barid Loghmani

Abstract. Two-point Taylor expansion as a certain case of Hermite interpolant is utilized to approximate the solution of multi-point boundary value problems. The properties of this interpolant, in addition to the use of differential equation under consideration and its boundary conditions help to reduce the computation of the solution to some algebraic equations.

1. Introduction

Multi-point boundary value problems (MBVPs) appear in the modeling of many situations in science and engineering (see [1] and references therein). If a system is modeled by different differential equations over subintervals of the main domain, then the solution should satisfy some conditions at the boundaries of these subintervals. The result of such a model is a MBVP. Consider the second order differential equation

$$y''(x) = f(x, y(x), y'(x)), \quad 0 \leq x \leq 1, \quad (1.1)$$

subject to one of the multi-point boundary conditions

$$y(0) = \sum_{i=1}^m \alpha_i y(\xi_i) + \lambda_0, \quad y(1) = \sum_{i=1}^m \beta_i y(\xi_i) + \lambda_1, \quad (1.2)$$

$$y'(0) = \sum_{i=1}^m \alpha_i y'(\xi_i) + \lambda_0, \quad y'(1) = \sum_{i=1}^m \beta_i y'(\xi_i) + \lambda_1, \quad (1.3)$$

$$y(0) = \sum_{i=1}^m \alpha_i y(\xi_i) + \lambda_0, \quad y'(1) = \sum_{i=1}^m \beta_i y'(\xi_i) + \lambda_1, \quad (1.4)$$

$$y'(0) = \sum_{i=1}^m \alpha_i y'(\xi_i) + \lambda_0, \quad y'(1) = \sum_{i=1}^m \beta_i y'(\xi_i) + \lambda_1, \quad (1.5)$$

where $\alpha_i \in [0, 1)$, $\beta_i \in [0, 1)$, $\xi_i \in (0, 1)$, λ_0 and λ_1 are constants. Also suppose that f in (1.1) satisfies the sufficient conditions to guarantee the existence and uniqueness of the solution of the problem. It should be noted that multi-point boundary conditions are not limited to conditions (1.2)-(1.5).

The aim of this paper is to propose an efficient numerical method to approximate the solution of second order differential equation (1.1) with certain multi-point boundary conditions of type

Keywords: Two-point Hermite interpolation, Multi-point boundary value problems.

AMS Mathematical Subject Classification [2010]: 65D05, 34B10.

(1.2)-(1.5). For this purpose, a particular case of Hermite interpolation method, namely two-point Hermite interpolation or two-point Taylor formula (TTF) is considered. This interpolation method utilizes the values of a function and its derivatives up to an adequate order at the endpoints of the domain to approximate the function. The needed data to construct the TTF approximation of the solution of a multi-point boundary value problem can be extracted by using the structure of the differential equation and its related boundary conditions.

2. The numerical method

The base of the numerical method proposed in this article, to approximate the solution of MBVPs, is an explicit form of the two-point Hermite interpolant. This explicit form is presented in the following theorem:

Theorem 2.1. [2] Suppose that $y \in \mathcal{C}^{2n}[0, 1]$, then y can be approximated using the polynomial interpolant

$$P_{2n-1}(y; x) := \sum_{i=0}^{n-1} \left(y^{(i)}(0) C_{n,i}(x) + (-1)^i y^{(i)}(1) C_{n,i}(1-x) \right), \quad (2.1)$$

where the basis functions $C_{n,i}$ are defined by

$$C_{n,i}(x) := \frac{x^i}{i!} (1-x)^n \sum_{j=0}^{n-i-1} \binom{n+j-1}{j} x^j, \quad i = 0, 1, \dots, n-1.$$

Furthermore, the error term of the approximation can be computed as

$$R_n(y; x) := y(x) - P_{2n-1}(y; x) = \frac{y^{(2n)}(\xi_x)}{(2n)!} x^n (1-x)^n, \quad \xi_x \in (0, 1).$$

Consider the second order differential equation (1.1) subject to multi-point boundary conditions (1.2). We present the method for this case and the other cases can be described in the same manner. If the values of a function and its first $n-1$ derivatives at $x=0$ and $x=1$ are available, the TTF approximation to this function can be constructed using (2.1). However, these values are unknown when we try to approximate the unknown solution of problem (1.1)-(1.2). Therefore, we intend to find a suitable way to estimate them. The available tools to achieve this goal are the differential equation under consideration and its related boundary conditions. Indeed, we keep $y(0)$, $y(1)$, $y'(0)$ and $y'(1)$ as unknowns and utilize (1.1) to compute $y''(0)$ and $y''(1)$ in this unknowns as

$$y''(0) = f(0, y(0), y'(0)), \quad y''(1) = f(1, y(1), y'(1)). \quad (2.2)$$

Moreover, differentiating (1.1) and utilizing (2.2) help to calculate $y'''(0)$ and $y'''(1)$ in terms of the same unknowns as before. The same thing happens by two times differentiating (1.1) to calculate $y^{(4)}(0)$ and $y^{(4)}(1)$. This process can be continued to calculate all the needed derivatives to obtain (2.1). Indeed, for $i = 2, 3, \dots, n-1$, all the values of $y^{(i)}(0)$ can be computed in two unknown values $y(0)$ and $y'(0)$, and $y^{(i)}(1)$ can be obtained in unknowns $y(1)$ and $y'(1)$. Therefore, we should construct a system of four algebraic equations and four unknowns $y(0)$, $y(1)$, $y'(0)$ and $y'(1)$ to attain the needed data to construct (2.1) as the approximate solution of (1.1)-(1.2). To this end, we can first substitute the approximation $P_{2n-1}(y; x)$ into the boundary conditions of the problem

to get

$$\begin{cases} y(0) = \sum_{i=1}^m \alpha_i P_{2n-1}(y; \xi_i) + \lambda_0, \\ y(1) = \sum_{i=1}^m \beta_i P_{2n-1}(y; \xi_i) + \lambda_1, \end{cases} \quad (2.3)$$

and then employ the residual function based on differential equation (1.1) which is defined by

$$\text{Res}(x) = P''_{2n-1}(y; x) - f(x, P_{2n-1}(y; x), P'_{2n-1}(y; x)),$$

and compose the following equations

$$\begin{cases} \text{Res}(x_1) = 0, \\ \text{Res}(x_2) = 0, \end{cases} \quad (2.4)$$

where $x_1, x_2 \in (0, 1)$ are two arbitrary points. The solution of (2.3)-(2.4) as a system of four equations and four unknowns provides $\tilde{y}(0)$, $\tilde{y}(1)$, $\tilde{y}'(0)$ and $\tilde{y}'(1)$ as the estimation of unknown values $y(0)$, $y(1)$, $y'(0)$ and $y'(1)$, respectively. The obtained estimations can be used to approximate the higher order derivatives of y at $x = 0$ and $x = 1$. Thus, the needed data to construct the TTF approximation (2.1) are available and the approximate solution of (1.1)-(1.2) can be achieved.

3. An application

The models of most small size bridges include two support points (left hand-side of Figure 1). These models can be formulated using a standard two-point boundary value problem. However, the model of a large size bridge can be formulated by a MBVP, if it is arranged using more than two supports (right hand-side of Figure 1) [3]. Two different types of conditions is possible near each endpoint of the bridge. When the position of the bridge at the supporting points near $x = 0$ and $x = 1$ is important for the designer, the suitable boundary conditions are (1.2). However, boundary conditions (1.5) describe the situation in which the designer tries to control the angles of the bridge at the supporting points near $x = 0$ and $x = 1$. Furthermore, boundary conditions (1.3) and (1.4) demonstrate cases that the designer does not take the same approach at the endpoints of the bridge. As an example, consider the MBVP [3]

$$\begin{cases} y''(x) + (x^3 + x + 1)(y(x))^2 = g(x), \\ y(0) = \frac{1}{6}y\left(\frac{2}{9}\right) + \frac{1}{3}y\left(\frac{7}{9}\right) - \frac{1}{6}\sin\left(\frac{14}{81}\right), \\ y(1) = \frac{1}{5}y\left(\frac{2}{9}\right) + \frac{1}{2}y\left(\frac{7}{9}\right) - \frac{7}{30}\sin\left(\frac{14}{81}\right), \end{cases} \quad (3.1)$$

where g is calculated such that the exact solution of the problem is $y(x) = \frac{1}{3}\sin(x - x^2)$.

The proposed method is applied to approximate the solution of (3.1). All the results are obtained by using Maple software on a Core(TM) i7 PC with 3.60 GHz of CPU and 8 GB of RAM. The maximum absolute error (E_∞) of the proposed TTF taking $(x_1, x_2) = (\frac{2}{9}, \frac{7}{9})$ for different values of n , and the used CPU time for each case are presented in Table 1. This table indicates the capability and acceptable accuracy of the TTF for solving problem (3.1). The presented TTF can be applied to approximate the solution of other practical MBVPs.

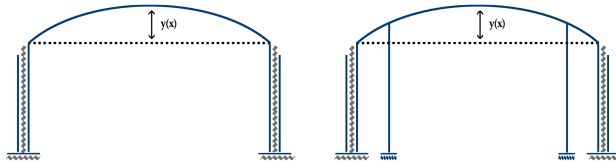


Figure 1: Simple models of bridges.

n	5	10	15
E_∞	3.59e-05	8.02e-14	2.22e-20
CPU time (s)	0.110	0.593	0.656

Table 1: Results for problem (3.1).

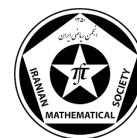
References

- [1] A. Saadatmandi, M. Dehghan, The use of Sinc-collocation method for solving multi-point boundary value problems, *Communications in Nonlinear Science and Numerical Simulation*, 17 (2012) 593–601.
- [2] G.M. Phillips, Explicit forms for certain Hermite approximations, *BIT Numerical Mathematics*, 13 (1973) 177–180.
- [3] F. Geng, M. Cui, Multi-point boundary value problem for optimal bridge design, *International Journal of Computer Mathematics*, 87 (2010) 1051–1056.

Nasibeh Karamollahi
 Department of Mathematics, Yazd University, Yazd, Iran
 email address: karamollahi.n@stu.yazd.ac.ir

Mohammad Heydari
 Department of Mathematics, Yazd University, Yazd, Iran
 email address: m.heydari@yazd.ac.ir

Ghasem Barid Loghmani
 Department of Mathematics, Yazd University, Yazd, Iran
 email address: loghmani@yazd.ac.ir



Numerical analysis application in virtual teaching with EJS software

Maedeh GholamAzad, Amir Pishkoo

Abstract. As a result of social distancing measures implemented around the world in the aftermath of the new coronavirus (COVID19) crisis, virtual educations have been forced to adjust their teaching patterns, potentially shifting from traditional in-person teaching to online education or virtual teaching (VT). A variety of methodologies and algorithms can be used to create VT. One of the most important is numerical analysis. Numerical analysis is a branch of mathematics and computer science concerned with developing, analyzing, and implementing numerical solutions to continuous mathematics problems. This article provides methodologies and approaches for designing the VT using various numerical analytic algorithms and the Easy Java Simulation (EJS) software.

1. Introduction

The new coronavirus (COVID19) epidemic has resulted in a significant reduction in student and intern operational opportunities. Science and technology have advanced and innovated as a result of this topic, and VT technology is evolving as time demands [1]. VT systems can be designed and manufactured using a variety of designs and methods. For simulation and improved understanding, EJS software is one of the most useful. The implementation language is EJS, which is one of the most widely used programming languages today [2]. EJS is an open-source program that allows you to create various simulations by inputting model equations and creating a graphical user interface (GUI) for the program [3]. Due to the COVID19 pandemic, several colleges have incorporated this type of tool to enhance their virtual courses in recent years. Face-to-face laboratory practice is typically not available to students in scientific and technical subjects due to the distant learning paradigm. As a result, distance learning processes require the usage of internet tools such as VT [4]. Runge-Kutta (4), Euler, Euler-Richardson, and other key solvers derived from the numerical analysis algorithm are included in EJS. The application of numerical analysis for the design of the VT utilizing the EJS program is shown in this paper. To begin, a brief overview of the EJS environment is provided. Following that, the EJS creates a simple physics example. The creation of a simple pendulum utilizing the Runge-Kutta(4) is the subject of this section's research. The following is how the paper is structured: Section 2 discusses the current state of EJS and how to create EJS apps using it. EJS's creation of the simple pendulum is discussed in Section 3.

2. About Easy Java Simulations

EJS is a free Java authoring tool that assists non-programmers in creating interactive simulations in Java or JavaScript, primarily for educational or learning reasons, and is a component of the Open

Keywords: Virtual Education, Numerical analysis, Algorithm, Easy Java Simulation.

AMS Mathematical Subject Classification [2010]: 65Sxx, 65S05, 65S99.

Source Physics project [4]. Despite its user-friendly interface, EJS contains all of the capabilities required for a full modeling cycle. Figure 1 depicts the main EJS environment.

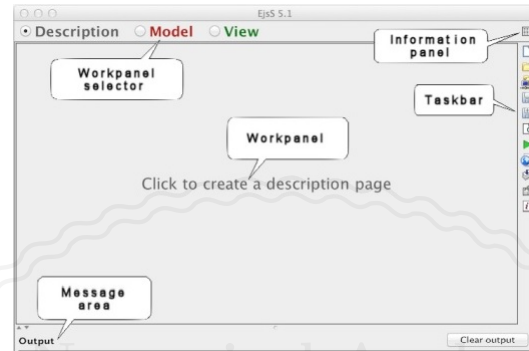


Figure 1: The EJS user interface with annotations.

The right-hand taskbar has icons for clearing, opening, searching, and saving files, configuring EJS, and displaying application information and help. It also has icons for running simulations and packaging one or more simulations into a single file. Right-clicking on taskbar icons brings up additional (but related) options that will be explained when needed.

EJS displays informational messages in the output section at the bottom of the interface. The work panels, which are located in the center of the interface, are where the modeling is done.

3. Designing a simple pendulum

The model is defined in the Model work panel so that EJS can transform it into a program. We investigate the motion of a basic pendulum in this simulation. You can alter the pendulum bob's mass, the length of the string, the acceleration due to gravity, and the pendulum's initial position. The pendulum's motion as a function of time can then be observed.

Identifying, defining, and initializing the variables that represent the system is a good first step when constructing a model.

We will use the Evolution panel frequently for models that are not based on ordinary differential equations since it allows us to build Java code that determines what changes are occurring in time (ODEs). The View is the third EJS work panel. This work panel enables us to develop a graphical interface with minimal scripting that integrates visualization, user interaction, and program control. The properties of image (the ODE with Runge-Kutta (4) solver) and the final simulation for the simple pendulum created by the View work panel are shown in Figures 2 and 3, respectively.

It's time to start the simulation by pressing the Run button on the taskbar. EJS creates and compiles Java code, gathers auxiliary and library files, and runs the produced program. It's all done with a single mouse click. To ensure that the model is in a consistent state, running a simulation initializes its variables and executes the fixed relations. When the user interface's play/pause button is pushed, the model's time evolution begins. (When the simulation is paused, the play/pause button displays the icon; when it is running, it displays the icon.) The software in our present example uses a numerical method to advance the harmonic oscillator differential equation by 0.05 time units before running the fixed relations code.

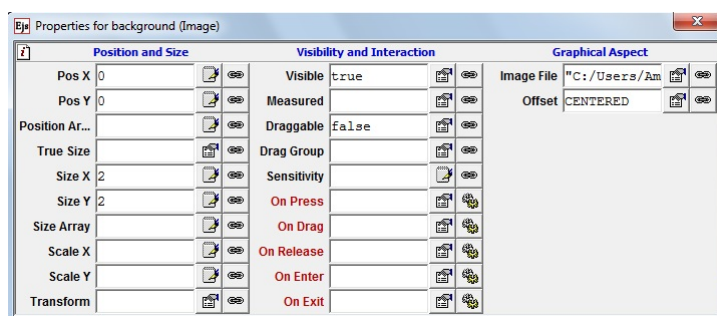


Figure 2: The properties of image.

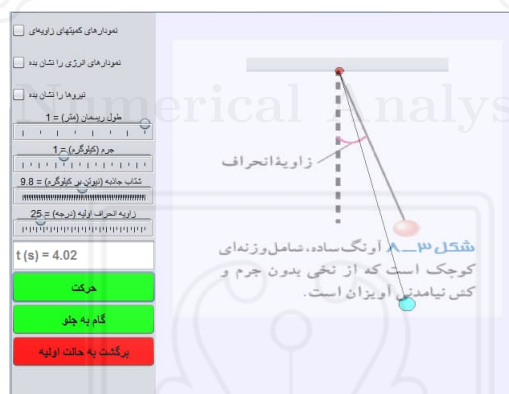


Figure 3: Final simulation of the simple pendulum.

References

- [1] J.C. Chan, T.K. Waddell, K. Yasufuku, S. Keshavjee, L.L. Donahoe, Maintaining technical proficiency in senior surgical fellows during the COVID-19 pandemic through virtual teaching. *Journal of Thoracic and Cardiovascular Surgery Open*, 8 (2021) 679–687.
- [2] G. Farias, K.E. Arzen, S. Dormido, F. Esquembre, Java Simulations of Embedded Control Systems, *Sensors* 10 (2010) 8585–8603.
- [3] J. Saenz, F. Esquembre, F.J. Garcia, LD.L.Torre, S. Dormido, An Architecture to use Easy Java-Javascript Simulations in New Devices. *IFAC-PapersOnLine series*, 48 (2015) 129–133.
- [4] W. Christian, F. Esquembre; Easy Java/Javascript Simulations Manual. Version 3.4. September (2005).

Maedeh GholamAzad

Department of Applied Mathematics, Faculty of Science, Azarbaijan Shahid Madani University, Tabriz, Iran
email address: m.gholamazad@azaruniv.ac.ir

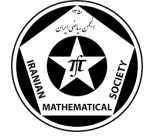
Amir Pishkoo

Physics and Accelerators Research School, Nuclear Science and Technology Research Institute (NSTRI), Tehran, Iran

email address: apishkoo@gmail.com



University of Guilan



A new numerical algorithm for the fractional model of human liver with new modified parameters

Hamed Ebrahimi, Jafar Biazar

Abstract. This paper investigates the dynamics of a fractional-order model of the human liver. The proposed model is examined via quasi-hat functions (QHF). Utilizing a method that incorporates the operational matrices of QHF is used to reduce the problem to several systems of two equations with two unknowns. Finally, an illustrative example is provided to confirm the accuracy and validity of the proposed scheme. We have studied the stability and convergence of the method for this system. However, these aspects are not covered here.

1. Introduction

A phthalein dye called bromsulphthalein (BSP) is used in liver function tests. Determining the rate of removal of the dye from the bloodstream gives a measure of liver function. [1]. In 2004, an integer-order model of human liver was studied and solved by Celechovska [2]. The fractional mathematical model of the dynamic behavior of liver function (couple system) with new parameters is presented as follows:

$$\begin{cases} {}^C_0 D_t^\alpha u_1(t) = -\delta^{\sin(\pi\alpha/2)} u_1(t) + \varepsilon^{\sin(\pi\alpha/2)} u_2(t), \\ {}^C_0 D_t^\alpha u_2(t) = \delta^{\sin(\pi\alpha/2)} u_1(t) - (\varepsilon^{\sin(\pi\alpha/2)} + \varphi^{\sin(\pi\alpha/2)}) u_2(t), \end{cases} \quad 0 < \alpha \leq 1, t \in [0, T], \quad (1.1)$$

with initial conditions $u_1(0) = \lambda$, $u_2(0) = 0$, and ${}^C_0 D_t^\alpha$ is the Caputo fractional operator [1]. Figure 1 illustrates the flow of BSP between the blood $u_1(t)$, liver $u_2(t)$, and bile. Normal liver function; not more than 5% of the dye should remain in the blood at the end of 45 min.

1.1. Definition of QHFs

These functions are established based on the idea of the hat functions [3]. Quasi-hat functions are defined as follows for i even, and $0 \leq i \leq n$:

$$\phi_i(t) = \begin{cases} \frac{1}{2h^2}(t - (i+1)h)(t - (i+2)h), & ih \leq t < (i+2)h, \\ 0, & \text{otherwise,} \end{cases} \quad (1.2)$$

when i is odd, and $1 \leq i \leq n-1$:

$$\phi_i(t) = \begin{cases} -\frac{1}{2h^2}(t - (i-1)h)(t - (i+2)h), & (i-1)h \leq t < (i+1)h, \\ 0, & \text{otherwise,} \end{cases} \quad (1.3)$$

Keywords: Numerical algorithms; Fractional modeling; Human liver; Fractional operational matrix.

AMS Mathematical Subject Classification [2010]: 65Yxx, 37N25, 65D15.

wherein $n \geq 2$ is an even positive integer, $h = \frac{T}{n}$. An arbitrary function $u_v(t)$, can be approximated by a linear combination of QHFs as the following:

$$\begin{aligned} u_v(t) &\simeq \sum_{i=0}^n a_{vi} \phi_i(t) = A_v^T \Phi(t), \quad \Phi(t) = [\phi_0(t), \phi_1(t), \dots, \phi_n(t)]^T, \\ A_v &= [a_{v0}, a_{v1}, \dots, a_{vn}]^T, \quad a_{vi} = u_v(ih), \quad i = 0, \dots, n, \quad v = 1, 2. \end{aligned} \quad (1.4)$$

1.2. Fractional order integral operator

Here, the Riemann-Liouville integral operator of order α is expressed, as well as its property [3]:

$$I_t^\alpha u(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} u(\tau) d\tau, \quad I_t^\alpha ({}_C D_t^\alpha u(t)) = u(t) - \sum_{i=0}^{n-1} u^{(i)}(0) \frac{t^i}{i!}, \quad n-1 < \alpha \leq n.$$

Theorem 1.1. Let $u_v(t)$ and $\Phi(t)$ be given by (1.4) and $\alpha > 0$, then

$$I_t^\alpha \Phi(t) \simeq Q^\alpha \Phi(t), \quad I_t^\alpha u_v(t) \simeq A_v^T Q^\alpha \Phi(t), \quad (1.5)$$

where Q^α is the $(n+1) \times (n+1)$ operational matrix of fractional integration of order α in the Riemann-Liouville integral sense as follows:

$$Q^{(\alpha)} = \frac{h^\alpha}{2\Gamma(\alpha+3)} \begin{pmatrix} 0 & \rho_1 & \rho_2 & \rho_3 & \rho_4 & \dots & \rho_{n-1} & \rho_n \\ 0 & \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \dots & \sigma_{n-1} & \sigma_n \\ 0 & 0 & 0 & \rho_1 & \rho_2 & \dots & \rho_{n-3} & \rho_{n-2} \\ 0 & 0 & 0 & \sigma_1 & \sigma_2 & \dots & \sigma_{n-3} & \sigma_{n-2} \\ 0 & 0 & 0 & 0 & 0 & \dots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & \rho_1 & \rho_2 \\ 0 & 0 & 0 & 0 & 0 & \dots & \sigma_1 & \sigma_2 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}, \quad (1.6)$$

where

$$\rho_1 = \alpha(2\alpha+3), \quad \rho_k = \left(k^{\alpha+1}(2k-3\alpha-6) + 2k^\alpha(\alpha+1)(\alpha+2) + (k-2)^{\alpha+1}(2-2k-\alpha) \right),$$

$$\begin{aligned} \sigma_1 &= 4(\alpha+1), \quad \sigma_k = (k-2)^{\alpha+1}(2k+\alpha-2) - 2(k-2)^\alpha(2+\alpha)(1+\alpha) - (k)^{\alpha+1}(2k-6-3\alpha), \\ &\quad k = 2, 3, \dots, n. \end{aligned}$$

Proof. Take into account the coefficient of $Q^\alpha = [\theta]_{ij}$, which is the value of $I_t^\alpha \phi_i(t)$ at the j h point, $i = 0, 1, \dots, n$. some simple manipulations completes the proof. \square

2. Description of numerical algorithm based on QHFs

In order to obtain numerical solutions of Eqs. (1.1) using QHFs, by applying (1.2)-(1.4) and substitution (1.5)-(1.6) into Eqs. (1.1) results in

$$\begin{aligned} A_1^T - \lambda E^T + \delta^{\sin(\pi\alpha/2)} A_1^T Q^\alpha - \varepsilon^{\sin(\pi\alpha/2)} A_2^T Q^\alpha &= 0, \\ A_2^T - \delta^{\sin(\pi\alpha/2)} A_1^T Q^\alpha + (\varepsilon^{\sin(\pi\alpha/2)} + \varphi^{\sin(\pi\alpha/2)}) A_2^T Q^\alpha &= 0, \quad 0 < \alpha \leq 1, \quad t \in [0, T], \end{aligned}$$

Table 1: Comparison between QHFs, and generalized Mittag–Leffler function method (GMLFM) for $u_1(t)$, $u_2(t)$ with Clinical data, $\alpha=1$.

Times	The amount of BSP in blood: $u_1(t)$			The amount of BSP in liver: $u_2(t)$		
t	Clinical data [2]	GMLFM [1]	QHFs; h=0.5	Clinical data [2]	GMLFM [1]	QHFs; h=0.5
0	250	250	250	0	0	0
5	184	192.24	191.95	65.8	56.35	56.68
10	141	151.36	150.96	106.5	93.6406	94.09
20	98	101.44	101.04	141.5	132.65	133.09
30	80	75.71	74.99	148.3	145.11	145.92
43	64	68.09	57.58	—	—	—

wherein $E = [1, 1, \dots, 1]^T$. This system has the dimension $2(n+1) \times 2(n+1)$. Suppose $Q^\alpha = [\theta]_{ij}$, $i, j = 0, \dots, n$. As shown in the operational matrix (1.6), and based on the initial values, we have the following Algorithm:

Step 1 : Inputs, n (even), α , T , $Q^\alpha = [\theta]_{ij}$, $i, j = 0, \dots, n$, $u_1(0) = \lambda_v$, $u_2(0) = 0$.

Step 2 : Set and solve recursive algebraic system k , (2×2) — $k = 1, 3, \dots, n-1$.

$$\begin{aligned}
 & a_{10} = \lambda, \quad a_{20} = 0. \\
 & \text{for } k = 1 : (2) : n-1 \\
 & \quad \text{Solution of the } k^{\text{th}} \ (2 \times 2) \text{ system } k, \text{ determines the unknown parameters.} \\
 & \quad \text{system } k : \begin{cases} a_{1k} + \delta^{\sin(\pi\alpha/2)} \left[\sum_{i=0}^k \theta_{ik} a_{1i} \right] - \varepsilon^{\sin(\pi\alpha/2)} \left[\sum_{i=0}^k \theta_{ik} a_{2i} \right] - \lambda = 0, \\ a_{2k} - \delta^{\sin(\pi\alpha/2)} \left[\sum_{i=0}^k \theta_{ik} a_{1i} \right] + (\varepsilon^{\sin(\pi\alpha/2)} + \varphi^{\sin(\pi\alpha/2)}) \left[\sum_{i=0}^k \theta_{ik} a_{2i} \right] = 0, \end{cases} \\
 & \quad \text{and we can get} \\
 & \quad a_{1k+1} = -\delta^{\sin(\pi\alpha/2)} \left[\sum_{i=0}^{k+1} \theta_{ik+1} a_{1i} \right] + \varepsilon^{\sin(\pi\alpha/2)} \left[\sum_{i=0}^{k+1} \theta_{ik+1} a_{2i} \right] + \lambda, \\
 & \quad a_{2k+1} = \delta^{\sin(\pi\alpha/2)} \left[\sum_{i=0}^{k+1} \theta_{ik+1} a_{1i} \right] - (\varepsilon^{\sin(\pi\alpha/2)} + \varphi^{\sin(\pi\alpha/2)}) \left[\sum_{i=0}^{k+1} \theta_{ik+1} a_{2i} \right], \\
 & \quad \text{end}
 \end{aligned}$$

Step 3 : Get a_{vi} and determine $u_v(t) \simeq \sum_{i=0}^n a_{vi} \phi_i(t)$, $v = 1, 2$, $i = 0, \dots, n$.

3. Simulation results and discussion

The numerical results (at different α) are presented in Table 1 and Figure 1 for $T = 60$. The values of the parameters are selected as $\delta = 0.054736$, $\varepsilon = 0.0152704$, $\varphi = 0.0093906$, and $\lambda = 250$ based on a realistic analysis in [2]. The relative error for $u_1(t)$ at the end of 43 min with $\alpha=1$ and $\alpha = 0.98$ are 0.1003 and 0.0320, respectively.

Conclusion: The dynamic behavior of this model via QHFs confirmed the behavior of clinical data. Based on this method, the liver and similar problems can be derived as $n/2$ algebraic systems involving two equations and two unknowns. Compared with the classic model with $\alpha = 1$, the model with $\alpha = 0.98$ gives more realistic results.

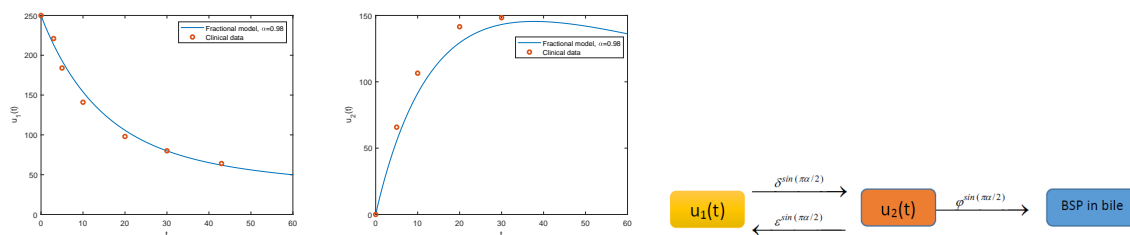


Figure 1: The QHFs solutions for $\alpha = 0.98, h = 0.5$ and Clinical data (Left: two plots), Flow scheme of the human liver math model (Right).

References

- [1] I.G. Ameen, N.H. Sweilam, H.M. Ali, A fractional-order model of human liver: Analytic-approximate and numerical solutions comparing with clinical data, *Alexandria Engineering Journal*, 60 (2021) 4797–4808.
- [2] L. Čelechovská, A simple mathematical model of the human liver, *Applied Mathematics*, 49 (2004) 227–246.
- [3] S. Nemati, P. M. Lima, Numerical solution of nonlinear fractional integro-differential equations with weakly singular kernels via a modification of hat functions. *Applied Mathematics and Computation*, 327 (2018) 79–92.

Hamed Ebrahimi

Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran
email address: hamed_ebrahimi@webmail.guilan.ac.ir

Jafar Biazar

Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, P.O. Box 41335-1914, P.C.4193822697, Rasht, Iran
email address: biazar@guilan.ac.ir



University of Guilan



An efficient method for solving the system of generalized Abel integral equations in three unknowns

Sayed Arsalan Sajjadi, Hashem Saberi Najafi, Hossein Aminikhah

Abstract. In this paper, we consider the product integration method for solving the system of generalized Abel integral equations in three unknowns. This is a mixed system of first kind Volterra integral equations with singular kernels. Finally, some numerical examples with smooth and non-smooth solutions are provided to test the efficiency of method.

1. Introduction

Systems of singular integral equations appear in many branches of scientific fields such as microscopy, seismology, radio astronomy, electron emission, atomic scattering, radar ranging, plasma diagnostics, X-ray radiography and optical fiber evaluation [6]. The general ideas and the essential features of these systems are of wide applicability [6]. In this paper, we consider the following system of Abel's generalized singular integral equations in three unknowns from [6]

$$\begin{aligned} g_1(x) &= \int_0^x \left((x-t)^{-\beta_{11}} u(t) + (x-t)^{-\beta_{12}} v(t) + (x-t)^{-\beta_{13}} w(t) \right) dt, \\ g_2(x) &= \int_0^x \left((x-t)^{-\beta_{21}} u(t) + (x-t)^{-\beta_{22}} v(t) + (x-t)^{-\beta_{23}} w(t) \right) dt, \\ g_3(x) &= \int_0^x \left((x-t)^{-\beta_{31}} u(t) + (x-t)^{-\beta_{32}} v(t) + (x-t)^{-\beta_{33}} w(t) \right) dt, \\ &, \quad x \in I := [0, T], \end{aligned} \tag{1.1}$$

where the singular kernels $(x-t)^{-\beta_{ij}}$ ($1 \leq i, j \leq 3$) and the continuous functions g_j ($1 \leq j \leq 3$) are given real-valued functions. The functions $u(t), v(t)$ and $w(t)$ are unknowns which should be determined. Also, $0 < \beta_{ij} < 1$ ($1 \leq i, j \leq 3$) and $g_j(0) = 0$ ($1 \leq j \leq 3$). We assume that

$\det \left(\begin{bmatrix} (x-t)^{-\beta_{11}} & (x-t)^{-\beta_{12}} & (x-t)^{-\beta_{13}} \\ (x-t)^{-\beta_{21}} & (x-t)^{-\beta_{22}} & (x-t)^{-\beta_{23}} \\ (x-t)^{-\beta_{31}} & (x-t)^{-\beta_{32}} & (x-t)^{-\beta_{33}} \end{bmatrix} \right) \neq 0$ in order to guarantee the existence and uniqueness of solution of (1.1) (see [6]).

In [6], the system (1.1) is examined using the Laplace transform method. The existence and uniqueness solution of the system (1.1) can be related to Theorem 6.1.14 from [2]. It's known that a first kind Volterra integral equation is an example of an ill-posed problem [2]. One of the most powerful ways for dealing with poorly behaved integrands is product integration [4]. The solutions of (1.1) maybe have singularity at the lower bound of the domain of integration. To deal with this non-smooth behavior, we use the product integration method to numerical solution of (1.1). Finally, we provide three numerical examples with smooth and non-smooth solutions to test the accuracy and efficiency of the presented method.

Keywords: Numerical solution, Singular integrals, Integral equations, ill-posed, Product integration.

AMS Mathematical Subject Classification [2010]: 45F15, 45Dxx.

2. Algorithm of method

The first kind Chebyshev polynomial $T_N(x)$ is defined by the following relation

$$T_N(x) = \cos(N \cos^{-1}(x)),$$

and satisfy in the following recursion relation [5]

$$T_N(x) = 2xT_{N-1}(x) - T_{N-2}(x), \quad N = 2, 3, \dots,$$

with

$$T_0(x) = 1, \quad T_1(x) = x.$$

The orthogonality of these polynomials with respect to the weight function $w(x) = (1 - x^2)^{(-\frac{1}{2})}$ as follows

$$\int_{-1}^1 T_M(x)T_N(x)w(x)dx = \begin{cases} 0, & M \neq N, \\ \frac{\pi}{2}, & M = N \neq 0, \\ \pi, & M = N = 0. \end{cases}$$

The Chebyshev Gauss quadrature points are given by [3]

$$z_j = -\cos\left(\frac{(2j+1)\pi}{2N+2}\right), \quad j = 0, \dots, N.$$

The Gauss quadrature formula

$$\int_{-1}^1 f(x)w(x)dx \approx f(z_0)w_0 + \sum_{j=1}^N f(z_j)w_j,$$

is exact for any polynomial of degree $\leq 2N + 1$.

We use the Lagrange interpolating polynomial to approximate $u(t)$, $v(t)$ and $w(t)$, as

$$I_N^C(u(t)) = \sum_{j=0}^N l_j(t)u(z_j), \quad (2.1)$$

$$I_N^C(v(t)) = \sum_{j=0}^N l_j(t)v(z_j), \quad (2.2)$$

$$I_N^C(w(t)) = \sum_{j=0}^N l_j(t)w(z_j), \quad (2.3)$$

where

$$l_i(t) = \prod_{j=0, j \neq i}^N \frac{t - z_j}{z_i - z_j}, \quad i = 0(1)N,$$

and define

$$W_{ijhk} = \int_0^{z_i} (z_i - t)^{-\beta_{hk}} l_j(t) dt, \quad j = 0(1)N, \quad i = 0(1)N, \quad 1 \leq h, k \leq 3.$$

Now, similar to the Nyström method [1], approximating the integrals of system (1.1), we have

$$\begin{aligned} g_1(x) &= \sum_{j=0}^N W_{11}u(z_j) + \sum_{j=0}^N W_{12}v(z_j) + \sum_{j=0}^N W_{13}w(z_j), \\ g_2(x) &= \sum_{j=0}^N W_{21}u(z_j) + \sum_{j=0}^N W_{22}v(z_j) + \sum_{j=0}^N W_{23}w(z_j), \\ g_3(x) &= \sum_{j=0}^N W_{31}u(z_j) + \sum_{j=0}^N W_{32}v(z_j) + \sum_{j=0}^N W_{33}w(z_j), \quad x \in I := [0, T] \end{aligned} \quad (2.4)$$

To find the solutions at the node points, let x run through the quadrature points $z_i (i = 0(1)N)$. This yields the following linear system

$$\begin{aligned} g_1(z_i) &= \sum_{j=0}^N W_{11}u(z_j) + \sum_{j=0}^N W_{12}v(z_j) + \sum_{j=0}^N W_{13}w(z_j), \\ g_2(z_i) &= \sum_{j=0}^N W_{21}u(z_j) + \sum_{j=0}^N W_{22}v(z_j) + \sum_{j=0}^N W_{23}w(z_j), \\ g_3(z_i) &= \sum_{j=0}^N W_{31}u(z_j) + \sum_{j=0}^N W_{32}v(z_j) + \sum_{j=0}^N W_{33}w(z_j), \quad i = 0(1)N. \end{aligned} \quad (2.5)$$

The above system is a system of $3(N+1)$ equations with same number of unknowns. Solving (2.5), we obtain the values of $u(z_j)$, $v(z_j)$ and $w(z_j)$.

3. Numerical examples

In this section, we present some examples to illustrate the validity of the presented method in section 2. We get the numerical results by Wolfram Mathematica 12.2.

Example 3.1. Consider the following system of Abel's generalized singular integral equations [6] with smooth solutions

$$\begin{aligned} g_1(x) &= \int_0^x \left((x-t)^{-1/2}u(t) + (x-t)^{-1/2}w(t) \right) dt, \\ g_2(x) &= \int_0^x \left((x-t)^{-2/3}v(t) + (x-t)^{-2/3}w(t) \right) dt, \\ g_3(x) &= \int_0^x \left((x-t)^{-3/4}u(t) + (x-t)^{-3/4}v(t) \right) dt, \quad x \in I := [0, 1], \end{aligned} \quad (3.1)$$

where $g_1(x)$, $g_2(x)$ and $g_3(x)$ are chosen such that $u(t) = 1$, $v(t) = t$ and $w(t) = t^2$.

Example 3.2. Consider the following system of Abel's generalized singular integral equations [6] with smooth solutions

$$\begin{aligned} g_1(x) &= \int_0^x \left((x-t)^{-2/3}u(t) + (x-t)^{-2/3}w(t) \right) dt, \\ g_2(x) &= \int_0^x \left((x-t)^{-3/4}v(t) + (x-t)^{-3/4}w(t) \right) dt, \\ g_3(x) &= \int_0^x \left((x-t)^{-2/5}u(t) + (x-t)^{-2/5}v(t) \right) dt, \quad x \in I := [0, 1], \end{aligned} \quad (3.2)$$

where $g_1(x), g_2(x)$ and $g_3(x)$ are chosen such that $u(t) = 1 + 2t + 3t^2$, $v(t) = 1 - 2t - 3t^2$ and $w(t) = 1 + 2t - 3t^2$.

Example 3.3. Consider the following system of Abel's generalized singular integral equations with non-smooth solutions

$$\begin{aligned} g_1(x) &= \int_0^x \left((x-t)^{-1/2}u(t) + (x-t)^{-1/2}v(t) + (x-t)^{-2/3}w(t) \right) dt, \\ g_2(x) &= \int_0^x \left((x-t)^{-3/4}u(t) + (x-t)^{-1/2}v(t) + (x-t)^{-2/5}w(t) \right) dt, \\ g_3(x) &= \int_0^x \left((x-t)^{-2/3}u(t) + (x-t)^{-1/4}v(t) + (x-t)^{-1/2}w(t) \right) dt, \\ x &\in I := [0, 1], \end{aligned} \quad (3.3)$$

where $g_1(x), g_2(x)$ and $g_3(x)$ are chosen such that $u(t) = \sqrt{t}$, $v(t) = \sin(\sqrt{t})$ and $w(t) = t^{1/3}$.

Tables 1, 2 and 3 contain the maximum errors for examples 3.1, 3.2 and 3.3, respectively. Figs. 1, 2 and 3 represent the error behaviors of the calculated solutions, corresponding to the examples 3.1, 3.2 and 3.3, respectively, at the grid points for $N = 12$.

Table 1: The obtained maximum errors $\|u - u_N\|_\infty$, $\|v - v_N\|_\infty$ and $\|w - w_N\|_\infty$ by presented method for Example 3.1.

N	2	4	8	16	32	64
$\ u - u_N\ _\infty$	0.0000329838	0.0000108216	1.106453×10^{-6}	1.115676×10^{-8}	1.14464×10^{-12}	1.22125×10^{-14}
$\ v - v_N\ _\infty$	0.0000355824	0.0000114222	1.135019×10^{-6}	1.121844×10^{-8}	1.13143×10^{-12}	1.37668×10^{-14}
$\ w - w_N\ _\infty$	0.0000329812	0.0000108213	1.106450×10^{-6}	1.115677×10^{-8}	1.12677×10^{-12}	1.17684×10^{-14}

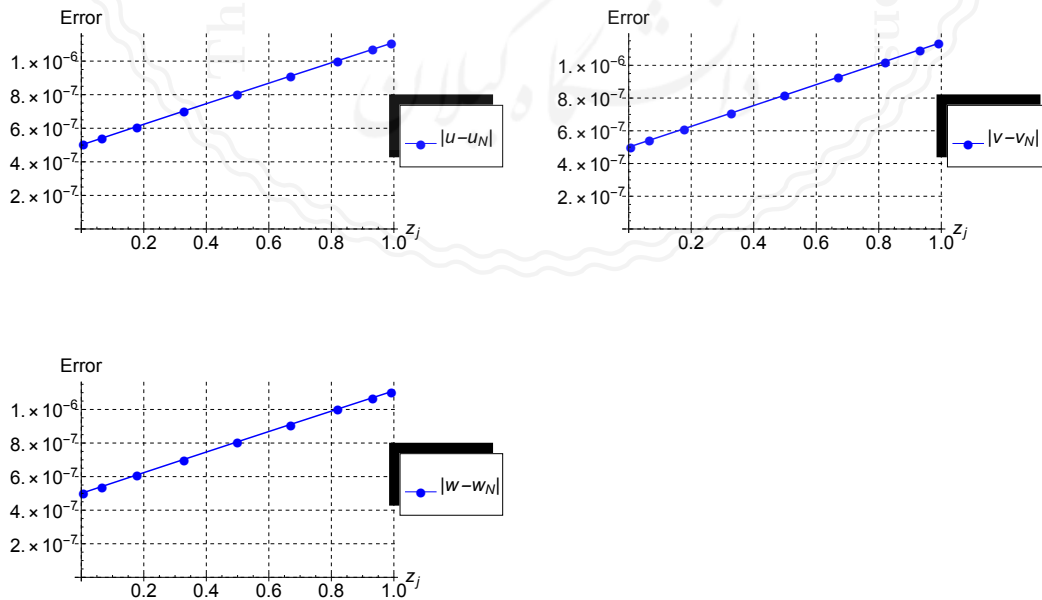


Figure 1: Plot of obtained errors $|u - u_N|$, $|v - v_N|$ and $|w - w_N|$ by the presented method with $N = 8$ for Example 3.1.

Table 2: The obtained maximum errors $\|u - u_N\|_\infty$, $\|v - v_N\|_\infty$ and $\|w - w_N\|_\infty$ by presented method for Example 3.2.

N	2	4	8	16	32	64
$\ u - u_N\ _\infty$	0.0000880819	0.0000309938	$3.19274993 \times 10^{-6}$	3.1946375×10^{-8}	3.2081×10^{-12}	6.1284×10^{-14}
$\ v - v_N\ _\infty$	0.0000880818	0.0000309938	$3.19274991 \times 10^{-6}$	3.1946371×10^{-8}	3.24629×10^{-12}	8.5709×10^{-14}
$\ w - w_N\ _\infty$	0.0000810976	0.0000294374	$3.11977989 \times 10^{-6}$	3.178955×10^{-8}	3.21156×10^{-12}	8.1394×10^{-14}

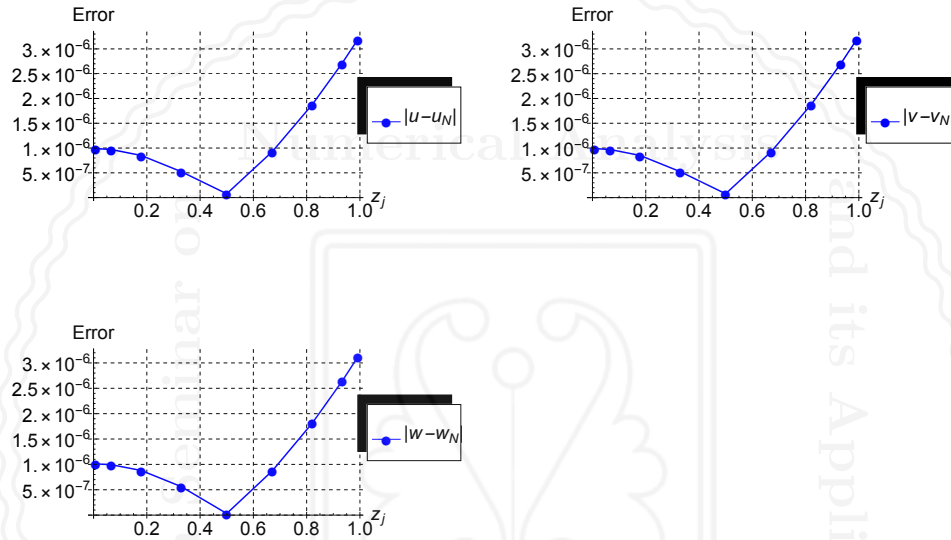


Figure 2: Plot of obtained errors $|u - u_N|$, $|v - v_N|$ and $|w - w_N|$ by the presented method with $N = 8$ for Example 3.2.

Table 3: The obtained maximum errors $\|u - u_N\|_\infty$, $\|v - v_N\|_\infty$ and $\|w - w_N\|_\infty$ by presented method for Example 3.3.

N	2	4	8	16	32	64
$\ u - u_N\ _\infty$	0.0201987	0.0112029	0.00601395	0.00314132	0.00161014	0.000816218
$\ v - v_N\ _\infty$	0.0157754	0.010693	0.00552607	0.00338234	0.00312386	0.00262143
$\ w - w_N\ _\infty$	0.0356525	0.0227428	0.0144483	0.00910235	0.00569364	0.00354716

4. Conclusion

In this paper, we used the product integration method to the numerical solution of Abel's generalized singular integral equations in three unknowns. The accuracy and efficiency of the presented method were tested by providing some numerical examples with smooth and

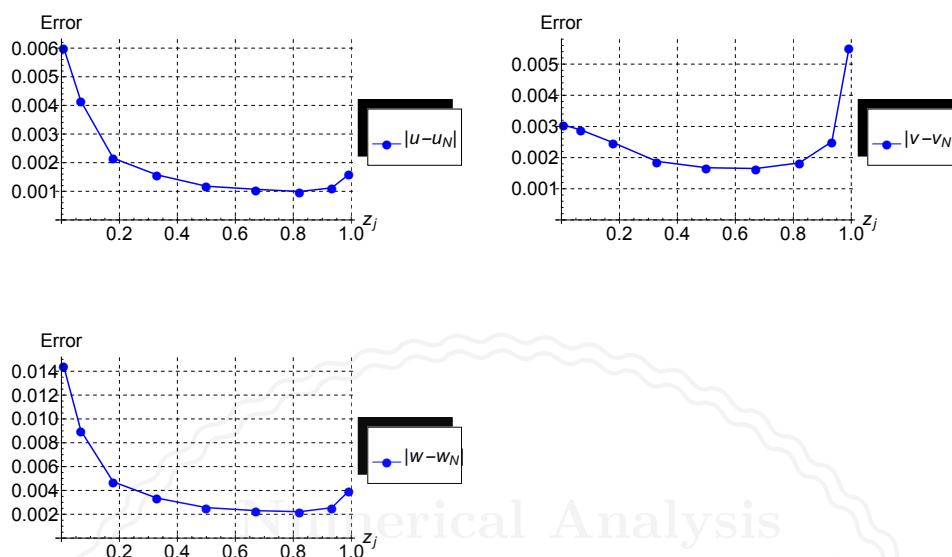


Figure 3: Plot of obtained errors $|u - u_N|$, $|v - v_N|$ and $|w - w_N|$ by the presented method with $N = 8$ for Example 3.3.

References

- [1] K.E. Atkinson, The numerical solution of integral equations of the second kind. *Cambridge Monographs on Applied and Computational Mathematics*, 1996.
- [2] H. Brunner, *Volterra Integral Equations: An Introduction to Theory and Applications*, Cambridge University Press, 2017.
- [3] J.S. Hesthaven, S. Gottlieb, D. Gottlieb, *Spectral Methods for Time-dependent Problems*, Cambridge University Press; 2007.
- [4] P. Linz, *Analytical and numerical methods for Volterra equations*, Society for Industrial and Applied Mathematics; 1985.
- [5] J.C. Mason, D.C. Handscomb, *Chebyshev polynomials*, CRC press, 2002.
- [6] A.M. Wazwaz, *Linear and nonlinear integral equations*, Berlin, Springer, 2011.

Sayed Arsalan Sajjadi

Department of Applied Mathematics and Computer Science, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

email address: arsalansajadi@gmail.com

Hashem Saberi Najafi

Department of Applied Mathematics and Computer Science, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

email address: hnajafi@guilan.ac.ir

Hossein Aminikhah

Center of Excellence for Mathematical Modelling, Optimization and Combinational Computing (MMOCC), University of Guilan, Rasht, Iran

email address: aminikhah@guilan.ac.ir



University of Guilan



Some boundary value problems on the Heisenberg Lie groups

Abdolrahman Razani, Farzaneh Safari

Abstract. The Heisenberg Lie groups are the simplest example of Carnot groups. It has a broad set of applications in many sciences such as quantum mechanics, ergodic theory, representation theory of nilpotent Lie group, foundation of abelian harmonic analysis and theory of partial differential equations. The existence of solutions of some boundary value problems in the Heisenberg Lie groups is proved via variational methods.

1. Introduction

The Heisenberg group historically originates in and still has its strongest ties to quantum physics: there it is a group of unitary operators acting on the space of states induced from those observable on a linear phase space, a simplistic vector space which are given by linear or by constant functions. So any Heisenberg group is a subgroup of a group of observable in certain simple examples of quantum mechanical systems.

As important topics where the Heisenberg group reveals itself as an essential factor are quantum mechanics, ergodic theory, representation theory of nilpotent Lie group, foundation of abelian harmonic analysis, and theory of partial differential equations where we are interested in the last one. We start with introducing notations and some definitions (See more details in [2–8] and references therein).

Thought this note, \mathbb{H}^n is the Heisenberg Lie group which has \mathbb{R}^{2n+1} as a background manifold and endowed with the following noncommutative law of product

$$(x, y, t) \circ (x', y', t') = (x + x', y + y', t + t' + 2(\langle y | x' \rangle - \langle x | y' \rangle)),$$

where $x, x', y, y' \in \mathbb{R}^n$, $t, t' \in \mathbb{R}$ and $\langle \cdot | \cdot \rangle$ denotes the standard inner product in \mathbb{R}^n . We denote by $|\cdot|_{\mathbb{H}^n}$ Korányi norm with respect to the parabolic dilation $\delta_\lambda \xi = (\lambda x, \lambda y, \lambda^2 t)$; i.e.

$$|\xi|_{\mathbb{H}^n} = (|z|^4 + t^2)^{\frac{1}{4}} = ((x^2 + y^2)^2 + t^2)^{\frac{1}{4}},$$

for $z = (x, y) \in \mathbb{R}^{2n}$ and $\xi = (z, t) \in \mathbb{H}^n$. A Korányi ball of center ξ_0 and radius r is defined by

$$B_{\mathbb{H}^n}(\xi_0, r) = \{\xi : |\xi^{-1} \circ \xi_0|_{\mathbb{H}^n} \leq r\},$$

and it satisfies the following equalities

$$|B_{\mathbb{H}^n}(\xi_0, r)| = |B_{\mathbb{H}^n}(0, r)| = r^Q |B_{\mathbb{H}^n}(0, 1)|,$$

Keywords: Heisenberg Lie groups, Laplacian operator.

AMS Mathematical Subject Classification [2010]: 35R03.

where $|U|$ denotes the $(2n+1)$ -dimensional Lebesgue measure of U and $Q = 2n+2$ is homogeneous dimension of \mathbb{H}^n . The Heisenberg gradient is given by

$$\nabla_{\mathbb{H}^n} = (X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_n),$$

where

$$T = \frac{\partial}{\partial t}, \quad X_i = \frac{\partial}{\partial x_i} + 2y_i \frac{\partial}{\partial t}, \quad Y_i = \frac{\partial}{\partial y_i} - 2x_i \frac{\partial}{\partial t}, \quad i = 1, 2, 3, \dots, n.$$

are vector fields that constitute a basis for the real Lie algebra of left-invariant vector fields on \mathbb{H}^n . For any horizontal vector field function $X = X(\xi)$, $X = \{x_i X_i + x'_i Y_i\}_{i=1}^n$, of class $C^1(\mathbb{H}^n, \mathbb{R}^{2n})$, we define the horizontal divergence of X by

$$\operatorname{div}_{\mathbb{H}^n} X := \sum_{i=1}^n [X_i(x_i) + Y_i(x'_i)].$$

Definition 1.1. (Horizontal curve) A piecewise smooth curve $y : [0, 1] \rightarrow \mathbb{H}^n$ is called a horizontal curve if $\dot{y}(t)$ belongs to the span of $\{X_i, Y_i\}_{i=1}^n$ a.e. in $[0, 1]$. The horizontal length of y is defined as follows

$$L_{\mathbb{H}^n}(y) = \int_0^1 \sqrt{(\dot{y}(t), \dot{y}(t))_{\mathbb{H}^n}} dt = \int_0^1 |\dot{y}(t)|_{\mathbb{H}^n} dt,$$

where

$$(X, Y)_{\mathbb{H}^n} = \sum_{i=1}^n (x_i y_i + x'_i y'_i),$$

for each $X = \{x_i X_i + x'_i Y_i\}_{i=1}^n$ and $Y = \{y_i X_i + y'_i Y_i\}_{i=1}^n$.

Carnot-Carathéodory distance of two points $\xi_1, \xi_2 \in \mathbb{H}^n$ is defined by

$$d_{cc}(\xi_1, \xi_2) = \inf \{L_{\mathbb{H}^n}(y) : y \text{ is a horizontal curve joining } \xi_1, \xi_2 \text{ in } \mathbb{H}^n\}.$$

Notice that according to Chow-Rashevsky theorem, for any two arbitrary points $\xi_1, \xi_2 \in \mathbb{H}^n$, there is a horizontal curve between them in \mathbb{H}^n , then the above definition is well-defined. d_{cc} is left invariant metric on \mathbb{H}^n and homogeneous of degree 1 with respect to dilations δ_λ , that is

$$d_{cc}(\delta_\lambda(\xi_1), \delta_\lambda(\xi_2)) = \lambda d_{cc}(\xi_1, \xi_2).$$

for all $\xi_1, \xi_2 \in \mathbb{H}^n$. We denote by $d\xi$ the Haar measure on \mathbb{H}^n that coincides with the $(2n+1)$ -Lebesgue measure, since the Haar measures on Lie groups are unique up to constant multipliers.

Here, we recall Hardy's inequality on Heisenberg group established in [9, Theorem 1.1].

Lemma 1.2. *Let $1 < p < Q$ and $u \in C_0^\infty(\mathbb{H}^n)$. Then*

$$\int_{\mathbb{H}^n} |\nabla_{\mathbb{H}^n} u|^p d\xi \geq \left(\frac{Q-p}{p}\right)^p \int_{\mathbb{H}^n} \frac{|u|^p}{d_{cc}^p} d\xi.$$

As usual, for any measurable set $\Omega \subset \mathbb{H}^n$, $n \geq 1$, we denote by $L^p(\Omega)$ the canonical Banach space, endowed with the norm

$$|u|_p = \left(\int_{\Omega} |u|^p d\xi \right)^{\frac{1}{p}}.$$

Also, the first-order Heisenberg Sobolev space on Ω is defined as follows

$$HW^{1,p}(\Omega) := \{u \in L^p(\Omega) : |\nabla_{\mathbb{H}^n} u| \in L^p(\Omega)\},$$

endowed with the norm

$$\|u\|_{1,p} = |u|_p + |\nabla_{\mathbb{H}^n} u|_p,$$

and we set $HW_0^{1,p}(\Omega) = \overline{C_0^\infty(\Omega), \|u\|_{1,p}}$ equipped with the norm

$$\|u\|_p = |\nabla_{\mathbb{H}^n} u|_p.$$

It is well-known $L^p(\Omega)$, $HW^{1,p}(\Omega)$ and $HW_0^{1,p}(\Omega)$ are separable, reflexive Banach space.

Definition 1.3. (Poincaré-Sobolev domain) An open set Ω of \mathbb{H}^n is said to be a Poincaré-Sobolev domain, if there exist a bounded open set $U \subset \mathbb{H}^n$, with $\Omega \subset \bar{\Omega} \subset U$, a covering $\{B\}_{B \in \mathfrak{F}}$ of Ω by Carnot-Carathéodory balls B and numbers $N > 0, \alpha \geq 1$ and $\nu \geq 1$ such that

- (i) $\sum_{B \in \mathfrak{F}} \mathbf{1}_{(a+1)B} \leq N \mathbf{1}_\Omega$ in U , where $\mathbf{1}_D$ is the characteristic function of a Lebesgue measurable subset D .
- (ii) there exists a (central) ball $B_0 \in \mathfrak{F}$ such that for all $B \in \mathfrak{F}$ there is a finite chain $B_0, B_1, \dots, B_{s(B)}$, with $B_i \cap B_{i+1} \neq \emptyset$ and

$$|B_i \cap B_{i+1}| \geq \frac{\max\{|B_i|, |B_{i+1}|\}}{N}, \quad i = 0, 1, \dots, s(B) - 1$$

and moreover, $B \subset \nu B_i$ for $i = 0, 1, \dots, s(B)$.

The next result is a special case of Theorem 1.3.1 in [1].

Theorem 1.4. (i) Let Ω be a bounded Poincaré-Sobolev domain in \mathbb{H}^n and let $1 \leq p \leq Q$. Then the embedding

$$HW_0^{1,p}(\Omega) \hookrightarrow L^\sigma(\Omega)$$

is compact for all σ , with $1 \leq \sigma < p^*$, where $p^* = \frac{pQ}{Q-p}$ is the critical Sobolev exponent related to p .

(ii) The Carnot-Carathéodory balls are Poincaré-Sobolev domains.

Remark 1.5. Combining Theorem 1.4, with the fact that the Carnot-Carathéodory distance and the Korányi distance are equivalent on \mathbb{H}^n , we get the following embedding is compact

$$HW_0^{1,p}(\Omega) \hookrightarrow L^\sigma(\Omega) \quad \text{for } 1 \leq \sigma < p^*,$$

when Ω is a Korányi ball and $1 \leq p \leq Q$. Furthermore, there exists $C_\sigma > 0$ such that

$$|u|_\sigma \leq C_\sigma \|u\|_p \quad \text{for } 1 \leq \sigma \leq p^*,$$

for all $u \in HW_0^{1,p}(\Omega)$.

Remark 1.6. From Hardy inequality mentioned in Lemma 1.2, since the Carnot-Carathéodory distance and the Korányi distance are equivalent on \mathbb{H}^n , we gain the following inequality

$$\int_\Omega |\nabla_{\mathbb{H}^n} u|^p d\xi \geq \left(\frac{Q-p}{p}\right)^p \int_\Omega \frac{|u|^p}{|\xi|_{\mathbb{H}^n}^p} d\xi,$$

for $1 < p < Q$ and $u \in HW_0^{1,p}(\Omega)$ where Ω is a Korányi ball. For convenience, we set $H = \left(\frac{Q-p}{p}\right)^p$ and so we deduce that

$$\int_\Omega \frac{|u|^p}{|\xi|_{\mathbb{H}^n}^p} d\xi \leq \frac{1}{H} \int_\Omega |\nabla_{\mathbb{H}^n} u|^p d\xi.$$

2. Main results

Here, we bring some problems on the Heisenberg groups to show the importance of Heisenberg groups in PDEs. The first one is a generalized Heisenberg p -Laplacian problem

$$\begin{cases} -\Delta_{\mathbb{H}^n, p} u + R(\xi)|u|^{p-2}u = f(u) - g(u) & \xi \in \Omega, \\ u > 0 & \xi \in \Omega, \\ u = 0 & \xi \in \partial\Omega, \end{cases}$$

in which $\Delta_{\mathbb{H}^n, p} u = \operatorname{div}_{\mathbb{H}^n}(|\nabla_{\mathbb{H}^n} u|^{p-2} \nabla_{\mathbb{H}^n} u)$ is Heisenberg p -Laplacian operator which comes from nonlinear phenomena specially in physics: rheology, glaciology, radiation of heat, plastic moulding etc. We can find enough conditions for f and g such that this problem has at least one weak solution. The next one is a (p, q) -Laplacian problem

$$\begin{cases} -\operatorname{div}_{\mathbb{H}^n}(|\nabla_{\mathbb{H}^n} u|^{p-2} \nabla_{\mathbb{H}^n} u) - \operatorname{div}_{\mathbb{H}^n}(|\nabla_{\mathbb{H}^n} u|^{q-2} \nabla_{\mathbb{H}^n} u) + \frac{|u|^{\gamma-2}u}{|\xi|^{\gamma}} = \lambda f(\xi, u), & \xi \in \Omega, \\ u = 0, & \xi \in \partial\Omega, \end{cases}$$

where f is a Carathéodory function with a growth condition and we can find enough conditions for the existence and multiplicity of solutions to this problem. We would like to point out that usually solutions to (p, q) -Laplacian problems are the steady state solutions of the reaction diffusion systems. Reaction-diffusion systems are mathematical models which correspond to several physical phenomena. This system has a wide range of applications in physics and related sciences like chemical reaction design, biophysics, plasma physics, geology, and ecology. This equations also arise in the study of soliton-like solutions of the nonlinear Schrödinger equation as a model for elementary particles for example waves in a discrete electrical lattice.

References

- [1] S.P. Ivanov, D.N. Vassilev, *Extremals for the Sobolev Inequality and the Quaternionic Contact Yamabe Problem*, World Scientific, Publishing Co. Pte. Ltd., Hackensack, 2011.
- [2] M.A. Ragusa, A. Razani, F. Safari, Existence of radial solutions for a $p(x)$ -Laplacian Dirichlet problem, *Advances in Difference Equations*, (2021) no. 1.
- [3] F. Safari, A. Razani, Existence of positive radial solutions for Neumann problem on the Heisenberg group, *Boundary Value Problems*, (2020), no. 88, <https://doi.org/10.1186/s13661-020-01386-5>.
- [4] F. Safari, A. Razani, Nonlinear nonhomogeneous Neumann problem on the Heisenberg group, *Applicable Analysis*, (2020) 2387–2400 <https://doi.org/10.1080/00036811.2020.1807013>.
- [5] F. Safari, A. Razani, Existence of radial solutions of the Kohn-Laplacian problem, *Complex Variables and Elliptic Equations*, (2020) 259–273 <https://doi.org/10.1080/17476933.2020.1818733>.
- [6] F. Safari, A. Razani, Positive weak solutions of a generalized supercritical Neumann problem, *Iranian Journal of Science and Technology, Transaction A: Science*, 44 (2020) 1891–1898.

- [7] F. Safari, A. Razani, Radial solutions for a general form of a p -Laplace equation involving nonlinearity terms, *Complex Variables and Elliptic Equations*, (2021) <https://doi.org/10.1080/17476933.2021.1991331>.
- [8] F. Safari, A. Razani, Existence of radial solutions for a weighted p -biharmonic problem with Navier boundary condition on the Heisenberg group, *Mathematica Slovaca*, (2021) accepted.
- [9] Q.H. Yang, Hardy type inequalities related to Carnot-Carathéodory distance on the Heisenberg group, *Proceedings of the American Mathematical Society*, 141 (2013) 351-362.

Abdolrahman Razani

Department of Pure Mathematics, Faculty of Science, Imam Khomeini International University, P.O. Box 34194-168181, Qazvin, Iran

email address: razani@sci.ikiu.ac.ir

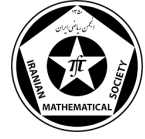
Farzaneh Safari

Department of Pure Mathematics, Faculty of Science, Imam Khomeini International University, P.O. Box 34194-168181, Qazvin, Iran

email address: f.safari@edu.ikiu.ac.ir



University of Guilan



Numerical method for distributed-order time-fractional sub-diffusion equations

Tahereh Eftekhari, Jalil Rashidinia

Abstract. In this paper, an efficient method for solving time-fractional sub-diffusion equations of distributed-order is presented. An error bound for the new method is obtained. Numerical experiments illustrate the efficiency of the proposed method.

1. Introduction

In the present research, we discuss time-fractional sub-diffusion equations (TFSDs) of distributed-order with initial and Dirichlet's boundary conditions, presented in the following form:

$${}_0^C D_t^{\sigma(\mu)} u(s, t) = \frac{\partial^2}{\partial s^2} u(s, t) + g(s, t), \quad (1.1)$$

$$u(s, 0) = a(s), \quad u(0, t) = b_0(t), \quad u(1, t) = b_1(t), \quad (1.2)$$

where, g is a known function, ${}_0^C D_t^{\sigma(\mu)}$ is the distributed-order derivative in the Caputo sense, and $\sigma(\mu)$ denotes the distribution of order $\mu \in (0, 1)$, such that $\sigma(\mu) \geq 0$ and $\int_0^1 \sigma(\mu) d\mu = L > 0$. Also, $(s, t) \in \Omega \times T$, where $\Omega = (0, 1)$, and $T = (0, t_f]$.

In recent years, numerical methods for solving fractional differential equations of distributed-order have attracted the attention of a large number of researchers. For a review on numerical techniques, see for instance [1–5] and references therein.

2. Preliminaries

2.1. The HBPSLs

The HBPSLs are defined as follows:

$$\theta_{n,m}(t) = \begin{cases} L_m \left(\frac{2^k}{t_f} t - 2n + 1 \right), & \frac{n-1}{2^{k-1}} t_f \leq t < \frac{n}{2^{k-1}} t_f, \\ 0, & \text{otherwise,} \end{cases} \quad (2.1)$$

for $n = 1, 2, \dots, 2^{k-1}$, $m = 0, 1, \dots, M-1$. Here $L_m(t) = \sum_{l=0}^m \eta_{l,m} t^l$, $t \in [-1, 1]$ is the Legendre polynomial of degree m and

$$\eta_{l,m} = 2^m \binom{m}{l} \binom{\frac{m+l-1}{2}}{m}.$$

Keywords: Distributed order fractional derivative, Sub-diffusion equations, Hybrid functions.

AMS Mathematical Subject Classification [2010]: 26A33, 65N35.

2.2. Function approximation

A function $u(s, t) \in L^2(\Omega \times T)$ can be approximated in terms of HBPSLs as follows

$$u(s, t) \simeq u_{k_1, M_1, k_2, M_2}(s, t) = \sum_{n=1}^{2^{k_1-1} M_1-1} \sum_{j=0}^{2^{k_2-1} M_2-1} \hat{u}_{n, m, n', m'} \theta_{n, m}(s) \theta_{n', m'}(t) = \Theta(s)^T \hat{U} \Theta(t),$$

where

$$\hat{u}_{n, m, n', m'} = \frac{\int_0^1 \int_0^{t_f} u(s, t) \theta_{n, m}(s) \theta_{n', m'}(t) \omega^{(\phi, \varphi)}(s) \omega^{(\phi, \varphi)}(t) dt ds}{\int_0^1 \int_0^{t_f} \theta_{n, m}^2(s) \theta_{n', m'}^2(t) dt ds},$$

and \hat{U} , $\Theta(s)$, $\Theta(t)$ are defined by

$$\hat{U} = \begin{pmatrix} \hat{u}_{1,0,1,0} & \hat{u}_{1,0,1,1} & \cdots & \hat{u}_{1,0,2^{k_2-1}, M_2-1} \\ \hat{u}_{1,1,1,0} & \hat{u}_{1,1,1,1} & \cdots & \hat{u}_{1,1,2^{k_2-1}, M_2-1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{u}_{2^{k_1-1}, M_1-1, 1, 0} & \hat{u}_{2^{k_1-1}, M_1-1, 1, 1} & \cdots & \hat{u}_{2^{k_1-1}, M_1-1, 2^{k_2-1}, M_2-1} \end{pmatrix},$$

$$\Theta(s) = [\theta_{1,0}(s), \dots, \theta_{1, M_1-1}(s), \dots, \theta_{2^{k_1-1}, 0}(s), \dots, \theta_{2^{k_1-1}, M_1-1}(s)]^T,$$

$$\Theta(t) = [\theta_{1,0}(t), \dots, \theta_{1, M_2-1}(t), \dots, \theta_{2^{k_2-1}, 0}(t), \dots, \theta_{2^{k_2-1}, M_2-1}(t)]^T.$$

3. The method of solution for TFSDEs of distributed-order

In this section, using the new method gives

$$\begin{aligned} & \frac{1}{2} \sum_{q=1}^N w_q \sigma \left(\frac{\varepsilon_q + 1}{2} \right) \left(\hat{\Theta}(s, 2)^T \hat{\Upsilon} \hat{\Theta}(t, \frac{1 - \varepsilon_q}{2}) - s \left(\hat{\Theta}(1, 2)^T \hat{\Upsilon} \hat{\Theta}(t, \frac{1 - \varepsilon_q}{2}) \right) \right. \\ & \left. + (1 - s) {}_0^C D_t^{\frac{\varepsilon_q + 1}{2}} b_0(t) + s {}_0^C D_t^{\frac{\varepsilon_q + 1}{2}} b_1(t) \right) - \Theta(s)^T \hat{\Upsilon} \hat{\Theta}(t, 1) - \frac{d^2}{ds^2} a(s) - g(s, t) \simeq 0. \end{aligned} \quad (3.1)$$

where $\hat{\Theta}$ is operational vector for the Riemann-Liouville integral operator of fractional order based on HBPSLs [1], which for a sake of brevity we will not able to explain them in detail.

Now we collocate the obtained equation at

$$s_{n_1} = \frac{2n_1 - 1}{2^{k_1} M_1}, \quad t_{n_2} = \frac{2n_2 - 1}{2^{k_2} M_2} t_f, \quad n_1 = 1, 2, \dots, 2^{k_1-1} M_1, \quad n_2 = 1, 2, \dots, 2^{k_2-1} M_2. \quad (3.2)$$

By the “fsolve” command of Maple 2018, we solve the above system and then the unknown $\hat{\Upsilon}$ can be determined. Finally, an approximate solution for (1.1), (1.2) can be obtained.

4. Error bound

Theorem 4.1 (see [1]). *Let $u \in C^{(2M+2)}(\Omega \times T)$. Suppose that $u(s, t)$ is the exact solution of (1.1), (1.2) and $u_{k, M}(s, t)$ is its approximate solution obtained by the HBPSLs method. Also, let $\|\sigma\|_2 \leq \gamma$. Therefore, we have the following error bound of HBPSLs method for the modified equation:*

$$\|E_k^M\|_2 \leq L \frac{\sqrt{t_f}}{2^{(k-1)M} M!}, \quad (4.1)$$

where L is a positive constant.

5. Illustrative examples

Here, one problem is presented and solved by our proposed method using Maple 2018 software. Also, we consider $k_1 = k_2 = 1$, $N = 8$, $\Omega \times T = (0, 1) \times (0, 1]$, and obtain the absolute errors, L_∞ -errors by

$$\max_{\substack{n_1=0,1,\dots,2^{k_1-1}M_1, \\ n_2=0,1,\dots,2^{k_2-1}M_2}} \{|u(s_{n_1}, t_{n_2}) - u_{k_1, M_1, k_2, M_2}(s_{n_1}, t_{n_2})|\},$$

respectively, where s_{n_1} and t_{n_2} are defined in (3.2).

Example 5.1. Consider the following TFDE of distributed-order [4]:

$$\int_0^1 \Gamma(4 - \mu) {}_0^C D_t^\mu u(s, t) d\mu = \frac{\partial^2}{\partial s^2} u(s, t) + \frac{s \sin(s) t^2 (t \ln t + 6t - 6)}{\ln t} - 2t^3 \cos(t),$$

$$u(s, 0) = u(0, t) = 0, \quad u(1, t) = t^3 \sin(1),$$

with $u(s, t) = t^3 s \sin(s)$.

In Table 1, the L_∞ -error is reported, for the methods of HBPSLs and CPs [4]. This table illustrates the accuracy of the HBPSLs method.

Table 1: Comparison of the L_∞ -errors, for Example 5.1.

$M_1 = M_2 = n = m$	HBPSLs	CPs [4]
1	6.558298e - 3	3.142e - 1
3	7.193385e - 6	3.469e - 3
5	6.667064e - 8	8.382e - 6
7	1.188375e - 10	1.275e - 8
9	2.010265e - 13	1.020e - 11
11	1.400000e - 15	3.709e - 14

References

- [1] T. Eftekhari, J. Rashidinia, A new operational vector approach for time-fractional sub-diffusion equations of distributed-order based on hybrid functions, *Mathematical Methods in the Applied Sciences*, In press (2022).
- [2] T. Eftekhari, J. Rashidinia, K. Maleknejad, Existence, uniqueness, and approximate solutions for the general nonlinear distributed-order fractional differential equations in a Banach space, *Advances in Difference Equations*, (2021), no. 1.
- [3] K. Maleknejad, J. Rashidinia, T. Eftekhari, Numerical solutions of distributed order fractional differential equations in the time domain using the Müntz-Legendre wavelets approach, *Numerical Methods for Partial Differential Equations*, 37 (2021) 707–731.
- [4] M.L.Morgado, M.Rebelo, L.L. Ferras, N.J. Ford, Numerical solution for diffusion equations with distributed order in time using a Chebyshev collocation method. *Applied Numerical Mathematics*, 114 (2017) 108–123.
- [5] J. Rashidinia, T. Eftekhari, K. Maleknejad, A novel operational vector for solving the general form of distributed order fractional differential equations in the time domain based on the second kind Chebyshev wavelets, *Numerical Algorithms*, 88 (2021) 1617–1639.

Tahereh Eftekhari

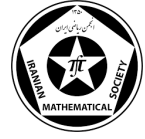
School of Mathematics, Iran University of Science & Technology (IUST), Narmak, Tehran 16846 13114, Iran
email address: t.eftekhari2009@gmail.com

Jalil Rashidinia

School of Mathematics, Iran University of Science & Technology (IUST), Narmak, Tehran 16846 13114, Iran
email address: rashidinia@iust.ac.ir



University of Guilan



RBF-FD method for a fractional inverse problem

Fatemeh Taghipour, Ahmad Shirzadi

Abstract. In this paper, a local meshless method is applied to the inverse source problem of distributed order time fractional diffusion equation. A first-order finite difference approach is suggested for discretizing temporal variable. Then, the resulting equations are fully discretized by a radial basis function-generated finite difference (RBF - FD) based method.

1. Introduction

Distributed-order derivatives have proven their applicability on several phenomena. The numerical approximation of the distributed-order time fractional diffusion equation was studied in many papers [1, 2]. However, there are still few studies related to inverse source problems for distributed-order time fractional diffusion equations. In this paper, we consider the following distributed time-fractional diffusion equation with the given initial and Dirichlet boundary conditions:

$$D_t^{w(\alpha)} u(\mathbf{x}, t) = \Delta u(\mathbf{x}, t) + F(\mathbf{x}, t), \quad \mathbf{x} = (x, y) \in \Omega \subset R^2, t \in [0, T], \quad (1.1)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega \quad (1.2)$$

$$u(\mathbf{x}, t) = g(\mathbf{x}, t), \quad \mathbf{x} = (x, y) \in \partial\Omega, t \in [0, T], \quad (1.3)$$

where g and u_0 are given sufficiently smooth functions and $D_t^{w(\alpha)}$ is time-fractional derivative of distributed order and defined by $D_t^{w(\alpha)} u(\mathbf{x}, t) = \int_0^1 w(\alpha) {}_0^C D_t^\alpha u(\mathbf{x}, t) d\alpha$. $w : [0, 1] \rightarrow R$ is a continuous non-negative weight function, with conditions $\forall \alpha \in [0, 1], w(\alpha) > 0$ and $\int_0^1 w(\alpha) d\alpha = W$ where W is a positive constant and ${}_0^C D_t^\alpha$ is the Caputo fractional derivative. If the source term $F(\mathbf{x}, t)$ can not be directly observed, it hence becomes unknown and then, we consider the additional condition

$$\int_{\Omega} w(\mathbf{x}) u(\mathbf{x}, t) d\mathbf{x} = \psi(t), \quad t \in [0, T], \quad (1.4)$$

where $w(\mathbf{x})$ is a given weight function. Suppose further that

$$F(\mathbf{x}, t) = f(\mathbf{x}, t)r(t), \quad (\mathbf{x}, t) \in \Omega \times [0, T], \quad (1.5)$$

Keywords: Inverse source problem, Distributed-order time fractional equation, Fractional diffusion equation, Radial basis function, Finite difference.

AMS Mathematical Subject Classification [2010]: 65N21, 35R11.

where $f(\mathbf{x}, t)$ represents the known source function component and $r(t)$ is an unknown time-dependent coefficient that is sought. If $w(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_*)$, then (1.4) becomes

$$u(\mathbf{x}_*, t) = \psi(t). \quad (1.6)$$

If we consider Eq. (1.1) and (1.5), with additional condition as Eq. (1.6), then the distributed order derivative of Eq. (1.6) is in the form $D_t^{\omega(\alpha)} u(\mathbf{x}_*, t) = D_t^{\omega(\alpha)} \psi(t)$. Then we get $D_t^{\omega(\alpha)} u(\mathbf{x}_*, t) = \Delta u(\mathbf{x}_*, t) + r(t)f(\mathbf{x}_*, t)$. We rewrite the recent relation as

$$r(t) = \frac{D_t^{\omega(\alpha)} u(\mathbf{x}_*, t) - \Delta u(\mathbf{x}_*, t)}{f(\mathbf{x}_*, t)}.$$

By replacing $r(t)$ in $D_t^{\omega(\alpha)} u(\mathbf{x}, t) = \Delta u(\mathbf{x}, t) + r(t)f(\mathbf{x}, t)$, we obtain:

$$D_t^{\omega(\alpha)} u(\mathbf{x}, t) = \Delta u(\mathbf{x}, t) + H(\mathbf{x}, t)\Delta u_* + K(\mathbf{x}, t), \quad (1.7)$$

where $H(\mathbf{x}, t) = -\frac{f(\mathbf{x}, t)}{f(\mathbf{x}_*, t)}$, $k(\mathbf{x}, t) = \frac{D_t^{\omega(\alpha)} u(\mathbf{x}_*, t)f(\mathbf{x}, t)}{f(\mathbf{x}_*, t)}$ and $\Delta u_* = \Delta u(\mathbf{x}_*, t)$. In the following section present an *RBF-FD* approach to numerically solve the above equation.

2. Discretization of the governing equation

To discretize the integral term in Equation (1.7), the Gauss-Legendre integration rule is used as

$$\int_{-1}^1 f(x)dx = \sum_{j=1}^n v_j f(x_j), \quad \forall f(x) \in C^\infty[-1, 1], \text{ where } x_j \text{ are roots of Legendre polynomial } P_n(x)$$

and v_j s are the weights. So, Eq. (1.7) becomes:

$$\sum_{s=1}^q v_s w(\alpha_s) {}^c D_t^{\alpha_s} u(\mathbf{x}, t) = \Delta u(\mathbf{x}, t) + H(\mathbf{x}, t)\Delta u_* + K(\mathbf{x}, t). \quad (2.1)$$

A discretization of the time interval $[0, T]$ is considered as $\{t_0, t_1, \dots, t_N\}$, where $t_n = n\tau$ and $\tau = \frac{T}{N}$, $n = 0, 1, \dots, N$. As described in [3], the following finite difference scheme is used for discretizing the fractional derivative:

$$\begin{aligned} {}^c D_t^{\alpha} u(\mathbf{x}, t_n) &= \frac{1}{\Gamma(1-\alpha)} \int_0^{t_n} (t_n - t)^{-\alpha} \frac{\partial u(\mathbf{x}, t)}{\partial t} dt \\ &= \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} \left[a_0 u^n - \sum_{k=1}^{n-1} (a_{k-n-1} - a_{k-n}) u^k - a_{n-1} u^0 \right]. \end{aligned} \quad (2.2)$$

where $a_l = (l+1)^{1-\alpha} - l^{1-\alpha}$.

Considering Eq. (1.7) at the point (x, y) and time instant t_n , and substituting (2.2) in (2.1), we obtain

$$\begin{aligned} \sum_{s=1}^q v_s w(\alpha_s) \frac{\tau^{-\alpha_s}}{\Gamma(2-\alpha_s)} \left[a_0^{\alpha_s} u^n - \sum_{k=1}^{n-1} (a_{k-n-1}^{\alpha_s} - a_{k-n}^{\alpha_s}) u^k - a_{n-1}^{\alpha_s} u^0 \right] \\ = \Delta u^n + H^n \Delta u_*^n + K^n \end{aligned} \quad (2.3)$$

where $u^n = u(\mathbf{x}, t_n)$. To describe the RBF-FD scheme, consider a set of N scattered nodes $\mathbf{x}_1, \dots, \mathbf{x}_N$ and the fractional differential operator L . For a given node, \mathbf{x}_1 , the objective is to

approximate $Lu(\mathbf{x}_1)$ as a linear combination of the values of u at the N scattered nodes, i.e., $Lu(\mathbf{x}_1) \approx \sum_{i=1}^N \alpha_i u(\mathbf{x}_i)$. To determine the weighting coefficients α_i , a local set of nodal points, called an stencil with n nodes and a set of basis functions $\phi_i(\mathbf{x})$, $i = 1, \dots, n$. In a local set, considering $L\phi_j(\mathbf{x}_i) = \sum_{i=1}^n \beta_i \phi_j(\mathbf{x}_i)$ $j = 1, 2, \dots, n$, a local system of n linear equations of n unknowns will be obtained. This system gives an equation corresponding to \mathbf{x}_i and finally a system of N equations with N unknown will be obtained. Solving this system of equations, the coefficients α_i will be obtained.

3. Test problem

Consider the following inverse source problem of the time-fractional diffusion equation:

$$\begin{cases} \int_0^1 \Gamma(3 - \alpha)_0^\alpha D_t^\alpha u(x, y, t) = \Delta u(x, y, t) + r(t)f(x, y, t), & 0 < x, y < 1, \quad 0 \leq t \leq 0.8 \\ u(x, y, 0) = 0, \end{cases} \quad (3.1)$$

where $r(t) = 2(\frac{t-1}{\ln t} + t)$, $f(x, y, t) = t \sin(x) \sin(y)$. The exact solution of the above problem is $u(x, y, t) = t^2 \sin(x) \sin(y)$. Table 1 present the RMS error vrsus the number of nodal points with four different values of time step τ . To see the accuracy of the method, the numerical results corresponding to $u(x, y, t)$ and $r(t)$ are present in Figure 1.

Table 1: RMS error for Test problem at time instant $T = 0.8$.

h	$\tau = \frac{T}{128}$	$\tau = \frac{T}{256}$	$\tau = \frac{T}{512}$	$\tau = \frac{T}{1024}$
1/8	2.0897×10^{-4}	9.4207×10^{-5}	4.2804×10^{-5}	1.9586×10^{-5}
1/16	1.6108×10^{-4}	7.2616×10^{-5}	3.2994×10^{-5}	1.5097×10^{-5}
1/32	7.8930×10^{-5}	3.5582×10^{-5}	1.6167×10^{-5}	7.3974×10^{-6}
1/64	5.1258×10^{-5}	2.3292×10^{-5}	1.0660×10^{-5}	4.9098×10^{-6}

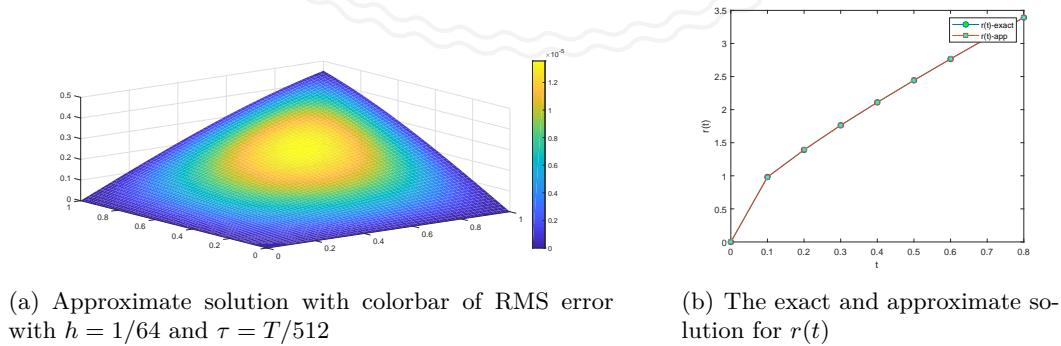


Figure 1:

References

- [1] K. Diethelm, N.J. Ford, Numerical analysis for distributed-order differential equations. *Journal of Computational and Applied Mathematics*, 225 (2009) 96–104.
- [2] J.T. Katsikadelis, Numerical solution of distributed order fractional differential equations, *Journal of Computational Physics*, 259 (2014) 11–22.
- [3] A. Mohebbi, M. Abbaszadeh, Compact finite difference scheme for the solution of time fractional advection-dispersion equation. *Numerical Algorithms*, 63 (2013) 431–452.

Fatemeh Taghipour

Department of Mathematics, Persian Gulf University, Bushehr, Iran

email address: f.taghipour@mehr.pgu.ac.ir

Ahmad Shirzadi

Department of Mathematics, Persian Gulf University, Bushehr, Iran

email address: shirzadi@pgu.ac.ir

Efficient algorithms to solve nonlinear Volterra integral equations of the first kind

Roya Montazeri, Jafar Biazar

Abstract. In this research, A new approach based on an alteration of usage of Optimal Homotopy Asymptotic Method (OHAM), is introduced that applies (OHAM) in multistage, let us call it, in short, as (MOHAM). This procedure is utilized to derive an approximate solution to nonlinear Volterra integral equations of the first kind (NVIEFK). To illustrate these approaches one example is presented. The results confirm the efficiency and ability of these methods for such equations. The results will be compared to find out which method, is more accurate. Advantages of applying MOHAM are also illustrated.

1. Introduction

The Volterra integral equations of the first kind appear in mathematical models of many phenomena in different disciplines and engineering branches such as the vehicular traffic, population dynamics, fluid dynamics, heat conduction problems, the kinetic theory of gases and economics [1, 4]. The sample equation under study is as the following.

$$\int_a^t \mathcal{K}(t, u) H(\psi(u)) du = g(t), \quad a \leq u \leq t \leq b. \quad (1.1)$$

Considering the nonlinear term as the following

$$H(\psi(t)) = \vartheta(t)$$

The canonical form of NVIEFK is as the following

$$\vartheta(t) = f(t) - \int_a^t \mathcal{K}(t, u) \vartheta(u) du, \quad a \leq t \leq b, \quad (1.2)$$

where ϑ is a function to be determined, $f \in l^2([a, b])$, and $\mathcal{K} \in l^2([a, b] \times [a, b])$.

In recent years, NVIEFK has been solved by several authors such as Erfanian and Mostahsan by an optimization method [5], Ma et. al used Sinc Nyström method [6], Singh et. al applied Haar wavelet method [7], and some others.

Now we propose the OHA and the MOHA methods for NVIEFK.

Keywords: Optimal homotopy asymptotic method (OHAM), Multistage optimal homotopy asymptotic method (MOHAM), Volterra integral equations of the first kind.

AMS Mathematical Subject Classification [2010]: 65D07, 65K05.

2. The OHAM and MOHAM applications

These two approaches are well defined in [2,3]. We are going to apply OHAM to the Volterra integral equation of the first kind. Applying the OHAM to Eq. (1.2), results in the following sequential equations

$$\begin{aligned} q^0 : \vartheta_0(t) &= f(t), \\ q^1 : \vartheta_1(t) &= \omega_1 \int_0^t \mathcal{K}(t, u) \vartheta_0(u) du, \\ &\vdots \\ q^k : \vartheta_k(t) &= (1 + \omega_1) \vartheta_{k-1}(t) + \sum_{i=2}^{k-1} \omega_i \vartheta_{k-i}(t) + \sum_{l=1}^k \omega_l \int_0^t \mathcal{K}(t, u) \vartheta_{k-l}(u) du. \end{aligned}$$

An approximation of the m th order is considered by

$$\tilde{\vartheta}(t, \omega_1, \omega_2, \dots, \omega_m) = \vartheta_0(t) + \sum_{k=1}^m \vartheta_k(t, \omega_1, \omega_2, \dots, \omega_m). \quad (2.1)$$

Substitution of Eq. (2.1) in Eq. (1.2), gives the following residual

$$\mathfrak{J}(t, \omega_1, \dots, \omega_m) = \tilde{\vartheta}(t) - f(t) + \int_a^t \mathcal{K}(t, u) \tilde{\vartheta}(u) du, \quad a \leq t \leq b.$$

By least-squares technique, we find optimal values of ω_i , $i = 1, 2, \dots, m$. An approximate solution of order m , is determined, when the parameters ω_i , $i = 1, 2, \dots, m$, are known.

In MOHAM, by partitioning the time interval, $[0, T]$, into N subintervals $[0, t_1], \dots, [t_{\gamma-1}, t_\gamma]$, where $t_\gamma = T$ and OHAM will be applied over each subintervals. The solution at the last point, in each subinterval, denotes an initial approximation to the solution, over the next interval. The process will continue until we achieve the pre-assigned time, T .

3. Numerical Experiments

Here, we are going to demonstrate the ability of the OHA and MOHA methods by one illustrative example. Also compare the results of OHA and MOHA methods will be compared and computations will be performed by Matlab Package.

Example 3.1. In this example we study the following equation

$$\int_0^t e^{t-u} \ln(\psi(u)) du = e^t - t - 1, \quad 0 \leq t \leq 1.$$

The exact solution is $\psi(t) = e^t$. By the new function $\vartheta(u) = \ln(\psi(u))$ and resolving the canonical form, one has,

$$\vartheta(t) = e^t - 1 - \int_0^t e^{t-u} \vartheta(u) du, \quad 0 \leq t \leq 1,$$

where $\psi(u) = e^{\vartheta(u)}$.

The solutions of the first and second orders, in order by MOHA and OHA methods and the exact solution are shown in Table 1, and Figure 1 A.E of OHA and MOHA methods are plotted in Figure 2.

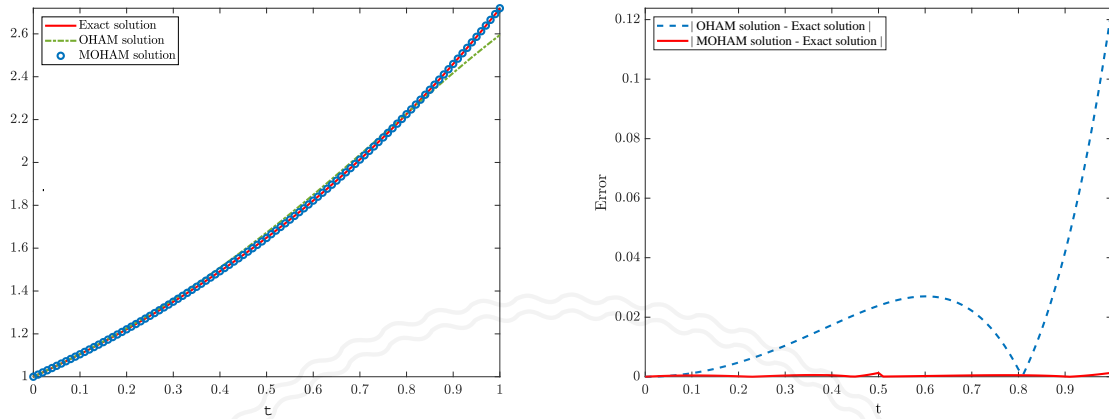


Figure 1: Exact and solutions by OHA and Figure 2: Presentation of A.E of OHA and MOHA methods.

Table 1: Exact and solutions by OHA and MOHA methods.

t_j	Exact	OHAM	A.E. OHAM	MOHAM	A.E. MOHAM
0.0	1.00000000	1.00000000	0.00000000	1.00000000	0.00000000
0.1	1.10517091	1.10636693	0.00119602	1.10474466	0.00042625
0.2	1.22140275	1.22617980	0.00477705	1.22124851	0.00015424
0.3	1.34985880	1.36031132	0.01045252	1.35026852	0.00040972
0.4	1.49182469	1.50920420	0.01737951	1.49230448	0.00047979
0.5	1.64872127	1.67259169	0.02387042	1.64743849	0.00128278
0.6	1.82211880	1.84912703	0.02700823	1.82187478	0.00024402
0.7	2.01375270	2.03591953	0.02216683	2.01330065	0.00045205
0.8	2.22554092	2.22799618	0.00783915	2.22503445	0.00050647
0.9	2.45960311	2.41774538	0.04185773	2.45951469	0.00008815
1.0	2.71828182	2.59445603	0.12382582	2.71962653	0.00134471

4. Conclusion and discussion

In the present study, NVIEFK have been solved using OHA and MOHA methods. Comparison with OHAM results higher accurate respect of applying MOHAM, especially for the nodes further from the initial point. The numerical experiments support this claim, figures are plotted to show the comparison between approximate and the exact solutions with of these experiments. Furthermore, MOHAM is reliable and effective for to obtain approximate solutions of the NVIEFK.

Acknowledgement

We are very grateful to anonymous referees for their careful reading and valuable comments which led to the improvement of this paper

References

- [1] M.A. Bartoshevich, A heat-conduction problem. *Journal of engineering physics*, 28 (1975) 240–244.
- [2] J. Biazar, R. Montazeri, The multistage optimal homotopy asymptotic method for nonlinear weakly singular Volterra integral equations, *Mathematical Sciences Letters*, 8 (2019) 1–9.
- [3] J. Biazar, R. Montazeri, Optimal homotopy asymptotic and multistage optimal homotopy asymptotic methods for solving system of Volterra integral equations of the second kind, *Journal of Applied Mathematics*, (2019) 1–17.
- [4] H.T. Davis, Introduction to Nonlinear Differential and Integral Equations, *Dover, Publications, New York*, 1962.
- [5] H.R. Erfanian, T. Mostahsan, Approximate solution of a class of nonlinear Volterra integral equations, *Journal of Mathematics and Computer Science*, 3 (2011) 278–286.
- [6] Y. Ma, J. Huang, C. Wang, H. Li, Sinc Nyström method for a class of nonlinear Volterra integral equations of the first kind, *Advances in Difference Equations*, 151 (2016) 1–15.
- [7] I. Singh, S. Kumar, Haar wavelet method for some nonlinear Volterra integral equations of the first kind, *Journal of Computational and Applied Mathematics*, 292 (2016) 541–552.

Roya Montazeri

Department of Mathematics, Payame Noor University, P. O. Box 19395-3697, Tehran, Iran
email address: montazer.roy@gmail.com

Jafar Biazar

Department of Applied Mathematics, Faculty of Mathematical sciences, University of Guilan, P.O.Box. 41635-19141, P.C.41938336997, Rasht, Iran
email address: biazar@guilan.ac.ir



Easy Java simulations software for numerical modeling

Amir Pishkoo, Maedeh GholamAzad

Abstract. “Easy Java simulations” is an application software for performing physical simulations. “Photoelectric effect” simulation is an example of simulation in EJS software. To perform this simulation in the introduction section, first, the three main work panels of the program are introduced. We have used Figure 1 as background (from the reference on page 117 of the Physics textbook (3), 12th grade, Mathematics and Physics, 3rd year of secondary education at high school in Iran, Chapter 5: Introduction to atomic physics), and using EJS turned the image of an experiment into a virtual experiment (executable Jar file). This Jar file is the final product of this research, while in the main results section, we explain the details of our work.

1. Introduction

Computer simulation is inextricably linked to computer modeling. A model is a mental representation of a physical system and its properties, and modeling is the process by which we create them. A computer simulation is a model implementation that allows us to test the model under various scenarios in order to learn more about its behavior. Easy Java/Javascript Simulations is a modeling tool that allows scientists, not just computer scientists, to create simulations in a variety of programming languages. Easy JavaScript Simulations is a free open-source tool that has acquired a significant role in physics instruction, with over a thousand simulations available in the ComPADRE digital library [1, 2].

To tackle simulation challenges, a variety of softwares are utilized, including

- MATLAB is a bundle of coding, modeling, and simulation tools produced by MathWorks.
- COMSOL Multiphysics is a multi-physical simulation pioneer and the most powerful simulation software available.
- GAMS is a coding and mathematical optimization high-level modeling system.
- *If the primary goal of the simulation is to visualize the problem and instructional objectives, the Easy Java Simulations (EjsS) application is an excellent choice for the user.*

EjsS automates tasks such as animation and numerical solution of ordinary differential equations. There are three modeling workpanels in Easy Java/Javascript Simulations [3, 4]. To create the model and its graphical user interface, we use a series of workpanels provided by the program.

Keywords: Simulation of ODE models, numerical modeling, animation.

AMS Mathematical Subject Classification [2010]: 65Y04, 34C60, 81T80.

2. EJS and three main workpanels

Working with EJS, we typically perform tasks such as selecting a numerical algorithm, specifying a differential equation or writing problem-solving codes and commands to run simulations, designing view elements that generate motion and animation.

The first panel, *Description*, allows us to build and modify a multimedia HTML-based narrative that describes the model.

The second work panel, *Model*, focuses on the modeling process. This panel is used to create model-specific variables, initialize them, and create algorithms that describe how the model changes over time.

The third work panel, *View*, is in charge of creating the graphical user interface (GUI) that allows users to manage and view the simulation's results. This program replicates Hertz's discovery of the photoelectric phenomenon [5] in 1887, which Einstein theoretically characterized in 1905. A metal is illuminated by a specific frequency of light (and energy) Fig.1.

Electrons are ejected and can produce a current if the frequency (energy) of the light is greater than the metal's work function, W . (which is shown in milliAmperes). If the energy of the light is greater than the workfunction, these photoelectrons will have kinetic energy. Electrons can be prevented from reaching the plate and current from flowing in a circuit if they are exposed to an electric potential.

3. Main results

In this simulation, the items of light waves (sine waves), electrons (set of particles), material button, and galvanometer must be designed and built. When the play/pause button on the user interface is pressed, the model's time evolution begins.

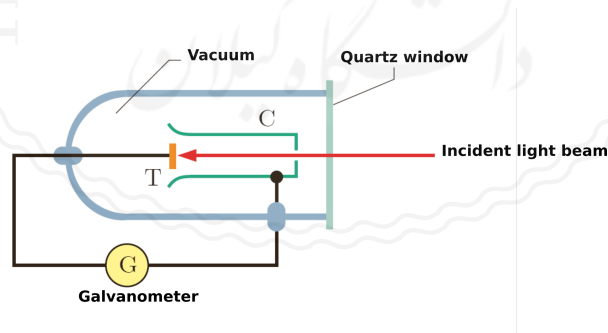


Figure 1: Experimental setup of photoelectric effect: snapshot of the final product of this research as a Jar file.

When you click on the Evolution panel, as shown in Fig.2, the ODE editor appears. The step size is determined by the increment. By advancing the state in discrete stages, numerical techniques approximate the exact ODE solution. Each view element has a set of internal attributes known as properties that govern how it appears and behaves.

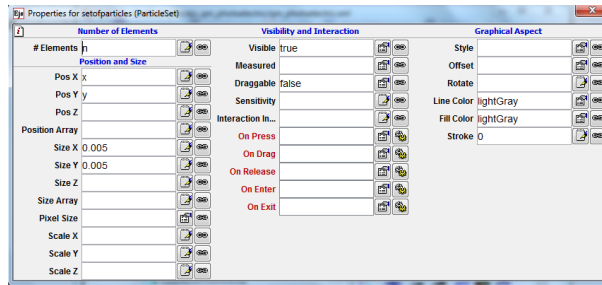


Figure 4: creating set of particles (electrons).

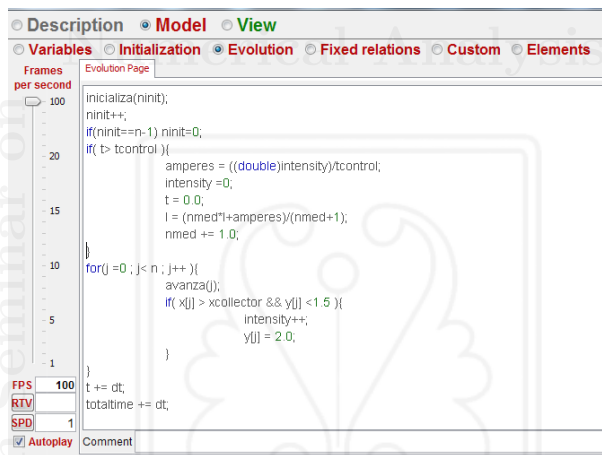


Figure 2: time evolution of the problem.

By double-clicking on the element in the tree, we can access the properties inspector table and change these attributes. A model variable can also be used to set an element's property (Fig.3, Fig.4, and Fig.5). The ability to connect (bind) a property to a variable without programming is critical for converting our static representation into a dynamic and interactive one.

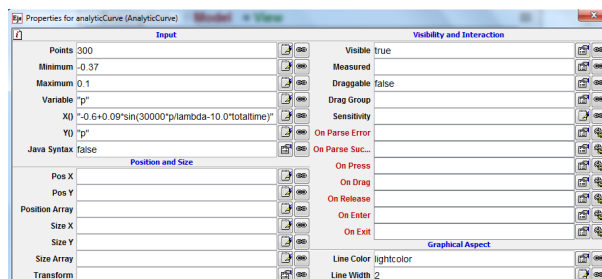


Figure 3: making Sine waves (light).

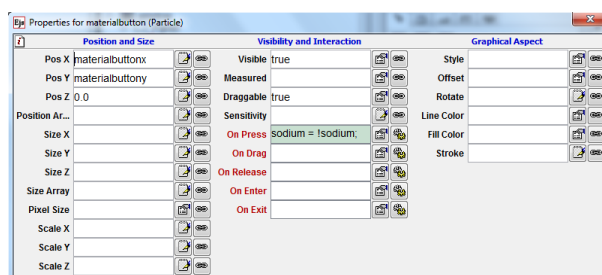


Figure 5: changing material button.

Acknowledgement

We'd like to thank the organizers of the numerical analysis seminar at the University of Guilan.

References

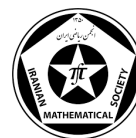
- [1] J. Chacon, E. Besada-Portas, G. Carazo-Barberoand, J.A. Lopez-Orozco, Enhancing EJS with extension plugins, *Electronics*, 10 (2021) no. 242.
- [2] W. Christian, F. Esquembre, Easy Java/Javascript simulations manual, 2015, https://www.um.es/fem/EjsWiki/uploads/Download/EjsS_Manual.pdf.
- [3] F. Esquembre, Easy Java Simulations: a software tool to create scientific simulations in java. *Computer Physics Communications*, 156 (2004) 199–204.
- [4] L. K. Wee, T. L. Lee, C. Chew, D.Wong, S. Tan, Understanding resonance graphs using easy Java simulations (EJS) and why we use EJS. *Physics Education*, 50 (2015) 2 189–196.
- [5] A.S. Raymond, J.W Jewett, *Physics for Scientists and Engineers with Modern Physics, Seventh Edition*, Thomson Brooks/Cole Publishing, 2007.

Amir Pishkoo

Physics and Accelerators Research School, Nuclear Science and Technology Research Institute, Tehran, Iran
email address: apishkoo@gmail.com; apishkoo@aeoi.org.ir

Maedeh GholamAzad

Department of Applied Mathematics, Azarbaijan Shahid Madani University, Tabriz, Iran
email address: m.gholamazad@azaruniv.ac.ir



A reduced-order difference potential algorithm based on POD method for parabolic interface problem

Mahboubeh Tavakoli Tameh, Fatemeh Shakeri

Abstract. We present a rapid and effective method for the numerical solution of the parabolic equation in domains with interfaces. The proposed approach combines the difference potentials method (DPM) with the proper orthogonal decomposition (POD) technique to improve the computational efficiency of the DPM. Numerical results confirm the efficiency and accuracy of the developed numerical method.

1. Introduction

The difference potentials method (DPM) is an efficient and strong technique for solving interface problems and problems defined on domains with complex geometry. This method was proposed by V. Ryaben'kii in 1969 and is a discrete counterpart of Calderon's potential theory in functional analysis [2, 3]. It combines some advantages of the boundary element method (BEM) and finite difference method (FDM) while avoids some drawbacks related to them. The advantages are the effectiveness of FDM in simple geometries and the dimension reduction of BEM. The avoided drawbacks by this method include the difficulty of FDM to handle the complex regions, and the requirement of BEM to the fundamental solution and evaluating singular integral kernels. In DPM, first the value of the solution is calculated at the discrete grid boundary (the grid points close to the boundaries of the original domains) by constructing pseudo-differential boundary equations [1]. Then, these values are used in the discrete generalized Green's formula to obtain the values of the solution in the original domain. The purpose of this paper is combining the second-order difference potentials method and the POD technique to improve the computational efficiency of DPM for solving the parabolic equations.

We consider the following parabolic interface problem

$$\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} = f, \quad (x, t) \in \Omega \times (0, T], \quad (1.1)$$

with initial condition as

$$u(x, 0) = u^0(x), \quad x \in \Omega, \quad (1.2)$$

boundary condition on the points $x = a$ and $x = b$ as

$$u(a, t) = \alpha(t), \quad u(b, t) = \beta(t), \quad (1.3)$$

Keywords: Difference potentials, Calderon's operators, Interface problems.

AMS Mathematical Subject Classification [2010]: 35Exx, 65M06, 82B24.

and interface condition at ζ as

$$u(\zeta, t)|_{\Omega_1} - u(\zeta, t)|_{\Omega_2} = \phi_1(t), \quad t \in (0, T], \quad (1.4)$$

$$u_x(\zeta, t)|_{\Omega_1} - u_x(\zeta, t)|_{\Omega_2} = \phi_2(t), \quad t \in (0, T], \quad (1.5)$$

where $\Omega := [a, b] \subset \mathbb{R}$ is a bounded domain separated by interface point ζ into two disjoint subdomains $\Omega_1 := [a, \zeta)$ and $\Omega_2 := (\zeta, b]$. The source function f is piecewise smooth but can have a jump along the interface, hence the solution of this problem may be discontinuous along ζ .

2. DPM algorithm

In this section, we describe the main steps of the algorithm based on DPM for governing interface problem. The details of DPM strategy are represented in [1, 3].

- **Step 1:** For each subdomain Ω_e , $e \in \{1, 2\}$ introduce an Auxiliary domain Ω_e^0 and formulate the Auxiliary Problem.
- **Step 2:** Discretize the Auxiliary Problem by the finite difference schemes (first-order in time and second order in space) as $L_{\Delta t, h}[u_i^{n+1}] = F_i^{n+1}$, $x_i \in N^+$.
- **Step 3:** At each time level t^{n+1} , for each subdomain Ω_e , calculate a Particular Solution, $u_i^{n+1} := G_{\Delta t, h}F_i^{n+1}$ as the solution of the Auxiliary Problem.
- **Step 4:** Compute $u_{\gamma_e}^{n+1}$ which is the solution at the discrete grid boundary γ_e (the grid points close to the interface) by solving the coupled system of Boundary Equations with Projections (BEP).
- **Step 5:** Calculate the Difference Potential $P_{N_e^+}u_{\gamma_e}^{n+1}$ from the obtained $u_{\gamma_e}^{n+1}$ in the previous step.
- **Step 6:** Finally, approximate the solution using the generalized Green's formula $u(x_i, t^{n+1}) \approx P_{N_e^+}u_{\gamma_e}^{n+1} + G_{\Delta t, h}F_i^{n+1}$.

3. A POD based reduced-order difference potential method

The POD method offers an orthogonal basis for representing a given data set. It is helpful to reduce the dimensions of the numerical computational models for time-dependent PDEs and save CPU time for large-scale scientific computing. Here, we use this technique to improve the efficiency of the DPM algorithm for solving the parabolic equation. We briefly represent the steps of POD-DPM as bellow

- **Step 1:** Find the solution sequence $\{u_i^n\}_{n=1}^L$ from the first steps of Particular Solutions $\{u_i^n\}_{n=1}^N$, ($1 \leq i \leq m$, $L \ll N$), where m and N are the number of space and time steps, respectively.
- **Step 2:** Then, formulate the snapshot matrices $A = (u_i^n)_{m \times L}$ and calculate the eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$, where $r = \text{rank}(A)$, and the eigenvectors ϕ_k ($k = 1, \dots, r$) of $A^t A$.
- **Step 3:** Select the number M ($M < r$) such that $\sqrt{\lambda_{M+1}} \leq \varepsilon$ for the error tolerance $\varepsilon = O(\Delta t, \Delta x^2)$. Then construct the POD basis $\Phi = (\varphi_1, \varphi_2, \dots, \varphi_M)$, where $\varphi_k = A\phi_k / \sqrt{\lambda_k}$ ($k = 1, 2, \dots, M$).

- **Step 4:** Then, obtain the POD-DPM by replacing u^n in the definition of Particular Solution with $u^{*n} = \Phi\theta^n$ ($n = L + 1, \dots, N$) where θ^n are vectors yet to be determined. Similar to the theorem represented in [4], the following theorem can be proved.

Theorem 3.1. *The following relation is hold for the error between the Particular Solution u^n from DPM and the u^{*n} from the POD-DPM*

$$\|u^n - u^{*n}\|_2 \leq E(n)\sqrt{\lambda_{M+1}}, \quad n = 1, \dots, N,$$

where $E(n) = 1$ ($n = 1, \dots, L$) and $E(n) = \exp(n - L)$ ($n = L + 1, \dots, N$).

- **Step 5:** Check accuracy and update POD basis to continue. If $E(n)\sqrt{\lambda_{M+1}} \leq \varepsilon$ the requirement accuracy is satisfied, else set $u^1 = u^{*(n-L)}, \dots, u^L = u^{*(n-1)}$ and go back to Step 2.

4. Numerical results

In this section, we present a numerical example to show the performance and accuracy of the proposed method for solving the parabolic interface problem. We consider $\Omega = [-0.5, 0.5]$ and $\zeta = 0$. Also, we select the Auxiliary domains $\Omega_1^0 = [-0.7, 0.2]$ and $\Omega_2^0 = [-0.2, 0.7]$ for subdomains Ω_1 and Ω_2 , respectively. Also, we set $L = 10$, final time $T = 6$ and time step size equal 0.001. Designed algorithm is implemented with MATLAB R2018a running on a desktop with Intel(R) Core(TM) i5-4200M CPU @ 2.50GHz 2.50 GHz and 6 GB memory. The exact solution of this example as studied in [1] is given by

$$u(x, t) = \begin{cases} x^8 \exp(-t), & -0.5 \leq x \leq 0, \\ \frac{1}{2}(\frac{1}{256} + x^8) \exp(-t), & 0 \leq x \leq 0.5, \end{cases} \quad (4.1)$$

Table 1: The error and convergence rate of DPM and POD-DPM.

N	Error _(DPM)	order	Error _(POD-DPM)	order
40	$7.6509e-08$	*	$7.1902e-08$	*
80	$1.9535e-08$	1.9696	$1.8185e-08$	1.9833
160	$5.0469e-09$	1.9526	$4.6280e-09$	1.9743
320	$1.3214e-09$	1.9333	$1.2088e-09$	1.9368

Table 2: Comparison between CPU time obtained of DPM and POD-DPM.

N	DPM	POD-DPM
160	5.147186	1.672115
320	17.099635	2.277276
640	73.871677	3.574449
1280	435.560937	6.816560

Since the exact solution is known, we calculate the source term, Dirichlet boundary conditions and interface jump conditions on the interface $\zeta = 0$ according to the given exact solution. The grid refinement analysis and computational time of DPM and POD-DPM are reported in Tables 1 and 2. We see that the numerical results of POD-DPM are in excellent agreement with those of DPM. Also, by comparing the CPU time of the POD-DPM with that of the DPM, the advantages of POD-DPM in computational efficiency can be found.

References

- [1] J. Albright, Y. Epshteyn, K.R. Steffen, High-order accurate difference potentials methods for parabolic problems, *Applied Numerical Mathematics*, 93 (2015) 87–106.
- [2] A.A.E. Reznik, Approximation of the potential surfaces of elliptic operators by difference potentials. *Doklady Akademii Nauk*, 263 (1982) 1318–1321.
- [3] V.S. Ryaben Kii, *Method of difference potentials and its applications*, Springer Science & Business Media, 2001.
- [4] B. Xu, X. Zhang, A reduced fourth-order compact difference scheme based on a proper orthogonal decomposition technique for parabolic equations. *Boundary Value Problems*, (2019) no. 130.

Mahboubeh Tavakoli Tameh

Faculty of Mathematics and Computer Sciences, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran.

email address: Mh.Tavakoli@aut.ac.ir

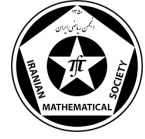
Fatemeh Shakeri

Faculty of Mathematics and Computer Sciences, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran.

email address: F.Shakeri@aut.ac.ir



University of Guilan



Feedback solution for high-nonlinear Hamilton-Jacobi-Bellman equation by a pseudospectral domain decomposition technique

Mehdi Ghasemi, Homa Shirmardi, Mohammad Maleki

Abstract. In this paper, we present a new pseudospectral scheme for solving a class of high-nonlinear optimal control problems. First of all, the Hamilton-Jacobi-Bellman equation is formulated. Then, we approximate the cost functional using the method of pseudospectral. The operational matrix of differentiation and the Gauss method are then utilized to reduce the optimal control problem to the solution of algebraic equations.

1. Introduction

Optimal control is one of the most important branches of the mathematics. For nonlinear systems the optimal state feedback control is obtained by the Hamilton-Jacobi-Bellman (HJB) equation [1]. This equation is difficult to solve, thus approximation techniques for obtaining solutions are important. In addition to linear systems, the nonlinear systems are also very important, because they are applicable in industry, say, in reactor systems, flight control systems and aircraft design [2]. In the year 2000, an upwind method for approximating the viscosity solutions of HJB is presented by Wang et al. [3]. Their method is based on an explicit finite difference scheme, the stability of method under some mild conditions is proved. Huang et al. [4] proposed a semi-meshless discretization scheme based on radial basis functions for approximating viscosity solutions of HJB.

2. HJB equation

A vast amount of literature exists on optimal control problems of the Bolza form

$$\begin{aligned} \min_{u \in \mathcal{U}} J(s, \mathbf{x}, u) &= \int_s^{T_f} L(t, \mathbf{y}(t), u(t)) dt + h(\mathbf{y}(T_f)) \\ \text{s.t. } \dot{\mathbf{y}}(t) &= f(t, \mathbf{y}(t), u(t)), \quad t \in (s, T_f], \quad \mathbf{y}(s) = \mathbf{x}, \end{aligned} \quad (2.1)$$

where $u(\cdot)$ is the control function, $y(\cdot)$ is the state function, $L(\cdot)$ is the running cost, $h(\mathbf{y}(T_f))$ is the terminal cost, $f(\cdot)$ is the vector-valued transition function, $g(\cdot)$ is the integrand function, $(s, x) \in [0, T_f] \times \mathbb{R}^n$, and \mathcal{U} is the set of admissible controls. We introduce the value function v defined by $v(s, \mathbf{x}) = \inf_{u \in \mathcal{U}} J(s, \mathbf{x}, u)$ using dynamic programming approach, the problem (2.1) can be formulated as HJB equation plus its terminal term condition

$$-v_s + \sup_{u \in \mathcal{U}} (-v_{\mathbf{x}} \cdot f(s, \mathbf{x}, u) - L(s, \mathbf{x}, u)) = 0, \quad (2.2)$$

Keywords: Nonlinear optimal control, Pseudospectral method, Hamilton-Jacobi-Bellman, Feedback solution.
AMS Mathematical Subject Classification [2010]: 49J20, 65M70.

$$v(T_f, \mathbf{x}) = h(\mathbf{x}), \quad (2.3)$$

where $v(\cdot)$ is the value function, $h(\mathbf{x})$ is terminal condition and \cdot stands for inner product. Here, we have two unknown functions, $u(\cdot)$ and $v(\cdot)$. The control u which achieves minimum is called optimal control, which we show it by u^* , if we could find a relation of the form $u^*(t) = g(t, \mathbf{x}(t))$

2.1. Approximation by interpolation

Let $-1 = \tau_0, \tau_1, \dots, \tau_N = 1$ be $N + 1$ distinct nodes in $[-1, 1]$, and $l_k(x)$, $k = 0, \dots, N$ be the Lagrange interpolation polynomials based on this nodes, which are defined as $l_k(x) = \prod_{j=0, j \neq k}^N \frac{x - \tau_j}{\tau_k - \tau_j}$ with the Kronecker property $l_k(\tau_j) = \delta_{kj}$. The polynomials $l_k(x)$, $k = 0, \dots, N$ form a basis for the space of the polynomials of degree less than or equal to N . A function $f(x)$ defined on $[-1, 1]$ may be approximated by Lagrange interpolation polynomials as $f(x) = \sum_{k=0}^N f(\tau_k) l_k(x)$. The above approximation can be written in the following matrix form $f(x) \simeq \Phi_N^T(x) f$ where $f = [f(\tau_0), \dots, f(\tau_N)]^T$ and $\Phi_N(x) = [\varphi_0(x), \dots, \varphi_N(x)]^T$. From the Kronecker property, we conclude $\Phi_N(\tau_j) = e_j$, $j = 0, \dots, N$ where e_j is the j th column of the identity matrix of dimension $N + 1$.

$$\tau_j = \cos\left(\frac{\pi j}{N}\right), \quad j = 0, \dots, N \quad (2.4)$$

3. The proposed Hybrid method

Theorem 3.1. Consider the problem (2.2)-(2.3) in one dimensional case, then solution can be written as:

$$v(s, x) \simeq \Phi_n^T(s) v \Phi_m(x), \quad (3.1)$$

which Φ_n and Φ_m are as follows $\Phi_n(s) = \begin{bmatrix} l_1(s) \\ l_2(s) \\ \vdots \\ l_n(s) \end{bmatrix}$, $\Phi_m(x) = \begin{bmatrix} l_1(x) \\ l_2(x) \\ \vdots \\ l_m(x) \end{bmatrix}$ where l_i is i th Lagrange function. By differentiating with respect to s and x , we will have

$$v_s(s, x) = \Phi_n^T(s) D_n v \Phi_m(x), \quad (3.2)$$

$$v_x(s, x) = \Phi_n^T(s) v D_m^T \Phi_m(x), \quad (3.3)$$

where D_n and D_m are differentiation matrices with respect to s and x .

Then, replacing (3.2)-(3.3) in HJB equation we get

$$-\Phi_n^T(s) D_n v \Phi_m(x) + \sup_{u \in \mathcal{U}} \{-\Phi_n^T(s) v D_m^T \Phi_m(y) f(s, x, u) - L(s, x, u)\} = 0, \quad (3.4)$$

Collocating this equation at points (τ_i, x_j) , $i = 1, \dots, n - 1$, $j = 1, \dots, m$ we conclude

$$-[D_n v]_{ij} + \sup_{u \in \mathcal{U}} \{-[v D_m^T]_{ij} f(s, x, u) - L(s, x, u)\} = 0, \quad i = 1, \dots, n - 1, \quad j = 1, \dots, m, \quad (3.5)$$

The above equation can be reformed to the following matrix form

$$-[D_n v]_{[1:n-1,:]} + \sup_{u \in \mathcal{U}} \{-[v D_m^T]_{[1:n-1,:]} f(s, x, u) - L(s, x, u)\} = 0. \quad (3.6)$$

Finally, we have a system of algebraic equations (3.6) with initial condition $v(1, x) = h(x)$, at points x_j , $j = 1, \dots, m$; which can be solved by iterative schemes like Newton method.

3.1. Domain decomposition

Let us in one dimensional case, for $\Omega = [a, b]$, take $\Omega = [a, a_0], [a_0, a_1] \cup \dots \cup [a_{n-1}, b]$ when a_0, a_1, \dots, a_{n-1} are the points that derivative does not exist. The following theorem shows the efficiency of the proposed method using domain decomposition technique.

Consider one dimensional case, for $\Omega = [a, b]$, take $\Omega = [a, a_0], [a_0, a_1] \cup \dots \cup [a_{n-1}, b]$ when a_0, a_1, \dots, a_{n-1} are the points that have no derivative, then the solution to (2.2)-(2.3) can be written as

$$v(s, x_1) = \begin{cases} v^1(s, x_1), & x_1 \in [a, a_0] \\ v^2(s, x_1), & x_1 \in [a_0, a_1] \\ \vdots \\ v^{n+1}(s, x_1), & x_1 \in [a_{n-1}, b]. \end{cases}$$

4. Illustrative examples

In this section, we begin by demonstrating the performance of the proposed method developed in Section. MATLAB function `ode45` is used to solve the system of differential equations with terminal conditions. This solver controls the error by two parameters `RelTol` and `AbsTol`. We set `RelTol=1e-11` and `AbsTol=1e-9`.

To assess the accuracy of the method, the following averaged absolute error is reported: $E_m = \frac{1}{m+1} \|v_{Exact}(t, \mathbf{x}) - v(t, \mathbf{x})\|_\infty$, where v_{Exact} and v are the exact and computed solutions, respectively.

4.1. Example 1

Consider the following system [4], for $x \in [-1, 1]$,

$$\begin{aligned} & \min_u -y(1) \\ \text{s.t.} \quad & \dot{y}(t) = y(t)u(t) \quad t \in [0, 1], \quad y(0) = x, \\ & u(\cdot) : [0, 1] \rightarrow [0, 1]. \end{aligned} \tag{4.1}$$

Corresponding HJB equation is

$$\begin{aligned} -v_t + \sup_{0 \leq u \leq 1} (-xu(t)v_x) &= 0, \\ v(1, x) &= -x. \end{aligned} \tag{4.2}$$

Collocating this equation, we conclude

$$-[D_n v]_{[1:n-1,:]} + \sup_{0 \leq u \leq 1} \left\{ -[v D_m^T]_{[1:n-1,:]} x_j u(t) \right\} = 0. \tag{4.3}$$

If the array (i, j) in matrix $[v D_m^T] x_j$ is positive, we set $u(t) = 0$, otherwise we set $u(t) = 1$. Since (4.3) is discontinuous in $x = 0$, we apply the multidomain strategy with $m = 1$ and $[-1, +1] = [-1, 0] \cup [0, 1]$. Then, we solve system of algebraic equations (4.3) with initial condition $v(1, x) = -x$, at points x_j , $j = 1, \dots, m$.

Table 1: Comparisons for the averaged absolute errors in Example 1.

Method in Ref. [4]		Method in Ref. [6]		Current method	
m	Error	m	Error	m	Error
5	1.575×10^{-2}	3	3.7898×10^{-10}	3	2.1526×10^{-12}
9	5.34×10^{-3}	5	2.2742×10^{-10}	5	2.1526×10^{-12}
14	2.18×10^{-3}	7	1.6239×10^{-10}	7	2.1526×10^{-12}
26	6.59×10^{-4}	9	1.2623×10^{-10}	9	2.1526×10^{-12}

Exact value function is as follows $v(t, x) = \begin{cases} -xe^{1-t}, & x > 0, \\ -x, & x \leq 0. \end{cases}$ Computed errors of approx-

imate value function from the methods in [4], [6] and the present one are given in Table 1; which demonstrates high accuracy of the method. When the number of nodes increases, it is expected that the pseudospectral solution will generate the optimal control solution with error close to zero.

5. Conclusions

In this paper, we discussed a new technique for the optimal control of high-nonlinear continuous-time systems based on pseudospectral method. The main advantages of this approach lie in good accuracy, very low numerical complexity, easy implementation and finding a feedback solution.

References

- [1] F.L. Lewis, V.L. Syrmos, *Optimal control*, John Wiley, (1995).
- [2] Z. Yan, J. Wang, Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks, *IEEE Transactions on industrial informatics*, 8 (2012) 746–756 .
- [3] S. Wang, F. Gao, K. L. Teo, An upwind finite-difference method for the approximation of viscosity solutions to Hamilton-Jacobi-Bellman equations, *IMA Journal of Mathematical Control and Information*, 17 (2000) 167–178 .
- [4] C. S. Huang, S. Wang, C. S. Chen, Z. C. Li, A radial basis collocation method for Hamilton-Jacobi-Bellman equations, *Automatica*, 42 (2006) 2201–2207 .
- [5] E. Cristiani, P. Martinon, Initialization of the shooting method via the Hamilton-Jacobi-Bellman approach, *Journal of Optimization Theory and Applications*, 146 (2010) 321–346 .
- [6] M. Mehrali-Varjani, M. Shamsi, A. Malek, Solving a class of Hamilton-Jacobi-Bellman equations using pseudospectral methods, *Kybernetika*, 54 (2018) 629–647.

Mehdi Ghasemi

Department of Mathematics, Shahrekord University, Shahrekord, Iran

email address: meh_ghasemi@yahoo.com

Homa Shirmardi

Department of Mathematics, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran

email address: homashirmardy@yahoo.com

Mohammad Maleki

Department of Mathematics, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran

email address: mm_maleki2005@yahoo.com



University of Guilan



A numerical method for solving multi-variable order fractional integro-differential equations using the Krall-Laguerre polynomials

Hadis Azin, Ali Habibirad

Abstract. In this work, the variable-order fractional Caputo derivative is used to define multi-variable order fractional integro-differential equation. The orthogonal Krall-Laguerre polynomials are used as basis functions to construct a numerical method for solving such problems. In the established method, these polynomials are applied to transform the main problem into an algebraic problem that can be easily solved. Some examples are considered to show the accuracy of the method.

1. Introduction

Fractional and variable order derivatives can be applied in characterizing hereditary properties of dynamical systems with uniform and non-uniform memory, respectively [1]. Therefore, developing numerical algorithms for solving variable-order fractional equations is significant. Some authors have proposed several numerical and approximation methods for solving such equations, the interested readers are referred to [2, 3]. In this paper, we follow study the multi-variable order nonlinear integro-differential equations as

$$\sum_{j=1}^s b_j(x) {}^c D_x^{\beta_j(x)} y(x) = \mu_1 \int_0^1 \mathbf{K}_1(x, t) (y(t))^{n_1} dt + \mu_2 \int_0^x \mathbf{K}_2(x, t) (y(t))^{n_2} dt + f(x, y), \quad (1.1)$$

with initial conditions $y^{(l)}(0) = y_l$ for $l = 0, 1, \dots, \max[\beta_j(x)]$ and $x \in [0, 1]$. Moreover, s, n_1 and n_2 are positive integer numbers, μ_1 and μ_2 are real numbers, $b_j(x)$, $\mathbf{K}_1(x, t)$ and $\mathbf{K}_2(x, t)$ are given known functions and ${}^c D_x^{\beta_j(x)} y(x)$, $(n - 1 < \beta_j(x) \leq n)$ is the $\beta_j(x)$ -th Caputo variable-order fractional derivative expressed by

$${}^c D_x^{\beta_j(x)} y(x) = \begin{cases} \frac{d^n y(x)}{dx^n}, & \beta_j(x) = n, \\ \frac{1}{\Gamma(n - \beta_j(x))} \int_0^x (x - t)^{n-1-\beta_j(x)} \frac{d^n y(t)}{dt^n} dt, & n - 1 < \beta_j(x) < n. \end{cases} \quad (1.2)$$

2. The Krall-Laguerre polynomials

Krall introduced the Krall orthogonal polynomials. These polynomials are eigenfunctions to fourth order linear differential equations [4]. The Krall-Laguerre polynomial of order $\alpha > 0$, $\mathbf{L}_{\alpha, k}(x)$ of

Keywords: Multi-variable order fractional integro-differential equation, Krall-Laguerre polynomials, Caputo fractional derivative.

AMS Mathematical Subject Classification [2010]: 34A08, 33C45, 41A10.

degree $k \in \mathbb{N} \cup \{0\}$ is given by

$$\mathbf{L}_{\alpha,k}(x) = \sum_{i=0}^k \frac{(-1)^i}{(i+1)!} \binom{k}{i} (i(\alpha + k + 1) + \alpha) x^i. \quad (2.1)$$

The set of Krall-Laguerre polynomials form an orthogonal system on $[0, +\infty)$ with

$$\langle \mathbf{L}_{\alpha,k}(x), \mathbf{L}_{\alpha,\bar{k}}(x) \rangle = \int_0^{+\infty} \mathbf{L}_{\alpha,k}(x) \mathbf{L}_{\alpha,\bar{k}}(x) e^{-x} dx + \frac{1}{\alpha} (\mathbf{L}_{\alpha,k}(0) \mathbf{L}_{\alpha,\bar{k}}(0)). \quad (2.2)$$

Theorem 2.1. Suppose that $\mathbf{L}_{\alpha,k}(x)$ are the functions introduced in (2.1) and $\beta(x) \in (n-1, n]$. Then, we have

$${}_0^c D_x^{\beta(x)} \mathbf{L}_{\alpha,k}(x) = \begin{cases} \mathbf{D}^{(n)} \mathbf{L}_{\alpha,k}(x), & \beta(x) = n, \\ \mathbf{D}^{\beta(x)} \mathbf{L}_{\alpha,k}(x), & n-1 < \beta(x) < n. \end{cases} \quad (2.3)$$

where

$$\begin{aligned} \mathbf{D}^{(n)} \mathbf{L}_{\alpha,k}(x) &= \sum_{i=1}^k \frac{(-1)^i}{(i+1)!} \binom{k}{i} (i(\alpha + k + 1) + \alpha) (i-n+1)! x^{i-n}, \\ \mathbf{D}^{\beta(x)} \mathbf{L}_{\alpha,k}(x) &= \sum_{i=n}^k \frac{(-1)^i}{(i+1)!} \binom{k}{i} (i(\alpha + k + 1) + \alpha) \frac{\Gamma(i+1)}{\Gamma(i+1-\beta(x))} x^{i-\beta(x)}. \end{aligned} \quad (2.4)$$

3. The proposed method

To construct a numerical technique for the problem given in (1.1), we approximate $y(x)$ as $y(x) \simeq \sum_{k=0}^m c_k \mathbf{L}_{\alpha,k}(x) := \mathbf{C}^T \Psi_{\alpha}(x)$ where \mathbf{C} is an $(m+1)$ unknown vector. Theorem 2.1 yield ${}_0^c D_x^{\beta(x)} y(x) \simeq \mathbf{C}^T \mathbf{D}^{\beta_j(x)} \Psi_{\alpha}(x)$. Substituting this relations into (1.1) results

$$\begin{aligned} \mathbf{R}(x) &:= \sum_{j=1}^s b_j(x) \mathbf{C}^T \mathbf{D}^{\beta_j(x)} \Psi_{\alpha}(x) - \mu_1 \int_0^1 \mathbf{K}_1(x, t) (\mathbf{C}^T \Psi_{\alpha}(t))^{n_1} dt \\ &\quad - \mu_2 \int_0^x \mathbf{K}_2(x, t) (\mathbf{C}^T \Psi_{\alpha}(t))^{n_2} dt - f(x, \mathbf{C}^T \Psi_{\alpha}(x)) \simeq 0, \\ \Lambda &:= \mathbf{C}^T \mathbf{D}^{(l)} \Psi_{\alpha}(0) - y_l \simeq 0 \quad l = 0, 1, \dots, \max[\beta_j(x)]. \end{aligned} \quad (3.1)$$

Eventually, we obtain a system of $(m+1)$ equations by inserting the shifted Chebyshev collocation points x_r in $[0, 1]$ for $r = 1, 2, \dots, m - \max[\beta_j(x)]$.

4. Numerical results

Example 4.1. Consider the problem

$$2 {}_0^c D_x^{\beta_1(x)} y(x) - {}_0^c D_x^{\beta_2(x)} y(x) = 56 \int_0^1 (x+t) y(t)^3 dt + f(x, y), \quad (4.1)$$

with initial conditions $y(0) = 1$ and $y'(0) = -3$. The exact solution is $y(x) = x^2 - 3x + 1$. Also, we obtain the source term f from the exact solution. In [5], this problem has been solved by considering $\beta_1(x) = \frac{x+3}{3}$ and $\beta_2(x) = \frac{x}{6}$. The absolute error obtained in [5] is 10^{-3} with $m = 2$ (the Bernstein polynomials) while, we get the exact solution with $m = 2$ and $\alpha = 1$ by using our method.

Example 4.2. consider the problem

$${}_0^c D_x^{\beta_1(x)} y(x) + {}_0^c D_x^{\beta_2(x)} y(x) = \int_0^1 xt(y(t))^2 dt + \int_0^x (x-t)y(t)dt + f(x, y), \quad (4.2)$$

with initial conditions $y(0) = y'(0) = 1$ and $\beta_1(x) = 1.8 - 0.45 \sin(x)$, $\beta_2(x) = 0.8 - 0.3 \cos(x)$ and f is extracted due to exact solution $y(x) = e^x$ as

$$\begin{aligned} f(x, y) = & x^{-\beta_1(x)} \left(\mathbf{E}_{1,1-\beta_1(x)}(x) - \frac{1}{\Gamma(1-\beta_1(x))} - \frac{x}{\Gamma(2-\beta_1(x))} \right) \\ & + x^{-\beta_2(x)} \left(\mathbf{E}_{1,1-\beta_2(x)}(x) - \frac{1}{\Gamma(1-\beta_2(x))} \right) + \frac{3-e^2}{4}x + 1 - y(x), \end{aligned} \quad (4.3)$$

where \mathbf{E} is the Mittag-Leffler function. The obtained results are provided numerically and graphically in Table 1 and Fig. 1 with $\alpha = 1$.

Table 1: The L_∞ errors for Example 4.2 with some values m .

m	L_∞	CPU time (s)
5	$1.9234E-04$	0.671
7	$1.0547E-08$	0.813
9	$5.6858E-10$	0.875
11	$4.3269E-13$	1.234

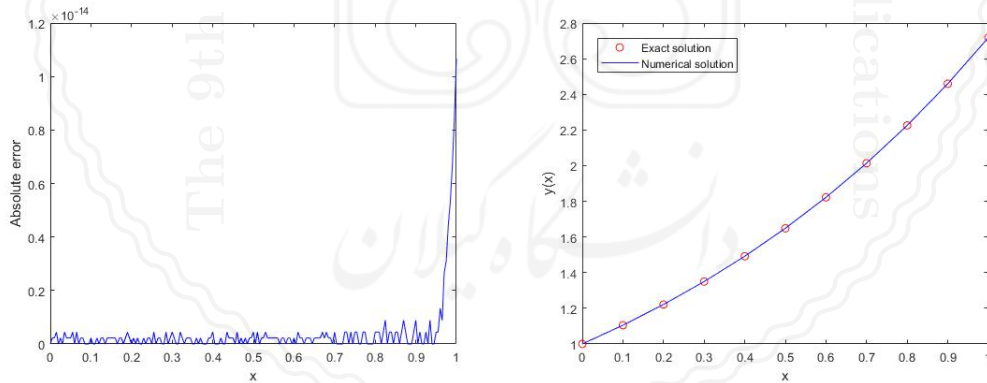


Figure 1: The graphs of absolute error (left), and the exact and approximate solutions (right) for Example 4.2 with $m = 12$.

5. Conclusion

In this article, the multi-variable order fractional integro-differential equation has been generated by using Caputo fractional derivative. The KrallLaguerre polynomials have been successfully extended for solving this equation. Using the presented method, the problem was reduced into a nonlinear system of algebraic equations and solved readily by the "fsolve" command of Maple software. The obtained numerical solutions maintain excellent accuracy in comparison with the exact solutions.

References

- [1] D. Baleanu, Mendes, A. Lopes *Handbook of fractional calculus with applications*, Applications in engineering, life and social sciences, 2019, <https://www.degruyter.com/document/doi/10.1515/9783110571929-fm/pdf>.
- [2] R. M. Ganji, H. Jafari, D. Baleanu, A new approach for solving multi variable orders differential equations with Mittag-Leffler kernel. *Chaos Solitons Fractals* 130 (2020) 109405.
- [3] H. Dehestani, Y. Ordokhani, M. Razzaghi, Pseudo-operational matrix method for the solution of variable-order fractional partial integro-differential equations. *Engineering with Computers*, 37 (2020) 1791–1806.
- [4] H.L. Krall, On orthogonal polynomials satisfying a certain fourth order differential equation, *Pennsylvania State College*, 34 (1940) no. 34.
- [5] N. S. Tuan, Nemati, R.M. Ganji, H. Jafari, Numerical solution of multi-variable order fractional integro-differential equations using the Bernstein polynomials, *Engineering with Computers*, 38 (2020) 139–147.

Hadis Azin
Department of Mathematics, University of Hormozgan, Bandar Abbas, Iran
email address: H.Azin1370@gmail.com

Ali Habibirad
Department of Mathematics, Shiraz University of Technology, Shiraz, Iran.
email address: a.habibirad@sutech.ac.ir



University of Guilan



Sparse two-greedy subspace Kaczmarz method with oblique projection for compressed sensing

Farshid Abdollahi, Fatemeh Pirayesh Dehkordi

Abstract. Kaczmarz method is one of alternating projection methods and is an iterative method for solving large-scale systems. In this article, we use the two-greedy subspace Kaczmarz method with oblique projection to generate sparse solution. The sparse system solution is obtained by oblique projecting the current solution on the hyperplane generated by two the active row. This method improves the convergence speed compared to other Kaczmarz method with oblique projection.

1. Introduction

Due to the signals in the real world are either sparse themselves or can be approximated by sparse signals, obtaining a sparse solution from a system of equations is of particular importance. This has led to the development of the theory of Compressed Sensing(CS) [2]. Here we use Iterative Hard Thresholding (IHT) Algorithm [5] because it has less computational complexity than other existing methods for CS. Two-greedy subspace Kaczmarz method with oblique projection can basically be used to generate a solution for consistent system of linear equations. But by designing a series of weights, the sparse solution can be obtained for a system of linear equations. The signal reconstruction problem is formulated as follows:

$$\begin{aligned} \min \|x\|_0, \\ \text{s.t : } Ax = y, \end{aligned} \quad (1.1)$$

where $x \in \mathbb{R}^n$ ($n < \infty$) is the signal that we want to reconstruct it, $\|x\|_0$ represents the number of nonzero entries of x , $A \in \mathbb{R}^{m \times n}$ is the measurement matrix, $m \ll n$ and $y \in \mathbb{R}^m$ is the measurement signal. This is a nonconvex optimization problem and has exponential computational complexity.

Due to the NP-hard of Problem (1.1), we consider the following problem, which is a convex problem:

$$\begin{aligned} \min \|x\|_1, \\ \text{s.t : } Ax = y, \end{aligned} \quad (1.2)$$

and plays an important role in CS theory. The IHT algorithm is an iterative method to reconstruct the original signal as follows:

$$x^{t+1} = H_k(x^{(t)} + \mu A^T(y - Ax^{(t)})), \quad (1.3)$$

where μ is the step size in each iteration and H_k is the hard thresholding operator (It sets all but k largest entries (in magnitude) of x to zero).

Keywords: Compressed Sensing, Kaczmarz Method.

AMS Mathematical Subject Classification [2020]: 65F10, 65F50, 15A29..

2. Two-greedy subspace Kaczmarz method with oblique projection

In [3], Two-Greedy Subspace Kaczmarz method is used to solve large-scale systems. In this method, two rows of measurement matrices are selected based on the ordered residual vector and current iterate orthogonal project onto the solution space given by these two rows. We use the Kaczmarz-type method with given oblique projection $d \in \mathbb{R}^n$, defined by

$$x^{k+1} = p_{H_i}^d(x^k). \quad (2.1)$$

The choice of direction is such that it increases the speed of classical Kaczmarz method. In this article, we use the direction

$$d = a_{i_{k+1}} - \frac{\langle a_{i_k}, a_{i_{k+1}} \rangle}{\|a_{i_k}\|} a_{i_k},$$

where a_i is the i -th row of the matrix A .

The index set of non-zero entries of vector x is denoted by $\text{supp}(x)$. For the original signal x with $|\text{supp}(x)| = k$, the support set S is defined as

$$S = \text{supp}_{|\max\{k', n-j+1\}|}(x^{(j)}), \quad (2.2)$$

where S selects $|\max\{k', n-j+1\}|$ large entries of $x^{(j)}$ in magnitude and k' is estimated support ($k' > k$). First we generate the following weight vector.

$$w_l^{(j)} = \begin{cases} 1, & l \in S, \\ \frac{1}{\sqrt{j}}, & l \in S^C. \end{cases} \quad (2.3)$$

The two selected rows s_j and t_j are replaced by $a'_{s_j} = w \odot a_{s_j}$ and $a'_{t_j} = w \odot a_{t_j}$, where \odot is denoted the element-wise product.

The iterative hard thresholding algorithm based on two-greedy subspace Kaczmarz method with oblique projection is expressed in Algorithm 1.

3. Numerical results

To illustrate the performance of the proposed method, we compare it with iterative hard thresholding Sparse Greedy Randomized Kaczmarz with oblique projection [1] and sparse maximal weighted residual Kaczmarz method with oblique projection [4]. We tested the convergence rate of signal reconstruction for the IHT-S2GSKO, IHT-SMWRKO and the IHT-SGRKO algorithms with a 256×512 random measurement matrix A with independent identically distributed (i.i.d.) Gaussian random entries. We tested this method for 100 different signals with sparsity between 30 and 60.

4. Conclusion

In this paper, we used two greedy subspace Kaczmarz method with oblique projection to improve the convergence rate of the iterative hard thresholding algorithm. Numerical results show this well.

Algorithm 1 IHT-S2GSKO

Input: A, b , support estimate k' , stopping threshold η, x^0

Output: \tilde{x} ;

Compute: $r_1 = b - Ax^{(0)}$

$s = \arg \max_{1 \leq i \leq m} |r_1^{(i)}|$ and $t = \arg \max_{i \in [m] \setminus s} |r_1^{(i)}|$

$x^1 = x^0 + \frac{b_s - \langle a_s, x^{(0)} \rangle}{\|a_s\|_2^2} (a_s)^T + \frac{b_t - \langle a_t, x^{(0)} \rangle}{\|a_t\|_2^2} (a_t)^T$

Initialization: $j = 2$

while ($\frac{\|x^{(j+1)} - x^{(j)}\|_2^2}{\|x^{(j)}\|_2^2} \leq \eta$) **do**

$r_j = b - Ax^{(j)}$

Select rows \hat{s}_j and \hat{t}_j that satisfy

$\hat{s}_j = \arg \max_{1 \leq i \leq m} |r_j^{(i)}|$ and $\hat{t}_j = \arg \max_{i \in [m] \setminus s_j} |r_j^{(i)}|$

Set the support estimate $S = \text{supp}_{\max\{k', n-j+1\}}(x^{(j)})$

Generate weight vector $w^{(j)}$ as $w_l^{(j)} = \begin{cases} 1 & l \in S \\ \frac{1}{\sqrt{j}} & l \in S^C \end{cases}$

$a'_l = w^{(j)} \odot a_l, \quad l = t, s, \hat{s}_j, \hat{t}_j$

$D_1 = \langle a'_s, a'_{\hat{s}_j} \rangle$ and $D_2 = \langle a'_{\hat{t}_j}, a'_{\hat{t}_j} \rangle$

$r_1 = b_{\hat{s}_j} - \langle a'_{\hat{s}_j}, x^j \rangle$ and $r_2 = b_{\hat{t}_j} - \langle a'_{\hat{t}_j}, x^j \rangle$

$v^1 = a'_{\hat{s}_j} - \frac{D_1}{\|a'_s\|_2^2} a'_s$ and $v^2 = a'_{\hat{t}_j} - \frac{D_2}{\|a'_{\hat{t}_j}\|_2^2} a'_{\hat{t}_j}$

$h^1 = \|v^1\|$ and $h^2 = \|v^2\|$

$\alpha_1 = \frac{r_{\hat{s}_j}}{h^1}$ and $\alpha_2 = \frac{r_{\hat{t}_j}}{h^2}$

$x^{(j+1)} = x^{(j)} + \alpha_1 v^1 + \alpha_2 v^2$

$s = \hat{s}_j, t = \hat{t}_j$

end while

$\tilde{x} = x^{(j)}$

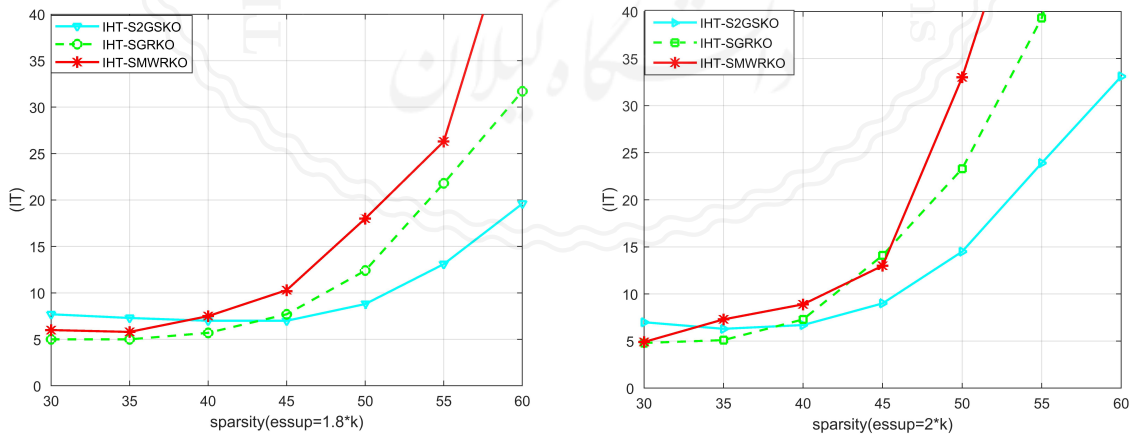


Figure 1: The average convergence rate of the IHT-S2GSKO, IHT-SMWRKO and the IHT-SGRKO method

References

- [1] Z.Z. Bai, W.T. Wu, On greedy randomized Kaczmarz method for solving large sparse linear systems, *SIAM Journal on Scientific Computing*, 40 (2018) A592–A606.

- [2] E.J. Candes T. Tao, Decoding by linear programming, *IEEE transactions on information theory*, 51 (2005) 4203–4215.
- [3] Y. Liu, C. Gu, Two greedy subspace Kaczmarz algorithm for image reconstruction, *IAENG International Journal of Applied Mathematics*, 50 (2020) no. 4.
- [4] S.F. McCormick, The methods of Kaczmarz and row orthogonalization for solving linear equations and least squares problems in Hilbert space, *Indiana University Mathematics Journal*, 26 (1977) 1137–1150.
- [5] Y. Wang, G. Li, An iterative hard thresholding algorithm based on sparse randomized Kaczmarz method for compressed sensing, *International Journal of Computational Intelligence and Applications*, 17 (2018) 1850015.

Farshid Abdollahi

Department of Mathematics, Shiraz University, Shiraz, Iran

email address: abdollahi@shirazu.ac.ir

Fatemeh Pirayesh Dehkordi

Department of Mathematics, Shiraz University, Shiraz, Iran

email address: f.pirayesh@shirazu.ac.ir



University of Guilan



An hp -version collocation method for weakly singular integral equations

Khadijeh Nedaiasl, Raziye Dehbozorgi

Abstract. We investigate the numerical solution of a class of nonlinear first kind integral equation with weakly singular kernel. An hp -version of collocation method based Jacobi polynomials are introduced and applied and the method is properly analyzed. The numerical result for a test problem with singular solution is presented.

1. Introduction

We deal with the numerical solution of the nonlinear weakly singular Volterra integral equation of the first kind

$$\mathcal{K}u(t) := \int_0^t (t-s)^{\alpha-1} \kappa(t,s) \psi(s, u(s)) ds = f(t), \quad 0 < \alpha \leq 1, \quad 0 \leq t \leq T < \infty, \quad (1.1)$$

where $\kappa(t,s)$, $\psi(s, u(s))$ in the kernel and $f(t)$ the right-hand side term are known and $u(t)$ is the unknown to be determined.

The weighted Lebesgue spaces are utilized as the suitable functional spaces. Let define the weight function $\chi^{\alpha,\beta}(x) := (1-x)^\alpha(1+x)^\beta$ on the interval $\Lambda := [-1, 1]$ for $\alpha, \beta > -1$. For $r \in \mathbb{N}$, $H_{\chi^{\alpha,\beta}}^r(\Lambda)$ is a weighted Sobolev space defined by

$$H_{\chi^{\alpha,\beta}}^r(\Lambda) = \left\{ v \mid v \text{ is measurable and } \|v\|_{r,\chi^{\alpha,\beta}} < \infty \right\},$$

where

$$\|v\|_{r,\chi^{\alpha,\beta}} = \left(\sum_{k=0}^r |v|_{k,\chi^{\alpha,\beta}}^2 \right)^{\frac{1}{2}}.$$

The semi-norm is defined as $|v|_{k,\chi^{\alpha,\beta}} = \|\partial_x^k v\|_{\chi^{\alpha+r,\beta+r}}$. For arbitrary real number $r = [r] + \theta$ with $\theta \in (0, 1)$, $H_{\chi^{\alpha,\beta}}^r(\Lambda)$ can be defined by the interpolation space as

$$H_{\chi^{\alpha,\beta}}^r(\Lambda) = [H_{\chi^{\alpha,\beta}}^{[r]}(\Lambda), H_{\chi^{\alpha,\beta}}^{[r]+1}(\Lambda)]_\theta.$$

More details can be seen in [4].

We recall the definition of the Riemann-Liouville integral operator ${}_0\mathcal{I}_x^r$

$${}_0\mathcal{I}_x^r u(x) = \int_0^x (x-t)^{r-1} u(t) dt. \quad (1.2)$$

Keywords: nonlinear operator, first kind Volterra integral equation, weakly singular operator, hp -version collocation method.

AMS Mathematical Subject Classification [2010]: 45H30; 45D05; 65L60; 65L70..

Theorem 1.1. Assume that the Eq. (1.1) satisfies the following assumptions

- i $f(t) \in H_{\chi^{\alpha-1,0}}^m(\Omega)$, $f(0) = 0$,
- ii $\kappa(s, t) \in C^m(\Omega \times \Omega)$ and $\kappa(t, t) \neq 0$ for all $t \in \Omega$,
- iii $\psi(s, u) \in H_{\chi^{\alpha-1,0}}^{m-1}(\Omega \times \mathbb{R})$,
- iv $\inf \left\{ \left| \frac{\partial \psi}{\partial u}(s, u) \right| \mid (s, u) \in \Omega \times \mathbb{R} \right\} \geq M > 0$,
- v $\psi(s, u)$ is Lipschitz continuous w.r. to u ,
- vi let $k(t) = \int_0^t \int_x^t (t-y)^{-\alpha} (y-x)^{\alpha-1} \kappa(y, x) \psi(x, u(x)) dy dx$, then $k \in H_{\chi^{\alpha-1,0}}^{m-1}(\Omega)$.

Then it has a unique solution u in $H_{\chi^{\alpha-1,0}}^{m-1}(\Omega)$.

1.1. Preliminaries

The shifted Jacobi-Gauss interpolation operator. Let us denote the standard Jacobi polynomial of degree k by $J_k^{\alpha,\beta}(x)$, for $\alpha, \beta > -1$. It is well-known that the set of Jacobi polynomials makes a complete orthogonal system with respect to the weight function $\chi^{\alpha,\beta}(x)$ which means that

$$\int_{\Lambda} J_k^{\alpha,\beta}(x) J_j^{\alpha,\beta}(x) \chi^{\alpha,\beta}(x) dx = \gamma_k^{\alpha,\beta} \delta_{k,j}, \quad (1.3)$$

wherein $\delta_{k,j}$ is the Kronecker function, and

$$\gamma_k^{\alpha,\beta} = \begin{cases} \frac{2^{\alpha+\beta+1} \Gamma(\alpha+1) \Gamma(\beta+1)}{\Gamma(\alpha+\beta+2)}, & k = 0, \\ \frac{2^{\alpha+\beta+1}}{2k+\alpha+\beta+1} \frac{\Gamma(k+\alpha+1) \Gamma(k+\beta+1)}{k! \Gamma(k+\alpha+\beta+1)}, & k \geq 1. \end{cases}$$

In order to work with these polynomials on the sub-intervals Ω_n properly, the shifted Jacobi polynomial of degree k is also defined as follows

$$J_{n,k}^{\alpha,\beta}(t) = J_k^{\alpha,\beta}\left(\frac{2t - t_{n-1} - t_n}{h_n}\right), \quad t \in \Omega_n, \quad k \geq 0. \quad (1.4)$$

Let $x_{n,j}^{\alpha,\beta}$ be the zeros of the standard Jacobi polynomial of degree k for $0 \leq j \leq M_n$ and $\omega_{n,j}^{\alpha,\beta}$ be the corresponding Christoffel numbers. Then we can define the shifted Jacobi-Gauss quadrature points on the interval Ω_n as follows

$$t_{n,j}^{\alpha,\beta} = \frac{1}{2}(h_n x_{n,j}^{\alpha,\beta} + t_{n-1} + t_n), \quad 0 \leq j \leq M_n. \quad (1.5)$$

Let $\mathcal{P}_M(\Omega)$ be the set of all polynomials of degree at most M on Ω . It is known from [1, 3] that for any $\phi(t) \in \mathcal{P}_{2M_n+1}(\Omega_n)$

$$\int_{\Omega_n} \phi(t) \chi_n^{\alpha,\beta}(t) dt = \left(\frac{h_n}{2}\right)^{\alpha+\beta+1} \sum_{j=0}^{M_n} \phi(t_{n,j}^{\alpha,\beta}) \omega_{n,j}^{\alpha,\beta}, \quad (1.6)$$

which leads to the result

$$\sum_{j=0}^{M_n} J_{n,p}^{\alpha,\beta}(t_{n,j}^{\alpha,\beta}) J_{n,q}^{\alpha,\beta}(t_{n,j}^{\alpha,\beta}) \omega_{n,j}^{\alpha,\beta} = \gamma_p^{\alpha,\beta} \delta_{p,q}. \quad (1.7)$$

For any $v \in C(\Omega_n)$, the shifted Jacobi-Gauss interpolation operator in the t -direction is defined as follows

$$\mathcal{I}_{t,M_n}^{\alpha,\beta} v(t_{n,j}^{\alpha,\beta}) = v(t_{n,j}^{\alpha,\beta}), \quad 0 \leq j \leq M_n, \quad (1.8)$$

and the following lemma reports an upper bound for the interpolation by (1.8).

Lemma 1.2. ([5]) For any $v \in H_{\chi_n^{\alpha,\beta}}^m(\Omega_n)$ with integer $1 \leq m \leq M_n + 1$ and $\alpha, \beta > -1$, we get

$$\|v - \mathcal{I}_{x,M_n}^{\alpha,\beta} v\|_{\chi_n^{\alpha,\beta}} \leq c \sqrt{\frac{\Gamma(M_n + 2 - m)}{\Gamma(M_n + 2 + m)}} \|\partial_x^m v\|_{\chi_n^{\alpha+m,\beta+m}}.$$

In particular, for any fixed m , we obtain

$$\|v - \mathcal{I}_{x,M_n}^{\alpha,\beta} v\|_{\chi_n^{\alpha,\beta}} \leq c(M_n + 1)^{-m} \|\partial_x^m v\|_{\chi_n^{\alpha+m,\beta+m}} \leq ch_n^m (M_n + 1)^{-m} \|\partial_x^m v\|_{\chi_n^{\alpha,\beta}}.$$

1.2. The hp -collocation method for weakly singular integral equations

For a fixed integer N , let $\Omega_h := \{t_n : 0 = t_0 < t_1 < \dots < t_N = T\}$ be as a mesh on Ω , $h_n := t_n - t_{n-1}$ and $h_{\max} = \max_{1 \leq n \leq N} h_n$. Moreover, denote $u^n(t)$ as the solution of Eq. (1.1) on the n -th subinterval of Ω , namely,

$$u^n(t) = u(t), \quad t \in \Omega_n := (t_{n-1}, t_n], \quad n = 1, 2, \dots, N.$$

By the above mesh, we rewrite the Eq. (1.1) as

$$\int_0^{t_{n-1}} (t-s)^{\alpha-1} \kappa(s, t) \psi(s, u(s)) ds + \int_{t_{n-1}}^t (t-s)^{\alpha-1} \kappa(s, t) \psi(s, u(s)) ds = f(t),$$

then for any $t \in \Omega_n$, this equation can be written as

$$\int_{t_{n-1}}^t (t-\tau)^{\alpha-1} \kappa(\tau, t) \psi(\tau, u^n(\tau)) d\tau = f(t) - \sum_{k=1}^{n-1} \int_{\Omega_k} (t-s)^{\alpha-1} \kappa(s, t) \psi(s, u^k(s)) ds. \quad (1.9)$$

Now, we transfer the interval (t_{n-1}, t) to Ω_n by the following linear transform

$$\tau = \sigma(\lambda, t) := t_{n-1} + \frac{(\lambda - t_{n-1})(t - t_{n-1})}{h_n}, \quad (1.10)$$

to get

$$\begin{aligned} & \left(\frac{t - t_{n-1}}{h_n}\right)^\alpha \int_{\Omega_n} (t_n - \lambda)^{\alpha-1} \kappa(\sigma(\lambda, t), t) \psi(\sigma(\lambda, t), u^n(\sigma(\lambda, t))) d\lambda = \\ & f(t) - \sum_{k=1}^{n-1} \int_{\Omega_k} (t-s)^{\alpha-1} \kappa(s, t) \psi(s, u^k(s)) ds. \end{aligned} \quad (1.11)$$

In the following, we mention some requirements considered in the next section. Let $\mathcal{I}_{\lambda, M_n}^{\alpha-1,0} : C(\Omega_n) \rightarrow \mathcal{P}_{M_n}(\Omega_n)$ be the Jacobi-Gauss interpolation operator. Now, we define a new Legendre-Gauss interpolation operator $\mathcal{I}_{\tau, M_n}^{\alpha-1,0} : C(t_{n-1}, t) \rightarrow \mathcal{P}_{M_n}(t_{n-1}, t)$ owing to the relation (1.10) with the following property

$$\mathcal{I}_{\tau, M_n}^{\alpha-1,0} g(\tau_{n,i}) = g(\tau_{n,i}), \quad 0 \leq i \leq M_n,$$

where $\tau_{n,i} := \tau_{n,i}(x) = \sigma(\lambda_{n,i}, t)$ and $\lambda_{n,i}$ are the $M_n + 1$ Jacobi-Gauss quadrature nodes in Ω_n . Clearly,

$$\mathcal{I}_{\tau, M_n}^{\alpha-1,0} g(\tau_{n,i}) = g(\sigma(\lambda_{n,i}, t)) = \mathcal{I}_{\lambda, M_n}^{\alpha-1,0} g(\sigma(\lambda_{n,i}, t)), \quad 0 \leq i \leq M_n,$$

and by Eq. (1.6), we get

$$\int_{t_{n-1}}^t (t-\tau)^{\alpha-1} \mathcal{I}_{\tau, M_n}^{\alpha-1,0} g(\tau) d\tau = \left(\frac{t-t_{n-1}}{2}\right)^{\alpha} \sum_{j=0}^{M_n} g(\tau_{n,j}) w_{n,j}. \quad (1.12)$$

Meanwhile, it is noticed that

$$\int_{t_{n-1}}^t (t-\tau)^{\alpha-1} (\mathcal{I}_{\tau, M_n}^{\alpha-1,0} g(\tau))^2 d\tau = \left(\frac{t-t_{n-1}}{2}\right)^{\alpha} \sum_{j=0}^{M_n} g^2(\tau_{n,j}) w_{n,j}. \quad (1.13)$$

These equations will be valid for the Legendre interpolation operator $\mathcal{I}_{M_n}^t$, if we take $\alpha = 1$ and $t = t_n$.

1.2.1 The hp -version of Jacobi-Gauss collocation method

In order to seek the solution $u_{M_n}^n(t) \in \mathcal{P}_{M_n}(\Omega_n)$ of Eq. (1.11) by hp -collocation method, at the first step this equation is fully discretized as

$$\begin{aligned} & \mathcal{I}_{M_n}^t \left(\left(\frac{t-t_{n-1}}{h_n} \right)^{\alpha} \int_{\Omega_n} (t_n - \lambda)^{\alpha-1} \mathcal{I}_{\lambda, M_n}^{\alpha-1,0} \kappa(\sigma(\lambda, t), t) \psi(\sigma(\lambda, t), u_{M_n}^n(\sigma(\lambda, t))) d\lambda \right) \\ & = \mathcal{I}_{M_n}^t (f(t)) - \mathcal{I}_{M_n}^t \left(\sum_{k=1}^{n-1} \int_{\Omega_k} (t-s)^{\alpha-1} \mathcal{I}_{s, M_k}^L \kappa(s, t) \psi(s, u_{M_k}^k(s)) ds \right), \quad t \in \Omega_n, \end{aligned} \quad (1.14)$$

where

$$\begin{aligned} \mathcal{I}_{M_n}^t u^n(t) &= u_{M_n}^n(t) = \sum_{p=0}^{M_n} \hat{u}_p^n L_{n,p}(t), \\ \mathcal{I}_{M_n}^t \mathcal{I}_{\lambda, M_n}^{\alpha-1,0} \left(\left(\frac{t-t_{n-1}}{h_n} \right)^{\alpha} \kappa(\sigma(\lambda, t), t) \psi(\sigma(\lambda, t), u_{M_n}^n(\sigma(\lambda, t))) \right) &= \sum_{p,q=0}^{M_n} a_{pq}^n L_{n,p}(t) J_{n,q}^{\alpha-1,0}(\lambda), \\ \sum_{k=1}^{n-1} \mathcal{I}_{M_n}^t \mathcal{I}_{s, M_k}^L \left(\int_{\Omega_k} (t-s)^{\alpha-1} \kappa(s, t) \psi(s, u_{M_k}^k(s)) ds \right) &= \sum_{k=1}^{n-1} \mathcal{I}_{M_n}^t \left(\sum_{q=0}^{M_k} \tilde{w}_{k,q}^L(t) \kappa(t_{k,q}^L, t) \psi(t_{k,q}^L, u_{M_k}^k(t_{k,q}^L)) \right) \\ &= \sum_{p=0}^{M_n} \sum_{k=1}^{n-1} \sum_{q=0}^{M_k} b_{pq}^k L_{n,p}(t), \end{aligned} \quad (1.15)$$

and

$$\mathcal{I}_{M_n}^t f(t) = \sum_{p=0}^{M_n} \hat{f}_p^n L_{n,p}(t). \quad (1.16)$$

Then, we get

$$\int_{\Omega_n} \frac{(t_n - \lambda)^{\alpha-1}}{h_n^{\alpha}} \mathcal{I}_{M_n}^t \mathcal{I}_{\lambda, M_n}^{\alpha-1,0} \left(\left(\frac{t-t_{n-1}}{h_n} \right)^{\alpha} \kappa(\sigma(\lambda, t), t) \psi(\sigma(\lambda, t), u_{M_n}^n(\sigma(\lambda, t))) \right) d\lambda = \mathcal{I}_{M_n}^t f(t). \quad (1.17)$$

It is evident from Eqs. (1.15)-(1.17) that

$$\begin{aligned}\hat{u}_p^n &= \frac{2p+1}{2} \sum_{i=0}^{M_n} u_{M_n}^n(t_{n,i}) L_{n,p}(t_{n,i}) w_{n,i}, \\ \hat{a}_{p0}^n &= \frac{2p+1}{2^{1+\alpha}} \sum_{i,j=0}^{M_n} (t_{n,i} - t_{n-1})^\alpha \kappa(\sigma(t_{n,j}^{\alpha-1,0}, t_{n,i}), t_{n,i}) \psi(\sigma(t_{n,j}^{\alpha-1,0}, t_{n,i}), u_{M_n}^n(\sigma(t_{n,j}^{\alpha-1,0}, t_{n,i}))) \\ &\quad L_{n,p}(t_{n,i}) w_{n,i} w_{n,j}^{\alpha-1,0}, \\ b_{pq}^k &= \frac{2p+1}{2} \sum_{i=0}^{M_n} \tilde{w}_{k,q}^L(t_{n,i}) \kappa(t_{k,q}^L, t_{n,i}) \psi(t_{k,q}^L, u_{M_k}^k(t_{k,q}^L)) L_{n,p}(t_{n,i}) w_{n,i}, \\ \hat{f}_p^n &= \frac{2p+1}{2} \sum_{i=0}^{M_n} f_{M_n}^n(t_{n,i}) L_{n,p}(t_{n,i}) w_{n,i}.\end{aligned}\tag{1.18}$$

With Eqs. (1.15)-(1.17), the equation (1.14) reads

$$\sum_{p=0}^{M_n} \hat{a}_{p0}^n L_{n,p}(t) = \sum_{p=0}^{M_n} \hat{f}_p^n L_{n,p}(t) + \sum_{p=0}^{M_n} \tilde{b}_p^n L_{n,p}(t),$$

where

$$\tilde{b}_p^n = \sum_{k=1}^{n-1} \sum_{q=0}^{M_k} b_{pq}^k.$$

Consequently, we compare the coefficients to obtain

$$\hat{a}_{p0}^n = \hat{f}_p^n + \tilde{b}_p^n, \quad 0 \leq p \leq M_n.\tag{1.19}$$

To evaluate the unknown coefficients u_p^n for any given n , we solve the nonlinear system (1.19) with the Newton iteration method. Finally, the approximate solution can be obtained as

$$u_M^N(t) = \sum_{n=1}^N \sum_{p=0}^{M_n} u_p^n L_{n,p}(t).\tag{1.20}$$

It is worth to notice that for the linear case of Eq. (1.1), the unknown coefficients \hat{u}_p^n for any given n can be obtained by the following linear system of equations

$$A\mathbf{u} = \mathbf{b} + \mathbf{c},\tag{1.21}$$

where the entries of the matrix $A = [a_{p,q}]_{p,q=0}^{M_n}$ are defined by

$$a_{p,q} = \frac{2p+1}{2^{1+\alpha}} \sum_{i,j=0}^{M_n} (t_{n,i} - t_{n-1})^\alpha \kappa(\sigma(t_{n,j}^{\alpha-1,0}, t_{n,i}), t_{n,i}) L_{n,q}(\sigma(t_{n,j}^{\alpha-1,0}, t_{n,i})) L_{n,p}(t_{n,i}) w_{n,i} w_{n,j}^{\alpha-1,0},$$

and

$$\mathbf{u} = (\hat{u}_0^n, \dots, \hat{u}_{M_n}^n)^T, \quad \mathbf{b} = (\tilde{b}_0^n, \dots, \tilde{b}_{M_n}^n)^T, \quad \mathbf{c} = (\hat{f}_0^n, \dots, \hat{f}_{M_n}^n)^T.$$

2. Some results on the analysis of the method

Theorem 2.1. Let u^n be the solution of Eq. (1.11) under the hypothesis of Theorem 1.1 and $u_{M_n}^n$ be the solution of Eq. (1.14). According the assumptions of Theorem 1.1, the function $\psi(\cdot, u)$ fulfills the Lipschitz condition with respect to the second variable, i.e.,

$$|\psi(\cdot, u_1) - \psi(\cdot, u_2)| \leq \gamma |u_1 - u_2|, \quad \gamma \geq 0. \quad (2.1)$$

Then, for any $1 \leq n \leq N$ and $m \leq M_{\min} + 1$,

$$B_1 = B_2 + B_3,$$

with

$$\begin{aligned} \|B_1\|_{\Omega_n}^2 &\leq ch_n T^{2\alpha-1} \sum_{k=1}^{n-1} \left(h_k^{2m} (M_k + 1)^{-2m} \|\partial_s^m \psi(s, u^k(s))\|_{\Omega_k}^2 \right. \\ &\quad \left. + \gamma^2 (\|e_k\|_{\Omega_k}^2 + h_k^{2m-1} (M_k + 1)^{-2m} \|\partial_t^m u\|_{\Omega_k}^2) \right) + ch_n^{2m} (M_n + 1)^{-2m} \|\partial^m f\|_{\Omega_n}^2, \end{aligned} \quad (2.2)$$

where

$$\begin{aligned} B_1 &= \mathcal{I}_{M_n}^t \left(\left(\frac{t - t_{n-1}}{h_n} \right)^\alpha \int_{\Omega_n} (t_n - \lambda)^{\alpha-1} (\mathcal{I}_{\lambda, M_n}^{\alpha-1, 0} (\kappa(\sigma(\lambda, t), t) \psi(\sigma(\lambda, t), u_{M_n}^n(\sigma(\lambda, t)))) \right. \\ &\quad \left. - \kappa(\sigma(\lambda, t), t) \psi(\sigma(\lambda, t), u^n(\sigma(\lambda, t)))) d\lambda \right), \\ B_2 &= f(t) - \mathcal{I}_{M_n}^t f(t), \\ B_3 &= \sum_{k=1}^{n-1} \mathcal{I}_{M_n}^t \left(\int_{\Omega_k} (t - s)^{\alpha-1} (\kappa(s, t) \psi(s, u^k(s)) - \mathcal{I}_{s, M_k}^L (\kappa(s, t) \psi(s, u_{M_k}^k(s)))) ds \right), \end{aligned} \quad (2.3)$$

and $e_k = u^k - u_{M_k}^k$ for $1 \leq k \leq N$.

Theorem 2.2. Assume that the Fréchet derivative of the operator Ku with respect to u is satisfied at $|(K'u)(t)| \geq l > 0$, then under the hypothesis of the Theorem 2.1, for sufficiently small h_{\max} the following error estimate is obtained

$$\begin{aligned} \|e_n\|^2 = \|u^n - u_{M_n}^n\|^2 &\leq \frac{c_\alpha}{\delta^2} \exp(c\gamma^2 T^{2\alpha}) \left(T^{2\alpha-1} \sum_{k=1}^{n-1} \left(h_k^{2m} (M_k + 1)^{-2m} \|\partial_s^m \psi(s, u(s))\|_{\Omega_k}^2 \right. \right. \\ &\quad \left. \left. + \gamma^2 h_k^{2m} (M_k + 1)^{-2m} \|\partial_t^m u\|_{\Omega_k}^2 \right) + h_n^{2m-1} (M_n + 1)^{-2m} (\|\partial^m f\|_{\Omega_n}^2 \right. \\ &\quad \left. + h_n^{2\alpha} \|\psi(\cdot, u_M^N(\cdot))\|_{H_{\chi^{\alpha-1,0}}^m(\Omega_n)}^2 + h_n^{2m+\alpha} (M_n + 1)^{-2m} (\gamma^2 \|u\|_{H_{\chi^{\alpha-1,0}}^m(\Omega_n)}^2 \right. \\ &\quad \left. + \|\psi(\cdot, u(\cdot))\|_{H_{\chi^{\alpha-1,0}}^m(\Omega_n)}^2) \right). \end{aligned} \quad (2.4)$$

Theorem 2.3. Assume that $u(t)$ be the exact solution of Eq. (1.1) and $u_M^N(t)$ be the global approximate solution obtained from Eq. (1.20). Under the hypothesis of Theorem 2.2, the following error estimate can be derived for sufficiently small h_{\max} as

$$\begin{aligned} \|u - u_M^N\|_\Omega &\leq \frac{c_\alpha}{\delta} \exp(c\gamma^2 T^{2\alpha}) h_{\max}^m (M_{\min} + 1)^{-m} \left(T^\alpha (\gamma \|\partial_t^m u\|_\Omega + \|\partial_s^m \psi(s, u(s))\|_\Omega) + \|\partial^m f\|_\Omega \right. \\ &\quad \left. + \gamma \|u\|_{H_{\chi^{\alpha-1,0}}^m(\Omega)} + \|\psi(\cdot, u(\cdot))\|_{H_{\chi^{\alpha-1,0}}^m(\Omega)} + h_{\max}^\alpha \|\psi(\cdot, u_M^N(\cdot))\|_{H_{\chi^{\alpha-1,0}}^m(\Omega)} \right). \end{aligned} \quad (2.5)$$

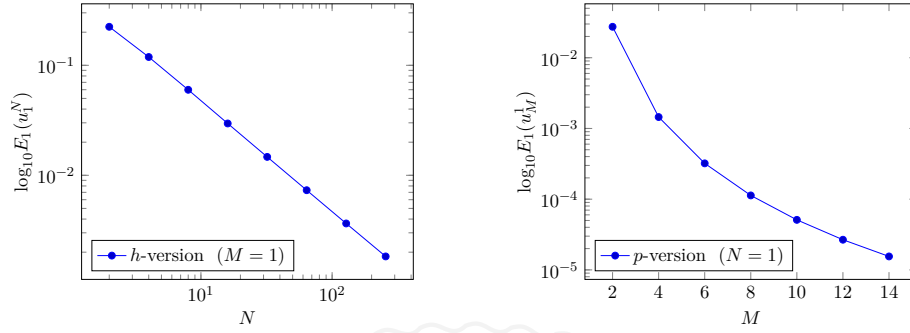


Figure 1: Plots of the $E_1(u_M^N)$ error in logarithmic scale for the h - and p -version collocation methods with $\alpha = 0.3$ for Example 2.4.

2.1. Numerical results

Example 2.4. Consider a test problem with singular solution

$$\int_0^t (t-s)^{\alpha-1} \exp(ts) u^2(s) ds = \left(\frac{1}{t}\right)^{-2\alpha} t^{2+\alpha} \Gamma(3+2\alpha) \Gamma(\alpha) {}_1F_1(3+2\alpha, 3+3\alpha, t^2), \quad t \in [0, 1],$$

where the function ${}_1F_1$ is called confluent hypergeometric function of the first kind. The exact solution $u(t) = t^{1+\alpha}$ belongs to $H_{\alpha-1,0}^2([0, 1])$. In this example, different merits of the hp -method are investigated. First and foremost, the superiority of the hp -version method against h - and p -version method with $\alpha = 0.3$ is demonstrated by Figure 1 and 2. The hp -version method allows us to adjust the parameters M and N to achieve the suitable solution. Figure 1 depicts h - and p -version methods in which the values of parameters M and N are equal to 1, respectively. Figure 2 shows the hp -version collocation method for each fixed $N = 1, 2, 4, 8$ when $h_n = h = \frac{1}{N}$ and various values of $M_n = M^*$ for $n = 1, \dots, N$.

Secondly, in order to compare the theoretical and the numerical solution, we consider hp -version with $M = 2$ and various N . Therefore, it is expected to have a rate near 2; namely, $\rho_N \approx m = 2$. This expectation is experimentally verified and shown in the left sub-figure of Figure 3.

Finally, we consider different values for α . According to Theorem 2.3, increasing the values of α affirmatively affects on the convergence rate which is verified by the numerical results on the right sub-figure of Figure 3.

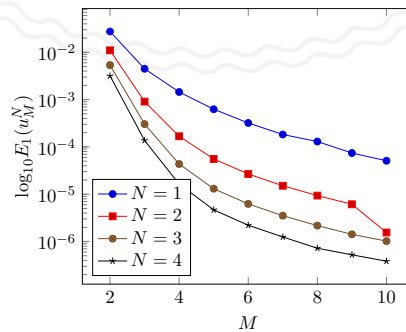


Figure 2: Plots of the $E_1(u_M^N)$ error in logarithmic scale for the hp -version collocation method with $\alpha = 0.3$ for Example 2.4.

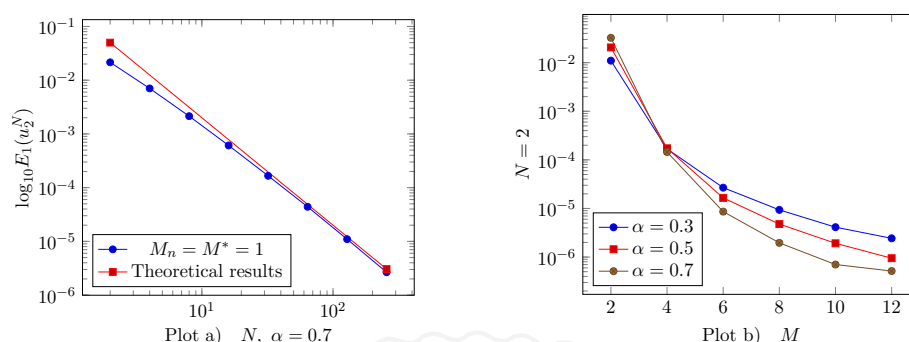


Figure 3: Plots of the $E_1(u_M^N)$ error in logarithmic scale: a) Comparison between theoretical and numerical results b) The results of hp -method for Example 2.4 for various α .

References

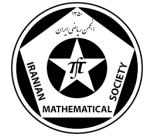
- [1] G. Ben-Yu, W. Li-Lian, Jacobi interpolation approximations and their applications to singular differential equations. *Advances in Computational Mathematics*, 14 (2001) 227–276.
- [2] R. Dehbozorgi, K. Nedaiasl, Numerical solution of nonlinear weakly singular Volterra integral equations of the first kind: An hp -version collocation approach. *Applied numerical mathematics* 161 (2021) 111–136.
- [3] B.Y. Guo, L.L. Wang, Jacobi approximations in non-uniformly Jacobi-weighted Sobolev spaces. *Journal of Approximation Theory*, 128 (2004) 1–41.
- [4] J.L. Lions, E. Magenes, *Non-homogeneous boundary value problems and applications*, Springer Science & Business Media, 2012.
- [5] Z.q. Wang, Y.l. Guo, L.j. Yi, An hp -version Legendre-Jacobi spectral collocation method for Volterra integro-differential equations with smooth and weakly singular kernels. *Mathematics of Computation*, 86 (2017) 2285–2324.

Khadijeh Nedaiasl
 Department of Mathematics, IASBS, Zanjan, Iran
 email address: knedaiasl85@gmail.com & nedaiasl@iasbs.ac.ir

Raziyeh Dehbozorgi
 Department of Mathematics, IASBS, Zanjan, Iran
 email address: r.dehbozorgi2012@gmail.com



University of Guilan



A numerical method based on Daubechies wavelet to solve a class of FDEs

Nasim Madah Shariati

Abstract. Solving Fractional Differential Equations(FDEs) is an important matter in various fields of sciences. In this paper, by using an extension of Block Pulse functions named quasi Block Pulse functions, fractional integration operational matrix of scaling functions of the Daubechies wavelet is introduced and through that a class of FDEs is solved numerically and acceptable results are obtained.

1. Introduction

Fractional Differential Equations (FDES) have a significant role in numerous fields of sciences. Some numerical methods have been used for solving FDEs such as Spectral Method [1], and wavelet [2]. In 1988 Ingrid Daubechies made an orthonormal basis from smooth wavelet with vanishing moments of order N and compactly support for scaling function which is $[0, 2N - 1]$. In this paper, we define quasi Block Pulse functions and by helping them, we introduce fractional integration operational matrix for scaling function of the Daubechies wavelet and through that we solve a class of FDEs.

Let ϕ as the scaling function of the Daubechies wavelet. Assume that, we use vanishing moments of order N . If $f \in V_j$ of multiresolution analysis and $t \in [a, b]$ that $a, b \in \mathbb{Z}$, in this case the function $f(t) \in L^2(\mathbb{R})$ can be approximated by

$$\mathcal{P}_j f(t) = \sum_{k=2^j a+2-2N}^{2^j b-1} c_{j,k} \phi_{j,k}(t), \quad a \leq t \leq b,$$

such that $\{\phi_{j,k} := 2^{j/2} \phi(2^j t - k)\}$ forms a base for subspace V_j of multiresolution analysis and [3]

$$c_{j,k} = \int_0^{2N-1} f(t) \phi_{j,k}(t) dt.$$

2. Quasi Block Pulse Functions

These functions are similar to Block Pulse functions, with the difference that grid points are arbitrarily selected that are not necessarily equidistant.

Keywords: Fractional differential equation, Scaling function, Daubechies wavelet, Quasi Block Pulse functions, Fractional integration operational matrix.

AMS Mathematical Subject Classification [2010]: 33F05, 34A08, 34A45.

Definition 2.1. For $m \in \mathbb{N}$ we define an m element partition of quasi Block Pulse functions on $[0, T)$:

$$b_i(t) = \begin{cases} 1, & t_i \leq t < t_{i+1}, \\ 0, & \text{otherwise,} \end{cases} \quad (2.1)$$

where $i = 1, \dots, m$, and $0 = t_1 < t_2 < \dots < t_m < t_{m+1} = T$.

The disjointness and orthogonality properties are hold for the functions $\{b_i\}_{i=1}^m$ defined in (2.1). In order to approximate functions, we use truncated series of QBP functions with m terms in the following way:

$$f(t) \approx \sum_{i=1}^m f_i b_i(t) = F^T \mathbf{B}_m(t) = \mathbf{B}_m(t)^T F, \quad (2.2)$$

where $F = [f_1, f_2, \dots, f_m]^T$ and $\Delta_i t = t_{i+1} - t_i$, we have $f_i = \frac{1}{\Delta_i t} \int_0^T f(t) b_i(t) dt$.

2.1. Fractional Integration Operational Matrix for the QBP Functions

We can write the Riemann-Liouville integral operator of order $\alpha > 0$ in the following form

$$I^\alpha f(t) = \frac{1}{\Gamma(\alpha)} t^{\alpha-1} * f(t), \quad 0 \leq t < T, \quad (2.3)$$

notation $*$ means the convolution product. From (2.2), we have

$$I^\alpha f(t) = \frac{1}{\Gamma(\alpha)} t^{\alpha-1} * f(t) \approx F^T \frac{1}{\Gamma(\alpha)} \{t^{\alpha-1} * \mathbf{B}_m(t)\}.$$

Taking the Laplace transform and then taking inverse Laplace transform results in

$$\frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} b_i(\tau) d\tau = \frac{1}{\Gamma(\alpha+1)} \left\{ (t-t_i)^\alpha u(t-t_i) - (t-t_{i+1})^\alpha u(t-t_{i+1}) \right\}, \quad (2.4)$$

here $u(t)$ is the unit step function. From definition of the QBP functions, for $j = 1, \dots, m$, we can write

$$(t-t_j)^\alpha u(t-t_j) = \sum_{i=1}^m d_{i,j} b_i(t) = [0, \dots, 0, d_{j,j}, d_{j+1,j}, \dots, d_{m,j}] \mathbf{B}_m(t),$$

where

$$d_{i,j} = \frac{1}{\Delta_i t} \int_{t_i-t_j}^{t_{i+1}-t_j} t^\alpha u(t) dt = \frac{(t_{i+1}-t_j)^{\alpha+1} - (t_i-t_j)^{\alpha+1}}{\Delta_i t(\alpha+1)}, \quad i, j = 1, \dots, m.$$

So from (2.4) we can write

$$\begin{aligned} \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} b_i(\tau) d\tau = \\ \frac{1}{\Gamma(\alpha+1)} [0, 0, \dots, 0, d_{i,i}, d_{i+1,i} - d_{i+1,i+1}, d_{i+2,i} - d_{i+2,i+1}, \dots, d_{m,i} - d_{m,i+1}] \mathbf{B}_m(t), \end{aligned}$$

and finally we conclude

$$I^\alpha \mathbf{B}_m(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \mathbf{B}_m(\tau) d\tau = \mathbf{F}_\alpha \mathbf{B}_m(t), \quad (2.5)$$

where

$$\mathbf{F}_\alpha = \frac{1}{\Gamma(\alpha+2)} \begin{bmatrix} \bar{d}_{1,1} & \bar{d}_{2,1} - \bar{d}_{2,2} & \bar{d}_{3,1} - \bar{d}_{3,2} & \dots & \bar{d}_{m,1} - \bar{d}_{m,2} \\ & \bar{d}_{2,2} & \bar{d}_{3,2} - \bar{d}_{3,3} & \dots & \bar{d}_{m,2} - \bar{d}_{m,3} \\ & & \bar{d}_{3,3} & \dots & \bar{d}_{m,3} - \bar{d}_{m,4} \\ & & & \ddots & \vdots \\ \mathbf{0} & & & & \bar{d}_{m,m} \end{bmatrix},$$

and

$$\bar{d}_{i,j} = \frac{(t_{i+1} - t_j)^{\alpha+1} - (t_i - t_j)^{\alpha+1}}{t_{i+1} - t_i}.$$

3. Operational Matrix of the Scaling function of the Daubechies Wavelet

Consider vector of the scaling functions for the Daubechies wavelet

$$\phi_m(t) = [\phi_{j,k_1}(t), \phi_{j,k_1+1}(t), \dots, \phi_{j,k_2}(t)]^T,$$

where $k_2 - k_1 = m - 1$. Grid points are in the form of $\mathbf{T} = \{t_i | i = 1, \dots, m\}$, and let matrix of the scaling functions of the Daubechies wavelet by

$$\Phi_m = [\phi_m(t_1), \phi_m(t_2), \dots, \phi_m(t_m)].$$

In grid points, we have

$$\phi_m(t) = \Phi_m \mathbf{B}_m(t), \quad (3.1)$$

and hence

$$\mathbf{B}_m(t) = (\Phi_m)^{-1} \phi_m(t). \quad (3.2)$$

3.1. Fractional Integration Operational Matrix for the Scaling Functions of the Daubechies Wavelet

Assume that fractional integration operational matrix for the scaling functions of the Daubechies wavelet has the following form

$$I^\alpha \phi_m(t) \approx \mathbf{P}_\alpha \phi_m(t). \quad (3.3)$$

In the grid points, from (2.5), (3.1), and (3.2), we can write

$$I^\alpha \phi_m(t) = I^\alpha \Phi_m \mathbf{B}_m(t) = \Phi_m I^\alpha \mathbf{B}_m(t) = \Phi_m \mathbf{F}_\alpha \mathbf{B}_m(t) = \Phi_m \mathbf{F}_\alpha (\Phi_m)^{-1} \phi_m(t).$$

So, we have

$$\mathbf{P}_\alpha = \Phi_m \mathbf{F}_\alpha (\Phi_m)^{-1}. \quad (3.4)$$

4. Numerical Method

In this section, we declare a numerical method based on the scaling functions of the Daubechies wavelet of order 2 named Daubechies Wavelet Method(DWM) to solve the FDEs in the form of

$${}_a \mathcal{D}_t^\alpha y(t) + f(t, y(t)) = 0, \quad a \leq t < T. \quad (4.1)$$

The notation \mathcal{D} points to Caputo fractional derivative and $\alpha \in \mathbb{Q}$. The initial conditions are

$$y^{(k)}(0) = 0, \quad k = 0, 1, \dots, [\alpha] - 1, \quad (4.2)$$

and we let $a = 0$ and $T = 1$. First, we let

$$\mathcal{D}^\alpha y(t) = \sum_{k=k_1}^{k_2} c_{j,k} \phi_{j,k}(t) = C^T \phi_m(t), \quad (4.3)$$

where $k_1 = 2 - 2N$, $k_2 = 2^j - 2$, and coefficients $c_{j,k}$'s should be determined. Then, by using the Riemann-Liouville integral operator of order α and imposing initial conditions, we get

$$y(t) = C^T I^\alpha \phi_m(t) = C^T \mathbf{P}_\alpha \phi_m(t). \quad (4.4)$$

By substituting (4.3), (4.4) in Eq. (4.1) and using (3.1) and (3.4), and considering dyadic points in interval $[0,1]$ as grid points, we get a linear or nonlinear system with $c_{j,k}$'s as unknowns. By solving this system and then using (4.4) the numerical solution is obtained.

5. Main results

Example 5.1. First, we solve the linear FDE

$$\mathcal{D}^\alpha y(t) + y(t) = t^4 - \frac{1}{2}t^3 - \frac{3}{\Gamma(4-\alpha)}t^{3-\alpha} + \frac{24}{\Gamma(5-\alpha)}t^{4-\alpha}, \quad 0 < \alpha < 1, \quad (5.1)$$

with initial conditions $y(0) = 0$, and $\alpha = 1/4$. The exact solution of Eq. (5.1) is [2]: $y(t) = t^4 - \frac{1}{2}t^3$. Results are indicated in Table 1.

Table 1: Maximum of absolute error in different resolutions, Example 5.1.

$\ e_j(t)\ _\infty$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
[2]	0.0045	0.0018	0.0007	0.0002
DWM	0.0028	9.4756e-04	2.7797e-04	7.6733e-05

Example 5.2. Now, we solve the nonlinear FDE

$$\mathcal{D}^\alpha y(t) + y^2(t) = t + \left(\frac{t^{\alpha+1}}{\Gamma(\alpha+2)} \right)^2, \quad 0 < \alpha \leq 2, \quad (5.2)$$

with initial conditions $y(0) = y'(0) = 0$. The exact solution of Eq. (5.2) is [1]: $y(t) = \frac{t^{\alpha+1}}{\Gamma(\alpha+2)}$. We solve for $\alpha = 1.5$. Results are indicated in Table 2.

Table 2: Maximum of absolute error in different resolutions, Example 5.2.

$\ e_j(t)\ _\infty$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
DWM	3.8802e-03	1.0023e-03	2.5399e-04	6.3883e-05

References

- [1] D. Baleanu, A. H. Bhrawy, T. M. Taha, A Modified Generalized Laguerre Spectral Method for Fractional Differential Equations on the Half Line, *Abstract and Applied Analysis*, 2013.
- [2] Y. Chen, M. Yi, C. Yu, Error analysis for numerical solution of fractional differential equation by Haar wavelets method, *Journal of Computational Science*, 3 (2012) 367–373.
- [3] W. Sweldens, R. Piessens, Quadrature formula and asymptotic error expansions for wavelet approximation of smooth functions, *SIAM Journal on Numerical Analysis*, 31 (1994) 1240–1264.

Nasim Madah Shariati

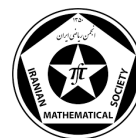
Department of Applied Mathematics and Computer Sciences, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

email address: nasim.maddah62@gmail.com





University of Guilan



Modified Ostrowski method for solving nonlinear equations and its dynamic

Vali Torkashvand, Manochehr Kazemi, Elham Hashemizadeh

Abstract. We have constructed a family of fourth-order methods which use three evaluations of $f(x_k)$, $f(w_k)$, and $f(y_k)$. They have an efficiency index equal to $4^{\frac{1}{3}} = 1.5874$ and are optimal in the sense of Kung and Traub's conjecture. Moreover, the dynamics of the proposed methods are displayed with some comparisons to other existing methods. Numerical comparison with existing optimal methods implies that the new class supplies a practical alternative for solving nonlinear equations.

1. Introduction

Most of the Mathematical problems that arise in science and engineering are very hard and sometime impossible to solve exactly. Therefore, it is indispensable to calculate approximate solutions based on numerical methods. The celebrated Newton's method which can be defined as $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$, is one of the oldest and the most applicable methods in the literature. This method has locally quadratically convergence for the simple roots and per iteration requires one evaluation of the function and its first derivative. Hence, many researchers have focused on constructing methods that do not require functional derivative evaluation (Steffensen-like methods) and have a higher degree of convergence than Newton's method. Ostrowski introduces the first optimal two-point method [5]. His method has a better efficiency index than Newton's method as follows.

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, k = 0, 1, 2, \dots, \\ x_{k+1} = y_k - \frac{\frac{f(x_k)}{f'(x_k)}}{2\frac{f(y_k)}{f'(y_k)} - \frac{f(x_k)}{f'(x_k)}}. \end{cases} \quad (1.1)$$

But these methods have a major weakness, one has to calculate the derivative of $f(x)$ at each approximation. A family of Steffensen like methods was derived in [2, 7, 8, 10] free from derivatives.

In this work, we turn Ostrowski's method into a Steffensen-like and solve the problem of computing the derivative function by the divided difference. The construction of the proposed class is based on the weight function approach.

The rest of the paper is organized as follows: We describe the structure of the without memory methods in Section 2. The numerical study presented in Section 3 confirms the theoretical results. We compare the basin of attraction of the proposed method with several existing methods in Section 4. Finally, we give the concluding remarks.

2. The Method and Its Convergence

To derive new methods, we approximate $f'(x_n)$ given in one-step (1.1) as follows:

$$w_k = x_k + \beta f(x_k), f'(x_k) \approx f[w_k, x_k] = \frac{f(w_k) - f(x_k)}{w_k - x_k}. \quad (2.1)$$

In following, the derivative $f'(x_n)$ in the second step will be approximated by

$$\frac{f[y_k, w_k]}{h(t_k)}, \quad (2.2)$$

where $h(t_k)$ is a differentiable function that depends real variable $t_k = \frac{f(y_k)}{f(x_k)}$. Therefore, we start from the scheme (1.1), the approximations (2.1), (2.2) and state the following two-point method

$$\begin{cases} w_k = x_k + \beta f(x_k), y_k = x_k - \frac{f(x_k)}{f[w_k, x_k]}, k = 0, 1, 2, \dots, \\ x_{k+1} = y_k - H(t_k) \frac{f(x_k)}{f(x_k) - 2f(y_k)} \frac{f(y_k)}{f[y_k, w_k]}. \end{cases} \quad (2.3)$$

Theorem 2.1. *Let $H, f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ be sufficiently differentiable functions and have a single root $x^* \in D$, for an open interval D . If the initial point x_0 is sufficiently close to x^* , then the sequence x_m generated by any method of the family (2.3) converges to x^* . If H is any function with $H(0) = 1, H'(0) = -1, |H''(0)| < \infty$ and $\beta \neq 0$ then the methods defined by (2.3) have convergence order at least 4.*

Proof. By using Taylor's expansion of $f(x)$ about x^* and taking into account that $f(x^*) = 0$, we obtain

$$f(x_k) = f'(x^*)(e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + O(e_k^5)). \quad (2.4)$$

Then, computing $e_{k,w} = w_k - x^*$, we attain $w_k = x_k + \beta f(x_k)$

$$e_{k,w} = e_k + e_k \beta f'(x^*)(1 + e_k(c_2 + e_k(c_3 + e_k c_4))) + O(e_k^5), \quad (2.5)$$

and

$$\begin{aligned} y_k = & x^* + (1 + \beta f'(x^*))e_k^2 + (-(2 + \beta f'(x^*)(2 + \beta f'(x^*))c_2^2) + (1 + \beta f'(x^*)(2 + \beta f'(x^*)) \\ & c_3 e_k^3 + ((4 + \beta f'(x^*)(5 + \beta f'(x^*)(3 + \beta f'(x^*))))c_2^3 - (7 + \beta f'(x^*)(10 + \beta f'(x^*) \\ & (7 + 2\beta f'(x^*))))c_2 c_3 + (1 + \beta f'(x^*)(3 + \beta f'(x^*)(3 + \beta f'(x^*)))c_4)e_k^4 + O(e_k^5). \end{aligned} \quad (2.6)$$

Using the Taylor expansion $H(t_k)$, we have

$$H(t_k) = H\left(\frac{f(y_k)}{f(x_k)}\right) = H(0) + H'(0)\left(\frac{f(y_k)}{f(x_k)}\right) + H''(0)\frac{\left(\frac{f(y_k)}{f(x_k)}\right)^2}{2}. \quad (2.7)$$

Thus, we get

$$\begin{aligned}
 x_{k+1} - x^* &= y_k - x^* - H(t_k) \frac{f(x_k)}{f(x_k) - 2f(y_k)} \frac{f(y_k)}{f[y_k, w_k]} \\
 &= -(-1 + h_0)(1 + \beta f'(x^*))c_2 e_k^2 + ((-2 + h_0 - h_1(1 + \beta f'(x^*))^2 - \beta f'(x^*) \\
 &\quad (2 + \beta f'(x^*)))c_2^2 - (-1 + h_0)(1 + \beta f'(x^*))(2 + \beta f'(x^*))c_3 e_k^3 (\beta f'(x^*))^2 \\
 &\quad + \left(\frac{1}{2}(8 - 2h_0 + 8h_1 - h_2 + f'(x^*)(10 + 6h_0 + 14h_1 - 10h_2)(6 + 8h_1 - 3h_2) \right. \\
 &\quad \left. (-\beta f'(x^*))^3(-2 + 2h_0 - 2h_1 + h_2)c_2^3 - (7 - h_0 + 4h_1 + 2\beta f'(x^*) \right. \\
 &\quad \left. (5 + h_0 + 5h_1) + (\beta f'(x^*))^2(7 + 2h_0 + 8h_1) + 2(\beta f'(x^*))^3(1 + h_1)c_2 c_3 \right. \\
 &\quad \left. - (-1 + h_0)(1 + \beta f'(x^*))(3 + \beta f'(x^*)(3 + \beta f'(x^*)))c_4\right)e_k^4 + O(e_k^5). \tag{2.8}
 \end{aligned}$$

By putting $h_0 = 1$, $h_1 = -1$, the final error expression is given by:

$$e_{k+1} = \frac{-1}{2}((1 + \beta f'(x^*))^2 c_2)((-2 + h_2 + f'(x^*)\beta(2 + h_2)c_2^2 + 2c_3))e_k^4 + O(e_k^5), \tag{2.9}$$

which finishes the proof of the theorem. \square

Some other simple forms of functions H can be:

$$\begin{cases} H_1(t) = 1 - t, H_2(t) = \frac{1}{1+t}, H_3(t) = (1 - \frac{t}{2})^2, H_4(t) = e^{-t}, \\ H_5(t) = \frac{1+2t}{1+3t}, H_6(t) = \cos(t) - \sin(t), H_7(t) = \text{Arccos}(t), H_8(t) = \frac{t^2+1}{1+t}. \end{cases} \tag{2.10}$$

3. Numerical results

The principal purpose of numerical examples is to verify the validity of the theoretical developments through a variety of test examples by use of Mathematica program. Numerical computations have been carried out using variable precision arithmetic in Mathematica 11 with 10000 significant digits. The computational order of convergence r_c [6] computed by the expressions

$$r_c = \frac{\log |f(x_k)/f(x_{k-1})|}{\log |f(x_{k-1})/f(x_{k-2})|}. \tag{3.1}$$

We compared proposed method (for $\beta_0 = 0.01$), Kung-Traub's method (KTM) [3] and Ostrowski's method (OM) [5]. The numerical values in Table 1 validate that the presented scheme TM4 performs better, not only for the absolute error in the root and the absolute value of the function as compared to without memory method. It should note that the condition for the convergence of repetitive methods is to select the appropriate initial conjecture root of the nonlinear equation. One can see more about this in reference [9].

$$\begin{cases} f_1(x) = x^5 + x^4 + 4x^2 - 15, \alpha \approx 1.34, x_0 = 1.1, \\ f_2(x) = x^3 + 4x^2 - 10, \alpha \approx 1.36, x_0 = 1, \\ f_3(x) = 10xe^{-x^2} - 1, \alpha \approx 1.67, x_0 = 1, \\ f_4(x) = \sin(5x)e^x - 2, \alpha \approx 1.36, x_0 = 1.5, \end{cases}$$

Table 1: Comparison of various iterative methods.

functions		TM4, $H_1(t)$	TM4, $H_2(t)$	TM4, $H_3(t)$	TM4, $H_4(t)$	TM4, $H_5(t)$	OM [5]	KTM [3]
$f_1, x_0 = 1.1$	$ x_{k+1} - x_k $	3.87e-1378	2.10e-1418	2.22e-1343	6.90e-1301	9.51e-1379	7.39e-35	5.39e-31
	$ f(x_{k+1}) $	1.24e-5509	2.14e-5669	5.66e-5370	1.17e-5199	2.76e-5510	1.01e-135	5.40e-120
	Iter	3	4	3	3	3	3	4
	r_c	4.00	4.00	4.00	4.00	4.00	4.00	4.00
$f_2, x_0 = 1$	$ x_{k+1} - x_k $	8.79e-1385	1.43e-1338	2.10e-1233	3.26e-1241	3.48e-1371	2.15e-34	3.36e-38
	$ f(x_{k+1}) $	9.20e-5537	6.50e-5352	1.51e-4931	1.02e-4965	1.129e-5481	5.78e-135	4.37e-64
	Iter	3	4	3	3	3	3	4
	r_c	4.00	4.00	4.00	4.00	4.00	4.00	4.00
$f_3, x_0 = 1$	$ x_{k+1} - x_k $	5.69e-1327	7.00e-1346	1.87e-1327	1.21e-1329	2.37e-1305	2.15e-34	7.32e-28
	$ f(x_{k+1}) $	2.27e-5305	1.20e-5382	2.00e-5307	2.30e-5316	1.42e-5218	5.78e-135	1.33e-108
	Iter	3	4	3	3	3	3	4
	r_c	4.00	4.00	4.00	4.00	4.00	4.00	4.00
$f_4, x_0 = 1.5$	$ x_{k+1} - x_k $	1.69e-1336	6.71e-1315	2.04e-1342	5.31e-1361	2.69e-1276	7.39e-35	1.24e-35
	$ f(x_{k+1}) $	3.92e-5342	9.36e-5256	8.27e-5366	3.73e-5440	2.22e-510	11.01e-135	5.11e-124
	Iter	3	4	3	3	3	3	4
	r_c	4.00	4.00	4.00	4.00	4.00	4.00	4.00

4. Basins of attraction of fourth-order derivative-free methods

In this section, to analyze the dynamic behavior of the proposed method, selecting the appropriate value of the parameter β and selecting the weight function with the maximum absorption region of a polynomial have been used. From the dynamic point of view, we take a 500×500 grid of the square $D = [-5, 5] \times [-5, 5] \in \mathbf{C}$. We have studied the dynamic behavior of the proposed methods by using the function $f(z) = z^3 - 1$. Various researchers have used basins of attraction to compare iteration schemes, for example, [1, 4].

We analog the attraction basin of the proposed method with two-step methods in Figure (1). Figure (1q) is Kung-Traub's method, which does not use the function-derivative where (1r) is the method with the proposed Kung-Traub derivative. According to the Figures, one has concluded that the basins of attraction of the proposed method are the best method because it has a vast and lighter basin of the attraction than all other mentioned methods ($H_1(t)$). Here the value of the free parameter $\beta = 0.001$ is considered.

5. Conclusion

In this paper, we used the idea of the weight function and turned Ostrowski's method into an optimal order method. The proposed methods are without-memory derivative-free. Numerical tests intend to verify the better performance of the proposed method over the others. According to the examples studied in Figures, we conclude that the weight function $H_1(t)$ and parameter $\beta = 0.001$ have the highest stability region and are competitive on other methods.

Further researches must be done to develop the proposed methods for system of nonlinear equations. These could be done in the next studies.

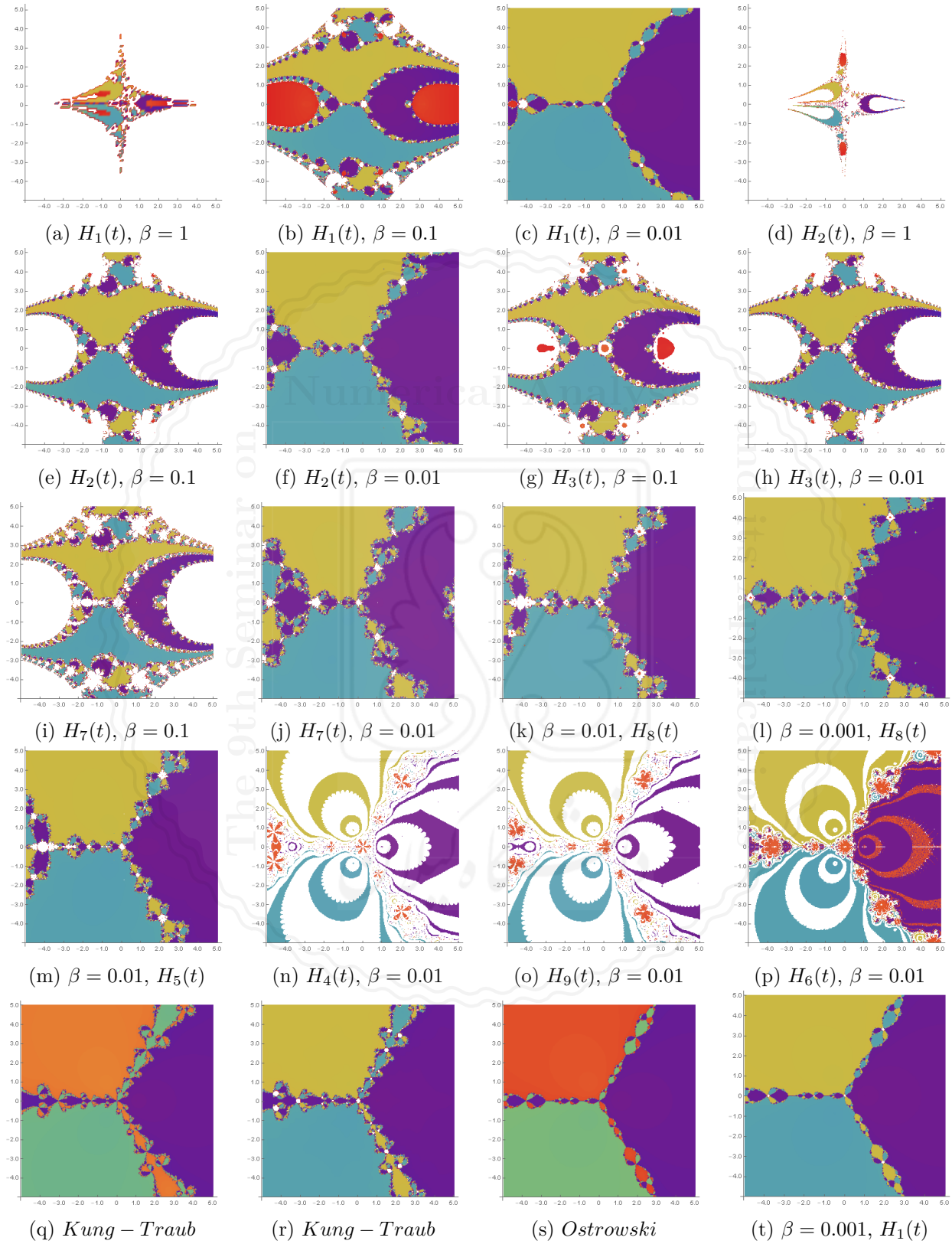


Figure 1: Basins of attraction for $f(z) = z^3 - 1$ for various methods

References

- [1] A. Cordero, F. Soleymani, J.R. Torregrosa, S. Shateyi, Basins of attraction for various Steffensen-type methods, *J. Appl. Math.* 2014 (2014) 1-17.
- [2] M. A. Hernández-Verón, S. Yadav, Á. A. Magreñán, E. Martínez, S. Singh, An Algorithm Derivative-Free to Improve the Steffensen-Type Methods. *Symm*, 14(4) (2021) 1-26.
- [3] H. T. Kung, J. F. Traub, Optimal order of one-point and multipoint iteration, *J. Assoc. Comput. Mach.* 21 (4) (1974) 643-651.
- [4] M. Moccari, T. Lotfi, On a two-step optimal Steffensen-type method: Relaxed local and semi-local convergence analysis and dynamical stability, *J. Math. Anal. Appl.* 468 (1) (2018) 240-269.
- [5] A.M. Ostrowski, Solution of equations and systems of equations, *Academic press*, New York, 1960.
- [6] M.S. Petković, B. Neta, L. D. Petković, J. Džunić, Multipoint Methods for Solving Nonlinear Equations, *Elsevier*, Amsterdam, 2013.
- [7] H. Ren, Q. Wu, W. Bi, A class of two-step Steffensen type methods with fourth-order convergence, *Appl. Math. Comp.* 209 (2) (2009) 206-210.
- [8] J. R. Sharma, S. Kumar, H. Singh, A new class of derivative-free root solvers with increasing optimal convergence order and their complex dynamics, *SeMA J.* (2022) 1-20.
- [9] V. Torkashvand, T. Lotfi, M.A. Fariborzi Araghi, A new family of adaptive methods with memory for solving nonlinear equations, *Math. scie.* 13 (1) (2019) 1-20.
- [10] V. Torkashvand, M. Kazemi, M. Moccari, Structure a family of three-step with-memory methods for solving nonlinear equations and their dynamics, *Math. Anal. Conv. Optim.* 2 (2) (2021) 119-137.

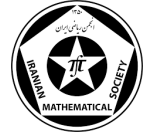
Vali Torkashvand
Department of Mathematics, Farhangian University, Tehran, Iran
email address: torkashvand1978@gmail.com

Manochehr Kazemi
Department of Mathematics, Ashtian Branch, Islamic Azad University, Ashtian, Iran
email address:

Elham Hashemizadeh
Department of Mathematics, Karaj Branch, Islamic Azad University, Karaj, Iran
email address:



University of Guilan



On the variable parameter Uzawa method for double saddle point systems

Mohammad Mahdi Izadkhah

Abstract. In this paper, we present variable parameter Uzawa method for solving double saddle point systems. We find the variable parameters of the proposed method, in a way that minimize some vector norms induced by symmetric positive definite matrices. Some numerical results are given to demonstrate the efficiency of the presented method.

1. Introduction

In this work, we consider the following large and sparse system of linear equations

$$\mathcal{A}\mathbf{u} \equiv \begin{pmatrix} A & B^T & C^T \\ B & 0 & 0 \\ C & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \equiv \mathbf{b}, \quad (1.1)$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite (SPD) matrix, $B \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{p \times n}$ have full row ranks, $x, b_1 \in \mathbb{R}^n$, $y, b_2 \in \mathbb{R}^m$ and $z, b_3 \in \mathbb{R}^p$. This is a class of double saddle point problems. The notation $\text{Ran}(A)$ stands for the range of A . For given vectors x, y and z of dimension n, m and p , respectively, $\mathbf{u} = (x; y; z)$ will denote a column vector of dimension $n + m + p$. we use $\langle \cdot, \cdot \rangle$ for usual inner product of two vectors. For a symmetric positive definite matrix G , we consider $\|x\|_G = \|G^{\frac{1}{2}}x\|_2$ for an arbitrary vector x , where $\|v\|_2 = \sqrt{\langle v, v \rangle}$ is Euclidean vector 2-norm. Linear systems of the form (1.1) arise from mixed finite element approximation of the potential fluid flow problems; see [1, 2] and the references therein for detailed descriptions of these problems. The following Proposition given in [1] represents the necessary and sufficient condition of the invertibility of the coefficient matrix \mathcal{A} in (1.1).

Proposition 1.1. *Let A be a SPD matrix and assume that B and C have full column ranks. Then a necessary and sufficient condition for the invertibility of the matrix \mathcal{A} in (1.1) is that $\text{Ran}(B^T) \cap \text{Ran}(C^T) = \{0\}$.*

2. Variable parameter Uzawa method

Uzawa's method has long been a popular technique for solving saddle point problems. We study possible extension of Uzawa's method to the double saddle point problem (1.1). To this end, we first split the coefficient matrix \mathcal{A} as follows

$$\mathcal{A} = \mathcal{M} - \mathcal{N}, \quad \mathcal{M} = \begin{pmatrix} A & 0 & 0 \\ B & -\alpha Q & 0 \\ C & 0 & -\beta M \end{pmatrix}, \quad \mathcal{N} = \begin{pmatrix} A & -B^T & -C^T \\ 0 & -\alpha Q & 0 \\ 0 & 0 & -\beta N \end{pmatrix}, \quad (2.1)$$

Keywords: Uzawa method, SOR iterative method, saddle point problem.

AMS Mathematical Subject Classification [2010]: 65F08, 65F10, 65F50.

in which the parameters $\alpha > 0$ and $\beta > 0$ and the matrices Q and $M = N$ are given and be positive definite. So, based on the splitting (2.1) the exact solution $\mathbf{u}^* = (x^*; y^*; z^*)$ satisfies

$$x^* = A^{-1}(b_1 - B^T y^* - C^T z^*), \quad (2.2)$$

$$y^* = y^* + \alpha Q^{-1}(Bx^* - b_2), \quad (2.3)$$

$$z^* = z^* + \beta M^{-1}(Cx^* - b_3). \quad (2.4)$$

By applying the first-order Richardson iterative method to the three linear equations (2.2)-(2.4), it follows

$$\begin{aligned} x^{(k+1)} &= A^{-1}(b_1 - B^T y^{(k)} - C^T z^{(k)}), \\ y^{(k+1)} &= y^{(k)} + \alpha Q^{-1}(Bx^{(k+1)} - b_2), \\ z^{(k+1)} &= z^{(k)} + \beta M^{-1}(Cx^{(k+1)} - b_3). \end{aligned} \quad (2.5)$$

We find the optimum parameters α and β such that the norms

$$\|\alpha Q^{-1}g^{(k)} - g^{(k)}\|_Q = \|\alpha Q^{-\frac{1}{2}}g^{(k)} - Q^{\frac{1}{2}}g^{(k)}\|_2, \quad \|\beta M^{-1}h^{(k)} - h^{(k)}\|_M = \|\beta M^{-\frac{1}{2}}h^{(k)} - M^{\frac{1}{2}}h^{(k)}\|_2$$

are minimized, respectively. Here $g^{(k)} = Bx^{k+1} - b_2$ and $h^{(k)} = Cx^{k+1} - b_3$. A direct calculation gives

$$\begin{aligned} \alpha &= \frac{\langle Q^{-1}g^{(k)}, g^{(k)} \rangle}{\langle g^{(k)}, g^{(k)} \rangle}, \\ \beta &= \frac{\langle M^{-1}h^{(k)}, h^{(k)} \rangle}{\langle h^{(k)}, h^{(k)} \rangle}. \end{aligned}$$

We are now ready to formulate the variable parameter Uzawa(VPU) method by (2.5) and motivated from [4] for the double saddle point problem (1.1).

Algorithm 1. (Variable parameter Uzawa method)

Given $x^{(0)} \in \mathbb{R}^n$, $y^{(0)} \in \mathbb{R}^m$ and $z^{(0)} \in \mathbb{R}^p$, the sequence $\mathbf{u}^{(k)} = (x^{(k)}; y^{(k)}; z^{(k)})$ is defined for $k = 1, 2, \dots$ as follows:

1. Set $x^{(k+1)} = A^{-1}(b_1 - B^T y^{(k)} - C^T z^{(k)})$.
2. Compute $g^{(k)} = Bx^{k+1} - b_2$ and $d^{(k)} = Q^{-1}g^{(k)}$. Then, compute the relaxation parameter

$$\alpha_k = \begin{cases} \frac{\langle d^{(k)}, g^{(k)} \rangle}{\langle g^{(k)}, g^{(k)} \rangle}, & g^{(k)} \neq 0, \\ 1, & g^{(k)} = 0. \end{cases}$$

$$\text{Set } y^{(k+1)} = y^{(k)} + \alpha_k d^{(k)}.$$

3. Compute $h^{(k)} = Cx^{k+1} - b_3$ and $s^{(k)} = M^{-1}h^{(k)}$. Then, compute the relaxation parameter

$$\beta_k = \begin{cases} \frac{\langle s^{(k)}, h^{(k)} \rangle}{\langle h^{(k)}, h^{(k)} \rangle}, & h^{(k)} \neq 0, \\ 1, & h^{(k)} = 0. \end{cases}$$

$$\text{Set } z^{(k+1)} = z^{(k)} + \beta_k s^{(k)}.$$

Remark 2.1. To further improvement of the computing efficiency of the VPU method, we can employ the Cholesky decomposition to solve the systems of linear equations with coefficient matrices A , Q and M , directly. For iterative scheme, one can use conjugate gradient method to some prescribed accuracy at each step.

3. Numerical experiments

We now describe some numerical experiments were carried out in order to show the efficiency and accuracy of the presented method. The computational study was done in the next problems.

Example 3.1. Let us consider the double saddle point system (1.1), where the entries of the matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times n}$, and $C \in \mathbb{R}^{p \times n}$ are defined as follows

$$A = (a_{ij}) = \begin{cases} i+1, & i=j \\ 1, & |i-j|=1 \\ 0, & \text{otherwise.} \end{cases}, \quad B = (b_{ij}) = \begin{cases} i, & j=n-m+i \\ 0, & \text{otherwise.} \end{cases}$$

and

$$C = (c_{ij}) = \begin{cases} i, & i=j \\ 0, & \text{otherwise.} \end{cases}$$

For this problem, the condition of Proposition 1.1 is satisfied. First, we set $Q = BA^{-1}B^T$ and $M = CA^{-1}C^T$, then we used Algorithm 1. to solve (1.1). The vector \mathbf{b} is chosen so that the components of the exact solution \mathbf{u} of (1.1) have values equal to 1. All runs are started with the initial zero vector and terminated if the current iterations satisfy $ERR = \frac{\|\mathbf{r}^{(k)}\|_2}{\|\mathbf{r}^{(0)}\|_2} \leq 10^{-4}$, or if the prescribed iteration number $k_{\max} = 2000$ is exceeded. Here, we define $\mathbf{r}^{(k)}$ as

$$\mathbf{r}^{(k)} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} - \begin{pmatrix} A & B^T & C^T \\ B & 0 & 0 \\ C & 0 & 0 \end{pmatrix} \begin{pmatrix} x^{(k)} \\ y^{(k)} \\ z^{(k)} \end{pmatrix}.$$

We compare the performance of our method with the SOR-like method [3] by reporting the number of iterations(minIT), the CPU time and the relative residual norm(ERR) in Talbe 1. we choose $\omega = 1.2538$ in the SOR-like method.

Table 1: CPU time, iteration number and EER

			VPU method				SOR-like method		
n	m	p	minIT	CPU(s)	ERR	ω	minIT	CPU(s)	ERR
50	30	10	79	0.0042	9.891e-05	1.2538	285	0.0625	9.996e-05
80	40	20	86	0.0054	9.766e-05	1.2538	429	0.2188	9.978e-05
100	50	40	183	0.0138	9.601e-05	1.2538	530	0.3281	9.914e-05
300	150	80	359	0.1195	9.920e-05	1.2538	1573	4.4688	9.983e-05

From the results reported in Table 1, we can conclude that minIT and computational CPU time are important items to demonstrate the efficiency of the VPU method in comparison with the SOR-like method [3].

References

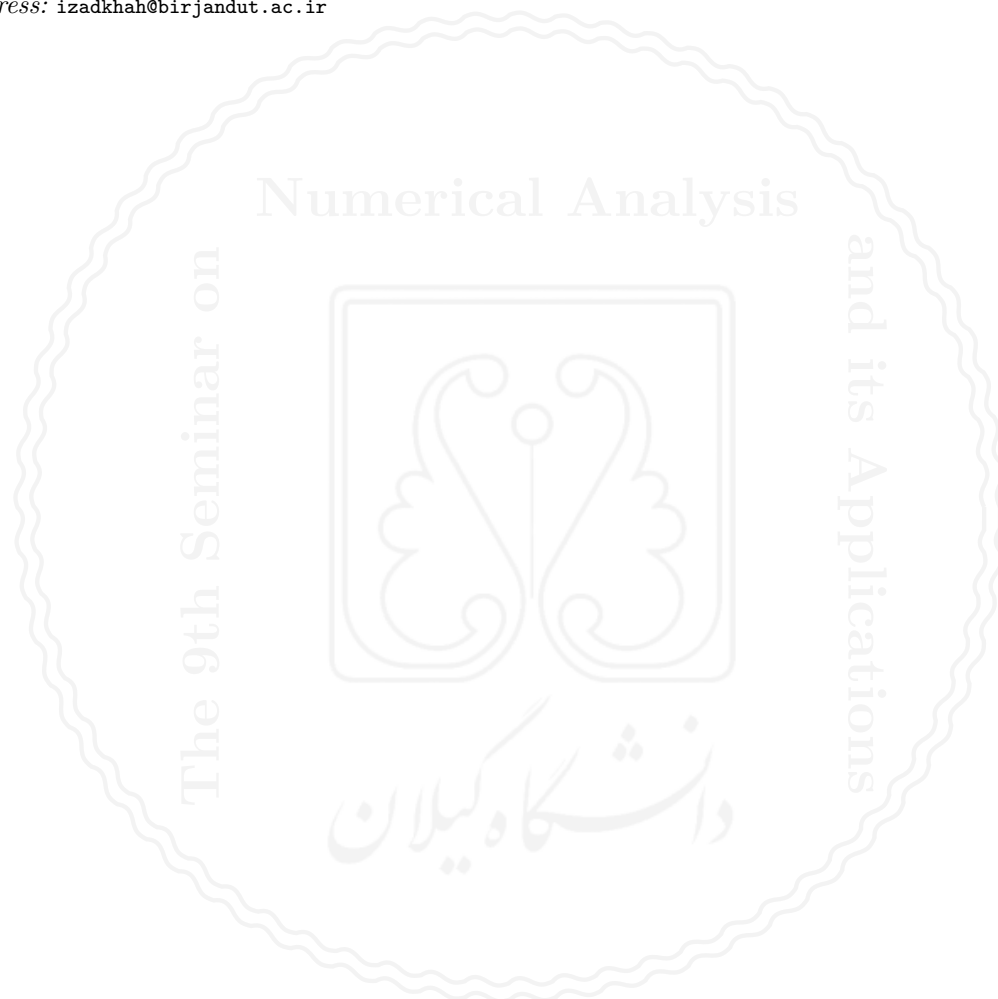
- [1] F.P.A. Beik, M. Benzi, Iterative methods for double saddle point systems, *SIAM Journal on Matrix Analysis and Applications*, 39 (2018) 902–921.
- [2] M. Benzi, F.P.A. Beik, Uzawa-type and augmented Lagrangian methods for double saddle point systems, *Structured Matrices in Numerical Linear Algebra*, Springer, Cham, 30 (2019) 215–236.

- [3] G.H.Golub, X. Wu, J.Y. Yuan, SOR-like methods for augmented systems, *BIT*, 41 (2001) 071–085.
- [4] N. Huang, Variable parameter Uzawa method for solving a class of block three-by-three saddle point problems, *Numerical Algorithms*, 85 (2020) 1233–1254.

Mohammad Mahdi Izadkhah

Department of Computer Science, Faculty of Computer & Industrial Engineering, Birjand University of Technology, Birjand, Iran

email address: izadkhah@birjandut.ac.ir





University of Guilan



The error analysis of an efficient numerical method for distributed Schrödinger equation

Fatemeh Hashemi, Esmail Hesameddini, Ameneh Taleei

Abstract. In this work, we study the error analysis of an efficient numerical method based on the finite difference approximation in time and the finite element method in spatial for a distributed-order time Schrödinger equation (DOT-SE). Firstly, the given problem is transformed into couple system of distributed-order fractional differential equations. Then, the fully discrete is formulated by using finite difference and finite element approximations. Moreover, unconditionally stability of this discrete scheme is proved.

1. Introduction

In this article, we study the error analysis of a distributed-order time fractional Schrödinger equation as follows

$$iD^{\omega(\alpha)}\psi(x, t) + \delta \frac{\partial^2}{\partial x^2}\psi(x, t) - v(x, t)\psi(x, t) - z(x, t) = 0, \quad (1.1)$$

with the initial condition $\psi(x, 0) = f$, $x \in [-l, l]$ and boundary condition $\psi(-l, t) = \psi(l, t) = 0$, $t \in [0, T]$. Also, the parameter δ is real constant, $D_t^{\omega(\alpha)}$ denotes the distributed-order fractional derivative, $\omega(\alpha)$ satisfy $0 < \int_0^1 \omega(\alpha) < \infty$ and $v(x, t)$ is general potential. In recent years, time-fractional Schrödinger equation has been used to describe many physical phenomena. Recently, many efforts have been made to develop effective numerical methods for solving fractional Schrödinger equation. The main goal of this paper is to study error analysis of FE-L1 method for DOT-SE. To do this, equation (1.1) can be written as the following coupled system:

$$\begin{cases} D_t^{\omega(\alpha)}\psi_I(x, t) - \delta \frac{\partial^2}{\partial x^2}\psi_R(x, t) + v(x, t)\psi_R(x, t) + z_R(x, t) = 0, \\ D_t^{\omega(\alpha)}\psi_R(x, t) + \delta \frac{\partial^2}{\partial x^2}\psi_I(x, t) - v(x, t)\psi_I(x, t) - z_I(x, t) = 0. \end{cases}$$

2. Numerical approximation

We first recall some definitions and lemmas which are needed in the numerical analysis.

Lemma 2.1. Let $0 < \alpha \leq 1$, define the nodes $\sigma_\alpha = \frac{1}{M_\alpha}$, $\alpha_m = m\sigma_\alpha$, $m = 0, 1, 2, \dots, M_\alpha$ in the interval $[0, 1]$ where $\alpha_0 = 0 < \alpha_1 < \dots < \alpha_{M_\alpha}$ and $s(\alpha) \in C^2(\Omega)$, then we have

$$\int_0^1 s(\alpha) d\alpha = \sigma_\alpha \sum_{m=0}^{M_\alpha} A_m s(\alpha_m) - \frac{\sigma_\alpha}{12} s^2(\xi), \quad \xi \in (0, 1),$$

where $A_m = \frac{1}{2}$ for $m = 0$ or M_α and $A_m = 1$ for otherwise.

Keywords: Schrödinger equation, distributed-order fractional equation, finite difference, L1-method, finite element.
AMS Mathematical Subject Classification [2010]: 13F55, 05E40, 05C65.

Lemma 2.2. Let $t_n = n\tau$, $n = 0, 1, 2, \dots, N$, and $T = N\tau$, we discretize the Caputo derivative by a difference approach as follows

$${}_0^C D_t^\alpha \psi(t_n) = \frac{1}{\Gamma(1-\alpha)} \int_0^{t_n} (t_n - s)^{-\alpha} \dot{\psi}(s) ds = \frac{1}{\tau_n^\alpha \Gamma(1-\alpha)} \sum_{j=0}^n b_j^n d_t \psi(t_{n-j}) + R_\tau^1 + R_\tau^2,$$

where $b_j^n = (j+1)^{1-\alpha} - j^{1-\alpha}$ and $d_t \psi(t_{n-j}) = \psi(t_{n-j}) - \psi(t_{n-j-1})$, $j = 0, 1, 2, \dots, n-1$. The truncated errors R_τ^1, R_τ^2 satisfy $R_\tau^2 = O(\tau^2)$ and $|R_\tau^1| \leq C\tau^{2-\alpha} \max |\frac{\partial^2 \psi(x,t)}{\partial t^2}|$.

The corresponding weak formulation couple system is to seek $\{\psi_I, \psi_R\} \in H_0^1 \times H_0^1$ for any $(\varphi_I, \varphi_R) \in H_0^1 \times H_0^1$, satisfying the following relation

$$\begin{cases} (D_t^{\omega(\alpha)} \psi_I(x, t), \varphi_R) + \delta(\frac{\partial \psi_R(x, t)}{\partial x}, \frac{\partial \varphi_R}{\partial x}) + v(x, t)(\psi_R(x, t), \varphi_R) + (z_R(x, t), \varphi_R) = 0, \\ (D_t^{\omega(\alpha)} \psi_R(x, t), \varphi_I) - \delta(\frac{\partial \psi_I(x, t)}{\partial x}, \frac{\partial \varphi_I}{\partial x}) - v(x, t)(\psi_I(x, t), \varphi_I) - (z_I(x, t), \varphi_I) = 0, \end{cases}$$

where the bilinear form $B(\cdot, \cdot)$ is defined by $B(\psi, \varphi) = \delta(\frac{\partial \psi}{\partial x}, \frac{\partial \varphi}{\partial x}) + v(x, t)(\psi, \varphi)$. Now, we define the FE space $X_h \subset H_0^1$ by $X_h = \{\varphi_h \in H_0^1 \cap C^0(\Omega) | \varphi_h|_{e_h} \in P_r(e_h), e_h \in \Omega_h\}$, where $\Omega_h = \{e_h | e_h \in T_h\}$ and T_h is a family of subdivisions $\Omega = [0, 1]$. Based on the given FE space X_h , the corresponding FE semi-discrete scheme is to find $\{\psi_{I,h}, \psi_{R,h}\} \in X_h \times X_h$ such that, for any $(\varphi_{I,h}, \varphi_{R,h}) \in X_h \times X_h$, we have

$$\begin{cases} (D_t^{\omega(\alpha)} \psi_{I,h}, \varphi_{R,h}) + B(\psi_{R,h}, \varphi_{R,h}) + (z_R(x, t_n), \varphi_{R,h}) = 0, & \forall \varphi_{R,h} \in X_h, \\ (D_t^{\omega(\alpha)} \psi_{R,h}, \varphi_{I,h}) - B(\psi_{I,h}, \varphi_{I,h}) - (z_I(x, t_n), \varphi_{I,h}) = 0, & \forall \varphi_{I,h} \in X_h. \end{cases} \quad (2.1)$$

To give the fully discrete analysis based on the spatial semi-discrete system, we need to approximate both distributed-order and fractional derivative in time.

Using Lemma 2.1 and Lemma 2.2, we can consider $L1$ type discretization for distributed-order fractional derivative at $t = t_n$, as follows

$$\begin{cases} (\frac{\sigma_\alpha}{\tau_n^\alpha \Gamma(1-\alpha)} \sum_{j=0}^n \sum_{m=0}^{M_\alpha} A_m \omega(\alpha_m) b_j^n d_t \psi_{I,h}^{n-j}, \varphi_{R,h}) + B(\psi_{R,h}^n, \varphi_{R,h}) + (z_R(x, t_n), \varphi_{R,h}) \\ = R_I^{\omega(\alpha)} + {}^{L1}R_I^\alpha, & \forall \varphi_{R,h} \in X_h, \\ (\frac{\sigma_\alpha}{\tau_n^\alpha \Gamma(1-\alpha)} \sum_{j=0}^n \sum_{m=0}^{M_\alpha} A_m \omega(\alpha_m) b_j^n d_t \psi_{R,h}^{n-j}, \varphi_{I,h}) - B(\psi_{I,h}^n, \varphi_{I,h}) - (z_I(x, t_n), \varphi_{I,h}) \\ = R_R^{\omega(\alpha)} + {}^{L1}R_R^\alpha, & \forall \varphi_{I,h} \in X_h. \end{cases} \quad (2.2)$$

2.1. Stability and error analysis

Theorem 2.3. The fully discrete variational formulation (2.2) is unconditionally stable.

Proof. To show the unconditionally stability of system (2.2), we have to prove the following relation

$$\|\psi_{I,h}^k\|^2 + \|\psi_{R,h}^k\|^2 \leq \|\psi_{I,h}^0\|^2 + \|\psi_{R,h}^0\|^2 + \frac{1}{\chi} (\|z_R^k\|^2 + \|z_I^k\|^2). \quad (2.3)$$

Setting $\{\varphi_{R,h}, \varphi_{I,h}\} = \{\psi_{I,h}^n, \psi_{R,h}^n\}$, $\chi = \sum_{m=0}^{M_\alpha} A_m \omega(\alpha_m) \frac{\sigma_\alpha}{\tau_n^\alpha \Gamma(1-\alpha)}$ in system (2.2) and since $\frac{1}{\tau_n^\alpha \Gamma(1-\alpha)} \sum_{j=0}^{n-1} b_j^n d_t \psi^{n-j} = \frac{1}{\tau_n^\alpha \Gamma(1-\alpha)} (b_0 \psi^n - \sum_{j=1}^{n-1} (b_j - b_{j-1}) \psi^{n-j} - b_{n-1} \psi^0)$, using the Cauchy-Schwartz inequality this system can be written as

$$\begin{cases} \chi \|\psi_{I,h}^n\|^2 + B(\psi_{R,h}^n, \psi_{I,h}^n) \leq \|z_R^n\| \|\psi_{I,h}^n\| + \chi \sum_{j=1}^{n-1} (b_{j-1} - b_j) \|\psi_{I,h}^{n-j}\| \|\psi_{I,h}^n\| \\ \quad + \chi b_{n-1} \|\psi_{I,h}^0\| \|\psi_{I,h}^n\|, \\ \chi \|\psi_{R,h}^n\|^2 - B(\psi_{I,h}^n, \psi_{R,h}^n) \leq \|z_I^n\| \|\psi_{R,h}^n\| + \chi \sum_{j=1}^{n-1} (b_{j-1} - b_j) \|\psi_{R,h}^{n-j}\| \|\psi_{R,h}^n\| \\ \quad + \chi b_{n-1} \|\psi_{R,h}^0\| \|\psi_{R,h}^n\|. \end{cases} \quad (2.4)$$

Due to $B(\psi_{R,h}^n, \psi_{I,h}^n) = -B(\psi_{I,h}^n, \psi_{R,h}^n)$ and $B(\psi, \varphi) \geq 0$ and with summing the both sides of system (2.4), we get

$$\begin{aligned} \chi(\|\psi_{I,h}^n\|^2 + \|\psi_{R,h}^n\|^2) &\leq \chi \sum_{j=1}^{n-1} (b_{j-1} - b_j) (\|\psi_{I,h}^{n-j}\| \|\psi_{I,h}^n\| + \|\psi_{R,h}^{n-j}\| \|\psi_{R,h}^n\|) \\ &\quad + \chi b_{n-1} (\|\psi_{I,h}^0\| \|\psi_{I,h}^n\| + \|\psi_{R,h}^0\| \|\psi_{R,h}^n\|) \\ &\quad + \|z_R^n\| \|\psi_{I,h}^n\| + \|z_I^n\| \|\psi_{R,h}^n\|. \end{aligned}$$

The Holder's inequality, implies that

$$\begin{aligned} \chi(\|\psi_{I,h}^n\|^2 + \|\psi_{R,h}^n\|^2)^{\frac{1}{2}} &\leq \chi \sum_{j=1}^{n-1} (b_{j-1} - b_j) (\|\psi_{I,h}^{n-j}\|^2 + \|\psi_{R,h}^{n-j}\|^2)^{\frac{1}{2}} \\ &\quad + \chi b_{n-1} (\|\psi_{I,h}^0\|^2 + \|\psi_{R,h}^0\|^2)^{\frac{1}{2}} \\ &\quad + (\|z_R^n\|^2 + \|z_I^n\|^2)^{\frac{1}{2}}. \end{aligned}$$

By using above inequality, we show that our claim (2.3) is valid. To do this the method of mathematical induction will be used. For $n = 1$, using (2.6), results in

$$\|\psi_{I,h}^1\|^2 + \|\psi_{R,h}^1\|^2 \leq \|\psi_{I,h}^0\|^2 + \|\psi_{R,h}^0\|^2 + \frac{1}{\chi} (\|z_R^1\|^2 + \|z_I^1\|^2). \quad (2.5)$$

which is true. Suppose that the relation (2.4) is true for $n = k - 1$. Using this hypothesis for $n = k$, one obtains

$$\|\psi_{I,h}^k\|^2 + \|\psi_{R,h}^k\|^2 \leq \left(\sum_{j=1}^{k-1} (b_{j-1} - b_j) + b_{k-1} \right) (\|\psi_{I,h}^0\|^2 + \|\psi_{R,h}^0\|^2) + \frac{1}{\chi} (\|z_R^k\|^2 + \|z_I^k\|^2),$$

since

$$\sum_{j=1}^n (b_{j-1} - b_j) + b_n = (1 - b_1) + \sum_{j=2}^{n-1} (b_{j-1} - b_j) + b_{n-1} = 1, \quad (2.6)$$

we get the following inequality, which is consistent with our claim

$$\|\psi_{I,h}^k\|^2 + \|\psi_{R,h}^k\|^2 \leq \|\psi_{I,h}^0\|^2 + \|\psi_{R,h}^0\|^2 + \frac{1}{\chi} (\|z_R^k\|^2 + \|z_I^k\|^2).$$

Therefore, the proof is completed. \square

In the following theorem, an error bound for the time discretization of system (2.1) is presented.

Theorem 2.4. Suppose that ψ_I, ψ_R , are the exact solutions of system (2.1). Then, its time discretization satisfies the following relation

$$\|\psi_I(t_n) - \psi_I^n\| + \|\psi_R(t_n) - \psi_R^n\| \leq \|R_I^{\omega(\alpha)} + {}^{L1}R_I^\alpha\| + \|R_R^{\omega(\alpha)} + {}^{L1}R_R^\alpha\|.$$

Proof. Since ψ_I, ψ_R are the exact solutions of system (2.1) by subtracting this system with its related time semi-discrete from, also setting

$$\begin{aligned} \varrho_R^n &= \psi_R(t_n) - \psi_R^n, \varrho_I^n = \psi_I(t_n) - \psi_I^n, \\ \chi &= \sum_{m=0}^{M_\alpha} A_m \omega(\alpha_m) \frac{\sigma_\alpha}{\tau_n^\alpha \Gamma(1-\alpha)}, \varphi_I = \varrho_R^n, \varphi_R = \varrho_I^n, \end{aligned}$$

and using the Cauchy-schwarz inequality, one obtains

$$\begin{cases} \chi \|\varrho_I^n\|^2 + B(\varrho_R^n, \varrho_I^n) \leq \|R_I^{\omega(\alpha)} + {}^{L1}R_I^\alpha\| \|\varrho_I^n\| + \chi \sum_{j=1}^{n-1} (b_{j-1} - b_j) \|\psi_{I,h}^{n-j}\| \|\varrho_I^n\| \\ \quad + \chi b_{n-1} \|\varrho_I^0\| \|\varrho_I^n\|, \\ \chi \|\varrho_R^n\|^2 - B(\varrho_I^n, \varrho_R^n) \leq \|R_R^{\omega(\alpha)} + {}^{L1}R_R^\alpha\| \|\varrho_R^n\| + \chi \sum_{j=1}^{n-1} (b_{j-1} - b_j) \|\varrho_R^{n-j}\| \|\varrho_R^n\| \\ \quad + \chi b_{n-1} \|\varrho_R^0\| \|\varrho_R^n\|. \end{cases}$$

Doing the some process in the proof of theorem 2.3, results in

$$\|\psi_I(t_n) - \psi_I^n\| + \|\psi_R(t_n) - \psi_R^n\| \leq \|R_I^{\omega(\alpha)} + {}^{L1}R_I^\alpha\| + \|R_R^{\omega(\alpha)} + {}^{L1}R_R^\alpha\|,$$

and the proof is completed. □

References

- [1] A. Ansari, Fundamental solution of a multi-dimensional distributed order fractional diffusion equation, *The European Physical Journal Plus*, 136 (2021) 1–23.
- [2] X. Gao, and F. Liu, H. Li, Y. Liu, I. Turner, B. Yin, A novel finite element method for the distributed-order time fractional Cable equation in two dimensions, *Computers and Mathematics with Applications*, 80 (2020) 923–939.
- [3] D. Li, J. Wang, J. Zhang, Unconditionally convergent L1-Galerkin FEMs for nonlinear time-fractional Schrodinger equations, *SIAM Journal on Scientific Computing*, 39 (2017) A3067–A3088.

Fatemeh Hashemi
Department of Mathematics, Shiraz University of Technology, Shiraz, Iran
email address: fa.hashemi@sutech.ac.ir

Esmail Hesameddini
Department of Mathematics, Shiraz University of Technology, Shiraz, Iran
email address: hesameddini@sutech.ac.ir

Ameneh Taleei
Department of Mathematics, Shiraz University of Technology, Shiraz, Iran
email address: a.taleei@sutech.ac.ir



University of Guilan



A robust spectral scheme for non-linear dynamical model of COVID-19 disease

Amin Faghieh, Payam Mokhtary

Abstract. In this paper, a high-order Galerkin approach based on the fractional Jacobi orthogonal functions is implemented for the numerical solution of a non-linear fractional dynamical system, in our case, modeling the recent pandemic due to corona virus (COVID-19). At the end, we have given a numerical approximation to illustrate the validity of the method.

1. Introduction

Due to the corona virus and related death toll, many studies have been recorded on mathematical modeling of outbreak of COVID-19 [2]. In [2], authors considered the following model of four compartment including healthy or susceptible population $z_1(t)$, the exposed class $z_2(t)$, the infected population $z_3(t)$ and the removed class $z_4(t)$ (death due to corona or natural) at time t (day) as

$$\begin{cases} D_C^\gamma z_1(t) = \bar{a} - \bar{K} z_3(t) z_1(t) (1 + \hat{\alpha} z_3(t)) - \bar{d}_0 z_1(t), \\ D_C^\gamma z_2(t) = \bar{K} z_3(t) z_1(t) (1 + \hat{\alpha} z_3(t)) - (\bar{d}_0 + \bar{\kappa}) z_2(t), \\ D_C^\gamma z_3(t) = \bar{b} + \hat{\alpha} z_2(t) - (\bar{\beta} + \bar{d}_0 + \bar{\delta}) z_3(t), \\ D_C^\gamma z_4(t) = \bar{\delta} z_3(t) - \bar{d}_0 z_4(t), \\ z_j(0) = z_j^{(0)}, \quad j = 1, 2, 3, 4, \quad t \in \chi = [0, T], \end{cases} \quad (1.1)$$

where $\bar{K} = \frac{R_0 \bar{d}_0 (\bar{d}_0 + \bar{\kappa}) (\bar{\beta} + \bar{d}_0 + \bar{\delta})}{\bar{\alpha} \bar{a}}$ is proportionality constant. The details of the parameters written in the model (1.1) and their values are given in Table 1. Here $\gamma = \frac{\eta}{\lambda} \in (0, 1)$ is a positive rational number described by the co-prime integers $\eta \geq 1$ and $\lambda \geq 2$, and T is a finite positive real number. D_C^γ is known as Caputo fractional derivative of order γ [1].

In this paper, we intend to provide a highly accurate fractional Jacobi Galerkin method for solving the non-linear fractional dynamical system (1.1).

2. Fractional Jacobi Galerkin method

The fractional Jacobi functions $J_n^{(\mu, \nu, \tau)}(t)$ with $\mu, \nu > -1$, $\tau \in (0, 1]$ and $t \in \chi$ are defined from the Jacobi polynomials $J_n^{(\mu, \nu)}(x)$ through the coordinate transform $x = 2(\frac{t}{T})^\tau - 1$ as follows [3]

$$J_n^{(\mu, \nu, \tau)}(t) = J_n^{(\mu, \nu)}\left(2\left(\frac{t}{T}\right)^\tau - 1\right) = \text{Span}\{1, t^\tau, \dots, t^{n\tau}\}.$$

Keywords: Mathematical model of COVID-19, Non-linear fractional dynamical system, Fractional Jacobi functions, Galerkin method.

AMS Mathematical Subject Classification [2010]: 34A09, 65L05, 65L20.

These functions are mutually orthogonal concerning the weight function $w^{(\mu,\nu,\tau)}(t) = t^{\tau\nu+\tau-1}(T^\tau - t^\tau)^\mu$. Further properties of fractional Jacobi functions can be found [3]. We set $\tau = \frac{1}{\lambda}$ and for $j = 1, 2, 3, 4$ consider the fractional Jacobi Galerkin solution as

$$z_{j,N}(t) = \sum_{i=0}^{\infty} v_{j,i} J_i^{(\mu,\nu,\tau)}(t) = \underline{v}_j \underline{J} = \underline{v}_j \underline{J} \underline{T}_t, \quad \underline{v}_j = [v_{j,0}, v_{j,1}, \dots, v_{j,N}, 0, \dots], \quad (2.1)$$

where

$$\underline{J} = [J_0^{(\mu,\nu,\tau)}(t), J_1^{(\mu,\nu,\tau)}(t), \dots, J_N^{(\mu,\nu,\tau)}(t), \dots]^T,$$

is the vector of fractional Jacobi functions, J is an infinite lower-triangular matrix and $\underline{T}_t = [1, t^\tau, \dots, t^{N\tau}, \dots]^T$.

Now, we give the following lemma which transforms $z_{1,N}(t)z_{3,N}(t)$ and $z_{1,N}(t)z_{3,N}^2(t)$ into a suitable matrix form.

Lemma 2.1. *The following relations hold*

$$z_{1,N}z_{3,N} = \underline{v}_1 \underline{J} \underline{M}_3 \underline{T}_t, \quad z_{1,N}z_{3,N}^2 = \underline{v}_1 \underline{J} \underline{M}_3^* \underline{T}_t, \quad (2.2)$$

where \underline{M}_3 and \underline{M}_3^* are the following infinite upper-triangular matrices

$$\underline{M}_3 = \begin{bmatrix} \underline{v}_3 J_0 & \underline{v}_3 J_1 & \underline{v}_3 J_2 & \dots \\ 0 & \underline{v}_3 J_0 & \underline{v}_3 J_1 & \dots \\ 0 & 0 & \underline{v}_3 J_0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad \underline{M}_3^* = \begin{bmatrix} \underline{v}_3 (J \underline{M}_3)_0 & \underline{v}_3 (J \underline{M}_3)_1 & \underline{v}_3 (J \underline{M}_3)_2 & \dots \\ 0 & \underline{v}_3 (J \underline{M}_3)_0 & \underline{v}_3 (J \underline{M}_3)_1 & \dots \\ 0 & 0 & \underline{v}_3 (J \underline{M}_3)_0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where J_s , and $(J \underline{M}_3)_s$, $s = 0, 1, \dots$ denotes the s -th column of the matrix J and $J \underline{M}_3$, respectively.

Since we intend to take the approach of solving the equivalence system of Volterra integral equations of (1.1), Computing $I^\gamma \underline{T}_t$ is required. Therefore, employing the relations (2.1), (2.2) and some simple manipulations yield

$$\begin{cases} \underline{v}_1 \underline{J} \underline{T}_t = -\bar{K} \underline{v}_1 \underline{J} \underline{M}_3 \underline{A} \underline{T}_t - \bar{K} \hat{\alpha} \underline{v}_1 \underline{J} \underline{M}_3^* \underline{A} \underline{T}_t - \bar{d}_0 \underline{v}_1 \underline{J} \underline{A} \underline{T}_t + \underline{f}_1 \underline{T}_t, \\ \underline{v}_2 \underline{J} \underline{T}_t = \bar{K} \underline{v}_1 \underline{J} \underline{M}_3 \underline{A} \underline{T}_t + \bar{K} \hat{\alpha} \underline{v}_1 \underline{J} \underline{M}_3^* \underline{A} \underline{T}_t - (\bar{d}_0 + \bar{\kappa}) \underline{v}_2 \underline{J} \underline{A} \underline{T}_t + \underline{f}_2 \underline{T}_t, \\ \underline{v}_3 \underline{J} \underline{T}_t = \hat{\alpha} \underline{v}_2 \underline{J} \underline{A} \underline{T}_t - (\bar{\beta} + \bar{d}_0 + \bar{\delta}) \underline{v}_3 \underline{J} \underline{A} \underline{T}_t + \underline{f}_3 \underline{T}_t, \\ \underline{v}_4 \underline{J} \underline{T}_t = \bar{\delta} \underline{v}_3 \underline{J} \underline{A} \underline{T}_t - \bar{d}_0 \underline{v}_4 \underline{J} \underline{A} \underline{T}_t + \underline{f}_4 \underline{T}_t. \end{cases} \quad (2.3)$$

Here $\underline{f}_j = [f_{j,0}, \dots, f_{j,N}, \dots]$, $j = 1, 2, 3, 4$ and \underline{A} are as follows

$$\underline{A} = \begin{bmatrix} 0 & \frac{1}{\Gamma(\gamma+1)} & 0 & \dots \\ \vdots & 0 & \frac{\Gamma(\tau+1)}{\Gamma(\gamma+\tau+1)} & 0 & \dots \\ \dots & \dots & \dots & \ddots & \ddots \end{bmatrix}, \quad \underline{f}_1 = [\bar{a}, 0, \dots] \underline{A} + [z_1^{(0)}, 0, \dots],$$

$$\underline{f}_2 = [z_2^{(0)}, 0, \dots], \quad \underline{f}_3 = [\bar{b}, 0, \dots] \underline{A} + [z_3^{(0)}, 0, \dots], \quad \underline{f}_4 = [z_4^{(0)}, 0, \dots].$$

It is noticed that I^γ is the Riemann-Liouville fractional integral operator of order γ [1]. Now, Projecting (2.3) onto $\langle J_0^{(\mu,\nu,\tau)}(t), \dots, J_N^{(\mu,\nu,\tau)}(t) \rangle$ and defining $\underline{v}_j = \underline{v}_j \underline{J} = [v_{j,0}, v_{j,1}, \dots, v_{j,N}]$, we

arrive at the following system of $4(N + 1)$ non-linear algebraic equations

$$\begin{cases} \underline{v}_1 = -\bar{K}\underline{v}_1\mathcal{M}_3^N A^N - \bar{K}\hat{\alpha}\underline{v}_1\mathcal{M}_3^{*N} A^N - \bar{d}_0\underline{v}_1 A^N + \underline{f}_1^N, \\ \underline{v}_2 = \bar{K}\underline{v}_1\mathcal{M}_3^N A^N + \bar{K}\hat{\alpha}\underline{v}_1\mathcal{M}_3^{*N} A^N - (\bar{d}_0 + \bar{\kappa})\underline{v}_2 A^N + \underline{f}_2^N, \\ \underline{v}_3 = \hat{\alpha}\underline{v}_2 A^N - (\bar{\beta} + \bar{d}_0 + \bar{\delta})\underline{v}_3 A^N + \underline{f}_3^N, \\ \underline{v}_4 = \bar{\delta}\underline{v}_3 A^N - \bar{d}_0\underline{v}_4 A^N + \underline{f}_4^N, \end{cases} \quad (2.4)$$

where the index N at the top of the vectors and matrices dictates the principle sub-vectors and sub-matrices of order $N + 1$ respectively. Due to the upper-triangular Toeplitz structure of matrices \mathcal{M}_3^N , \mathcal{M}_3^{*N} and the structure of matrix A^N , the unknown components of the unknown vectors \underline{v}_j , $j = 1, 2, 3, 4$, are derived through the following recurrence relations

$$\begin{cases} \underline{v}_{1,0} = f_{1,0}, & \underline{v}_{2,0} = f_{2,0}, & \underline{v}_{3,0} = f_{3,0}, & \underline{v}_{4,0} = f_{4,0}, \\ \underline{v}_{1,i} = -\bar{K}G_{1,3} - \bar{K}\hat{\alpha}\hat{G}_{1,3} - \bar{d}_0F_1 + f_{1,i0}, \\ \underline{v}_{2,i} = \bar{K}G_{1,3} + \bar{K}\hat{\alpha}\hat{G}_{1,3} - (\bar{d}_0 + \bar{\kappa})F_2 + f_{2,i}, \\ \underline{v}_{3,i} = \hat{\alpha}F_2 - (\bar{\beta} + \bar{d}_0 + \bar{\delta})F_3 + f_{3,i}, \\ \underline{v}_{4,i} = \bar{\delta}F_3 - \bar{d}_0F_4 + f_{4,i}, & i = 1, 2, \dots, N, \end{cases}$$

in which $G_{1,3}$ and $\hat{G}_{1,3}$ are non-linear functions of the elements

$$\underline{v}_{1,0}, \underline{v}_{1,1}, \dots, \underline{v}_{1,N}, \underline{v}_{2,0}, \underline{v}_{2,1}, \dots, \underline{v}_{2,N}.$$

Also, F_s , $s = 2, 3, 4$ are linear functions of the elements $\underline{v}_{s,0}, \underline{v}_{s,1}, \dots, \underline{v}_{s,N}$. Consequently, obtaining \underline{v}_j , $j = 1, 2, 3, 4$, the approximate solutions (2.1) can be characterized by solving $\underline{v}_j = \underline{v}_j J$. Indeed, our robust and intelligent implementation let us evaluate the unknowns by some recurrence relations without the need to solve non-linear block algebraic system (2.4) directly.

3. Numerical example

Defining

$$\mathbf{E}(N) = \max_{j \in \{1, 2, 3, 4\}} \|e_{j,N}\|_{w(\mu, \nu, \tau)}, \quad e_{j,N} = z_j(t) - z_{j,N}(t),$$

we approximate the solutions of (1.1) by setting the initial conditions $z_1(0) = 0.323$, $z_2(0) = 0.21$, $z_3(0) = 0.22$, $z_4(0) = 0.21$ scaled in million and $\gamma = \frac{1}{2}$. The numerical results are illustrated in Table 2.

Table 1: Description of the Parameters of model (1.1).

Parameter	Description	Value [2]
\bar{a}	The population whose test is negative	0.00250281 millions
\bar{d}_0	Natural death rate	0.0000004/million
\bar{b}	The population whose test is positive	0.006656 millions
$\bar{\beta}$	Death due to Corona	0.0109
$\bar{\kappa}$	The rate constant characterizing the infection	0.000024
$\hat{\alpha}$	Rate at which recovered individuals lose immunity	0.00009/million
$\bar{\delta}$	Recovered rete	0.75

Table 2: The numerical results for various values of μ , ν and N .

N	$\mu = \nu = -\frac{1}{2}$		$\mu = 0, \nu = \frac{1}{2}$	
	$E(N)$	$CPU-time$	$E(N)$	$CPU-time$
40	4.01×10^{-6}	2.27	1.61×10^{-5}	2.23
80	8.95×10^{-9}	25.53	3.06×10^{-8}	25.72
160	5.57×10^{-14}	312.39	1.61×10^{-13}	313.95

References

- [1] K. Diethelm, *The Analysis of Fractional Differential Equations*, Springer, Berlin, 2010.
- [2] M. Rahman, M. Arfan, K. Shah, J. F. Gomez-Aguilar, Investigating a nonlinear dynamical model of COVID-19 disease under fuzzy caputo, random and ABC fractional order derivative, *Chaos Solitons Fractals*, 140 (2020) no. 110232.
- [3] M. A. Zaky, E. H. Doha, J. A. Tenreiro Machado, A spectral framework for fractional variational problems based on fractional Jacobi functions, *Applied Numerical Mathematics*, 132 (2018) 51-72.

Amin Faghieh

Department of Mathematics, Faculty of Science, Sahand University of Technology, Tabriz, Iran

email address: amin.fagheh71@gmail.com

Payam Mokhtary

Department of Mathematics, Faculty of Science, Sahand University of Technology, Tabriz, Iran

email address: mokhtary.payam@gmail.com



University of Guilan



A new preconditioner for the SOR method to solve multi-linear systems

Afsaneh Hasanpour, Maryam Mojarab

Abstract. The preconditioned techniques play a significant role in solving linear and multi-linear systems. In this paper, we present a new preconditioner of the tensor splitting SOR method for solving multi-linear systems with \mathcal{M} -tensors. Numerical examples confirm our theoretical results and the efficiency of our new preconditioner. It is apparent that the preconditioner can improve the method in reducing the number of iterations and the CPU time.

1. Introduction

Recently, solving the multi-linear systems seems to be attractive in many engineering and scientific computing researches [2], like image processing [3], tensor complementarity problem, and numerical partial differential equations [4].

Suppose that $\mathcal{A} \in \mathbb{R}^{[m,n]}$ is an order m dimension n tensor and b is a vector in \mathbb{R}^n . Consider the following tensor equation

$$\mathcal{A}x^{m-1} = b, \quad (1.1)$$

where $\mathcal{A}x^{m-1} \in \mathbb{R}^n$ is defined as

$$(\mathcal{A}x^{m-1})_i = \sum_{i_2 i_3 \dots i_n=1}^n a_{i i_2 i_3 \dots i_n} x_{i_2} x_{i_3} \dots x_{i_n}, \quad i = 1, 2, \dots, n,$$

where x_i denotes the i th component of $x \in \mathbb{R}^n$. Many theoretical analysis and algorithms have been proposed for solving (1.1) [4].

The role of the preconditioning technique is clear in solving linear and multi-linear systems and if a suitable preconditioner is chosen, the convergence rate of the method can be improved. Some preconditioner for solving multi-linear systems were introduced in [5, 6]. In this paper, we propose a new preconditioner for the SOR method which is constructed by combination of two existing preconditioners.

This paper is organized as follows. In Section 2, a new preconditioner is proposed, and the preconditioned SOR method is constructed. Some numerical examples are given in Section 3 which illustrate the effectiveness and superiority of the new preconditioned iterative method. Finally, Section 4 consists of conclusion.

Keywords: SOR method, Preconditioner, Multi-linear system, \mathcal{M} -tensor.

AMS Mathematical Subject Classification [2010]: 65F08, 65F10.

2. Main Results

Li et al. [5] and Liu et al. [6] proposed two tensor preconditioners of the form $P_\alpha = I + S_\alpha$ and $P_\beta = I + G_\beta$, where:

$$S_\alpha = \begin{pmatrix} 0 & -\alpha_1 a_{12\dots 2} & 0 & \cdots & 0 \\ 0 & 0 & -\alpha_2 a_{23\dots 3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\alpha_{n-1} a_{(n-1)n\dots n} \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

and

$$G_\beta = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ -\beta_1 a_{21\dots 1} & 0 & 0 & \cdots & 0 \\ -\beta_2 a_{31\dots 1} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\beta_{n-2} a_{(n-1)1\dots 1} & 0 & 0 & \cdots & 0 \\ -\beta_{n-1} a_{n1\dots 1} & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

and $\alpha = (\alpha_i)$, $\beta = (\beta_i)$ where α_i and β_i in \mathbb{R} are parameters for $i = 1, \dots, n-1$. In this paper we combine this two preconditioner and propose a more effective preconditioner $P = I + S_\alpha + G_\beta$.

Consider the multi-linear system (1.1). Without loss of generality, we take all the diagonal entries of the tensor \mathcal{A} equal to 1. Applying a nonsingular matrix P as a preconditioner, we have

$$P\mathcal{A}x^{m-1} = Pb. \quad (2.1)$$

We consider $\hat{\mathcal{A}} = (I + S_\alpha + G_\beta)\mathcal{A}$ and $\hat{b} = (I + S_\alpha + G_\beta)b$. Let

$$\hat{\mathcal{A}} = \hat{\mathcal{D}} - \hat{\mathcal{L}} - \hat{\mathcal{F}}, \quad \text{and} \quad \mathcal{A} = \mathcal{I}_m - \mathcal{L} - \mathcal{F},$$

with $\hat{\mathcal{D}} = \hat{D}\mathcal{I}_m$, $\hat{\mathcal{L}} = \hat{L}\mathcal{I}_m$, where \hat{D} is the diagonal matrix and $-\hat{L}$ is the strictly lower triangle matrix of $M(\hat{\mathcal{A}})$. Thus

$$\hat{\mathcal{A}} = \mathcal{I}_m - \mathcal{L} - \mathcal{F} + S_\alpha \mathcal{I}_m - S_\alpha \mathcal{L} - S_\alpha \mathcal{F} + G_\beta \mathcal{I}_m - G_\beta \mathcal{L} - G_\beta \mathcal{F}.$$

The SOR-type method for solving (2.1) can be written as

$$x_k = (\mathcal{T}_p x_{k-1}^{m-1} + q_p)^{[\frac{1}{m-1}]}, \quad k = 1, 2, \dots,$$

where

$$\mathcal{T}_p = M(\mathcal{E}_p)^{-1} \mathcal{F}_p, \quad \mathcal{E}_p = \frac{1}{\omega} (\hat{\mathcal{D}} - \omega \hat{\mathcal{L}}), \quad \mathcal{F}_p = \frac{1}{\omega} ((1 - \omega) \hat{\mathcal{D}} + \omega \hat{\mathcal{F}}), \quad q_p = M(\mathcal{E}_p)^{-1} Pb.$$

Theorem 2.1. Let $\mathcal{A} \in \mathbb{R}^{[m,n]}$ be a strong \mathcal{M} -tensor. Then for the new preconditioner P , $\hat{\mathcal{A}} = P\mathcal{A}$ is a strong \mathcal{M} -tensor for $\alpha_i, \beta_i \in [0, 1]$.

Proof. Since the off-diagonal entries of $\hat{\mathcal{A}}$ are non-positive for $\alpha_i, \beta_i \in [0, 1]$, so $\hat{\mathcal{A}}$ is a \mathcal{Z} -tensor. According to this fact that \mathcal{A} is a strong \mathcal{M} -tensor and $\mathcal{A} = \mathcal{E} - \mathcal{F}$ is a weak regular splitting, $\hat{\mathcal{A}} = \hat{\mathcal{E}} - \hat{\mathcal{F}}$ can be a weak regular splitting by considering $\hat{\mathcal{E}} = (I + S_\alpha + G_\beta)\mathcal{E}$ and $\hat{\mathcal{F}} = (I + S_\alpha + G_\beta)\mathcal{F}$. Therefore $\hat{\mathcal{A}}$ is a strong \mathcal{M} -tensor. \square

3. Numerical examples

In this section, numerical examples are given to show the efficiency of the preconditioned SOR method. The stopping criterion $\| \mathcal{A}x^{m-1} - b \| \leq 10^{-10}$ is used and a maximum of 1000 iterations is allowed. We take the starting vector x_0 and right-hand side vector b equal to $\text{ones}(n, 1)$. Finding the optimal parameter ω , we search from 0.01 to 2 in the interval of 0.01. All the examples were executed in double precision in MATLAB R2014a. We show the number of iterations by “Iter” and the CPU time in second by “time” for the new preconditioned *SOR* (*PSOR*) and the SOR and former preconditioned *SOR* (*P_αSOR* and *P_βSOR*) methods, respectively.

Example 3.1. Consider $\mathcal{A} \in \mathbb{R}^{[3,n]}$ and $b \in \mathbb{R}^n$ in which

$$\begin{cases} a_{111} = a_{nnn} = 1, & a_{122} = a_{n(n-1)(n-1)} = -0.5, \\ a_{iii} = \frac{\theta^2}{h^2} + \frac{\mu_1}{h} + \eta, & i = 2, 3, \dots, n-1, \\ a_{i(i-1)i} = a_{i(i-1)(i-1)} = -\frac{\theta^2}{4h^2} + \frac{\mu_2^2}{2h}, & i = 2, 3, \dots, n-1, \\ a_{i(i+1)i} = a_{i(i+1)(i+1)} = -\frac{\theta^2}{4h^2} + \frac{\mu_2^2}{2h}, & i = 2, 3, \dots, n-1, \end{cases}$$

where

$$\theta = 0.2, \quad \mu_1 = 0.04, \quad \eta = 0.04, \quad \mu_2 = -0.04, \quad h = \frac{2}{n}.$$

From [1], it is found that \mathcal{A} is a strong \mathcal{M} -tensor. Numerical results in Table 1 with different sizes of \mathcal{A} represent that the new preconditioned method is better than original ones and former preconditioned *SOR* methods for solving \mathcal{M} -tensor equation. .

Table 1: Numerical results of Example 3.1 with $\omega_{opt} = 1.3$

n	<i>PSOR</i>		<i>SOR</i>		<i>P_αSOR</i>		<i>P_βSOR</i>	
	Iter	time	Iter	time	Iter	time	Iter	time
10	46	0.020	47	0.025	46	0.023	47	0.025
20	89	0.023	103	0.032	89	0.024	103	0.026
30	114	0.030	158	0.038	114	0.036	158	0.033
40	117	0.034	208	0.043	117	0.036	208	0.039
50	107	0.041	257	0.076	107	0.052	257	0.067

4. Conclusion

In this paper, we proposed a new preconditioner of the tensor splitting *SOR* method for solving multi-linear systems with \mathcal{M} -tensors. Numerical results show that our new preconditioner is more effective than the original ones and prior preconditioner that we combined them.

References

- [1] P. Azimzadeh , E. Bayraktar, High order bellman equations and weakly chained diagonally dominant tensors, *SIAM Journal on Matrix Analysis and Applications*, 40 (2019) 276–298.
- [2] L. Cui , W. Li , M. Ng, Primitive tensors and directed hypergraphs. *Linear Algebra and its Applications*, 471 (2015), 96–108.

- [3] L. Cui , C. Chen, W. Li , M. Ng, An eigenvalue problem for even order tensors with its applications. *Linear Multilinear Algebra*, 64 (2016) 602–621.
- [4] W. Ding , Y. Wei, Solving multi-linear system with M-tensors. *Journal of Scientific Computing*, 68 (2016) 689–715.
- [5] W. Li , D. Liu , S. Vong (2018) Comparison results for splitting iterations for solving multi-linear systems. *Applied Numerical Mathematics*, 134 (2018) 105–121.
- [6] D. Liu , W. Li , S. Vong, A new preconditioned SOR method for solving multi-linear systems with an M-tensor. *Calcolo*, 57 (2020) no.15.

Afsaneh Hasanpour

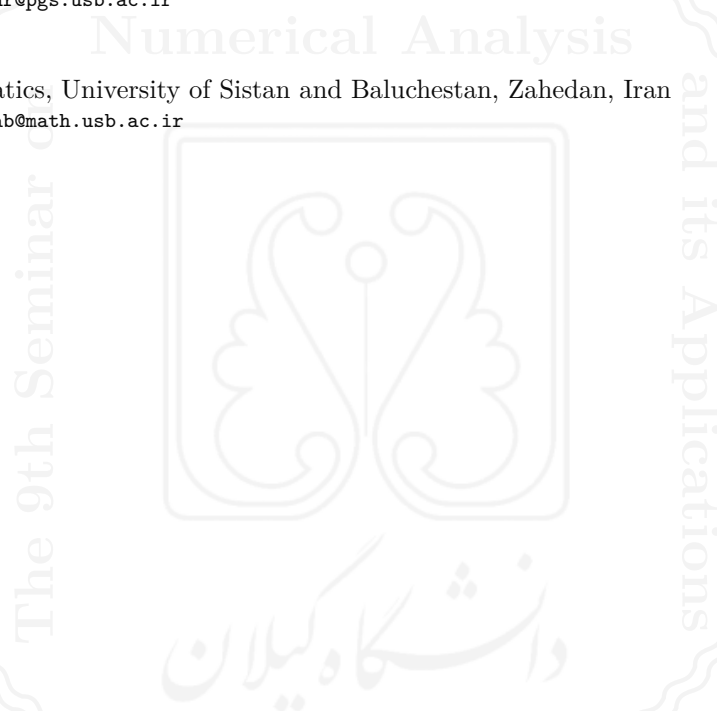
Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran

email address: a_hasanpour@pgs.usb.ac.ir

Maryam Mojarab

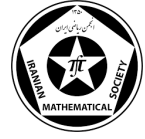
Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran

email address: ma_mojarrab@math.usb.ac.ir





University of Guilan



Accurately approximating flat functions by a new transcendental RBF

Maryam Mohammadi, Mohammad Heidari

Abstract. It is well-known that the RBF method is increasingly more accurate on steeper gradient surfaces and has difficulty accurately approximating flat functions. The apparent reason is that the flat surfaces are represented by linear combinations of very large shape parameters c . But as c becomes large, so does the condition number. In this paper, we first show that the Powers RBF r is the best candidate for approximating univariate functions having zero curvature everywhere, like lines. So for approximating flat functions having very small values of curvature, we introduce a new transcendental RBF based on the hyperbolic tangent function as a smooth approximant to $\phi(r) = r$ with higher accuracy and better convergence properties than the MQ RBF by decreasing shape parameter c .

1. Introduction

Given a set of n distinct points $\{x_j\}_{j=1}^n \subset \mathbb{R}^d$ and corresponding data values $\{f_j\}_{j=1}^n$, the RBF interpolant is given by

$$s(x) = \sum_{j=1}^n \lambda_j \phi(\|x - x_j\|), \quad (1.1)$$

where $\phi(r)$, $r \geq 0$, is some radial function (cf. e.g. [5]). The expansion coefficients λ_j are determined from the interpolation conditions $s(x_j) = f_j$, $j = 1, \dots, n$, which leads to the symmetric linear system $A\lambda = f$, where $A = [\phi(\|x_i - x_j\|)]_{1 \leq i, j \leq n}$. The existence of a solution is assured for positive definite RBFs and also for conditionally positive definite RBFs by adding a lower degree polynomial to (1.1). We can introduce a shape parameter as $\phi(\frac{r}{c})$ allowing to scale the basis function ϕ making it flatter as $c \rightarrow \infty$ and spiky as $c \rightarrow 0$. So for accurately approximating flat functions, we need to use too large shape parameters which leads to highly ill-conditioned and even singular coefficient matrices [2]. In this paper, we first show that the Powers RBF r is the best candidate for approximating univariate functions having zero curvature everywhere, like lines. Then we introduce a new transcendental RBF based on the hyperbolic tangent function converging so much faster than the MQ RBF to r by decreasing c . So one can approximate both steep and gentle gradient surfaces by not too much small values of c .

2. Main results

Let κ and τ denote curvature and torsion, respectively. The fundamental theorem of curve theory is given as follows [3].

Keywords: RBF interpolation, Transcendental RBF, Flat surfaces, Curvature.

AMS Mathematical Subject Classification [2010]: 65D05, 65D12, 65D20.

Theorem 2.1. *Two space curves C and C^* with nonzero curvature are congruent (i.e., differ by the composition of a translation and a rotation) if and only if the corresponding arclength parametrizations $\alpha, \alpha^* : [a, b] \rightarrow \mathbb{R}^3$ have the property that $\kappa(s) = \kappa^*(s)$ and $\tau(s) = \tau^*(s)$ for all $s \in [a, b]$.*

In the sequel, we just consider the univariate functions i.e parametric vector-valued curves $\alpha : [a, b] \rightarrow \mathbb{R}^2$. Investigating multivariate flat functions need a vast discussion on one of the most important concepts in the geometry of surfaces called Gaussian curvature [4], which is beyond the scope of this note and leave it to our further works.

Corollary 2.2. *If $\alpha, \alpha^* : I \rightarrow \mathbb{R}^2$ are plane curves such that $\kappa_\alpha = \kappa_{\alpha^*}$, then α and α^* are congruent.*

Proposition 2.3. *A space curve is a line if and only if its curvature is 0 everywhere.*

Theorem 2.4. *Powers RBF $\phi(r) = r$ exactly approximates lines.*

Proof. By considering $\phi(r) = r$, the RBF interpolant (1.1) is given as

$$s(x) = \sum_{j=1}^n \lambda_j |x - x_j|.$$

Now, according to the Corollary 2.2 and Proposition 2.3, it suffices to show that $\kappa(s(x)) = 0$. For $x_1 < \dots < x_i < x < x_{i+1} < \dots < x_n$, we have

$$\kappa(s(x)) = \frac{\left| \frac{d^2}{dx^2} (s(x)) \right|}{\left(1 + \left(\frac{d}{dx} (s(x)) \right)^2 \right)^{\frac{3}{2}}} = \frac{0}{\left(1 + \left(\lambda_1 + \dots + \lambda_i - (\lambda_{i+1} + \dots + \lambda_n) \right)^2 \right)^{\frac{3}{2}}} = 0.$$

The case $x = x_1$ and $x = x_n$ can be proved similarly. \square

Theorem 2.4 says that working with smooth approximants to $\phi(r) = r$ with high accuracy and fast convergence properties leads to accurate approximations of flat functions having low small values of curvature. In [1], we introduced the following new globally supported and infinitely differentiable transcendental RBF, abbreviated by “RTH RBF”

$$\phi(r) = r \tanh \left(\frac{r}{c} \right), \quad (2.1)$$

which is conditionally negative definite of order 1. It has the property

$$\lim_{c \rightarrow 0^+} r \tanh \left(\frac{r}{c} \right) = r,$$

with much better convergence properties than the Multiquadric RBF (see Theorems 2.3 and 2.4 in [1]). In the next section, we show that the RTH RBF interpolation method leads to accurate results for functions with both steep and gradient regions.

3. Numerical Results

We now provide two examples which approves successful role of RTH RBF in approximating functions with zero and small values of curvature. We take different values of the shape parameter c , $n = 200$ equidistant center points and $m = 125$ equidistant evaluation points. We use the maximum absolute error norm $L_\infty = \max_{1 \leq i \leq m} |f_i - \tilde{f}_i|$, where f and \tilde{f} represent the exact and approximate solutions, respectively.

Example 3.1. In the first test problem, we approximate the direct line

$$f_1(x) = x + 1, \quad x \in [-1, 1].$$

The L_∞ error norms of the RTH RBF interpolation are given in Table 1. The relative error distribution is plotted in Figure 1 using the RTH RBF interpolation method with $c = 0.1$. Numerical results show that the results are in good agreement with the exact solution.

Table 1: L_∞ error norms of RTH RBF interpolation; Example 3.1.

c	1	0.5	0.1	0.01	0.001	0.0001
L_∞	1.6×10^{-8}	1.01×10^{-8}	4.4×10^{-9}	3.1×10^{-9}	1.5×10^{-11}	0

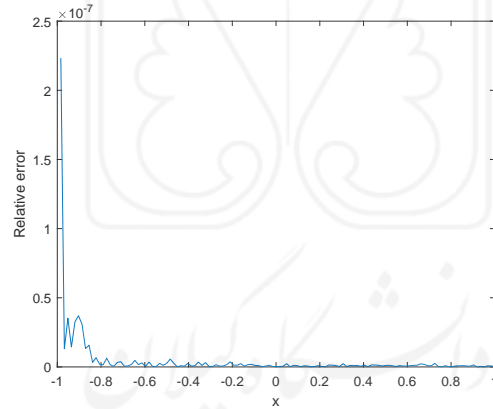


Figure 1: Relatives errors using the RTH RBF interpolation method for $c = 0.1$; Example 3.1.

Example 3.2. In the next test problem, we approximate the following function having both steep and flat regions with corner features.

$$f_2(x) = \tanh(60x - 0.01), \quad x \in [-1, 1].$$

The exact and approximate solutions of $f_2(x)$ using the RTH RBF interpolation method with $c = 0.1$ and $c = 0.01$ are shown in Figure 2. Figure 2-(A) shows some oscillatory behaviour for $c = 0.1$ in flat regions. But decreasing the shape parameter c completely removes oscillations and approves the theory. The relative error distribution is plotted in Figure 3 using the RTH RBF interpolation method with $c = 0.01$. Results show that we can simulate both steep and flat regions with intermediate values of the shape parameter c .

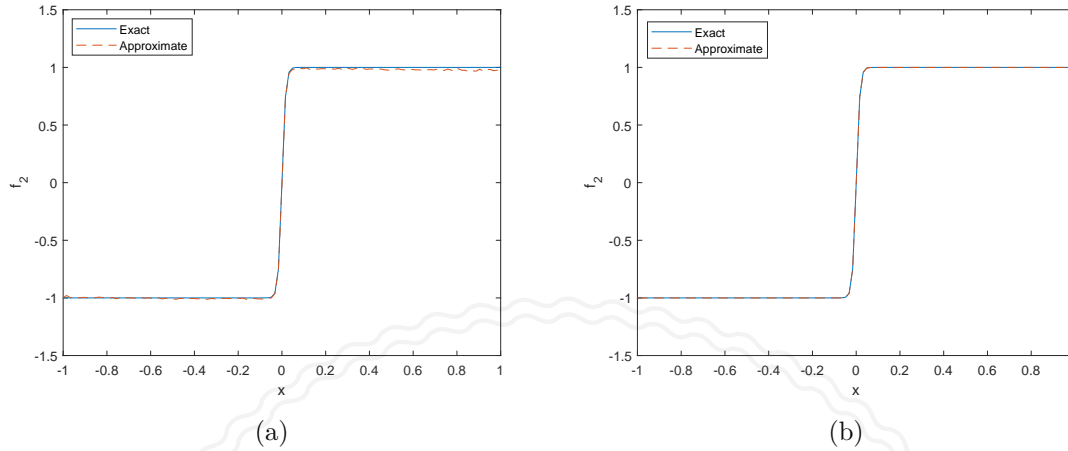


Figure 2: Exact and approximate solutions of $f_2(x)$ using the RTH RBF interpolation method for (A) $c = 0.1$ and (B) $c = 0.01$; Example 3.2.

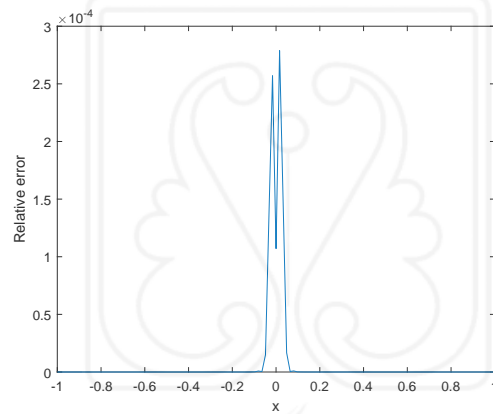


Figure 3: Relatives errors using the RTH RBF interpolation method for $c = 0.01$; Example 3.2.

4. Conclusion

In this paper, a new globally supported and infinitely differentiable transcendental RBF named “RTH RBF” is used for interpolation. Its fast convergence to the Powers RBF $\phi(r) = r$ leads to accurate results for both steep and gentle gradient surfaces by not too much small values of the shape parameter c . The given theory can be proved for the multivariate interpolation and used for high dimensional surfaces. We leave this to our further works.

References

- [1] M. Heidari, M. Mohammadi, S. De-Marchi, A shape preserving quasi-interpolation operator based on a new transcendental RBF, *Dolomites Research Notes on Approximation*, 14 (2021) 56–73.
- [2] E.J. Kansa, Multiquadrics-A scattered data approximation scheme with applications to computational fluid-dynamics-II solutions to parabolic, hyperbolic and elliptic partial differential equations, *Computers & mathematics with applications*, 19 (1990) 147–161.

- [3] B. Oneill, *Elementary differential geometry*, Elsevier, 2006.
- [4] T. Shifrin, *Differential geometry: a first course in curves and surfaces*, University of Georgia, 2015.
- [5] H. Wendland, *Scattered data approximation*, Cambridge university press, 2004.

Maryam Mohammadi

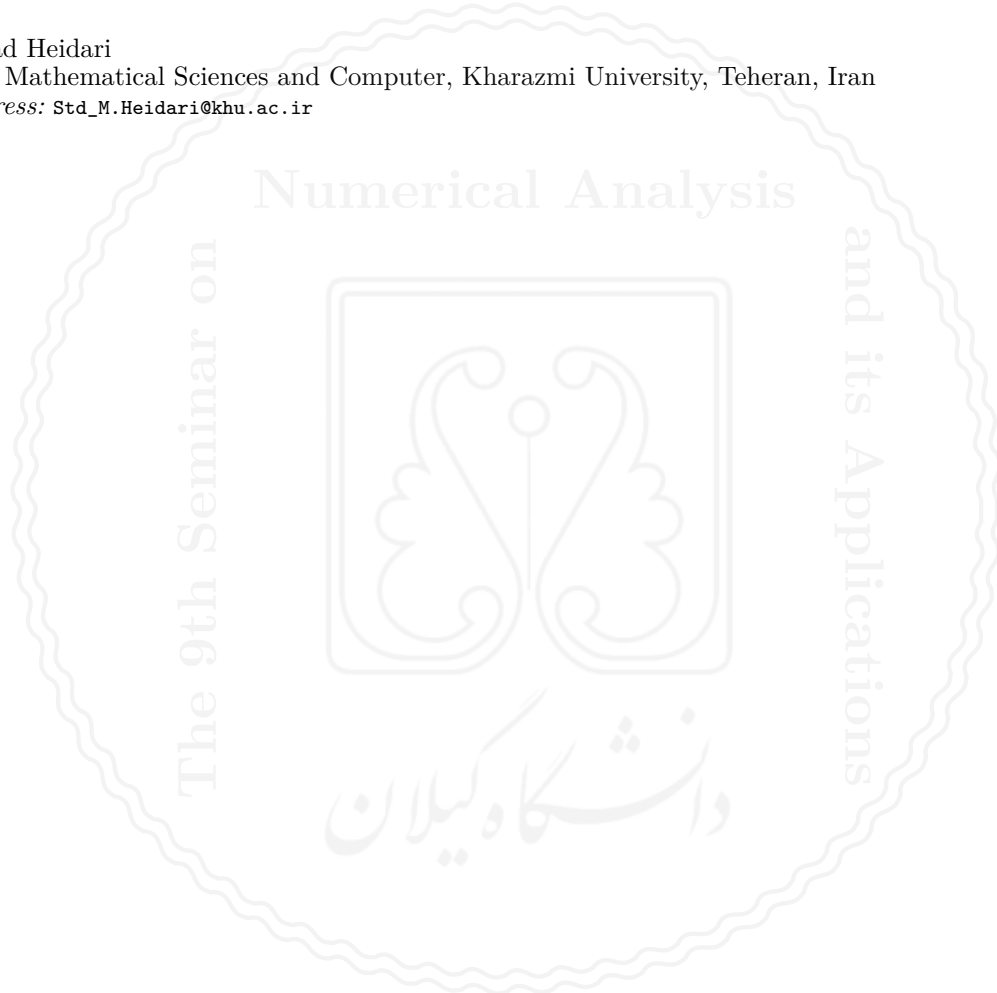
Faculty of Mathematical Sciences and Computer, Kharazmi University, Teheran, Iran

email address: m.mohammadi@khu.ac.ir

Mohammad Heidari

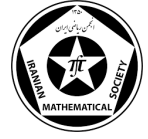
Faculty of Mathematical Sciences and Computer, Kharazmi University, Teheran, Iran

email address: Std_M.Heidari@khu.ac.ir





University of Guilan



Extended minimal residual biconjugate gradient stabilized method for generalized coupled Sylvester tensor equations

Mohammad Mahdi Izadkhah

Abstract. In this paper, a BiCG-like iterative method-minimal residual biconjugate gradient stabilized (MRBiCGStab)- is extended for solving the generalized coupled Sylvester tensor equations. The presented method uses tensor computations with no matricizations included. The reported numerical experiments show the performance of the proposed method.

1. Introduction

In this paper, we are concerned with the generalized coupled Sylvester tensor equations of the form

$$\sum_{j=1}^n \mathcal{X}_j \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN} = \mathcal{C}_i, \quad i = 1, 2, \dots, n, \quad (1.1)$$

where the matrices $A_{ijl} \in \mathbb{C}^{I_{ijl} \times I_{ijl}}$, $(i, j = 1, 2, \dots, n, \quad l = 1, 2, \dots, N)$, and the tensors $\mathcal{C}_i \in \mathbb{C}^{I_{i1} \times \cdots \times I_{iN}}$ ($i = 1, 2, \dots, n$) are known and the tensors $\mathcal{X}_i \in \mathbb{C}^{I_{i1} \times \cdots \times I_{iN}}$ ($i = 1, 2, \dots, n$) are unknown. In fact, for a positive integer N , an order N tensor $\mathcal{A} = (a_{i_1 \dots i_N}) (1 \leq i_j \leq I_j, j = 1, \dots, N)$ is a multidimensional array with $I_1 I_2 \cdots I_N$ entries. \mathcal{O} with all entries zero denotes the zero tensor. We use k -mode product $\times_k (k = 1, 2, \dots, N)$ in (1.1) that will be defined later. In the sequel, some basic definitions which will be used, are given from [3].

Definition 1.1. The operator $\times_k (k = 1, 2, \dots, N)$ represent the k -mode product of a tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ with a matrix $A \in \mathbb{C}^{J \times I_k}$ defined as

$$(\mathcal{X} \times_k A)_{i_1 i_2 \dots i_{k-1} j i_{k+1} \dots i_N} = \sum_{i_k=1}^{I_k} x_{i_1 i_2 \dots i_{k-1} i_k i_{k+1} \dots i_N} a_{j i_k}.$$

Definition 1.2. For a tensor $\mathcal{A} = (a_{i_1 \dots i_N j_1 \dots j_M}) \in \mathbb{C}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$, let $\mathcal{B} = (b_{i_1 \dots i_M j_1 \dots j_N}) \in \mathbb{C}^{J_1 \times \cdots \times J_M \times I_1 \times \cdots \times I_N}$ be the conjugate transpose of \mathcal{A} , where $b_{i_1 \dots i_M j_1 \dots j_N} = \bar{a}_{j_1 \dots j_N i_1 \dots i_M}$. The tensor \mathcal{B} is denoted by \mathcal{A}^* . When $b_{i_1 \dots i_M j_1 \dots j_N} = a_{j_1 \dots j_N i_1 \dots i_M}$, \mathcal{B} is called the transpose of \mathcal{A} , denoted by \mathcal{A}^T .

Definition 1.3. Let N and M be positive integers. The inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$ is defined by

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \sum_{j_1=1}^{J_1} \cdots \sum_{j_M=1}^{J_M} x_{i_1 \dots i_N j_1 \dots j_M} \bar{y}_{j_1 \dots j_M i_1 \dots i_N}.$$

Keywords: tensor equations, MRBiCGStab, iterative method, k -mode product.

AMS Mathematical Subject Classification [2010]: 15A10, 15A69, 15A72, 65F10.

So the tensor norm generated by this inner product is $\|\mathcal{X}\| = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$ which is called the tensor Frobenius norm.

Definition 1.4. Let $\mathcal{X}_i, \mathcal{Y}_i \in \mathbb{C}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$, for $i = 1, 2, \dots, n$. If we put $\mathfrak{X} = (\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n)$ and $\mathfrak{Y} = (\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_n)$, then $\langle \mathfrak{X}, \mathfrak{Y} \rangle = \sum_{i=1}^n \langle \mathcal{X}_i, \mathcal{Y}_i \rangle$ is an inner product and the associated norm is defined by $\|\mathfrak{X}\|_* = \sqrt{\langle \mathfrak{X}, \mathfrak{X} \rangle} = \sqrt{\sum_{i=1}^n \|\mathcal{X}_i\|^2}$.

We say that \mathfrak{X} and \mathfrak{Y} are orthogonal if and only if $\langle \mathfrak{X}, \mathfrak{Y} \rangle = 0$.

Definition 1.5. Let $\mathbb{H}_i = \mathbb{C}^{I_{i1} \times \dots \times I_{iN}}$ ($i = 1, 2, \dots, n$). Then

$$\mathcal{L} : \mathbb{H}_1 \times \mathbb{H}_2 \times \dots \times \mathbb{H}_n \longrightarrow \mathbb{H}_1 \times \mathbb{H}_2 \times \dots \times \mathbb{H}_n$$

$$\mathcal{L}(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n) = \begin{pmatrix} \sum_{j=1}^n \mathcal{X}_j \times_1 A_{1j1} \times_2 A_{1j2} \cdots \times_N A_{1jN} \\ \sum_{j=1}^n \mathcal{X}_j \times_1 A_{2j1} \times_2 A_{2j2} \cdots \times_N A_{2jN} \\ \vdots \\ \sum_{j=1}^n \mathcal{X}_j \times_1 A_{nj1} \times_2 A_{nj2} \cdots \times_N A_{njN} \end{pmatrix}.$$

According to Definition 1.5, the linear system (1.1) can be expressed as $\mathcal{L}(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n) = \mathcal{C}$, where $\mathcal{C} = (\mathcal{C}_1^T, \mathcal{C}_2^T, \dots, \mathcal{C}_n^T)^T$. We use Definition 1.4 and operator \mathcal{L} in Definition 1.5 to construct the k -th residual tensor of EMRBiCGStab method $\mathcal{R}_k = \mathcal{C} - \mathcal{L}(\mathcal{X}_{1,k}, \mathcal{X}_{2,k}, \dots, \mathcal{X}_{n,k})$ where $\mathcal{X}_{i,k}, i = 1, 2, \dots, n$ are k -th approximations for the tensor solutions $\mathcal{X}_i, i = 1, 2, \dots, n$.

2. Extended MRBiCGStab Method

In the special case of (1.1), Beik et al. have developed the gradient-based iterative method [1]. We would like to mention that in the BiCGStab method for solving the system $Ax = b$, the product of the linear polynomials are considered in construction of the residual form. In order to overcome the stagnation of convergence of BiCGStab in some discretized dominated problems, quadratic stabilization polynomials are used in the MRBiCGStab algorithm [4]. The MRBiCGStab method computes an approximation x_{2k} whose residual is of the form $r_{2k}^* = Q_{2k}^*(A)P_{2k}(A)r_0$, where $Q_0^*(t) = 1$ and for $k \geq 1$ $Q_{2k}^*(t) = (1 + w_1t + w_2t^2)(1 + w_3t + w_4t^2) \cdots (1 + w_{2k-1}t + w_{2k}t^2)$ and the parameters w_{2k-1} and w_{2k} are determined at the k -th iteration so that $\|r_{2k}^*\|$ is minimized. Based on this algorithm, we propose the Extended MRBiCGStab algorithm according to the tensor form for solving (1.1). Since the $\|\mathcal{R}_{k+2}^*\|_*$ is minimized over two dimensional vector space \mathbb{R}^2 , it may be expected that the extended MRBiCGStab converges faster than the BiCGStab in which residual norm is minimized over one dimensional vector space \mathbb{R} .

3. Numerical example

In this section, we give numerical results of the EMRBiCGStab method to solve the following problem

$$\mathcal{X} \times_1 A_1 \times_2 A_2 \times_3 A_3 + \mathcal{Y} \times_1 B_1 \times_2 B_2 \times_3 B_3 = \mathcal{C}_1,$$

$$\mathcal{X} \times_1 E_1 \times_2 E_2 \times_3 E_3 + \mathcal{Y} \times_1 F_1 \times_2 F_2 \times_3 F_3 = \mathcal{C}_2,$$

which show the effectiveness of the proposed algorithm. The matrices $A_i, B_i, E_i, F_i, i = 1, 2, 3$ have been chosen from [2] by the MATLAB function `rand` of appropriate size, and construct the right-hand side tensors $\mathcal{C}_i, i = 1, 2$ such that the exact solutions $\mathcal{X}^*, \mathcal{Y}^* \in \mathbb{C}^{m \times n \times l}$ would be tensors with all entries equal to one. The initial guess was taken to be the zero tensors and the stopping criterion $\frac{\|\mathcal{R}_k^*\|_*}{\|\mathcal{R}_0^*\|_*} \leq 10^{-6}$ or Max-iteration = 2000 were used. The corresponding convergence histories of EMRBiCGStab method with EBiCG [2] method are depicted in Figure 1.

Algorithm 1 The Extended MRBiCGStab method for solving (1.1)

1. **Input:** matrices A_{ijl} and tensors $\mathcal{X}_{j,0}, \mathcal{C}_i$, for $i, j = 1, 2, \dots, n$ and $l = 1, 2, \dots, N$
 2. Compute $\mathcal{R}_{i,0}^* = \mathcal{C}_i - \sum_{j=1}^n \mathcal{X}_{j,0} \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN}$, $i = 1, 2, \dots, n$
 3. Put $\mathcal{P}_{i,0}^* = \mathcal{R}_{i,0}^*$, $i = 1, 2, \dots, n$
 4. Choose arbitrary tensors $\tilde{\mathcal{R}}_{i,0}$ such that $\sum_{i=1}^n \langle \tilde{\mathcal{R}}_{i,0}, \mathcal{R}_{i,0}^* \rangle \neq 0$, for $i = 1, 2, \dots, n$
 6. For $k = 0, 2, \dots, 2m, \dots$ until $\|\mathcal{R}_{k+2}^*\|_*$ small enough Do
 7. Compute $\mathcal{V}_{i,k}^* = \sum_{j=1}^n \mathcal{P}_{j,k}^* \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN}$, for $i = 1, 2, \dots, n$
 8. $\alpha_k = \frac{\sum_{i=1}^n \langle \mathcal{R}_{i,k}^*, \tilde{\mathcal{R}}_{i,0} \rangle}{\sum_{i=1}^n \langle \mathcal{V}_{i,k}^*, \tilde{\mathcal{R}}_{i,0} \rangle}$
 9. $\bar{\mathcal{X}}_{i,k+1} = \mathcal{X}_{i,k} + \alpha_k \mathcal{P}_{i,k}^*$, for $i = 1, 2, \dots, n$
 10. $\bar{\mathcal{R}}_{i,k+1} = \mathcal{R}_{i,k}^* - \alpha_k \mathcal{V}_{i,k}^*$, for $i = 1, 2, \dots, n$
 11. Compute $\hat{\mathcal{R}}_{i,k+1} = \sum_{j=1}^n \bar{\mathcal{R}}_{j,k+1} \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN}$, $i = 1, 2, \dots, n$
 12. $\beta_k = -\alpha_k \frac{\sum_{i=1}^n \langle \hat{\mathcal{R}}_{i,k+1}, \tilde{\mathcal{R}}_{i,0} \rangle}{\sum_{i=1}^n \langle \mathcal{R}_{i,k}^*, \tilde{\mathcal{R}}_{i,0} \rangle}$
 13. $\bar{\mathcal{P}}_{i,k+1} = \bar{\mathcal{R}}_{i,k+1} + \beta_k \mathcal{P}_{i,k}^*$, for $i = 1, 2, \dots, n$
 14. $\hat{\mathcal{P}}_{i,k+1} = \hat{\mathcal{R}}_{i,k+1} + \beta_k \mathcal{V}_{i,k}^*$, for $i = 1, 2, \dots, n$
 15. $\hat{\mathcal{P}}_{i,k+1}^* = \sum_{j=1}^n \hat{\mathcal{P}}_{j,k+1} \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN}$, for $i = 1, 2, \dots, n$
 16. $\alpha_{k+1} = \frac{\sum_{i=1}^n \langle \hat{\mathcal{R}}_{i,k+1}, \tilde{\mathcal{R}}_{i,0} \rangle}{\sum_{i=1}^n \langle \hat{\mathcal{P}}_{i,k+1}^*, \tilde{\mathcal{R}}_{i,0} \rangle}$
 17. $\bar{\mathcal{X}}_{i,k+2} = \bar{\mathcal{X}}_{i,k+1} + \alpha_{k+1} \bar{\mathcal{P}}_{i,k+1}$, for $i = 1, 2, \dots, n$
 18. $\bar{\mathcal{R}}_{i,k+2} = \bar{\mathcal{R}}_{i,k+1} - \alpha_{k+1} \hat{\mathcal{P}}_{i,k+1}$, for $i = 1, 2, \dots, n$
 19. $\hat{\mathcal{R}}_{i,k+2} = \hat{\mathcal{R}}_{i,k+1} - \alpha_{k+1} \hat{\mathcal{P}}_{i,k+1}^*$, for $i = 1, 2, \dots, n$
 20. $\hat{\mathcal{R}}_{i,k+2}^* = \sum_{j=1}^n \hat{\mathcal{R}}_{j,k+2} \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN}$, for $i = 1, 2, \dots, n$
 21. $w_{k+1} = \frac{(\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2}^* \rangle)(\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2}^* \rangle) - (\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \bar{\mathcal{R}}_{i,k+2} \rangle)(\sum_{i=1}^n \langle \hat{\mathcal{R}}_{i,k+2}^*, \hat{\mathcal{R}}_{i,k+2}^* \rangle)}{(\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \bar{\mathcal{R}}_{i,k+2} \rangle)(\sum_{i=1}^n \langle \hat{\mathcal{R}}_{i,k+2}^*, \hat{\mathcal{R}}_{i,k+2}^* \rangle) - (\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2}^* \rangle)(\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2}^* \rangle)}$
 22. $w_{k+2} = \frac{(\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2} \rangle)(\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2}^* \rangle) - (\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \bar{\mathcal{R}}_{i,k+2} \rangle)(\sum_{i=1}^n \langle \hat{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2} \rangle)}{(\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2}^* \rangle)(\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2}^* \rangle) - (\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+2}, \bar{\mathcal{R}}_{i,k+2} \rangle)(\sum_{i=1}^n \langle \hat{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2} \rangle)}$
 23. $\mathcal{X}_{i,k+2} = \bar{\mathcal{X}}_{i,k+2} - w_{k+1} \bar{\mathcal{R}}_{i,k+2} - w_{k+2} \hat{\mathcal{R}}_{i,k+2}$, for $i = 1, 2, \dots, n$
 24. $\mathcal{R}_{i,k+2}^* = \bar{\mathcal{R}}_{i,k+2} + w_{k+1} \hat{\mathcal{R}}_{i,k+2} + w_{k+2} \hat{\mathcal{R}}_{i,k+2}^*$, for $i = 1, 2, \dots, n$
 25. $\beta_{k+1} = -\alpha_{k+1} \frac{\sum_{i=1}^n \langle \hat{\mathcal{R}}_{i,k+2}^*, \tilde{\mathcal{R}}_{i,0} \rangle}{\sum_{i=1}^n \langle \bar{\mathcal{R}}_{i,k+1}, \tilde{\mathcal{R}}_{i,0} \rangle}$
 26. $\mathcal{P}_{i,k+2}^* = \mathcal{R}_{i,k+2}^* + \beta_{k+1} (\bar{\mathcal{P}}_{i,k+1} + w_{k+1} \hat{\mathcal{P}}_{i,k+1} + w_{k+2} \hat{\mathcal{P}}_{i,k+1}^*)$, for $i = 1, 2, \dots, n$
 27. EndDo
 28. **Output:** solution \mathcal{X}_j for (1.1)
-

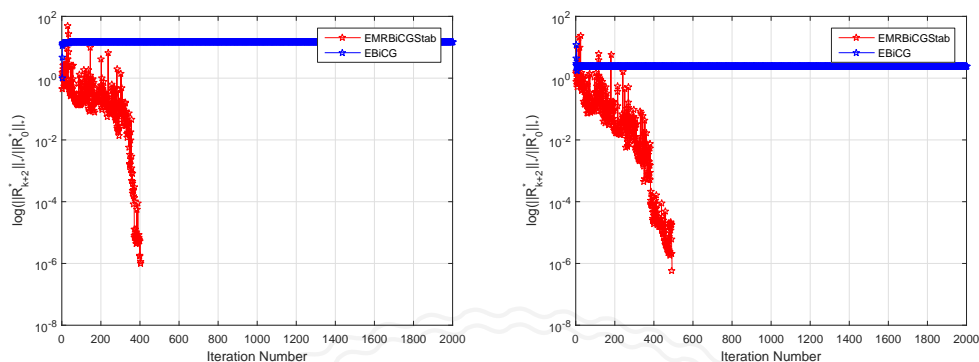


Figure 1: Convergence histories of REE for $m = 6, n = 4, l = 3$ (left) and $m = 8, n = 4, l = 3$ (right)

References

- [1] F.P.A. Beik, S. Ahmadi-Asl, Residual norm steepest descent based iterative algorithms for Sylvester tensor equations, *Journal of Mathematical Modeling*, 2 (2015) 115–131.
- [2] E.D. Khosravi, S. Karimi, Extended conjugate gradient squared and conjugate residual squared methods for solving the generalized coupled Sylvester tensor equations, *Transactions of the Institute of Measurement and Control*, 43 (2021) 519–527.
- [3] T.G. Kolda, B.W. Bader, ; Tensor decompositions and applications, *SIAM Review*, 51 (2009) 455–500.
- [4] J.H. Yun, M.S. Joo, BCG like methods for solving nonsymmetric linear systems, *Journal of The Chungcheong Mathematical Society*, 8 (1995) 55–69.

Mohammad Mahdi Izadkhah

Department of Computer Science, Faculty of Computer & Industrial Engineering, Birjand University of Technology, Birjand, Iran

email address: izadkhah@birjandut.ac.ir



University of Guilan



An efficient Lucas wavelet Galerkin method for solving time-delay optimal control problems

Sedigheh Sabermahani, Yadollah Ordokhani, Farshid Nourian, Mehrdad Lakestani

Abstract. Here, we present a numerical scheme to solve optimal control problems with time-varying delay system. This method is based on Lucas wavelets and Galerkin method. Operational matrices of integration and delay for Lucas wavelets are proposed. Then, Galerkin method is used to solve the mentioned problems. Numerical results are included to demonstrate the efficiency of the present technique.

1. Introduction

One of important class of delay problems is optimal control problems that are used to model many of the phenomena. Furthermore, there are several numerical methods to solve delay optimal control problems such as Variational iteration method [1] and finite difference method [2]. In recent years, the construction and application of different wavelets such as Bernoulli wavelet [3], Fibonacci wavelet [4], Legendre wavelet [5] has been shown to be a powerful mathematical tool for discretization of selected problems.

In this work, we apply the extended Lucas wavelets for solving fractional delay optimal control problems. To this end, we approximate the fractional derivative of the state variables and control variables in terms of these wavelets. We present new fractional integration and delay operational matrices for these functions. Then, by employing the operational matrices and Galerkin method, the problems under consideration are converted into systems of algebraic equations. The validity of the established methods is studied in one example.

2. Lucas wavelets and properties

The Lucas wavelets are defined over the interval $[0, 1]$ in [6]. We present a new presentation of these functions in the following form

$$\psi_{n,m}(t) = \frac{2^{\frac{k-1}{2}}}{\sqrt{w_m}} \tilde{L}_m(2^{k-1}t - n + 1) \chi_{[\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}}]}(t), \quad n = 1, 2, \dots, 2^{k-1}, \quad m = 0, 1, \dots, M-1, \quad (2.1)$$

and $\chi_{[\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}}]}(t)$ is the characteristic function, $w_m = \int_0^1 \tilde{L}_m^2(t) dt$, and $\tilde{L}_m(t), m = 0, 1, \dots, M-1$ denotes the Lucas polynomials. $n = 1, 2, \dots, 2^{k-1}$ and k is a positive integer. Also, $\tilde{L}_0(t) = 2$, $\tilde{L}_m(t) = \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \frac{m}{m-i} \binom{m-i}{i} t^{m-2i}$.

Keywords: Lucas wavelet, Galerkin method, optimal control problem, time-varying delay system.

AMS Mathematical Subject Classification [2010]: 65T60, 44A45, 49N05.

2.1. Integration operational matrix of Lucas wavelets

Let

$$\Psi(t) = [\psi_{1,0}, \psi_{1,1}, \dots, \psi_{1,M-1}, \psi_{2,0}, \psi_{2,1}, \dots, \psi_{2,M-1}, \dots, \psi_{2^{k-1},0}, \dots, \psi_{2^{k-1},M-1}]^T,$$

be Lucas wavelets vector. The integration operational matrix of Lucas wavelets \tilde{P} is defined as $\int_0^t \Psi(s)ds \simeq \tilde{P}\Psi(t)$, each of element of this matrix is derived in the following process

$$\begin{aligned} \psi_{n,m}(t) &= \frac{2^{\frac{k-1}{2}}}{\sqrt{w_m}} \tilde{L}_m(2^{k-1}t - n + 1) \chi_{[\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}}]}(t) \\ &= \frac{2^{\frac{k-1}{2}}}{\sqrt{w_m}} \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=0}^{m-2i} \binom{m-i}{i} \frac{m}{m-i} 2^{kj-j} t^j (1-n)^{m-2i-j} \chi_{[\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}}]}(t). \end{aligned}$$

Now, using the above relation, we get

$$\begin{aligned} \int_0^t \psi_{n,m}(s)ds &= \frac{2^{\frac{k-1}{2}}}{\sqrt{w_m}} \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=0}^{m-2i} \binom{m-i}{i} \frac{m}{m-i} 2^{kj-j} (1-n)^{m-2i-j} \int_0^t s^j \chi_{[\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}}]}(s)ds \\ &= \frac{2^{\frac{k-1}{2}}}{\sqrt{w_m}} \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=0}^{m-2i} \binom{m-i}{i} \frac{m}{m-i} 2^{kj-j} (1-n)^{m-2i-j} \theta_j(t), \\ \theta_j(t) &= \int_0^t s^j \chi_{[\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}}]}(s)ds = \int_{\frac{n-1}{2^{k-1}}}^t s^j ds \chi_{[\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}}]}(t) + \int_{\frac{n-1}{2^{k-1}}}^{\frac{n}{2^{k-1}}} s^j ds \chi_{[\frac{n}{2^{k-1}}, 1]}(t), \end{aligned}$$

Here, we expand this function in terms of Lucas wavelets as $\theta_j(t) = \sum_{s=1}^{2^{k-1}} \sum_{p=0}^{M-1} b_{s,p} \psi_{s,p}(t)$. Therefore, we achieve $\int_0^t \psi_{n,m}(s)ds = \sum_{s=1}^{2^{k-1}} \sum_{p=0}^{M-1} \Theta_{s,p}^{n,m} \psi_{s,p}(t)$,

$$\Theta_{s,p}^{n,m} = \frac{2^{\frac{k-1}{2}}}{\sqrt{w_m}} \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=0}^{m-2i} \binom{m-i}{i} \frac{m}{m-i} 2^{kj-j} (1-n)^{m-2i-j} b_{s,p} \quad (2.2)$$

2.2. Delay operational matrix of Lucas wavelets

We suppose that $\tau = \frac{s}{2^{k-1}}$, then $\Psi(t - \tau) = \tilde{D}\Psi(t)$, $t > \tau$. For this approach, we have

$$\begin{aligned} \psi_{n,m}(t - \tau) &= \begin{cases} \frac{2^{\frac{k-1}{2}}}{\sqrt{w_m}} \tilde{L}_m(2^{k-1}(t - \tau) - n + 1), & \frac{n-1}{2^{k-1}} \leq t - \tau < \frac{n}{2^{k-1}}, \\ 0, & \text{otherwise} \end{cases} \\ &= \begin{cases} \frac{2^{\frac{k-1}{2}}}{\sqrt{w_m}} \tilde{L}_m(2^{k-1}t - (n + s) + 1), & \frac{n-1+s}{2^{k-1}} \leq t < \frac{n+s}{2^{k-1}}, \\ 0, & \text{otherwise} \end{cases} = \psi_{m,n+s}(t). \end{aligned}$$

3. Numerical method

Here, we consider

$$\begin{cases} X'(t) = A(t)X(t) + B(t)X(t - \tau) + G(t)U(t) + H(t)U(t - \tau), & t \in [0, 1], \\ X(0) = X_0, \\ X(t) = \phi_1(t), & t \in [-\tau, 0), \\ U(t) = \phi_2(t), & t \in [-\tau, 0). \end{cases} \quad (3.1)$$

where $X(t) \in R^l$, $U(t) \in R^q$, ($l \geq q$) $A(t), B(t), G(t)$ and $H(t)$ are continuous matrices with appropriate dimensions, X_0 is constant vector and $\phi_1(t)$ and $\phi_2(t)$ are known functions defined on the interval $[-\tau, 0)$. In Eq. (3.1), we derive $X(t)$ and optimal control $U(t)$ which are satisfied in conditions of problem while extremizing J

$$J = X^T(1)Q(1)X(1) + \int_0^1 [X^T(t)Q(t)X(t) + U^T(t)R(t)U(t)]dt, \quad (3.2)$$

where $Q(t), R(t)$ are matrix functions with appropriate dimensions. Also, $Q(t)$ is a symmetric positive-semi-definite matrix and $R(t)$ is a symmetric positive-definite matrix. Assume that $X(t) = [X_1(t), X_2(t), \dots, X_l(t)]^T$, $U(t) = [U_1(t), U_2(t), \dots, U_q(t)]^T$, and $[\tilde{\Psi}_l(t) = I_l \otimes \Psi(t)$, $\tilde{\Psi}_q(t) = I_q \otimes \Psi(t)$, where I_l, I_q are the l -dimensional and q -dimensional identity matrices and \otimes denotes Kronecker product. Moreover $\tilde{\Psi}_l(t)$ and $\tilde{\Psi}_q(t)$, $i = 1, 2, \dots, l$, $j = 1, 2, \dots, q$ are $l2^{k-1}M \times l$ and $q2^{k-1}M \times q$ matrices. We approximate as $X'_i(t) = X_i^T \Psi(t)$, $U_j(t) = U_j^T \Psi(t)$. Therefore, we get $X'(t) = X^T \tilde{\Psi}_l(t)$, $U(t) = U^T \tilde{\Psi}_q(t)$, and for $X(t)$, we can see that $X(t) = X^T \hat{P} \tilde{\Psi}_l(t) + \hat{E}^T \tilde{\Psi}_l(t)$, where $\hat{P} = I_l \otimes \tilde{P}$, $\hat{E} = I_l \otimes \tilde{E}$ and $X(0) \simeq \tilde{E}^T \Psi(t)$. Also, using delay operational matrix of Lucas wavelets, we expand $X(t - \tau)$ and $U(t - \tau)$ in terms of them as

$$X(t - \tau) = \begin{cases} \phi_1(t - \tau), & 0 \leq t \leq \tau \\ X^T \hat{P} \hat{D}_l \tilde{\Psi}_l(t) + \hat{E}^T \hat{D}_l \tilde{\Psi}_l(t), & \tau \leq t \leq 1, \end{cases}$$

$$U(t - \tau) = \begin{cases} \phi_2(t - \tau), & 0 \leq t \leq \tau \\ U^T \hat{D}_q \tilde{\Psi}_q(t), & \tau \leq t \leq 1, \end{cases}$$

where $\hat{D}_q = I_q \otimes \tilde{D}$, $\hat{D}_l = I_l \otimes \tilde{D}$ and \tilde{D} is delay operational matrix of Lucas wavelets. Moreover, we approximate $A(t), B(t), G(t)$, and $H(t)$. We substitute the approximations in the system then we derive $\tilde{\Phi}(t)$. The resulting equation can be solved using Galerkin method $Y = \langle \tilde{\Phi}, \Psi \rangle$. Also, we substitute the mentioned approximations and the performance index J , so we have $J^* = J + \lambda^T Y$, where $\lambda = [\lambda_{n,m}]$ $n = 1, 2, \dots, 2^{k-1}$, $m = 0, 1, \dots, M - 1$ are the unknown multipliers coefficients. For deriving extremum of J^* , the necessary condition is that the following equations hold $\frac{\partial J^*}{\partial X} = 0$, $\frac{\partial J^*}{\partial U} = 0$, $\frac{\partial J^*}{\partial \lambda} = 0$. We can solve these equations using "FindRoot" package in Mathematica software.

4. Numerical results

Example. Consider $J = \frac{3}{2}X^2(2) + \frac{1}{2} \int_0^2 U^2(t)dt$, subject to the time-delay system

$$X'(t) = X(t) + X(t - 1) + U(t), \quad 0 \leq t \leq 2, \quad X(t) = 1, \quad -1 \leq t \leq 0,$$

in which the analytic solution for $U(t)$ is $U(t) = \begin{cases} \delta(e^{2-t} + \frac{(1-t)e^{1-t}}{\delta e^{2-t}}), & 0 \leq t \leq 1, \\ 1 \leq t \leq 2. \end{cases}$ and with $\delta = -0.3932$, $J \simeq 3.1017$ [4]. This example solved by several numerical techniques such as variational iteration method [1] and finite difference method [2] with $h = 0.01$. In Table 1, these numerical results are compared to the results obtained using the present method for different values of k, M .

Table 1: Comparison of the value of J , in Example.

Numerical methods	J
Variational iteration method [1]	3.1091
Finite difference method [2]	3.102519
Present technique ($k=3$, $M=6$)	3.10078
Present technique ($k=3$, $M=8$)	3.10101
Exact value	3.1017

References

- [1] S. M. Mirhosseini-Alizamini, S. Effati, A. Heydari, An iterative method for suboptimal control of linear time-delayed systems. *Systems & Control Letters*, 82 (2015) 40–50.
- [2] A. Jajarmi, M. Hajipour, An efficient finite difference method for the time-delay optimal control problems with time-varying delay, *Asian Journal of Control*, 19 (2017) 554–563.
- [3] P. Rahimkhani, Y. Ordokhani, E. Babolian, An efficient approximate method for solving delay fractional optimal control problems. *Nonlinear Dynamics*, 86 (2016) 1649–1661.
- [4] S. Sabermahani, Y. Ordokhani, S.A. Yousefi, Fibonacci wavelets and their applications for solving two classes of time-varying delay problems. *Optimal Control Applications and Methods*, 41 (2020) 395–416.
- [5] F. Khellat, Optimal control of linear time-delayed systems by linear Legendre multiwavelets, *Journal of Optimization Theory and Applications*, 143 (2009) 107–121.
- [6] H. Dehestani, Y. Ordokhani, M. Razzaghi, Combination of Lucas wavelets with Legendre-Gauss quadrature for fractional Fredholm–Volterra integro-differential equations, *Journal of Computational and Applied Mathematics*, 382 (2021) no. 113070.

Sedigheh Sabermahani

Department of Mathematics, Faculty of Mathematical Sciences, Alzahra University, Tehran, Iran
email address: s.saber@alzahra.ac.ir

Yadollah Ordokhani

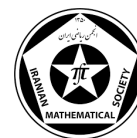
Department of Mathematics, Faculty of Mathematical Sciences, Alzahra University, Tehran, Iran
email address: ordokhani@alzahra.ac.ir

Farshid Nourian

Faculty of Mathematics, Statistics, and Computer Science, University of Tabriz, Tabriz, Iran
email address: f.nourian@tabrizu.ac.ir

Mehrdad Lakestani

Faculty of Mathematics, Statistics, and Computer Science, University of Tabriz, Tabriz, Iran
email address: lakestani@tabrizu.ac.ir



Numerical solution of mixed fractional heston partial differential equation

Fereshteh Goldoust

Abstract. In this work the Mixed Fractional Heston partial differential equation option pricing model considered and Legendre wavelet method (LWM) use to solve this equation with reducing feature the PDEs problem into the solution of ODEs system. The wavelet base is used in approximation due to its simplicity and efficiency. Some numerical schemes has compared with the LWM in result.

1. Introduction

Since the Black-Scholes model was introduced forty years ago, practitioners and academics have been proposing refinements there of in order to take into account the specific behaviour of market data. In particular, stochastic volatility models, turning the constant Black-Scholes instantaneous volatility of returns into a stochastic process, have been studied and used heavily [1–3]. The Heston Model is one of the most widely used stochastic volatility (SV) models today. Its attractiveness lies in the powerful duality of its tractability and robustness relative to other SV models [4]. To remedy this issue, several authors have suggested the addition of jumps, either in the form of an independent Levy process or within the more general framework of affine processes [5–7]. In continuous time, this long-memory behaviour has been modelled through fractional Brownian motion with Hurst exponent strictly greater than $1/2$ [8, 9]. Fractional Brownian motion has its pitfalls though, since it is not a semimartingale, and yields arbitrage opportunities [10].

2. Mixed Fractional Heston Model

One of the most widely used random processes in financial market modeling is the use of memory processes. Which includes fractional Brownian motion and its derivative process. This process is the generalization of a parameter of standard Brownian motion, and this parameter is named after the English researcher Harold Edwin Hurst with $H \in (0, 1)$. In the case where $H = \frac{1}{2}$, the Brownian motion of the fraction corresponds to the standard Brownian motion [11–13].

2.1. Fractional Brownian motion

The purpose of adding the long-run dependency feature to the geometric Brownian model is to obtain a European stock price dependent on a stock whose dynamics follow the fractional geometric model. The fractional geometric Brownian model is

$$dS(t) = \mu S(t)dt + \sigma S(t)dB^H(t), \quad (2.1)$$

Keywords: Stochastic differential equation, Partial differential equation, Mixed Fractional Heston Model, Legendre Wavelet Method (LWM).

AMS Mathematical Subject Classification [2010]: 13F55, 05E40, 05C65 .

Where μ is the expected return on the stock and σ , stock price volatility and $B^H(t)$ are the Brownian deficit.

2.2. Mixed Fractional Heston partial differential equation

To fit a real market, generalizations to the geometric Brownie model are needed, one of which is random volatility models. In this type of model, volatility is considered as a positive process. So if the price (discounted) of the commodity is at risk, in the differential equation

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t). \quad (2.2)$$

Now, if the turbulence of the Discounted Feynman-Kac Theorem [14–16] follows the Cox-Ingersell process, where μ , expected stock returns, k , average volatility return rate, η , long-term volatility average, σ , Heston model volatility, $B_{1,t}$ and $B_{2,t}$ are standard Brownian motions with a correlation coefficient of ρ , which are $\rho \in (-1, 1)$, And $S(t)$ and $V(t)$ are stock prices and volatility rates, respectively.

$$\begin{cases} dS(t) = \mu S(t)dt + \sqrt{V(t)}S(t)dM^{1,H}(t), \\ dV(t) = k(\eta - V(t))dt + \sigma\sqrt{V(t)}dM^{2,H}(t), \\ dM^{1,H}(t)dM^{2,H}(t) = \rho(dt + \alpha^2 H t^{2H-1}dt). \end{cases} \quad (2.3)$$

Where $M_t^{1,H}$ and $M_t^{2,H}$, are fractional mixed Brownian with $H \in (\frac{3}{4}, 0)$, is a Hurst parameter.

By applying ITOs lemma The Mixed Fractional Heston partial differential equation pricing model as follows

$$\begin{aligned} \frac{\partial u}{\partial t} + v s^2(t)(1 + \alpha^2 H t^{2H-1}) \frac{\partial^2 u}{\partial s^2} + \sigma^2 v(1 + \alpha^2 H t^{2H-1}) \frac{\partial^2 u}{\partial v^2} \\ + \sigma v s \rho(1 + \alpha^2 H t^{2H-1}) \frac{\partial^2 u}{\partial s \partial v} - \eta \frac{\partial u}{\partial v} - r u + r s \frac{\partial u}{\partial s}. \end{aligned} \quad (2.4)$$

3. Metodology

The wavelet basis is constructed from a single function, which is called the mother wavelet. Legendre differential equation [17–20]. One dimension Legendre wavelets over the interval $[0, 1]$ defined as

$$\psi_{n,m}(x) = \begin{cases} \sqrt{\frac{k}{(m + \frac{1}{2})2^2}} P_m(2^k x - 2n + 1), & \frac{n-1}{2^{k-1}} \leq x \leq \frac{n}{2^{k-1}}, \\ 0, & o.w. \end{cases} \quad (3.1)$$

With $n = 1, 2, \dots, 2k - 1, m = 0, 1, 2, \dots, M - 1$. In Eq. (3.1) $\{P_m\}$'s are ordinary Legendre functions of order m is defined over the interval $[-1, 1]$. Legendre wavelet is an orthonormal set as

$$\int_0^1 \psi_{n,m}(x) \psi_{n',m'}(x) dx = \delta_{n,n'} \delta_{m,m'}. \quad (3.2)$$

Any element $f \in L^2([0, 1])$, may be expanded as

$$f(x) \cong \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} C_{n,m} \psi_{n,m}(x), \quad (3.3)$$

With boundary condition $u(s, v, 0) = \beta_1(s, v, t)$ and $\frac{\partial u(s, v, 0)}{\partial t} = \beta_2(s, v, t)$,

$$\frac{\partial u}{\partial t} = C_1^T(s, v, t)\psi(s, v, t), \quad (3.4)$$

$$u(s, v, t) = C_1^T(s, v, t)P\psi(s, v, t) + \beta_2(u, s, t). \quad (3.5)$$

Substituting (3.4) and (3.5) in (2.4), we obtain

$$\begin{aligned} C^T(s, v, t) = & \alpha_1 \left(\frac{\partial^2 C^T(s, v, t)}{\partial s^2} P\psi(s, v, t) + \frac{d^2 \beta_2(u, s, t)}{ds^2} d^T \psi \right) \\ & + \alpha_2 \left(\frac{\partial^2 C^T(s, v, t)}{\partial v^2} P\psi(s, v, t) + \frac{d^2 \beta_2(u, s, t)}{dv^2} d^T \psi \right) \\ & + \alpha_3 \left(\frac{d}{ds} \frac{d C^T(s, v, t)}{dv} P\psi(s, v, t) + \frac{d}{ds} \frac{d \beta_2(u, s, t)}{dv} d^T \psi \right) \\ & + \alpha_4 \left(\frac{\partial C^T(s, v, t)}{\partial s} P\psi(s, v, t) + \frac{d \beta_2(u, s, t)}{ds} d^T \psi \right) \\ & + \alpha_5 \left(\frac{\partial C^T(s, v, t)}{\partial v} P\psi(s, v, t) + \frac{d \beta_2(u, s, t)}{dv} d^T \psi \right) \\ & - \alpha_6 (C^T(s, v, t) P\psi + \beta_2 d^T \psi), \end{aligned}$$

which $1 = d^T \psi(s, v, t)$.

4. Results

The Mixed Fractional Heston partial differential equation pricing model can be solved for unknown coefficients of the vector, in this case, Adomian decomposition method have used [21]. Consequently, the solution can be calculated $C(s, v, t)$.

Table 1: The values of parameters.

ρ	δ	η	α	H
0.06	0.04	0.12	0.2	3/4
0.1	0.9	0.2	0.5	3/4

5. Conclusion

The Legendre wavelet method has been applied to solve the Mixed Fractional Heston partial differential equation pricing model which these PDE equations have been derived from stochastic differential equations by using one of the important stochastic calculus Lemma named Ito. Also in this work results of LWM have been compared with the Adomian Decomposition Method (ADM).

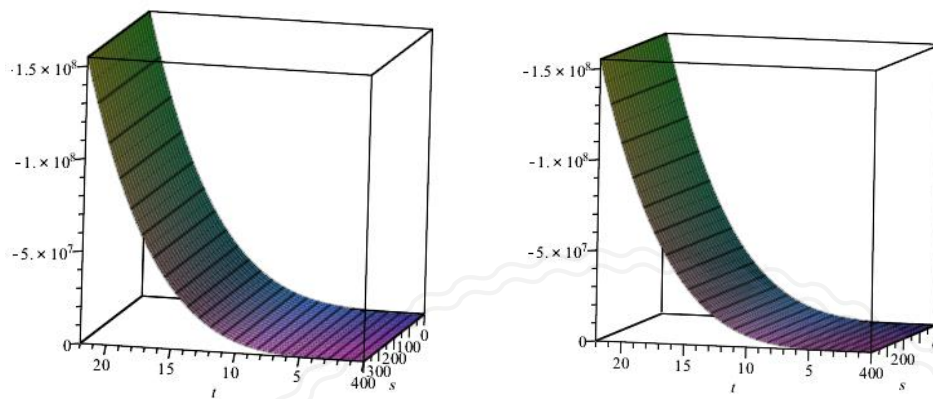


Figure 1: Solution Mixed Fractional Heston PDE model by LWM & ADM

References

- [1] J.P. Fouque, G. Papanicolaou, R. Sircar, K. Solna, *Multiscale Stochastic Volatility for Equity, Interest Rate, and Credit Derivatives*, CUP, 2011.
- [2] M. Fukasawa, Asymptotic analysis for stochastic volatility: martingale expansion, *Finance and Stochastics*, 15 (2011) 635-654.
- [3] J. Gatheral, *The Volatility Surface: a practitioner's guide*, Wiley, 2006.
- [4] N. Moodley, *The Heston Model: A Practical Approach*, 2005, 1-53
- [5] D.S. Bates, Jumps and stochastic volatility: exchange rate processes implicit in Deutsche Mark options. *Review of Financial Studies*, 9 (1996) 69-107.
- [6] A. Jacquier, M. Keller-Ressel, A. Mijatoviffic, Large deviations and stochastic volatility with jumps: asymptotic implied volatility for affine models. *Stochastics*, 85 (2013) 321-345.
- [7] C.W.J. Granger, R. Joyeux, An introduction to long memory time series models and fractional differencing. *Journal of Time Series Analysis*, 1 (1980) 15-39.
- [8] F. Comte, E. Renault, Long memory in continuous-time stochastic volatility models, *Math Finance*, 8 (1998) 291-323, .
- [9] F. Comte, L. Coutin, E. Renault, Affine fractional stochastic volatility models, *Annals of Finance*, 8 (2012) 337-378.
- [10] C. Bayer, P.K. Friz, J. Gatheral, Pricing Under Rough Volatility, *Quantitative Finance*, 16 (2016) 887-904.
- [11] B. Yan, *Option Pricing under the generalized mixed fractional Brownian motion model*, Master Thesis, University of Wollongong, 2014.
- [12] M. Zili, On the mixed fractional Brownian motion, *Journal of Applied Mathematics and Stochastic Analysis*, Vol. (2006), Article ID 32435, 9 pages.

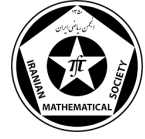
- [13] A.D. Wentzell, A course in the theory of stochastic processes, *SIAM Review*, 2 (1982) 361–362.
- [14] G. Adomian, *Stochastic differential equations*, Kluwer, Boston, 1994.
- [15] J. D. Duffy, *Finite difference method in financial engineering*, 2006.
- [16] S. E. Shreve, *Stochastic calculus for finance- continuous time model*, springer, 2003.
- [17] J.S. Gu, W.S. Jiang, The Haar Wavelets operational matrix of integration, *International Journal of Systems Science*, 27 (1996) 623–628.
- [18] M. Razzaghi, S. Yousefi, Legendre Wavelets Direct method for Variational problems, *Mathematics and Computers in Simulation*, 53 (2000) 185–192.
- [19] R.Y. Chang, M.L. Wang, Shifted Legendre directs method for Variational problems, *Journal of Optimization Theory and Application*, 39 (1983) 22–307.
- [20] L. Nanshan, E.B. Lin, Legendre wavelet method for numerical solutions of partial differential equations, *Wiley Interscience*, (2009) 85–88.
- [21] J. Biazar , F. Goldoust, F. Mehrdoust, On the Numerical Solutions of Heston Partial Differential Equation, *Mathematical Sciences Letters*, 4 (2015) 63–68.

Fereshteh Goldoust

Department of Applied Mathematics, Bandar Anzali Branch, Islamic Azad University, Bandar Anzali, Iran
email address: feriigolii@gmail.com, f.goldoust@iaubanz.ac.ir,



University of Guilan



Stable Gaussian elimination algorithm for ill-conditioned banded toeplitz

Nasser Akhoundi

Abstract. In this paper, banded Toeplitz matrices generated by $f(\theta) = (2(1 - \cos(\theta - \tilde{\theta})))^d$ are studied. The function f is a real non-negative function with a zero of order $2d$ at $\tilde{\theta}$. Hence the generating matrix is an ill-conditioned Hermitian positive definite matrix. We propose an $O(n)$ recursive Gaussian elimination algorithm to solve the linear systems with these matrices. Numerical experiments show that our proposed method is faster and more stable than the stable Levinson algorithm.

1. Introduction

Toeplitz matrices $T_n = (t_{i-j})_{i,j=1,\dots,n}$ can be interpreted as Fourier coefficients of the generating function $f(\theta) = \sum_{-\infty}^{\infty} t_k e^{ik\theta}$, defined on $[-\pi, \pi]$, i.e., $t_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-ik\theta} d\theta$ for $k = 0, \pm 1, \pm 2, \dots$. If the generating function $f(\theta)$ is non-negative and real, then T_n is Hermitian positive definite (HPD) matrix, furthermore for the even function f , The Toeplitz matrix T_n is real and symmetric.

In this paper we assume that

$$f(\theta) = (2(1 - \cos(\theta - \tilde{\theta})))^d, \quad (1.1)$$

where $\tilde{\theta} \in [-\pi, \pi]$ is a constant number. In this case we name the generated Toeplitz matrix by $T_{\tilde{\theta},n}^{(d)}$. $T_{\tilde{\theta},n}^{(d)}$ is banded HPD, with bandwidth $2d - 1$. For the special case $\tilde{\theta} = 0$ we omit the subscript $\tilde{\theta}$ in $T_{\tilde{\theta},n}^{(d)}$ and we name it $T_n^{(d)}$. These linear systems arise in the discretization of differential equations. In [3] the authors show that for general generating function f where $|f|$ has zeros of even order, their singularity lies in banded Toeplitz generated from 1.1 and these banded Toeplitz matrices can be perform as a good preconditioner. Function f in (1.1) has zero of order $2d$ at $\tilde{\theta}$, hence the condition number of these matrices can be very large. As an example for $d = 2$, in [1], the authors showed that the condition number of these matrices is about $16^2(\frac{n+2}{3\pi})^4$.

2. Main Results

We use the following theorem [4] to describe our Gaussian elimination method.

Theorem 2.1. Let $T_{\tilde{\theta},n}^{(d)}$ be an n -by- n , banded Toeplitz matrix with bandwidth $2d - 1$ generated by $f(\theta) = (2(1 - \cos(\theta - \tilde{\theta})))^d$. Then

$$T_{\tilde{\theta},n}^{(d)} = T_{\tilde{\theta},n}^{(d-1)} T_{\tilde{\theta},n}^{(1)} - w \mathbf{a}_{\tilde{\theta}}^{(d-1)} \mathbf{e}_1^T - \bar{w} J \bar{\mathbf{a}}_{\tilde{\theta}}^{(d-1)} \mathbf{e}_n^T, \quad (2.1)$$

Keywords: Toeplitz matrices, Fast Toeplitz solver, Levinson Algorithm.

AMS Mathematical Subject Classification [2010]: 65F05, 15B05, 65F15.

where

$$\mathbf{a}_{\tilde{\theta}}^{(d-1)} = Z^T T_{\tilde{\theta},n}^{(d-1)}(1:n, 1),$$

and Z is the down shift matrix $z_{ij} = \delta_{i-j,1}$ ($\delta_{i,j}$ is the Kronecker delta).

Let $w = e^{i\tilde{\theta}}$ and define the unitary diagonal matrix $\Omega = \text{diag}(1, w, w^2, \dots, w^{n-1})$, then $T_{\tilde{\theta},n}^{(1)} = \Omega^T T_n^{(1)} \Omega$. Hence if we let $\tilde{\theta} = 0$ in (2.1) we have

$$T_n^{(d)} = T_n^{(d-1)} T_n^{(1)} - \mathbf{a}^{(d-1)} \mathbf{e}_1^T - J \mathbf{a}^{(d-1)} \mathbf{e}_n^T, \quad (2.2)$$

where $\mathbf{a}^{(d-1)} = Z^T T_n^{(d-1)}(1:n, 1)$. We define

$$H_n^{(d-1)} = T_n^{-(d-1)} T_n^{(d)} \quad (2.3)$$

$$= T_n^{(1)} - w \mathbf{c}^{(d-1)} \mathbf{e}_1^T - \bar{w} J \mathbf{c}^{(d-1)} \mathbf{e}_n^T \quad (2.4)$$

$$= \begin{pmatrix} -c_1 - 2 & -\mathbf{e}_1^T & -c_n \\ \mathbf{d} & T_{n-2}^{(1)} & J\mathbf{d} \\ -c_{n-1} & -\mathbf{e}_{n-2}^T & -c_1 - 2 \end{pmatrix}, \quad (2.5)$$

where $T_n^{(d-1)} \mathbf{c}^{(d-1)} = \mathbf{a}^{(d-1)}$ and $\mathbf{c}^{(d-1)} = (c_1 \quad \mathbf{d} \quad c_n)^T$. Relation (2.1), is the core of our algorithm to solve the linear system $T_n^{(d)} x^{(d)} = b$. In fact we can define the following recursive linear systems to solve $T_n^{(d)} x = b$.

$$T_n^d x^{(d)} = b \Rightarrow T_n^{-(d-1)} T_n^d x^{(d)} = T_n^{-(d-1)} b = x^{(d-1)} \quad (2.6)$$

$$H_n^{(d-1)} x^{(d)} = x^{(d-1)} \Rightarrow \dots \quad (2.7)$$

$$H_n^{(1)} x^{(d)} = x^{(1)} \quad (2.8)$$

To construct the linear system $H_n^{(d-1)} x^{(d)} = x^{(d-1)}$ and then solving it, we need to compute $T_n^{(d-1)} x^{(d-1)} = b$, and $T_n^{(d-1)} \mathbf{c}^{(d-1)} = \mathbf{a}^{(d-1)}$. Again we can compute them recursively. Lastly we need to solve $T_n^{(1)} \mathbf{c}^{(1)} = \mathbf{a}^{(1)}$ and $T_n^{(1)} x^{(1)} = b$. As we know, $T_n^{(1)} = \text{tridiag}(-1, 2, -1)$. By simple calculation we can see that, the regular Gaussian elimination for this matrix can be performed as follows.

The complexity of Algorithm 1 is $O(n)$, and for all $k > 1$, $1 < \mathbf{a}(k) < 2$, so this algorithm is stable. The procedure to compute $T_n^{(d)} \mathbf{x}^{(d)} = \mathbf{b}$ is described in Algorithm 2.

The matrix $H_n^{(d-1)} = T_n^{(1)} - \mathbf{c}^{(d-1)} \mathbf{e}_1^T - J \mathbf{c}^{(d-1)} \mathbf{e}_n^T$ coincides with the matrix $T_n^{(1)}$ except the first and last column. If we define the permutation matrix E_n as

$$E_n = \begin{pmatrix} 0_{n-2} & I_{n-2} & 0_{n-2} \\ 1 & 0_{n-1}^T & 0 \\ 0 & 0_{n-1}^T & 1 \end{pmatrix}, \quad (2.9)$$

then $\tilde{H}_n = E_n H_n E_n^T$ has the following structure

$$\tilde{H}_n = \begin{pmatrix} T_{n-2}^{(1)} & \mathcal{R} \\ \mathcal{S} & \mathcal{T} \end{pmatrix}, \quad (2.10)$$

where $\mathcal{R} = (\mathbf{d} \quad J\mathbf{d})$, $\mathcal{S} = (-\mathbf{e}_1 \quad -\mathbf{e}_{n-2})^T$, and

$$\mathcal{T} = \begin{pmatrix} -b_1 - 2 & -b_n \\ -b_n & -b_1 - 2 \end{pmatrix}. \quad (2.11)$$

Hence, we can solve the linear system $H_n^{(d-1)} x^{(d-1)} \mathbf{x}^{(d)} = \mathbf{x}^{(d-1)}$ in line 9 Algorithm 2 by the Gaussian elimination (Algorithm 1.) in $O(n)$.

Algorithm 1 Gaussian elimination algorithm to solve $T_n^{(1)} \mathbf{x} = \mathbf{b}$

```

1 Set  $\mathbf{a}(1) = 2$ .
2 For  $k = 2 : n$ 
3    $\mathbf{a}(k) = 2 - \frac{1}{\mathbf{a}(k-1)}$ 
4    $\mathbf{b}(k) = \mathbf{b}(k) - \mathbf{a}(k-1) * \mathbf{b}(k-1)$ 
5 EndFor
6  $\mathbf{x}(n) = \mathbf{b}(n)/\mathbf{a}(n)$ .
7 For  $(k = n-1 : -1 : 1)$ 
8    $\mathbf{x}(k) = (\mathbf{b}(k) + \mathbf{x}(k+1))/\mathbf{a}(k)$ 
9 EndFor

```

Algorithm 2 Recursive procedure to solve $T_n^{(d)} \mathbf{x}^d = \mathbf{b}$

```

1: procedure  $\mathbf{x}^{(d)} \leftarrow \text{Recursive}(\mathbf{b}, d)$ 
2:   If  $(d == 1)$ 
3:     return  $\mathbf{x}^{(1)} = T_n^{-(1)} \mathbf{b}$  (by Algorithm 1.)
4:   else
5:     Define  $\mathbf{a}^{(d-1)} = Z^T T_n^{(d-1)} e_1$ 
6:     Compute  $\mathbf{x}^{(d-1)} = \text{Recursive}(\mathbf{b}, d-1)$ 
7:     Compute  $\mathbf{c}^{(d-1)} = \text{Recursive}(\mathbf{a}^{(d-1)}, d-1)$ 
8:     Define  $H_n^{(d-1)}$  as defined in (2.4)
9:     Compute  $\mathbf{x}^{(d)} = H_n^{-(d-1)} \mathbf{x}^{(d-1)}$ 
10:    return  $\mathbf{x}^{(d)}$ .
8:   end if
9: end procedure

```

Table 1: Comparison the results of GBD and GLev Algorithm for $T_n^{(2)}$ and $T_n^{(4)}$

n	Results for $T_n^{(2)}$				Results for $T_n^{(4)}$			
	GBD		Glev		GBD		Glev	
	$\ \mathbf{r}_n\ $	CPU	$\ \mathbf{r}_n\ $	CPU	$\ \mathbf{r}_n\ $	CPU	$\ \mathbf{r}_n\ $	CPU
2^9	1.32×10^{-12}	0.0013	4.68×10^{-12}	0.0101	2.86×10^{-10}	0.0053	9.42×10^{-10}	0.0112
2^{10}	1.59×10^{-11}	0.0018	2.02×10^{-11}	0.0221	7.34×10^{-9}	0.0038	6.23×10^{-9}	0.0342
2^{11}	6.56×10^{-11}	0.0054	7.80×10^{-11}	0.0494	5.96×10^{-8}	0.0066	6.85×10^{-8}	0.0513
2^{12}	2.48×10^{-10}	0.0152	3.14×10^{-10}	0.1452	4.96×10^{-7}	0.0161	4.68×10^{-4}	0.1501
2^{13}	1.02×10^{-9}	0.0325	1.25×10^{-9}	0.7443	4.02×10^{-6}	0.0710	5.17×10^{-4}	0.7321
2^{14}	4.04×10^{-9}	0.1253	5.11×10^{-9}	2.8594	3.20×10^{-5}	0.2238	6.21×10^{-4}	3.001
2^{15}	1.60×10^{-8}	0.4203	1.70×10^{-8}	11.3476	2.57×10^{-4}	0.7596	0.0084	11.4521
2^{16}	6.3628×10^{-8}	1.3627	4.2896×10^{-8}	53.8154	0.0021	2.8229	0.0338	54.9231

3. Numerical Experiments

In this section, we perform some numerical examples to show the efficiency of the proposed method. All tests were carried out in double precision with a MATLAB code. In this section, we compare our method GBD (Gaussian elimination for banded Toeplitz matrices) with GLev (general Levinson algorithm) [2].

References

- [1] M. Barrera, A. Böttcher, S. M. Grudsky, E. A. Maximenko, Eigenvalues of even very nice Toeplitz matrices can be unexpectedly erratic, *In The diversity and beauty of applied operator theory*, Birkhäuser, 2018, (pp. 51-77).
- [2] P. Favati, G. Lotti, O. Menchi, . Stability of the Levinson algorithm for Toeplitz-like systems, *SIAM journal on matrix analysis and applications*, 31 (2010) 2531–2552.
- [3] S. Hon, S. Serra-Capizzano, A. Wathen, Band-Toeplitz preconditioners for ill-conditioned Toeplitz systems, *BIT Numerical Mathematics*, 62 (2022) 465–491.
- [4] N. Akhoundi, direct method to solve banded ill-conditioned Toeplitz matrices, submitted

Nasser Akhoundi

Department of Mathematics, Faculty of Mathematics and Computer Science, University Damghan, Damghan, Iran

email address: akhoundi@du.ac.ir



University of Guilan



Perturbed Simpson-type inequality via h -convex functions

Ali Barani, Naser Abbasi

Abstract. In this paper the celebrated Simpson-type inequality for functions whose n -th derivatives in absolute value are h -convex is introduced.

1. Introduction

One of the most important results in numerical analysis for approximate integrals is Simpson's inequality as follows

$$\left| \int_a^b f(x)dx - \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \right| \leq \frac{1}{2880} \|f^{(4)}\|_{\infty} (b-a)^5, \quad (1.1)$$

where the function $f : [a, b] \rightarrow \mathbb{R}$ is supposed to be four times continuously differentiable on (a, b) and

$$\|f^{(4)}\|_{\infty} := \sup_{x \in (a, b)} |f^{(4)}(x)| < \infty,$$

see [7]. Then several generalization of (1.1) appeared in the literature by imposing the convexity and generalized convexity conditions on $|f^{(n)}|$ for $n = 1, 2, 3, \dots$ (see for example [1, 3]). The perturbed Simpson's formula investigated by Liu in [4] as follows.

Theorem 1.1. Let $f : [a, b] \rightarrow \mathbb{R}$ be n -times continuously differentiable on (a, b) . Then

$$\begin{aligned} (-1)^n \int_a^b S_n(x) f^{(n)}(x) dx &= \int_a^b f(x) dx - \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \\ &+ \sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} \frac{(i-1)(b-a)^{2i+1}}{3(2i+1)!2^{2i-1}} f^{(2i)}\left(\frac{a+b}{2}\right), \end{aligned} \quad (1.2)$$

where $\lfloor \frac{n-1}{2} \rfloor$ is the integer part of $\frac{n-1}{2}$ and

$$S_n(x) := \begin{cases} \frac{(x-a)^n}{n!} - \frac{(b-a)(x-a)^{n-1}}{6(n-1)!} & , x \in [a, \frac{a+b}{2}], \\ \frac{(x-b)^n}{n!} - \frac{(b-a)(x-b)^{n-1}}{6(n-1)!} & , x \in (\frac{a+b}{2}, b]. \end{cases}$$

Then several applications and improvements appeared in literature in [5, 6, 9] and references therein. Note that for $n = 1, 2, 3, 4$ there are no perturbation terms so the perturbed Simpson's inequality is obtained from Theorem 1.1 as follows.

Keywords: Simpson-type inequality, h -convex function, s -convex function.

AMS Mathematical Subject Classification [2010]: 26D15, 26A51.

Theorem 1.2. Let $f : [a, b] \rightarrow \mathbb{R}$ be n -times continuously differentiable on (a, b) such that $n > 4$ and $\|f^{(n)}\|_\infty := \sup_{x \in (a, b)} |f^{(n)}(x)| < \infty$. Then

$$\left| \int_a^b f(x) dx - \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] + \sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} \frac{(i-1)(b-a)^{2i+1}}{3(2i+1)!2^{2i-1}} f^{(2i)}\left(\frac{a+b}{2}\right) \right| \leq \|f^{(n)}\|_\infty \frac{(n-2)(b-a)^{n+1}}{3(n+1)!2^n}. \quad (1.3)$$

On the other hand the class of h -convex functions was introduced by S. Varosanec in [2] which contains the some important subclasses for example convex, quasi convex, P -convex and s -convex functions.

Definition 1.3. The function $f : I \subset \mathbb{R} \rightarrow \mathbb{R}$ is said to be h -convex for a function $h : J \rightarrow (0, +\infty)$ with $(0, 1) \subseteq J$, if for every $x, y \in I$ and $t \in (0, 1)$ one has

$$f((1-t)x + ty) \leq h(1-t)f(x) + h(t)f(y).$$

Then several authors investigated more generalizations and applications on this notion, see for example [8]. Motivated by above results in this paper we obtain several Simpson-type inequalities for functions whose n -th derivatives in absolute value are h -convex.

2. Main results

To proceed we set

$$\sigma(a, b) := f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) + \sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} \frac{(i-1)(b-a)^{2i+1}}{3(2i+1)!2^{2i-1}} f^{(2i)}\left(\frac{a+b}{2}\right).$$

The following lemma is a consequence of Theorem 1.1.

Lemma 2.1. Let $f : [a, b] \rightarrow \mathbb{R}$ be n -times continuously differentiable on (a, b) . Then

$$(-1)^n \int_0^1 \gamma_n(t) f^{(n)}((1-t)a + tb) dt = \int_a^b f(x) dx - \sigma(a, b), \quad (2.1)$$

where

$$\gamma_n(t) := \begin{cases} \frac{(b-a)^{n+1}}{n!} \left(t^n - \frac{nt^{n-1}}{6} \right) & , t \in [0, \frac{1}{2}], \\ \frac{(-1)^n (b-a)^{n+1}}{n!} \left((1-t)^n - \frac{n(1-t)^{n-1}}{6} \right) & , t \in (\frac{1}{2}, 1]. \end{cases}$$

Now, we obtain a version of Simpson's type inequality for h -convex functions.

Theorem 2.2. Let $f : [a, b] \rightarrow \mathbb{R}$ be n -times continuously differentiable on (a, b) such that $n > 4$. Suppose that $f^{(n)} \in L[a, b]$ and $|f^{(n)}|$ is a h -convex function on $[a, b]$. Then

$$\left| \int_a^b f(x) dx - \sigma(a, b) \right| \leq \frac{(b-a)^{n+1}}{n!} \left[\int_0^{\frac{1}{2}} \left(\frac{n}{2} t^{n-1} - t^n \right) (h(1-t) + h(t)) dt \right] (|f^{(n)}(a)| + |f^{(n)}(b)|). \quad (2.2)$$

In Theorem 2.2, if $h(t) = t$, for every $t \in [0, 1]$ then we get a version of Simpson's type inequality related to convex functions.

Theorem 2.3. Let $f : [a, b] \rightarrow \mathbb{R}$ be n -times continuously differentiable on (a, b) such that $n > 4$. Suppose that $f^{(n)} \in L[a, b]$ and $|f^{(n)}|$ is a convex function on $[a, b]$. Then

$$\left| \int_a^b f(x) dx - \sigma(a, b) \right| \leq \frac{[(n-2)(|f^{(n)}(a)| + |f^{(n)}(b)|)]}{3(n+1)!2^{n+1}} (b-a)^{n+1}. \quad (2.3)$$

For quasi convex case we derive the next theorem from Theorem 2.2, which investigate in [5, p. 49].

Theorem 2.4. Let $f : [a, b] \rightarrow \mathbb{R}$ be n -times continuously differentiable on (a, b) such that $n > 4$. Suppose that $f^{(n)} \in L[a, b]$ and $|f^{(n)}|$ is a quasiconvex function on $[a, b]$. Then

$$\left| \int_a^b f(x) dx - \sigma(a, b) \right| \leq \frac{(n-2) \max \{|f^{(n)}(a)|, |f^{(n)}(b)|\}}{3(n+1)!2^n} (b-a)^{n+1}. \quad (2.4)$$

Pick $h(t) = t^s$, for every $t \in [0, 1]$ and some $s \in (0, 1)$ in Theorem 2.2 then we deduce the next result.

Theorem 2.5. Let $f : [a, b] \rightarrow [0, +\infty)$ be n -times continuously differentiable on (a, b) such that $n > 4$. Suppose that $f^{(n)} \in L[a, b]$ and $|f^{(n)}|$ is a s -convex function on $[a, b]$, for some $f \in (0, 1)$. Then we have

$$\begin{aligned} & \left| \int_a^b f(x) dx - \sigma(a, b) \right| \\ & \leq \left[I(n, s) + \frac{n^2 + ns - s}{(n+s)(n+s+1)2^{n+s+1}} \right] \frac{(|f^{(n)}(a)| + |f^{(n)}(b)|) (b-a)^{n+1}}{n!}, \end{aligned} \quad (2.5)$$

where $I(n, s) := \int_0^{\frac{1}{2}} (\frac{n}{2}t^{n-1} - t^n)(1-t)^s dt$.

References

- [1] M. Alomari, S. Hussain, Two inequalities of Simpson type for quasiconvex functions and applications, *Applied Mathematics E-Notes*, 11 (2011) 110–117.
- [2] S.S. Dragomir, Inequalities of Hermite-Hadamard type for h -convex functions on linear spaces, *Proyecciones Journal of Mathematics*, 34 (2015) 323–341.
- [3] I. Franjic, J. Pecaric, Schur-convexity and the Simpson formula, *Applied Mathematics Letters*, 24 (2011) 1565–1568.
- [4] Z. Liu, An inequality of Simpson type, *Proceedings of the Royal Society of London Ser. A*, 461 (2005) 2155–2158.
- [5] Z. Liu, On generalizations of some classical integral inequalities, *Journal of Mathematica Inequalities*, 7 (2013) 255–269.
- [6] Z. Liu, On inequalities related to Simpson quasiconvex functions, *Problemy Analiza-Issues of Analysis*, 4 (2015) 45–64.

- [7] V.I. Krylov, Approximate calculation of integrals, *The Macmillan Company*, New York, 1962.
- [8] A. Olbrys, Representation theorems for h -convexity, *Journal of Mathematical Analysis and Applications*, 426 (2015) 986-994.
- [9] Y. Shi, Z. Liu, Some sharp Simpson type inequalities and applications, *Applied Mathematics E-Notes*, 9 (2009) 205-215.
- [10] S. Varosanec, On h -convexity, *Journal of Mathematical Analysis and Applications*, 326 (2007) 303-311.

Ali Barani

Department of Mathematics, Lorestan University, Faculty of Science, Lorestan University, Khoramabad, Iran
email address: barani.a@lu.ac.ir

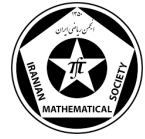
Naser Abbasi

Department of Mathematics, Lorestan University, Faculty of Science, Lorestan University, Khoramabad, Iran
email address: abasi.n@lu.ac.ir





University of Guilan



The irreducible representation of general linear methods

Rana Akbari, Gholamreza Hojjati, Ali Abdi

Abstract. In this paper, we discuss equivalence and reducibility consents of the general linear methods (GLMs) for ordinary differential equations which are useful in studding the stability properties of the methods. Also, we give a representation of linear multistep methods as irreducible GLMs. Some examples of reducible methods and their equivalents along with some irreducible GLMs are examined.

1. Introduction

A general linear method (GLM) of order p and stage order q for the numerical solution of the autonomous initial value problem (IVP)

$$\begin{aligned} y'(x) &= f(y(x)), & x \in [x_0, \bar{x}], \\ y(x_0) &= y_0, \end{aligned} \quad (1.1)$$

utilizes r input and output values, and s stage values. The vectors $y^{[n-1]} = [y_i^{[n-1]}]_{i=1}^r$ and $y^{[n]} = [y_i^{[n]}]_{i=1}^r$ are respectively collection of the quantities imported at the beginning of step number n and the quantities exported at the end of this step. Also, $Y^{[n]} = [Y_i^{[n]}]_{i=1}^s$ is an approximation of stage order q to the vector $y(x_{n-1} + ch) = [y(x_{n-1} + c_i h)]_{i=1}^s$. A GLM used to solve (1.1) takes the form [1]

$$\begin{aligned} Y^{[n]} &= h(A \otimes I_m)f(Y^{[n]}) + (U \otimes I_m)y^{[n-1]}, \\ y^{[n]} &= h(B \otimes I_m)f(Y^{[n]}) + (V \otimes I_m)y^{[n-1]}. \end{aligned} \quad (1.2)$$

where A , U , B and V are the coefficients matrices of the method. This formulation might be reducible. Hence, Butcher and Hill discovered [2] a compact representation of linear multistep methods (LMMs) with $r = k$ inputs and $s = 1$ stage. The main focus of this paper is on the irreducible representation of GLMs. We define equivalence and reducibility concepts of GLMs. Also, an irreducible GLM representation is presented for linear multistep methods. Our last step will be to look at examples of GLMs and examine their irreducibility.

2. Reducibility of GLMs

It is possible for two numerical methods to have somewhat different appearances, but when applied to an IVP, they ultimately give the same answer. It allows us to define the equivalence between the two GLMs.

Keywords: General linear methods, Linear multistep methods, Equivalence, Reducibility.

AMS Mathematical Subject Classification [2010]: 65L05.

Definition 2.1. Two GLMs are equivalent if they yield the same numerical solution for all initial value problems, for a small enough stepsize h .

Identifying whether two GLMs are equivalent requires discussion of the concept of reducibility. If a GLM can be partitioned in such a way that $s = s_1 + s_2$ and $r = r_1 + r_2 + r_3$ with $s_2 + r_2 + r_3 > 0$, so that it has the following sparsity pattern

$$\left[\begin{array}{cc|ccc} A_{11} & 0 & U_{11} & 0 & U_{13} \\ A_{21} & A_{22} & U_{21} & U_{22} & 0 \\ \hline B_{11} & 0 & V_{11} & 0 & V_{13} \\ B_{21} & B_{22} & V_{21} & V_{22} & V_{23} \\ 0 & 0 & 0 & 0 & V_{33} \end{array} \right], \quad (2.1)$$

then we say that the method is reducible; otherwise the method is said to be irreducible [3, 4].

3. The LMM as an irreducible GLM

The k -step LMMs with the stepsize $h > 0$ may be written as

$$y_n = \sum_{j=1}^k \alpha_j y_{n-j} + h \sum_{j=0}^k \beta_j f(y_{n-j}). \quad (3.1)$$

These methods can be expressed as GLMs with $s = 1$, $r = 2k$ and the coefficients matrices

$$\left[\begin{array}{c|ccccccccc} \beta_0 & \alpha_1 & \cdots & \alpha_{k-1} & \alpha_k & \beta_1 & \cdots & \beta_{k-1} & \beta_k \\ \beta_0 & \alpha_1 & \cdots & \alpha_{k-1} & \alpha_k & \beta_1 & \cdots & \beta_{k-1} & \beta_k \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \vdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \vdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \vdots & 1 & 0 \end{array} \right] = \left[\begin{array}{c|c} A & U \\ \hline B & V \end{array} \right]. \quad (3.2)$$

Taking $s_1 = 1$, $s_2 = 0$ and $r_1 = k$, $r_2 = 0$, $r_3 = k$ as an assumption, this representation is reducible. Special cases of LMMs may be represented in a more compact way. Butcher and Hill [2] discovered a compact representation of LMMs with $r = k$ inputs and $s = 1$ stage. Defining $y^{[n-1]}$ with only k inputs as

$$y_i^{[n-1]} = \sum_{j=k-i+1}^k (\alpha_j y_{n+k-i-j} + h \beta_j f(y_{n+k-i-j})), \quad (3.3)$$

the LMMs can be written in the form

$$\begin{aligned} y_n &= h \beta_0 f(y_n) + \sum_{j=1}^k (\alpha_j y_{n-j} + h \beta_j f(y_{n-j})) = h \beta_0 f(y_n) + y_k^{[n-1]}, \\ y_i^{[n]} &= \sum_{j=k-i+1}^k (\alpha_j y_{n+1+k-i-j} + h \beta_j f(y_{n+1+k-i-j})) = \alpha_{k-i+1} y_n + h \beta_{k-i+1} f(y_n) \\ &\quad + \sum_{j=k-i+2}^k (\alpha_j y_{n+1+k-i-j} + h \beta_j f(y_{n+1+k-i-j})) = \alpha_{k-i+1} \beta_0 + \beta_{k-i+1} h f(y_n) + \alpha_{k-i+1} y_k^{[n-1]} + y_{i-1}^{[n-1]}. \end{aligned}$$

Hence, we have representation

$$\left[\begin{array}{c|ccc} \beta_0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ \alpha_k \beta_0 + \beta_k & 0 & 0 & 0 & \cdots & 0 & \alpha_k \\ \alpha_{k-1} \beta_0 + \beta_{k-1} & 1 & 0 & 0 & \cdots & 0 & \alpha_{k-1} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \alpha_2 \beta_0 + \beta_2 & 0 & 0 & 0 & \ddots & 0 & \alpha_2 \\ \alpha_1 \beta_0 + \beta_1 & 0 & 0 & 0 & \cdots & 1 & \alpha_1 \end{array} \right] = \quad (3.4)$$

4. Examples

Example 4.1. The GLM form of the modified extended backward differentiation formulas (MEBDFs) [5] can be written with the coefficients matrices

$$A = \begin{bmatrix} \hat{\beta}_k & 0 & 0 \\ -\hat{\alpha}_{k-1} \hat{\beta}_k & \hat{\beta}_k & 0 \\ \beta_k - \hat{\beta}_{k-1} & \beta_{k+1} & \hat{\beta}_k \end{bmatrix},$$

$$U = \begin{bmatrix} -\hat{\alpha}_{k-1} & -\hat{\alpha}_{k-2} & \cdots & -\hat{\alpha}_1 & -\hat{\alpha}_0 \\ \hat{\alpha}_{k-1} \hat{\alpha}_{k-1} - \hat{\alpha}_{k-2} & \hat{\alpha}_{k-1} \hat{\alpha}_{k-2} - \hat{\alpha}_{k-3} & \cdots & \hat{\alpha}_{k-1} \hat{\alpha}_1 - \hat{\alpha}_0 & \hat{\alpha}_{k-1} \hat{\alpha}_0 \\ -\hat{\alpha}_{k-1} & -\hat{\alpha}_{k-2} & \cdots & -\hat{\alpha}_1 & -\hat{\alpha}_0 \end{bmatrix},$$

$$B = \begin{bmatrix} \beta_k - \hat{\beta}_{k-1} & \beta_{k+1} & \hat{\beta}_k \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, V = \begin{bmatrix} -\hat{\alpha}_{k-1} & -\hat{\alpha}_{k-2} & \cdots & -\hat{\alpha}_1 & -\hat{\alpha}_0 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

Since this method cannot be interpreted with the sparsity pattern (2.1), $s_2 = r_2 = r_3 = 0$, so it is irreducible.

Example 4.2. The 2-step LMM given by

$$\left[\begin{array}{c|cc} A & U \\ B & V \end{array} \right] = \left[\begin{array}{c|cc} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{array} \right],$$

is reducible with the assumption of $r_2 = 1$ and $s_2 = r_3 = 0$, and it corresponds to backward Euler method

$$\left[\begin{array}{c|c} 1 & 1 \\ 1 & 1 \end{array} \right].$$

References

- [1] K. Burrage, J.C. Butcher, Non-linear stability for a general class of differential equation methods, *BIT*, 20 (1980) 185–203.
- [2] J.C. Butcher, A.T. Hill, Linear multistep methods as irreducible general linear methods, *BIT*, 46 (2006) 5–19.

- [3] J.C. Butcher, The equivalence of algebraic stability and AN-stability, *BIT*, 27 (1987) 510–534.
- [4] L.L. Hewitt, A.T. Hill, Algebraically stable general linear methods and the G-matrix, *BIT*, 49 (2009) 93–111.
- [5] Z. Jackiewicz, *General Linear Methods for Ordinary Differential Equations*, Wiley, New Jersey, 2009.

Rana Akbari

Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

email address: r.akbari@tabrizu.ac.ir

Gholamreza Hojjati

Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

email address: ghohjjati@tabrizu.ac.ir

Ali Abdi

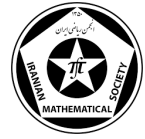
Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

email address: a_abdi@tabrizu.ac.ir





University of Guilan



A modification of Grünwald-Letnikov derivative in image processing

Hoda Jalalinejad, Ali Tavakoli

Abstract. In order to better show the rate of changes of the derivative in image processing, we need to redefine the Grünwald-Letnikov derivative. In this paper, we highlight the problems of Grünwald-Letnikov derivative in image processing and based on, we present a new definition of Grünwald-Letnikov derivative to improve these problems.

1. Introduction

The fractional differential equation had been studied over the last 300 years. Recently, the fractional differential equation has used for image and signal processing. In 2003, Mathieu et al. applied the fractional differentiation for edge detection [5]. Gao et al. in [2] applied an improved fractional differential operator based on a piecewise quaternion for image enhancement. Furthermore, in [4], the generalized fractional image denoising algorithm based on Srivastava-owa fractional differential operator is introduced for image denoising. The Grünwald-Letnikov derivative is used for image enhancement in [6]. In [1] Gao et al. by development of the real fractional differential and its applications in the signal processing, extended the quaternion fractional differential (QFD), based on Grünwald-Letnikov and apply it to edge detection of colour image. In this paper, we investigate some problems of Grünwald-Letnikov derivative in image processing and then, improve them with a new definition.

2. Preliminaries

In this section, we introduce the Grünwald-Letnikov derivative and discuss why we want to modify this definition.

Definition 2.1. The Grünwald-Letnikov derivative for one variable function f is defined as follows:

$$D_{G-L}^{\alpha} f(x) = \lim_{h \rightarrow 0} \frac{1}{h^{\alpha}} \sum_{r=0}^{[t-a]} (-1)^r \binom{\alpha}{r} f(x - rh),$$

where

$$\binom{\alpha}{r} = \frac{\Gamma(\alpha + 1)}{\Gamma(r + 1)\Gamma(\alpha - r + 1)},$$

and Γ is the Gamma function and a is a real constant.

Keywords: Grünwald-Letnikov derivative, image processing, edge detection..

AMS Mathematical Subject Classification [2010]: 97M10, 1072.

In image processing, the value of $f(x, y)$ is called the color intensity of image at point (x, y) where x and y are spatial coordinates. Therefore, the Grünwald-Letnikov derivative in two dimension in x -direction can be defined as follow [3, 6]:

$$D_{G-L}^{\alpha} f_x(x, y) = [f(x, y) - \alpha f(x-1, y) + \frac{\alpha(\alpha-1)f(x-2, y)}{2}]. \quad (2.1)$$

Similarly, the Grünwald-Letnikov derivative is defined in y -direction. Hence, the Grünwald-Letnikov fractional derivative is defined by:

$$D_{G-L}^{\alpha} f(x, y) = \sqrt{(D_{G-L}^{\alpha} f_x(x, y))^2 + (D_{G-L}^{\alpha} f_y(x, y))^2}. \quad (2.2)$$

To summarize, we present two examples to show that it is better to modify the definition of Grünwald-Letnikov derivative.

Example 2.2. Let $f(x-1, y) = f(x-2, y) = f(x, y) = 250$. By (2.1) we get

$$D_{G-L}^{\alpha} f_x(x, y) = 250 - \alpha 250 + \frac{\alpha(\alpha-1)}{2} 250 = (1-\alpha)(2-\alpha)125,$$

that implies

$$0 < D_{G-L}^{\alpha} f_x(x, y) < 250.$$

In the special case $\alpha = 1/2$, we have $D_{G-L}^{\alpha} f_x(x, y) = 93.75$.

Example 2.3. Let $f(x-2, y) = f(x-1, y) = f(x, y) = 1$. We have

$$D_{G-L}^{\alpha} f_x(x, y) = 1 - 1\alpha + \frac{\alpha(\alpha-1)}{2} = (1-\alpha)(2-\alpha)/2.$$

Again, for $0 < \alpha \leq 1$, we have

$$0 \leq D_{G-L}^{\alpha} f_x(x, y) < 1.$$

In this examples, since the value of f is constant in x -direction, we have expect no change or a few change in the derivative of f in x -direction. However, we see the value of $D_{G-L}^{\alpha} f(x, y)$ severely depends on the intensity of f compared to the difference of f and their x -neighbourhoods.

Therefore, in order to better represent the rate of changes of the derivative, we modify the definition of Grünwald-Letnikov derivative.

3. Main result

In this section, we express a modified definition of Grünwald-Letnikov derivative. By considering

$$X(x, y) = |f(x, y) - \alpha f(x-1, y) + \frac{\alpha(\alpha-1)}{2} f(x-2, y)|,$$

and

$$M(x, y) = \frac{1}{s^n} \min\{f(x, y), f(x-1, y), f(x-2, y)\},$$

where $s \geq 255$ is an integer number and $0 \leq n \leq 1$ is a real number, the equation of the line passing through of two points $(0, M(x, y))$ and $(s, 0)$ is

$$Y(x, y) = M(x, y) \left(\frac{s - X(x, y)}{s} \right).$$

Now, we define the modified Grünwald-Letnikov derivative in x-direction as follow:

$${}_m D_{G-L}^\alpha f_x(x, y) = \frac{f(x, y) - \alpha f(x-1, y) + \frac{\alpha(\alpha-1)}{2} f(x-2, y)}{Y(x, y) + 1}, \quad (3.1)$$

that the value 1 is added to $Y(x, y)$ to avoid of vanishing the denominator. By (3.1), we get

$${}_m D_{G-L}^\alpha f_x(x, y) = \frac{s^{n+1} A}{\theta(s - |A|) + s^{n+1}},$$

where $\theta = \min\{f(x, y), f(x-1, y), f(x-2, y)\}$ and $A = D_{G-L}^\alpha f_x(x, y)$.

By definition the modified Grünwald-Letnikov derivative in y-direction, the modified Grünwald-Letnikov fractional derivative can be defined by

$${}_m D_{G-L}^\alpha f(x, y) = \sqrt{({}_m D_{G-L}^\alpha f_x(x, y))^2 + ({}_m D_{G-L}^\alpha f_y(x, y))^2}. \quad (3.2)$$

Now, we compute the modified Grünwald-Letnikov derivative for the preceding examples.

For Examples Example 2.2, we have

$$0 \leq {}_m D_{G-L}^\alpha f_x(x, y) < \frac{250s^{n+1}}{250(s-250) + s^{n+1}},$$

in which $0 < \alpha \leq 1$. The special case $\alpha = 1/2$, $s = 255$ and $n = 1$ yields ${}_m D_{G-L}^\alpha f_x(x, y) = 57.8725$.

For Example 2.3,

$$0 \leq {}_m D_{G-L}^\alpha f_x(x, y) < \frac{s^{n+1}}{(s-1) + s^{n+1}} < 1,$$

in which $0 < \alpha \leq 1$.

4. Numerical results

In this section, we show that the modified Grünwald-Letnikov fractional derivative can be efficiently applied for edge detection.

Example 4.1. (Edge detection) Consider Figure 1(a) as an original image. Figure 1(b) shows its Grünwald-Letnikov derivative defined by (2.2) and Figure 1(c) shows its modified Grünwald-Letnikov derivative defined by (3.2). In both Figures 1(b,c), we put $\alpha = 0.5$. Also, for modified Grünwald-Letnikov derivative, $s = 255$ and $n = 0.5$ selected. As it is seen the modified Grünwald-Letnikov derivative shows only the edges of the main figure while Grünwald-Letnikov derivative shows the whole of figure with low intensity.

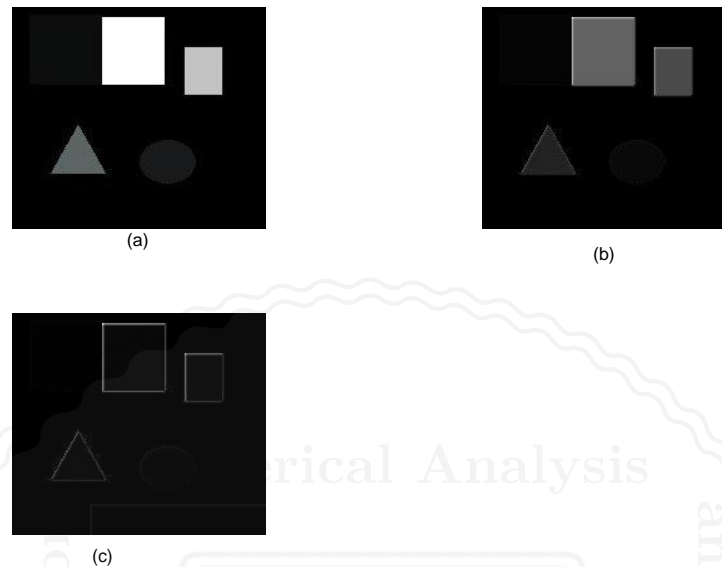


Figure 1: An original image (a), its Grünwald-Letnikov derivative (b) and its modified Grünwald-Letnikov derivative (c), corresponding to Example 4.1.

References

- [1] C.B. Gao, J.L. Zhou, J.R. Hu, F.N. Lang, Edge detection of color image based on quaternion fractional differential, *IET Image Processing*, 5 (2011) 261–272.
- [2] C.B. Gao, J.L. Zhou, X.Q. Zheng, F.N. Lang, Image enhancement based on improved fractional differentiation, *Journal of Computational Information Systems*, 7 (2011) 257–264.
- [3] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, 3rd ed., Prentice Hall, New Jersey, 2007.
- [4] H.A. Jalab, R.W. Ibrahim, Fractional masks based on generalized fractional differential operator for image denoising, *World Academy of Science, Engineering and Technology International Journal of Computer, Information Science and Engineering*, 7 (2013) 124–129.
- [5] B. Mathieu, P. Melchior, A. Oustaloup, Ch. Ceyral, Fractional Differentiation for Edge Detection, *Signal Process.*, 83 (2003) 2421–2432.
- [6] Y.F. Pu, J.L. Zhou, X. Yuan, Fractional Differential Mask: A Fractional Differential-Based Approach for Multiscale Texture Enhancement, *IEEE Transactions on Image Processing*, 19 (2010) 491–511.

Hoda Jalalinejad

Department of Mathematics, Bandar Abbas Branch, Islamic Azad University, Bandar Abbas, Iran.

email address: hoda.jalalinezhad@gmail.com

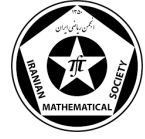
Ali Tavakoli

Mathematics Department, University of Mazandaran, Iran

email address: a.tavakoli@umz.ac.ir



University of Guilan



A spectral collocation method for solving the nonlinear weakly singular FIPDE

Mehran Taghipour, Hossein Aminikhah

Abstract. This work focuses on finding the numerical solution of the nonlinear time-fractional partial integro-differential equation (FIPDE). For this purpose, we use the operational matrices based on Pell polynomials to approximate fractional Caputo derivative, nonlinear, and integro-differential terms, and by collocation points, we transform the problem to a system of nonlinear equations. This nonlinear system can be solved by the `fsolve` command in Matlab.

1. Introduction

In the present work, we propose a numerical scheme to solve the nonlinear time-fractional partial integro-differential equation with a weakly singular kernel

$${}^C\mathcal{D}_t^\alpha u(x, t) + u(x, t)u_x(x, t) = \int_0^t (t-s)^{\beta-1} u_{xx}(x, s) ds + g(x, t), \quad x \in [0, L], \quad t \in [0, T], \quad (1.1)$$

with initial and boundary conditions

$$u(0, t) = \phi_1(t), \quad u(L, t) = \phi_2(t), \quad 0 < t \leq T, \quad (1.2)$$

$$u(x, 0) = \psi(x), \quad 0 < x \leq L, \quad (1.3)$$

where $0 < \alpha, \beta < 1$, $g(x, t) \in C([0, L] \times [0, T])$, and ${}^C\mathcal{D}_0^\alpha$ is the Caputo fractional derivative with respect to t . This problem appears in the modeling of heat transfer materials with memory, population dynamics and nuclear reaction theory.

To the best of the author's knowledge, little work has been done on problem (1.1). For example, Guo et al. [1] proposed a finite difference scheme for solving the problem (1.1)–(1.3). In the case of $\alpha = 1$, Zheng et al. [2] described three semi-implicit compact finite difference schemes for problem (1.1)–(1.3). This stimulates us to propose a numerical method for solving the problem (1.1)–(1.3).

2. Numerical method

We first provide some definitions.

Definition 2.1. The Caputo derivatives of order $0 < \alpha < 1$ of a suitably smooth function $u(x, t)$ on $(0, b) \times (0, T)$ is defined by

$${}^C\mathcal{D}_t^\alpha u(x, t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} u'(x, \tau) d\tau. \quad (2.1)$$

Keywords: Pell polynomials, Spectral collocation method, Caputo fractional derivative.

AMS Mathematical Subject Classification [2010]: 65M70, 65R10, 34K37.

Pell polynomials can be generated by the following recurrence relation:

$$P_{n+2}(x) = 2xP_{n+1}(x) + P_n(x), \quad P_0(x) = 0, \quad P_1(x) = 1. \quad (2.2)$$

Using Pell polynomials, we can approximate a continuous function $u(x)$ as follows:

$$u(x) \approx u_N(x) = \sum_{i=0}^N c_{i+1} P_{i+1}(x) = C^T \mathcal{P}_N(x), \quad (2.3)$$

where

$$C = [c_1, c_2, \dots, c_{N+1}]^T, \quad \mathcal{P}_N(x) = [P_1(x), P_2(x), \dots, P_{N+1}(x)]^T. \quad (2.4)$$

Analogously, we can express a continuous function $u(x, t)$ on $[0, L] \times [0, T]$ as follows:

$$u(x, t) \approx u_{NM}(x, t) = \sum_{i=0}^N \sum_{j=0}^M c_{i+1, j+1} P_{i+1}(x) P_{j+1}(t) = \mathcal{P}_N(x)^T W \mathcal{P}_M(t) = C^T \mathcal{P}_{NM}(x, t), \quad (2.5)$$

where W is a $(N+1) \times (M+1)$ matrix, C and $\mathcal{P}_{NM}(x, t)$ are $(N+1)(M+1) \times 1$ vectors

$$C = [c_{11}, c_{12}, \dots, c_{1M+1}, c_{21}, \dots, c_{2M+1}, \dots, c_{N1}, \dots, c_{N+1M+1}]^T,$$

$$\mathcal{P}_{NM}(x, t) = [P_{11}(x, t), \dots, P_{1M+1}(x, t), \dots, P_{N+11}(x, t), \dots, P_{N+1M+1}(x, t)]^T,$$

and $P_{ij}(x, t) = P_i(x)P_j(t)$ are two variable Pell polynomials.

Also we can rewrite $\mathcal{P}_N(x), \mathcal{P}_M(t)$ in an equivalent forms as

$$P_N(x) = Q_x T(x) \quad (2.6)$$

$$P_M(t) = Q_t T(t), \quad (2.7)$$

where

$$T(x) = [1, x, x^2, \dots, x^N]^T, \quad T(t) = [1, t, t^2, \dots, t^M]^T \quad (2.8)$$

and matrices Q_x and Q_t are $(N+1) \times (n+1), (M+1) \times (n+1)$ have the formulas:

$$Q_x = \begin{pmatrix} q_{0,0} & 0 & 0 & 0 & \cdots & 0 \\ 0 & q_{1,1} & 0 & 0 & \cdots & 0 \\ q_{2,0} & 0 & q_{2,2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{n,1} & 0 & q_{n,3} & \cdots & 0 & q_{n,N} \end{pmatrix}, \quad Q_t = \begin{pmatrix} q_{0,0} & 0 & 0 & 0 & \cdots & 0 \\ 0 & q_{1,1} & 0 & 0 & \cdots & 0 \\ q_{2,0} & 0 & q_{2,2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{n,1} & 0 & q_{n,3} & \cdots & 0 & q_{n,M} \end{pmatrix}$$

with

$$(q_{i,j}) = \begin{cases} \binom{i - \lfloor \frac{i-j}{2} \rfloor}{\lfloor \frac{i-j}{2} \rfloor} 2^{i-2\lfloor \frac{i-j}{2} \rfloor}, & \text{if } i \geq j, i, j = \text{odd or } i, j = \text{even} \\ 0, & \text{otherwise.} \end{cases}$$

We approximate the Caputo fractional derivative as follows:

$$\begin{aligned} {}^C_0 \mathcal{D}_t^\alpha u(x, t) &\approx {}^C_0 \mathcal{D}_t^\alpha u_{NM}(x, t) = {}^C_0 \mathcal{D}_t^\alpha \mathcal{P}_N(x)^T W \mathcal{P}_M(t) = \mathcal{P}_N(x)^T W ({}^C_0 \mathcal{D}_t^\alpha \mathcal{P}_M(t)), \\ &= \mathcal{P}_N(x)^T W Q_t ({}^C_0 \mathcal{D}_t^\alpha T(t)) \\ &= \mathcal{P}_N(x)^T W Q_t \left[0, \frac{\Gamma(2)}{\Gamma(2-\alpha)} t^{1-\alpha}, \frac{\Gamma(3)}{\Gamma(3-\alpha)} t^{2-\alpha}, \dots, \frac{\Gamma(M+1)}{\Gamma(M+1-\alpha)} t^{M-\alpha} \right]^T, \\ &= \mathcal{P}_N(x)^T W Q_t M_\alpha Q_t^{-1} \mathcal{P}_M(t). \end{aligned} \quad (2.9)$$

Next, we approximate the nonlinear and integro-differential terms in equation (1.1).

$$u(x, t)u_x(x, t) \approx u_{NM}(x, t)u_{MNx}(x, t) = \mathcal{P}_N(x)^T W \mathcal{P}_M(t) \mathcal{P}_N(x)^T Q_x^{-T} D^T Q_x^T W \mathcal{P}_M(t). \quad (2.10)$$

For integro-differential term, we have

$$\int_0^t (t-s)^{\beta-1} u_{xx}(x, s) ds \approx \int_0^t (t-s)^{\beta-1} \mathcal{P}_N''(x)^T W \mathcal{P}_M(s) ds \quad (2.11)$$

$$\begin{aligned} &= \int_0^t (t-s)^{\beta-1} (Q_x D'' Q_x^{-1} \mathcal{P}_N(x))^T W \mathcal{P}_M(s) ds \\ &= \mathcal{P}_N(x)^T Q_x^{-T} D''^T Q_x^T W \int_0^t (t-s)^{\beta-1} Q_t T(s) ds \\ &= \mathcal{P}_N(x)^T Q_x^{-T} D''^T Q_x^T W Q_t \int_0^t \frac{[1, s, \dots, s^M]^T}{(t-s)^{1-\beta}} ds. \end{aligned} \quad (2.12)$$

On the other hand, the following relationship is valid:

$$\int_0^t \frac{x^k}{(t-x)^{1-\beta}} dx = \frac{\Gamma(k+1)\Gamma(\beta)}{\Gamma(k+\beta+1)} t^{k+\beta}, \quad 0 < \beta < 1, \quad k = 0, 1, 2, \dots \quad (2.13)$$

So, by substituting (2.13) into (2.12), we have

$$\int_0^t (t-s)^{\beta-1} u_{xx}(x, t) ds \approx \mathcal{P}_N(x)^T Q_x^{-T} D''^T Q_x^T W Q_t S T^\beta(t), \quad (2.14)$$

where

$$S = \text{diag} \left(\frac{\Gamma(\beta)}{\Gamma(\beta+1)}, \frac{\Gamma(2)\Gamma(\beta)}{\Gamma(\beta+2)}, \dots, \frac{\Gamma(M+1)\Gamma(1-\beta)}{\Gamma(\beta+M+1)} \right),$$

and

$$T^\beta(t) = [t^\beta, t^{\beta+1}, \dots, t^{\beta+M}].$$

Hence, using relations (2.9), (2.10), and (2.14), we obtain the following equations:

$$\begin{cases} R_1(x, t) = \mathcal{P}_N(x)^T W Q_t M_\alpha Q_t^{-1} \mathcal{P}_M(t) + \mathcal{P}_N(x)^T W \mathcal{P}_M(t) \mathcal{P}_N(x)^T Q_x^{-T} D^T Q_x^T W \mathcal{P}_M(t) \\ \quad + \mathcal{P}_N(x)^T Q_x^{-T} D''^T Q_x^T W Q_t S T^\beta(t) \approx 0 \\ R_2(x) = \mathcal{P}_N(x)^T W \mathcal{P}_M(0) - \psi(x) \approx 0 \\ R_3(t) = \mathcal{P}_N(0)^T W \mathcal{P}_M(t) - \phi_1(t) \approx 0 \\ R_4(t) = \mathcal{P}_N(L)^T W \mathcal{P}_M(t) - \phi_2(t) \approx 0 \end{cases} \quad (2.15)$$

Now, we collocate equations (2.15) with the points $x_i = \frac{(2i+1)}{2N+2}$ and $t_j = \frac{(2j+1)}{2M+2}$ to obtain

$$\begin{cases} R_1(x_i, t_j) \approx 0 & i = 0, 1, \dots, N-2, \quad j = 0, 1, \dots, M-1, \\ R_2(x_i) \approx 0 & i = 0, 1, \dots, N, \\ R_3(t_j) \approx 0 & j = 0, 1, \dots, M-1, \\ R_4(t_j) \approx 0 & j = 0, 1, \dots, M-1. \end{cases} \quad (2.16)$$

In view of (2.16), we have a nonlinear system of equations. By solving this system, the unknown matrix W can be determined. The numerical solution of the equations (1.1)–(1.3) can be obtained by substituting the matrix W into the approximation (2.5).

3. Numerical experiment

In this section, we use the L_∞ error norm and L_2 error norm

$$\|e\|_\infty = \max_{0 \leq i \leq N, 0 \leq j \leq M} |u(x_i, t_j) - u_{NM}(x_i, t_j)|, \quad \|e\|_2 = \left(h \sum_{j=0}^n (e_j^m)^2 \right)^{\frac{1}{2}},$$

where u and u_{NM} are the exact and approximate solutions of (1.1)–(1.3), respectively.

Example 3.1. Consider the equations (1.1)–(1.3) with the exact solution $u(x, t) = t^3 \sin(\pi x)$. The source term is

$$g(x, t) = \left(\frac{6t^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{6\pi^2\Gamma(\beta)t^{3+\beta}}{\Gamma(4+\beta)} \pi t^6 \cos(\pi x) \right) \sin(\pi x).$$

Absolute errors for $\alpha = 0.5$ and different values of β are reported in Table 1. Furthermore, the norm of errors and CPU times are reported in Table 2.

Table 1: Absolute errors for $\alpha = 0.5$ and different β for Example 3.1

(x_i, t_i)	$\alpha = 0.5, N = 11$	$\alpha = 0.5, N = 11$	$\alpha = 0.5, N = 9$	$\alpha = 0.5, N = 9$
	$\beta = 0.1, M = 4$	$\beta = 0.3, M = 4$	$\beta = 0.7, M = 4$	$\beta = 0.9, M = 4$
(0.1, 0.1)	5.2665e-11	7.5026e-11	9.4157e-10	1.5935e-09
(0.2, 0.2)	8.8713e-10	7.0142e-10	1.7084e-09	4.5505e-09
(0.3, 0.3)	4.6062e-09	3.6713e-09	5.8173e-08	1.3764e-09
(0.4, 0.4)	1.4766e-08	1.2353e-08	3.6620e-07	1.2902e-07
(0.5, 0.5)	3.6426e-08	3.1853e-08	1.2736e-06	6.7104e-07
(0.6, 0.6)	7.6243e-08	6.9345e-08	3.4065e-06	2.2596e-06
(0.7, 0.7)	1.4258e-07	1.3432e-07	7.7630e-06	6.0393e-06
(0.8, 0.8)	2.4555e-07	2.3833e-07	1.5689e-05	1.3673e-05
(0.9, 0.9)	3.8165e-07	3.7801e-07	2.5814e-05	2.4145e-05
(1, 1)	7.3344e-15	1.1944e-16	1.6793e-15	2.1690e-14

Table 2: Norm of errors for $\alpha = \beta = 0.5$ and CPU time for Example 3.1

	$M = 4, N = 6$	$M = 4, N = 7$	$M = 4, N = 8$	$M = 4, N = 9$	$M = 4, N = 10$
$\ e\ _\infty$	1.7427e-03	1.7093e-03	3.7497e-05	3.7046e-05	5.5452e-07
CPU	0.7787s	1.3037s	1.4066s	1.5607s	1.9541s

References

- [1] J. Guo, X. Da, Q. Wenlin, A finite difference scheme for the nonlinear time-fractional partial integro-differential equation. *Mathematical Methods in the Applied Sciences*, 43 (2020) 3392–3412.
- [2] X. Zheng, Q. Wenlin, C. Hongbin, Three semi-implicit compact finite difference schemes for the nonlinear partial integro-differential equation arising from viscoelasticity, *International Journal of Modelling and Simulation* 41 (2021) 234–242.

Mehran Taghipour

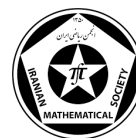
Department of Applied Mathematics and Computer Science, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

email address: mtp20222@yahoo.com

Hossein Aminikhah

Department of Applied Mathematics and Computer Science, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

email address: aminikhah@guilan.ac.ir



Numerical solution of non-linear reaction-diffusion equations using meshless radial point Hermite interpolation method

Masoud Pendar, Kamal Shanazari

Abstract. In this work, we present a numerical method to solve two-dimensional non-linear reaction-diffusion equations. Meshless and collocation techniques using radial basis functions (RBFs) with the help of radial point Hermite interpolation (RPHI) method are employed to construct the so called shape functions. Due to the use of meshless method, no mesh generation is required in the spatial domain. Time discretization is performed using the finite difference method and Taylor expansion is utilized for the non-linear part. The accuracy and efficiency of the method are examined by a numerical example. The numerical results show that this procedure is stable through the time.

1. Introduction

The mesh-based methods such as finite element method, finite difference method and boundary element method are still important tools in solving engineering problems and numerical analysis. However, in the recent decades, meshless methods such as collocation method based on radial basis functions (RBFs) have become alternatives for the numerical solutions of partial differential equations (PDEs) [2]. Hermite radial point interpolation method is a combination of point interpolation and Hermite interpolation technique based on RBFs. In this method, in addition to the unknown function, its derivative at the field nodes are considered as independent variables to construct the shape functions. This provides a suitable tool for imposing the Neumann and Robin boundary conditions which results in accurate numerical solutions. For more details see for example [3, 4] and references therein. Besides the steady state equations, the radial point Hermite interpolation method has been applied to time-dependent linear problems [3–5]. Many physical and engineering phenomena are modeled by time-dependent non-linear equations. In this work, we apply the proposed method to the non-linear reaction-diffusion equations which is introduced in subsection 2.2.

2. Main Results

2.1. Meshless RPHI method

In this section, we briefly introduce some important features and notations of meshless RPHI method. For more details we refer to [5]. Suppose that $u(\mathbf{x})$ is defined in a domain Ω , represented by a set of field nodes. In the RPHI method, the approximate function at a point of interest

Keywords: Nonlinear reaction-diffusion equation, Meshless method, Radial basis function, Radial point Hermite interpolation method.

AMS Mathematical Subject Classification [2010]: 35Q79, 49M30, 65N35.

\mathbf{x} contains the nodal values of $u(\mathbf{x})$ and its derivatives at the field nodes as follows [5]:

$$\begin{aligned} u(\mathbf{x}) &= \sum_{j=1}^n \varphi_j(\mathbf{x})a_j + \sum_{j=1}^n \frac{\partial \varphi_j(\mathbf{x})}{\partial x} a_j^x + \sum_{j=1}^n \frac{\partial \varphi_j(\mathbf{x})}{\partial y} a_j^y + \sum_{j=1}^m p_j(\mathbf{x})b_j \\ &= \Phi^T(\mathbf{x})\mathbf{a} + \Phi_{,x}^T(\mathbf{x})\mathbf{a}^x + \Phi_{,y}^T(\mathbf{x})\mathbf{a}^y + P^T(\mathbf{x})\mathbf{b}, \end{aligned} \quad (2.1)$$

where $,x$ and $,y$ represent the partial derivatives with respect to x and y respectively, $\varphi_j(\mathbf{x})$ denotes an RBF, n is the number of basis functions, $p_j(\mathbf{x})$ is the augmented monomial which is built using Pascal's triangle, m is the number of polynomial basis functions. In this work, we choose $\varphi(r) = r^4 \ln(r)$, namely Thin plate spline (TPS), as the RBF. In order to determine the unknown coefficients in (2.1), n field points are included in a support domain that is formed for the point of interest at \mathbf{x} . If we use the interpolation conditions for $u(\mathbf{x})$ and its derivatives then, in a matrix form, we can write

$$\begin{aligned} U &= \Phi_n \mathbf{a} + \Phi_n^x \mathbf{a}^x + \Phi_n^y \mathbf{a}^y + P_m \mathbf{b}, \quad U_x = \Phi_n^x \mathbf{a} + \Phi_n^{xx} \mathbf{a}^x + \Phi_n^{xy} \mathbf{a}^y + P_m^x \mathbf{b} \\ U_y &= \Phi_n^y \mathbf{a} + \Phi_n^{yx} \mathbf{a}^x + \Phi_n^{yy} \mathbf{a}^y + P_m^y \mathbf{b}. \end{aligned} \quad (2.2)$$

There are $m + 3n$ unknown coefficients and $3n$ equations in (2.2). Since TPS is a conditionally positive definite RBF, we can add m equations as $P_m^T \mathbf{a} + P_m^{xT} \mathbf{a}^x + P_m^{yT} \mathbf{a}^y = 0$ and combine them with (2.2) so that we obtain a $(m + 3n) \times (m + 3n)$ system of equations as

$$\tilde{U}_s = \begin{pmatrix} U \\ U_x \\ U_y \\ 0 \end{pmatrix} = \begin{pmatrix} \Phi_n & \Phi_n^x & \Phi_n^y & P_m \\ \Phi_n^x & \Phi_n^{xx} & \Phi_n^{xy} & P_m^x \\ \Phi_n^y & \Phi_n^{yx} & \Phi_n^{yy} & P_m^y \\ P_m^T & P_m^{xT} & P_m^{yT} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{a}^x \\ \mathbf{a}^y \\ \mathbf{b} \end{pmatrix} = G \tilde{\mathbf{a}}_s, \quad (2.3)$$

where G is a symmetric matrix and also invertible. Solving equations (2.3) we get $\tilde{\mathbf{a}}_s = G^{-1} \tilde{U}_s$ and inserting it in (2.1), we obtain $u(\mathbf{x}) = \tilde{\Phi}^T(\mathbf{x}) \tilde{U}_s$, where

$$\tilde{\Phi}^T(\mathbf{x}) = \begin{pmatrix} \phi_1(\mathbf{x}) \dots \phi_n(\mathbf{x}) & \phi_1^x(\mathbf{x}) \dots \phi_n^x(\mathbf{x}) & \phi_1^y(\mathbf{x}) \dots \phi_n^y(\mathbf{x}) & \phi_1^p(\mathbf{x}) \dots \phi_m^p(\mathbf{x}) \end{pmatrix}. \quad (2.4)$$

The first $3n$ functions in (2.4) are called radial point Hermite interpolation shape functions. For the unknown function $u(\mathbf{x})$ we can write

$$u(\mathbf{x}) = \sum_{j=1}^n \phi_j(\mathbf{x})u_j + \sum_{j=1}^n \phi_j^x(\mathbf{x})u_j^x + \sum_{j=1}^n \phi_j^y(\mathbf{x})u_j^y. \quad (2.5)$$

Let the total number of nodes covering the domain Ω is N , then we can replace n in (2.5) by N . So we obtain the operational matrices for calculating the derivative of function u in (2.5) as $U_x^{(s)} = D_x^{(s)}U$, $U_y^{(s)} = D_y^{(s)}U$. For more details see [5].

2.2. The nonlinear reaction-diffusion equation

We consider the following two-dimensional reaction-diffusion equation:

$$\alpha \left(\frac{\partial u}{\partial t} \right) + \nabla^2 u + \beta g(u) = f(\mathbf{x}, t), \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^2, \quad t \in (0, T]. \quad (2.6)$$

with the Neumann boundary condition $\frac{\partial u}{\partial \mathbf{n}} = h_1(\mathbf{x}, t)$, $\mathbf{x} \in \partial\Omega$, $t \in (0, T]$ and the initial condition $u(\mathbf{x}, 0) = h_2(\mathbf{x})$, $\mathbf{x} \in \Omega \cup \partial\Omega$. where ∇ is the gradient differential operator, α and β are known constants, g is the non-linear known function, \mathbf{n} is the outward normal vector on the boundary, T denotes the total time, f , h_1 and h_2 are known functions and the function u is unknown.

2.3. Time discretization

We apply a Crank-Nicolson scheme on (2.6) as

$$\alpha\left(\frac{u^{n+1} - u^n}{\delta t}\right) + \frac{1}{2}(\nabla^2 u^{n+1} + \nabla^2 u^n) + \frac{1}{2}\beta(g(u^{n+1}) + g(u^n)) = f^{n+1}, \quad (2.7)$$

where δt denotes the time step size and $u^n = u(\mathbf{x}, t_n)$ represents u at the time level n . Equation (2.7) is a non-linear PDE for the unknown function $u^n(\mathbf{x})$. We need to solve the stationary PDE (2.7) at time level $n + 1$. If we write the Taylor expansion of $g(u^{n+1}) = g(u(\mathbf{x}, t_n + \delta t))$ around t_n and insert it in (2.7), then we have

$$\alpha u^{n+1} + \frac{\delta t}{2} \nabla^2 u^{n+1} = \delta t \alpha u^n - \frac{\delta t}{2} \nabla^2 u^n - \delta t \beta g(u^n) - \frac{1}{2}(\delta t)^2 \beta g'_t(u^n) + f^{n+1}. \quad (2.8)$$

This removes the non-linear term and, consequently, in a matrix form we can write

$$A[U]^{n+1} = B[U]^n + C_n, \quad (2.9)$$

where A and B are $3N \times 3N$ matrices and C_n is a vector of size $3N \times 1$. Using the initial condition as the value of u at the time level n , we can evaluate the function u at the time level $n + 1$ by solving the linear system (2.9). To impose the boundary conditions we apply the method introduced in [4, 5].

2.4. Numerical example

In this subsection, we examine the performance of the proposed method, by considering a numerical example. To measure the accuracy, we use the maximum absolute error $\|u\|_\infty = \max\{|u_e(\mathbf{x}_i) - u_a(\mathbf{x}_i)|, 1 \leq i \leq N\}$.

Example. We consider the following 2-D non-linear reaction-diffusion equation [1]:

$$\frac{\partial u}{\partial t} + \nabla^2 u + \cos(u(x, y, t)) = \cos(e^{-2t} \sin(x) \sin(y)), \quad (x, y, t) \in [0, \pi]^2 \times (0, T],$$

with the boundary and initial conditions as $u_x(0, y, t) = h_1$, $u_x(\pi, y, t) = -h_1$ and $u_y(x, 0, t) = h_2$, $u_y(x, \pi, t) = -h_2$ where $h_1(y, t) = e^{-2t} \sin(y)$, $h_2(x, t) = e^{-2t} \sin(x)$. The exact solution of the above equation is given by $u(x, y, t) = e^{-2t} \sin(x) \sin(y)$. The error values at $T = 2s$ for various lengths of h are presented in Table 1. Also, the order of convergence in space and in time are presented in the columns denoted by O.C.(h) and O.C.(t) respectively.

Table 1: Error values in approximating u , u_x and u_y for various h and order of convergence in space and time.

$t = 2s, h$	$\ U\ _\infty$	O.C.(h)	$\ U_x\ _\infty$	$\ U_y\ _\infty$
$\frac{\pi}{5}$	$5.4e - 3$	—	$3.8e - 2$	$3.8e - 2$
$\frac{\pi}{10}$	$2.9e - 3$	0.89	$3.7e - 3$	$3.7e - 3$
$\frac{\pi}{20}$	$1.9e - 4$	3.93	$2.7e - 4$	$2.7e - 4$
$\frac{\pi}{40}$	$1.5e - 5$	3.66	$1.9e - 5$	$1.9e - 5$
$h = \frac{\pi}{20}, t(s)$		O.C.(t)		
0.5	$1.6e - 3$	—	$2.8e - 3$	$3.8e - 2$
1	$7.4e - 4$	1.1	$1.2e - 3$	$9.5e - 3$
2	$1.9e - 4$	1.96	$3.1e - 4$	$1.9e - 4$
4	$7.8e - 6$	4.69	$9.6e - 6$	$7.4e - 5$
8	$4.9e - 9$	10.63	$5.1e - 9$	$4.8e - 8$

Conclusions

We applied the RPHI method to a non-linear reaction-diffusion equation. Time discretization was performed by Crank-Nicholson method. For the non-linear part, a Taylor expansion method with respect to time was used. The results showed that this method is stable and by increasing the number of nodes the approximation error can be reduced.

References

- [1] W. Liao, Z. Jianping, Q.K. Abdul, A fourth-order compact algorithm for nonlinear reaction-diffusion equations with Neumann boundary conditions, *Numerical Methods for Partial Differential Equations*, 22 (2006) 600–616.
- [2] G.R. Liu, G. Yuan-Tong, *An introduction to meshfree methods and their programming*, Springer Science Business Media, 2005.
- [3] Y. Liu, Y. C. Hon, K. M. Liew, A meshfree Hermite-type radial point interpolation method for Kirchhoff plate problems, *International Journal for Numerical Methods in Engineering*, 66 (2006) 1153–1178.
- [4] Y.E. Seblani, E. Shivanian, New insight into meshless radial point Hermite interpolation through direct and inverse 2-D reaction-diffusion equation, *Engineering with Computers*, 37 (2021) 3605–3613.
- [5] E. Shivanian, Pseudo spectral meshless radial point Hermit interpolation versus Pseudo spectral meshless radial point interpolation, *International Journal of Computational Methods*, 17 (2020) 1950023.

Masoud Pendar

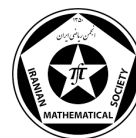
Department of Mathematics, Faculty of Science, U. of Kurdistan, Sanandaj, Iran

email address: m.pendar@uok.ac.ir

Kamal Shanazari

Department of Mathematics, Faculty of Science, U. of Kurdistan, Sanandaj, Iran

email address: k.shanazari@uok.ac.ir



A novel shifted Jacobi operational matrix method: An application for solving nonlinear multi-term fractional variable-order differential equations

Hamidreza Khodabandehlo, Elyas Shivanian

Abstract. This paper presents the generalized multi-term fractional variable-order differential equations. In this article, a novel shifted Jacobi operational matrix technique is introduced for solving a class of these equations via reducing the main problem to an algebraic system of equations that can be solved numerically. The suggested technique is successfully developed for the aforementioned problem. Comprehensive numerical experiments are presented to demonstrate the efficiency, generality, accuracy of proposed scheme and the flexibility of this method. Comparing the results of the current method (*NSJOM*) with the exact solution, indicating the efficiency and validity of this method. Note that the procedure is easy to implement and this technique will be considered as a generalization of many numerical schemes.

1. Introduction

Fractional calculus analysis and applications are an active and rapidly growing area for research in the last three decades. At present, due to their extensive applications in diverse scientific disciplines, such as physics, regular changes in thermodynamics, etc., it has become an important tool ([1] and its resources). The increasing development of appropriate and efficient method to solve *FDEs* has aroused more interest of researchers in this field. In recent years, many attempts have been made to solve the *FDEs*, we refer the interested reader to [3–5] and etc. In this paper, the our aim is to generalize the orthogonal polynomials in the base of solution. In fact, we present a new shifted Jacobi operational matrix for the fractional derivative to solve the nonlinear multi-term variable order *FDEs* which as follow:

$$\sum_{j=1}^n \alpha_j D^{\eta_j(t)} w(t) = F(t, w(t), D^{\eta_1(t)} w(t), D^{\eta_2(t)} w(t), \dots, D^{\eta_n(t)} w(t)), \quad (1.1)$$

$0 \leq t \leq T, w^{(i)}(0) = 0, i = 0, 1, 2, \dots, n-1, n \in \mathbb{N}$, where $\alpha_j \in \mathbb{R} (j = 1, 2, \dots, n), 0 < T$ and $D^{\eta_j(t)} w(t) (j = 1, 2, \dots, n)$ are the Caputo's derivative of variable-order fractional.

2. Fundamentals and preliminaries

This article is based on Caputo definition because, as well as know, only the Caputo sense has the same form as integer-order differential equations in initial conditions.

Keywords: Shifted Jacobi Operational Matrix technique, Multi-term variable-order FDE..

AMS Mathematical Subject Classification [2010]: 13F55, 05E40, 05C65.

Definition 2.1. The Caputo derivative with fractional variable-order $\eta(t)$ for $w(t) \in C^m[0, T]$ are given respectively as [2]:

$$D^{\eta(t)}w(t) = \frac{1}{\Gamma(1-\eta(t))} \int_{0+}^t \frac{w'(s)}{(t-s)^{\eta(t)}} ds + \frac{w(0^+) - w(0^-)}{\Gamma(1-\eta(t))} t^{-\eta(t)}.$$

Definition 2.2. Denote $P_{T,i}^{(\alpha,\beta)}(t); \alpha > -1, \beta > -1$ as the n -th order Shifted Jacobi polynomial in t defined on $[0, T]$, and [2]:

$$P_{T,n}^{(\alpha,\beta)}(t) = \sum_{k=0}^n (-1)^{n-k} \frac{\Gamma(\alpha+n+1)\Gamma(\alpha+\beta+k+n+1)}{\Gamma(\alpha+\beta+n+1)\Gamma(\alpha+1+k)\Gamma(k+1)\Gamma(n-k+1)} T^k t^k.$$

2.1. Function approximation by shifted Jacobi polynomials

The function $w(t)$, square integrable with respect to $\omega_T^{(\alpha,\beta)}(t)$ in $[0, T]$, can be expanded as the following expression [2, 5]:

$$w(t) = \sum_{i=0}^{\infty} a_i P_{T,i}^{(\alpha,\beta)}(t), a_i = \frac{1}{h_{T,j}^{(\alpha,\beta)}} \int_0^T \omega_T^{(\alpha,\beta)} P_{T,i}^{(\alpha,\beta)}(t) w(t) dt, i = 0, 1, \dots \quad (2.1)$$

So, we can estimate the approximate solution by taking $(N+1)$ -terms of the series in Eq. (2.1) and we will have

$$w(t) \simeq w_N(t) = \sum_{i=0}^N a_i P_{T,i}^{(\alpha,\beta)}(t) = A^T \Phi_{T,N}(t), \quad (2.2)$$

where $A = [a_0, a_1, \dots, a_N]^T$, and $\Phi_{T,N}(t) = [P_{T,0}^{(\alpha,\beta)}(t), P_{T,1}^{(\alpha,\beta)}(t), \dots, P_{T,N}^{(\alpha,\beta)}(t)]^T$. Here, we suppose that $S(t) = [1, t, t^2, t^3, \dots, t^N]^T$. By equation (2.2), the vector $\Phi_{T,N}(t)$ can be presented as $\Phi_{T,N}(t) = B_{(\alpha,\beta)} S(t)$, where $B_{(\alpha,\beta)}$ is a square matrix of order $(N+1) \times (N+1)$. Hence, we get

$$S(t) = B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t). \quad (2.3)$$

3. Shifted Jacobi Polynomials Operational Matrix (SJOM)

At first, $D^{\eta_i(t)} \Phi_{T,N}(t), (i = 1, 2, \dots, n)$ can be deduced as the following: since $\Phi_{T,N}(t) = B_{(\alpha,\beta)} S(t)$, then we have

$$D^{\eta_i(t)} \Phi_{T,N}(t) = D^{\eta_i(t)} (B_{(\alpha,\beta)} S(t)) = B_{(\alpha,\beta)} D^{\eta_i(t)} [1, t, \dots, t^N]^T, i = 1, 2, \dots, n. \quad (3.1)$$

Then

$$\begin{aligned} D^{\eta_i(t)} \Phi_{T,N}(t) &= B_{(\alpha,\beta)} D^{\eta_i(t)} (S(t)) = B_{(\alpha,\beta)} \left[0, \frac{\Gamma(2)t^{(1-\eta_i(t))}}{\Gamma(2-\eta_i(t))}, \dots, \frac{\Gamma(N+1)t^{(N-\eta_i(t))}}{\Gamma(N+1-\eta_i(t))} \right]^T \\ &= B_{(\alpha,\beta)} \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & \frac{\Gamma(2)t^{-\eta_i(t)}}{\Gamma(2-\eta_i(t))} & 0 & \dots & 0 \\ 0 & 0 & \frac{\Gamma(3)t^{-\eta_i(t)}}{\Gamma(3-\eta_i(t))} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \frac{\Gamma(N)t^{-\eta_i(t)}}{\Gamma(N-\eta_i(t))} \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ \vdots \\ t^N \end{bmatrix} \\ &= B_{(\alpha,\beta)} Q_i(t) S(t), i = 1, 2, \dots, n. \end{aligned} \quad (3.2)$$

Table 1: Absolute errors of $w(t)$ with $N = 3$ and $T = 1$ for Ex.1 by *NSJOM*.

$t \in [0, T]$	$\alpha = 1, \beta = 1$	$\alpha = 0, \beta = 0$	$\alpha = 0.5, \beta = 0.5$
0	8.4655×10^{-16}	6.9389×10^{-17}	1.8041×10^{-16}
0.2	1.0339×10^{-15}	6.9389×10^{-18}	2.0122×10^{-16}
0.4	1.0963×10^{-15}	8.3267×10^{-17}	1.9428×10^{-16}
0.6	1.0825×10^{-15}	1.3878×10^{-16}	1.9428×10^{-16}
0.8	9.7144×10^{-16}	1.6653×10^{-16}	1.1102×10^{-16}
1.0	9.0206×10^{-16}	1.1102×10^{-16}	1.5265×10^{-16}
CPU time	0.3432s	0.2028s	0.3276s

Using Eq.(2.3), then $D^{\eta_i(t)}\Phi_{T,N}(t) = B_{(\alpha,\beta)}Q_i(t)B_{(\alpha,\beta)}^{-1}\Phi_{T,N}(t), i = 1, 2, \dots, n$. The operational matrix of $D^{\eta_i(t)}\Phi_{T,N}(t), (i = 1, 2, \dots, n.)$ is $B_{(\alpha,\beta)}Q_i(t)B_{(\alpha,\beta)}^{-1}$. Now, we can estimate the multi-order fractional of the approximated function that obtained in Eq. (2.2). Finally, we use $t_j(j = 0, 1, 2, \dots, m.)$ wher they are the roots of $P_{T,m+1}^{(\alpha,\beta)}(t)$. Then Eq. (1.1) can be converted into the following algebraic system

$$\sum_{i=1}^n \alpha_i (A^T B_{(\alpha,\beta)} Q_i(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)) = F(t_j, A^T \Phi_{T,N}(t_j), (A^T B_{(\alpha,\beta)} Q_1(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)), (A^T B_{(\alpha,\beta)} Q_2(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)), \dots, (A^T B_{(\alpha,\beta)} Q_n(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)), A^T \Phi_{T,N}(t_j - \tau)), j = 0, 1, 2, \dots, m. \quad (3.3)$$

So, the system in Eq. (3.3) can be solved numerically for determining the unknown vector A . Therefore, the numerical solution that presented in Eq. (2.2) can be obtained.

4. Numerical experinces

Example 4.1. Consider the following multi-order fractional *DE*

$$D^{\eta_1(t)}w(t) + D^{\eta_2(t)}w(t) + D^{\eta_3(t)}w(t) - w(t) - 2w(t)^3 = \left(\frac{\Gamma(4)w(t)^{3-\eta_1(t)}}{\Gamma(4-\eta_1(t))} - \frac{\Gamma(3)w(t)^{2-\eta_1(t)}}{\Gamma(3-\eta_1(t))}\right) + \left(\frac{\Gamma(4)w(t)^{3-\eta_2(t)}}{\Gamma(4-\eta_2(t))} - \frac{\Gamma(3)w(t)^{2-\eta_2(t)}}{\Gamma(3-\eta_2(t))}\right) + \left(\frac{\Gamma(4)w(t)^{3-\eta_3(t)}}{\Gamma(4-\eta_3(t))} - \frac{\Gamma(3)w(t)^{2-\eta_3(t)}}{\Gamma(3-\eta_3(t))}\right) - (t^3 - t^2) - 2(t^3 - t^2)^3, w^{(0)}(0) = 0. \quad (4.1)$$

Note that $w(t) = t^3 - t^2$ is the exact solution and $0 \leq t \leq T, T = 1, \eta_1(t) = \frac{t}{3}, \eta_2(t) = \frac{t}{2}$ and $\eta_3(t) = \frac{2t}{5}$. The absolute errors (at some nodal points) of this method, also the CPU time needed for our method for different values of α and β are shown in Table. (1). From this Table, it is observed that the numerical results which obtained via our technique, are much closer to the true solution.

References

- [1] K. Diethelm, N.J. Ford, A. D. Freed, Detailed error analysis for a fractional Adams method, *Numerical Algorithms*, 36 (2004) 31-52.

- [2] A.A. El-Sayed, D. Baleanu, P. Agarwal, A novel Jacobi operational matrix for numerical solution of multi-term variable-order fractional differential equations, *Journal of Taibah University for Science*, 14 (2020) 963-974.
- [3] H.R. Khodabandehlo, E. Shivanian, S. Abbasbandy, Numerical solution of nonlinear delay differential equations of fractional variable-order using a novel shifted Jacobi operational matrix, *Engineering with Computers*, 18 May 2021.
- [4] R. Gorenflo, Fractional calculus, Fractals and Fractional Calculus in Continuum Mechanics, Springer Verlag, Wien, 1997, pp. 277–290.
- [5] A.H. Bhrawy, M.A. Zaky, A method based on the Jacobi tau approximation for solving multi-term time-space fractional partial differential equations, *Journal of Computational Physics*, 281 (2014) 876–895.

Hamidreza Khodabandehlo

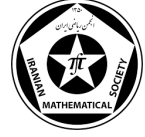
Department of Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin, Iran
email address: khodabandehlo.hamidreza@yahoo.com

Elyas Shivanian

Department of Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin, Iran
email address: shivanian@sci.ikiu.ac.ir



University of Guilan



A difference scheme for fourth-order fractional partial integro-differential equation

Mehran Taghipour, Hossein Aminikhah

Abstract. This work presents a difference scheme by considering cubic B-spline Quasi-interpolation for the numerical solution of a fourth-order time-fractional integro-differential equation with a weakly singular kernel. The fractional derivative of the mentioned equation has been described in the Caputo sense. Time fractional derivative is approximated by a scheme of order $O(\tau^{2-\alpha})$ and the Riemann-Liouville fractional integral term is discretized by the fractional trapezoidal formula. The spatial second derivative has been approximated using the second derivative of the cubic B-spline Quasi-interpolation. The discrete scheme leads to the solution of a system of linear equations.

1. Introduction

In this work, we consider the fourth-order time-fractional integro-differential equation (PIDE) with a weakly singular kernel as follows [1]:

$$\begin{cases} {}^C\mathcal{D}_{0,t}^\alpha u(x,t) - u_{xx}(x,t) - \mathcal{I}^{(\beta)} u_{xx}(x,t) + u_{xxxx}(x,t) = f(x,t), & (x,t) \in \Omega, \\ u(x,0) = u^0(x), & 0 \leq x \leq L, \\ u(0,t) = u(L,t) = u_{xx}(0,t) = u_{xx}(L,t) = 0, & 0 < t \leq T, \end{cases} \quad (1.1)$$

where $\Omega = (0, L) \times (0, T]$, $0 < \alpha, \beta < 1$, $f(x,t)$ is source term and $u^0(x)$ is given smooth function. In fact, problem (1.1) is equivalent to

$$\begin{cases} {}^C\mathcal{D}_{0,t}^\alpha u(x,t) - v(x,t) - \mathcal{I}^{(\beta)} v(x,t) + v_{xx}(x,t) = f(x,t), & (x,t) \in \Omega, \\ v(x,t) = u_{xx}(x,t), & 0 < x < L, \quad 0 < t \leq T, \\ u(x,0) = u^0(x), & 0 \leq x \leq L, \\ u(0,t) = u(L,t) = v(0,t) = v(L,t) = 0, & 0 < t \leq T. \end{cases} \quad (1.2)$$

In (1.2), ${}^C\mathcal{D}_{0,t}^\alpha$ is fractional derivative operator in caputo sense and $\mathcal{I}^{(\beta)}$ is defined as follows

$$\mathcal{I}^{(\beta)} u_{xx}(x,t) = \frac{1}{\Gamma(\beta)} \int_0^t (t-s)^{\beta-1} u_{xx}(x,s) ds, \quad t > 0, \quad (1.3)$$

where $\Gamma(\cdot)$ is the Gamma function. Equation (1.1), can be found in the modeling of floor systems, window glasses, airplane wings, and bridge slabs.

Keywords: B-spline Quasi-interpolation, Time-fractional partial integro-differential equation, Weakly singular kernel.

AMS Mathematical Subject Classification [2010]: 65M12, 65M70, 65R10, 34K37.

2. Numerical method

The domain is divided into a uniform grid of mesh points (x_j, t_k) with $x_j = jh$, $h = \frac{L}{M}$, $0 \leq j \leq M$ and $t_k = k\tau$, $\tau = \frac{T}{N}$, $0 \leq k \leq N$. The values of the function u at the grid points are denoted $u(x_i, t_k)$ and U_i^k is the approximate solution at the point (x_i, t_k) .

Definition 2.1. The Caputo derivatives of order α are defined by

$${}_C\mathcal{D}_{a,x}^\alpha f(x) = \frac{1}{\Gamma(n-\alpha)} \int_a^x \frac{f^n(t)}{(x-t)^{\alpha-n+1}} dt, \quad a < x. \quad (2.1)$$

Lemma 2.2. (L_1 approximation) Let $\alpha \in (0, 1)$ and $u(., t) \in C_t^2([0, T])$ then the following approximation formula holds

$${}_C\mathcal{D}_{0,t}^\alpha u(x, t_k) = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} \left[b_0 u(x, t_k) - \sum_{j=1}^{k-1} (b_{k-j-1} - b_{k-j}) u(x, t_j) - b_{k-1} u(x, t_0) \right] + R, \quad (2.2)$$

in which

$$b_j = [(l+1)^{1-\alpha} - l^{1-\alpha}], \quad 0 \leq l \leq k-1, \quad (2.3)$$

$$|R| \leq C\tau^{2-\alpha} \quad (2.4)$$

Lemma 2.3. Let $\beta \in (0, 1)$ and $u(., t)$ is suitably smooth on $(0, T)$ then for the $\mathcal{I}^{(\beta)}$ there holds that

$$\mathcal{I}^{(\beta)} u(x, t_k) = \sum_{j=0}^k a_{j,k} u(x, t_j) + O(\tau^2), \quad (2.5)$$

where

$$a_{j,k} = \frac{\tau^\beta}{\Gamma(\beta+2)} \begin{cases} (k-1)^{\beta+1} - (k-1-\beta)k^\beta, & j=0, \\ (k-j+1)^{\beta+1} + (k-1-j)^{\beta+1} - 2(k-j)^{\beta+1}, & 1 \leq j \leq k-1, \\ 1, & j=k. \end{cases}$$

Definition 2.4. Suppose for a nonnegative integer p and some integer j that $\xi_{j-p-1} \leq \xi_{j-p} \leq \dots \leq \xi_j$ are $p+2$ real numbers taken from a knot sequence ξ . The j -th B-spline $B_{j,p,\xi} : \mathbb{R} \rightarrow \mathbb{R}$ of degree p is identically zero if $\xi_{j-p-1} = \xi_j$ and otherwise defined recursively by

$$B_{j,p,\xi}(x) = \frac{x - \xi_{j-p-1}}{\xi_{j-1} - \xi_{j-p-1}} B_{j-1,p-1,\xi}(x) + \frac{\xi_j - x}{\xi_j - \xi_{j-p}} B_{j,p-1,\xi}(x), \quad (2.6)$$

starting with

$$B_{i,0,\xi}(x) = \begin{cases} 1, & \text{if } x \in [\xi_{i-1}, \xi_i), \\ 0, & \text{otherwise.} \end{cases}$$

Let λ_j be a linear functional defined on $C[a, b]$ that can be computed from values of f at some set of points in $[a, b]$. We have the following definition.

Definition 2.5. [2] A formula of the form

$$Q_p f(x) := \sum_{j=1}^{n+p} (\lambda_j f) B_{j,p,\xi}(x), \quad (2.7)$$

is called a B-spline quasi-interpolation formula of degree p .

Suppose $a = x_0 < \dots < x_n = b$ are equally spaced points in the interval $[a, b]$. We have the following theorem.

Theorem 2.6. [2] Given a function f defined on $[a, b]$, let

$$\lambda_j f := \begin{cases} f(x_0), & j = 1, \\ \frac{1}{18}(7f(x_0) + 18f(x_1) - 9f(x_2) + 2f(x_3)), & j = 2, \\ \frac{1}{6}(-f(x_{j-3}) + 8f(x_{j-2}) - f(x_{j-1})), & 3 \leq j \leq n+1, \\ \frac{1}{18}(2f(x_{n-3}) - 9f(x_{n-2}) + 18f(x_{n-1}) + 7f(x_n)), & j = n+2, \\ f(x_n), & j = n+3. \end{cases} \quad (2.8)$$

Then (2.7) defines a linear operator mapping $C[a, b]$ into the space of splines spanned by the B-splines with $Q_p s = s$ for all cubic polynomials s .

For approximate derivatives of f by derivatives of $Q_3 f$ up to the order h^3 , we can evaluate the value of f' and f'' at x_j by $(Q_3 f)'(x) = \sum_{j=1}^{n+3} (\lambda_j f) B_{j,p,\xi}(x)'(x)$ and $(Q_3 f)''(x) = \sum_{j=1}^{n+3} (\lambda_j f) B_{j,p,\xi}(x)''(x)$. We set $Y = (f_0, f_1, \dots, f_n)^T$, $Y' = (f'_0, f'_1, \dots, f'_n)^T$ and $Y'' = (f''_0, f''_1, \dots, f''_n)^T$ where $f'_j = (Q_3 f)'(x_j)$, $j = 1, \dots, n$ and $f''_j = (Q_3 f)''(x_j)$, $j = 1, \dots, n$. The first and the second derivatives of $Q_3(f)$ are calculated as

$$f'_j = \sum_{j=1}^{n+3} (\lambda_j f) B_{j,p,\xi}(x)'(x), \quad j = 0, 1, \dots, n, \quad (2.9)$$

$$f''_j = \sum_{j=1}^{n+3} (\lambda_j f) B_{j,p,\xi}(x)''(x), \quad j = 0, 1, \dots, n. \quad (2.10)$$

Therefore, we can display the approximation of f' and f'' in the following matrix form

$$Y' = \frac{1}{h} D_1 Y, \quad Y'' = \frac{1}{h^2} D_2 Y, \quad (2.11)$$

where $D_1, D_2 \in \mathbb{R}^{(n+1) \times (n+1)}$ are pentadiagonal matrices.

Considering (1.2) at the point (x_i, t_k) , one has

$$\begin{aligned} \mathcal{D}_{0,t}^\alpha u(x_i, t_k) - v(x_i, t_k) - \mathcal{I}^{(\beta)} v(x_i, t_k) + v_{xx}(x_i, t_k) &= f(x_i, t_k), \\ v(x_i, t_k) &= u_{xx}(x_i, t_k), \quad 1 \leq i \leq M-1, \quad 1 \leq k \leq N. \end{aligned} \quad (2.12)$$

Using (2.2), (2.5) and (2.9), (2.10), (2.12) can be approximated by

$$\frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} [b_0 u_i^k - \sum_{j=1}^{k-1} (b_{k-j-1} - b_{k-j}) u_i^j - b_{k-1} u_i^0] - v_i^k - \sum_{j=0}^k a_{j,k} v_i^j + \sum_{j=0}^M \frac{d_{ij}^2}{h^2} v_j^k = f_i^k + (R_1)_i^k, \quad (2.13)$$

$$v_i^k = \sum_{j=0}^M \frac{d_{ij}^2}{h^2} u_j^k + (R_2)_i^k, \quad 1 \leq i \leq M-1, \quad 1 \leq k \leq N, \quad (2.14)$$

where $|(R_1)_i^k| \leq C(\tau^{2-\alpha} + h^2)$ and $|(R_2)_i^k| \leq Ch^2$. So that in each time step we encounter the following system of linear equations

$$AU^k = F^k. \quad (2.15)$$

3. Numerical experiment

In this section, we use the following error norm

$$\|e(\tau, h)\| = \|e^N\| = \left(\Delta x \sum_{j=1}^M (e_j^N)^2 \right)^{\frac{1}{2}},$$

where $e_j^k = u(x_j, t_k) - U_j^k$.

Example 3.1. Consider the equations (1.1)-(1.3) with the exact solution $u(x, t) = t^{\alpha+\beta} \sin(\pi x)$. The source term is

$$f(x, t) = \left(\frac{\Gamma(\alpha + \beta + 1)}{\Gamma(\beta + 1)} t^{-\alpha} + \frac{\pi^2 \Gamma(\alpha + \beta + 1)}{\Gamma(2\beta + 1)} t^\beta + \pi^2 + \pi^4 \right) t^{\alpha+\beta} \sin(\pi x).$$

In Table 1, we record the absolute errors and convergence orders in spatial direction for different

Table 1: L2-norm errors and order of convergence for $\alpha = 0.1, 0.3, 0.5$ and $\beta = 0.1, 0.15, 0.45$ for Example 3.1

h	τ	$\alpha = 0.1$	$\beta = 0.1$	τ	$\alpha = 0.3$	$\beta = 0.15$	τ	$\alpha = 0.5$	$\beta = 0.45$
	$\frac{1}{55}$	$\ e^N\ $	$r_1(\tau, h)$	$\frac{1}{135}$	$\ e^N\ $	$r_1(\tau, h)$	$\frac{1}{15}$	$\ e^N\ $	$r_1(\tau, h)$
$\frac{1}{10}$		9.6233e-03			9.9334e-03			1.0321e-02	
$\frac{1}{20}$		2.6307e-03	1.8711		2.6851e-03	1.8873		2.7430e-03	1.9116
$\frac{1}{40}$		6.4975e-04	2.0175		6.6955e-04	2.0037		6.7759e-04	2.0173
$\frac{1}{80}$		1.5008e-04	2.1141		1.6518e-04	2.0191		1.6533e-04	2.0350
$\frac{1}{160}$		2.5577e-05	2.5528		4.0002e-05	2.0459		3.8777e-05	2.0921
$\frac{1}{320}$		5.4572e-06	2.2286		8.8637e-06	2.1741		7.3685e-06	2.3958

values of α and β . We have used the following formula to calculate the convergence rate:

$$r_1(\tau, h) = \log_2 \left(\frac{\|e(\tau, 2h)\|}{\|e(\tau, h)\|} \right).$$

References

- [1] D. Xu, , Q. Wenlin, G. Jing, A compact finite difference scheme for the fourth-order time-fractional integro-differential equation with a weakly singular kernel, *Numerical Methods for Partial Differential Equations*, 36 (2020) 439–458.
- [2] L.L. Schumaker, *Spline Functions: Basic Theory*. Wiley Interscience, New York, 1981.

Mehran Taghipour

Department of Applied Mathematics and Computer Science, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

email address: mtp20222@yahoo.com

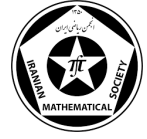
Hossein Aminikhah

Department of Applied Mathematics and Computer Science, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

email address: aminikhah@guilan.ac.ir



University of Guilan



A meshless method for solving the fractional Schnakenberg model used to biological pattern formation

Marzieh Raei

Abstract. This paper develops a meshless collocation technique based on the radial basis functions to simulate a famous two-dimensional reaction-diffusion system so-called Schnakenberg model with Riesz space fractional derivatives in developmental biology. The Schnakenberg system is one of the popular Turing pattern formations applied in biological pattern formation.

1. Introduction

The reaction-diffusion system has the beneficial property that causes generate pattern formation in biology. Schnakenberg proposed one of the interesting reaction-diffusion models in 1979 based on the hypothetical mechanism consisting of trimolecular autocatalytic reactions [1]. In this work, a two-dimensional reaction-diffusion Schnakenberg model by replacing the second order space derivatives with Riesz fractional derivative is specifically investigated as follows

$$\begin{cases} \frac{\partial u(x,y,t)}{\partial t} = \frac{\partial^\mu u(x,y,t)}{\partial |x|^\mu} + \frac{\partial^\mu u(x,y,t)}{\partial |y|^\mu} + \gamma(\alpha - u(x,y,t) + u^2(x,y,t)v(x,y,t)), & (x,y,t) \in \Omega \times (0,T), \\ \frac{\partial v(x,y,t)}{\partial t} = d_v \left(\frac{\partial^\nu v(x,y,t)}{\partial |x|^\nu} + \frac{\partial^\nu v(x,y,t)}{\partial |y|^\nu} \right) + \gamma(\beta - u^2(x,y,t)v(x,y,t)), & (x,y,t) \in \Omega \times (0,T), \end{cases} \quad (1.1)$$

with boundary and initial conditions

$$u(x,y,t) = 0, \quad v(x,y,t) = 0, \quad (x,y,t) \in \partial\Omega \times (0,T), \quad (1.2)$$

$$u(x,y,0) = u^0(x,y), \quad v(x,y,0) = v^0(x,y), \quad (x,y) \in \Omega, \quad (1.3)$$

where $u(x,y,t)$ and $v(x,y,t)$ are the two chemical concentrations, α and β are positive constants, γ is the positive scale parameter determined adequate to the dimensional of the computational domain, and d_v is the positive diffusion coefficient.

Moreover, $\frac{\partial^\mu u}{\partial |x|^\mu}$ and $\frac{\partial^\mu u}{\partial |y|^\mu}$ are Riesz fractional derivatives of order $1 < \mu \leq 2$ and $\frac{\partial^\nu u}{\partial |x|^\nu}$ and $\frac{\partial^\nu u}{\partial |y|^\nu}$ are Riesz fractional derivatives of order $1 < \nu \leq 2$. In general form, the Riesz fractional derivative of order $n - 1 < \sigma \leq n$ in x direction is defined on $[a, b]$ as follows

$$\frac{\partial^\sigma}{\partial |x|^\sigma} u(x,y,t) = -\frac{1}{2 \cos \frac{\pi\sigma}{2}} [{}_a D_x^\sigma u(x,y,t) + {}_x D_b^\sigma u(x,y,t)] \quad (1.4)$$

The left and right Riemann-Liouville fractional derivatives in above formula can be defined as follows

$$\begin{aligned} {}_a D_x^\sigma u(x,y,t) &= \frac{1}{\Gamma(n-\sigma)} \frac{\partial^n}{\partial x^n} \int_a^x \frac{u(\eta,y,t)}{(x-\eta)^{\sigma-n+1}} d\eta \\ {}_x D_b^\sigma u(x,y,t) &= \frac{-1}{\Gamma(n-\sigma)} \frac{\partial^n}{\partial x^n} \int_x^b \frac{u(\eta,y,t)}{(\eta-x)^{\sigma-n+1}} d\eta. \end{aligned} \quad (1.5)$$

Keywords: Schnakenberg model, biological pattern formation, Riesz fractional derivative, meshless method, radial basis functions.

AMS Mathematical Subject Classification [2010]: 35K99, 34M99.

In this work, we apply an implicit finite difference procedure for time discretization. Then in the time-independent system, to approximate the Riesz fractional derivatives, the Grünwald-Letnikov formula is employed. The numerical results verify the accuracy and efficiency of the suggested numerical method.

Riemann-Liouville fractional derivatives as follows

2. Numerical Procedure

In this section, first, an implicit finite difference formulation is suggested to discrete the problem (1.1) in time direction. For this purpose, the time interval $[0, T]$ is uniformly decomposed into M sub-intervals $\bigcup_{j=0}^{M-1} [t^j, t^{j+1}]$ where $t^j = j\tau$, $j = 0, \dots, M$ and $\tau = T/M$ is time step size. The time integer derivative can be discretized at two sequential time levels $n+1$ and n as follows

$$\frac{\partial u(\mathbf{x}, t^{n+1})}{\partial t} = \frac{u^{n+1} - u^n}{\tau} + O(\tau), \quad \frac{\partial v(\mathbf{x}, t^{n+1})}{\partial t} = \frac{v^{n+1} - v^n}{\tau} + O(\tau), \quad (2.1)$$

Then by substituting $t = t^{n+1}$ in the system (1.1) and using the relations (2.1), the following relation is obtained:

$$\begin{cases} (1 + \tau\gamma)u^{n+1} - \tau\left(\frac{\partial^\mu u^{n+1}}{\partial |x|^\mu} + \frac{\partial^\mu u^{n+1}}{\partial |y|^\mu}\right) = \tau\gamma\alpha + \tau\gamma G^{n+1} + u^n, \\ v^{n+1} - \tau d_v\left(\frac{\partial^\nu v^{n+1}}{\partial |x|^\nu} + \frac{\partial^\nu v^{n+1}}{\partial |y|^\nu}\right) = \tau\gamma\beta - \tau\gamma G^{n+1} + v^n, \end{cases} \quad (2.2)$$

where $u^n = u(x, y, t^n)$ and $v^n = v(x, y, t^n)$ and $G^{n+1} = u^2(x, y, t^{n+1})v(x, y, t^{n+1})$.

Moreover, the first-order shifted Grünwald-Letnikov operators could be applied to approximate the left and right Riemann-Liouville fractional derivatives (1.5). Therefore, the space interval $[a, b]$ should uniformly decompose into N sub-intervals $\bigcup_{i=0}^N [x_i, x_{i+1}]$ where $x_i = a + ih_x$, $i = 0, \dots, N$ and $h_x = (b - a)/N$. Thus by substituting $x = x_i$ in the time-independent relations (2.2), we have [2]

$$\begin{aligned} {}_a D_x^\sigma u(x_i, y, t^{n+1}) &= \frac{1}{h_x^\sigma} \sum_{k=0}^{i+1} \omega_k^\sigma u(x_{i-k+1}, y, t^{n+1}) + O(h_x), \\ {}_x D_b^\sigma u(x_i, y, t^{n+1}) &= \frac{1}{h_x^\sigma} \sum_{k=0}^{N-i+1} \omega_k^\sigma u(x_{i+k-1}, y, t^{n+1}) + O(h_x). \end{aligned} \quad (2.3)$$

Furthermore, the space Riesz fractional could approximate by using the discretized formulae (2.3) in x and y directions for fractional orders μ and ν . Therefore, by placing the shifted Grünwald-Letnikov approximation in the time-independent relations (2.2), the resulting finite difference equations are obtained as follows

$$\begin{cases} (1 + \tau\gamma)u_{i,j}^{n+1} - \tau\left(C_{x\mu}\left(\sum_{k=0}^{i+1} \omega_k^\mu u_{i-k+1,j}^{n+1} + \sum_{k=0}^{N-i+1} \omega_k^\mu u_{i+k-1,j}^{n+1}\right) + C_{y\mu}\left(\sum_{k=0}^{i+1} \omega_k^\mu u_{j,i-k+1}^{n+1} + \sum_{k=0}^{N-i+1} \omega_k^\mu u_{j,i+k-1}^{n+1}\right)\right) = \tau\gamma\alpha + \tau\gamma G^{n+1} + u_{i,j}^n, & j = 1, 2, \dots, N, \\ v_{i,j}^{n+1} - \tau d_v\left(C_{x\nu}\left(\sum_{k=0}^{i+1} \omega_k^\nu u_{i-k+1,j}^{n+1} + \sum_{k=0}^{N-i+1} \omega_k^\nu u_{i+k-1,j}^{n+1}\right) + C_{y\nu}\left(\sum_{k=0}^{i+1} \omega_k^\nu u_{j,i-k+1}^{n+1} + \sum_{k=0}^{N-i+1} \omega_k^\nu u_{j,i+k-1}^{n+1}\right)\right) = \tau\gamma\beta - \tau\gamma G^{n+1} + v_{i,j}^n, & j = 1, 2, \dots, N, \end{cases} \quad (2.4)$$

where $u_{i,j}^{n+1} = u^{n+1}(x_i, y_j)$, $C_{x\mu} = \frac{1}{2h_x^\mu \cos(\frac{\pi\mu}{2})}$, $C_{y\mu} = \frac{1}{2h_y^\mu \cos(\frac{\pi\mu}{2})}$, $C_{x\nu} = \frac{1}{2h_x^\nu \cos(\frac{\pi\nu}{2})}$ and $C_{y\nu} = \frac{1}{2h_y^\nu \cos(\frac{\pi\nu}{2})}$.

Now, we could perform the collocation meshless method based on radial basis function on the discretized relations (2.6) to approximate the numerical solutions $u(x, y, t)$ and $v(x, y, t)$. Therefore, we briefly review the meshless collocation method based on RBFs. In this method, the numerical solution of governing equations can be approximated by a linear combination of the RBFs as follows

linear combination of the particular solutions of the given RBFs as follow:

$$u(\mathbf{x}) \simeq \hat{u}(\mathbf{x}) = \sum_{i=1}^n \lambda_i \Phi_i(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (2.5)$$

where $\Phi_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i\|$ and $\{\lambda_i\}$ are the undetermined coefficients. Also, in the current work, Φ is thin plate spline (TPS) radial basis function. By applying the collocation method for discretized equations (2.6) and boundary conditions (1.2), the following linear system of equations is obtained

$$\left\{ \begin{aligned} & \left[(1 + \tau\gamma) \sum_{l=1}^N \Phi_l(x_i, y_j) - \tau \left(C_{x\mu} \left(\sum_{k=0}^{i+1} \omega_k^\mu \sum_{l=1}^N \Phi_l(x_{i-k+1}, y_j) + \sum_{k=0}^{N-i+1} \omega_k^\mu \sum_{l=1}^N \Phi_l(x_{i+k-1}, y_j) \right) \right. \right. \\ & \quad \left. \left. + C_{y\mu} \left(\sum_{k=0}^{i+1} \omega_k^\mu \sum_{l=1}^N \Phi_l(x_j, y_{i-k+1}) + \sum_{k=0}^{N-i+1} \omega_k^\mu \sum_{l=1}^N \Phi_l(x_j, y_{i+k-1}) \right) \right) \right] \lambda_l^u \\ & = \tau\gamma\alpha + \tau\gamma G^{n+1} + u_{i,j}^n, \quad i, j = 1, 2, \dots, N_i, \\ & \left[\sum_{l=1}^N \Phi_l(x_i, y_j)_{i,j} - \tau d_v \left(C_{x\nu} \left(\sum_{k=0}^{i+1} \omega_k^\nu \sum_{l=1}^N \Phi_l(x_{i-k+1}, y_j) + \sum_{k=0}^{N-i+1} \omega_k^\nu \sum_{l=1}^N \Phi_l(x_{i+k-1}, y_j) \right) \right. \right. \\ & \quad \left. \left. + C_{y\nu} \left(\sum_{k=0}^{i+1} \omega_k^\nu \sum_{l=1}^N \Phi_l(x_j, y_{i-k+1}) + \sum_{k=0}^{N-i+1} \omega_k^\nu \sum_{l=1}^N \Phi_l(x_j, y_{i+k-1}) \right) \right) \right] \lambda_l^v \\ & = \tau\gamma\beta - \tau\gamma G^{n+1} + v_{i,j}^n, \quad i, j = 1, 2, \dots, N_i, \\ & \sum_{l=1}^N \Phi_l(x_i, y_j)_{i,j} \lambda_l^u = 0, \quad i, j = N_i + 1, 2, \dots, N, \\ & \sum_{l=1}^N \Phi_l(x_i, y_j)_{i,j} \lambda_l^v = 0, \quad i, j = N_i + 1, 2, \dots, N. \end{aligned} \right. \quad (2.6)$$

Therefore, the numerical solutions could achieve by solving the $2N \times 2N$ linear system of equations.

3. Numerical results

As a benchmark problem, the Schnakenberg Model (1.1) on unit square $\Omega = (0, 1)^2$ with Dirichlet boundary conditions (1.2) for both u and v variables. Moreover, The initial conditions are considered by taking small random perturbations around the steady-state (u_s, v_s) given by $u_s = \alpha + \beta$ and $v_s = \frac{\beta}{(\alpha + \beta)^2}$. To investigate the convergence of the presented numerical method, the following error estimation is considered:

$$E_u^k = \|u_\tau - u_{2\tau}\|_\infty.$$

The CPU time and error estimation concerning time step size τ at $T = 3$ by letting $\alpha = 0.1$, $\beta = 0.9$, $\gamma = 660$, and $d_v = 8.6676$ for the space fractional derivative orders $\mu = \nu = 1.5$ is presented in Figure 1. The time evolution of the activator concentration u with fix values of the parameters for different time levels are demonstrated in Figure 2.

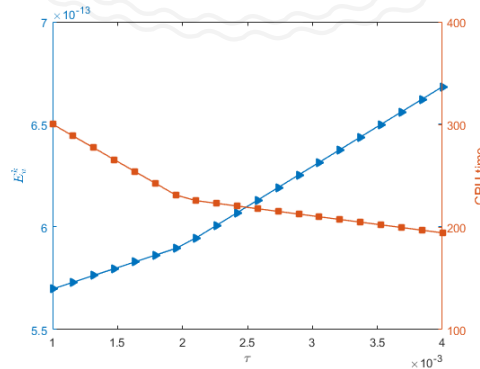


Figure 1: Error estimation and CPU time with respect to time step size τ .

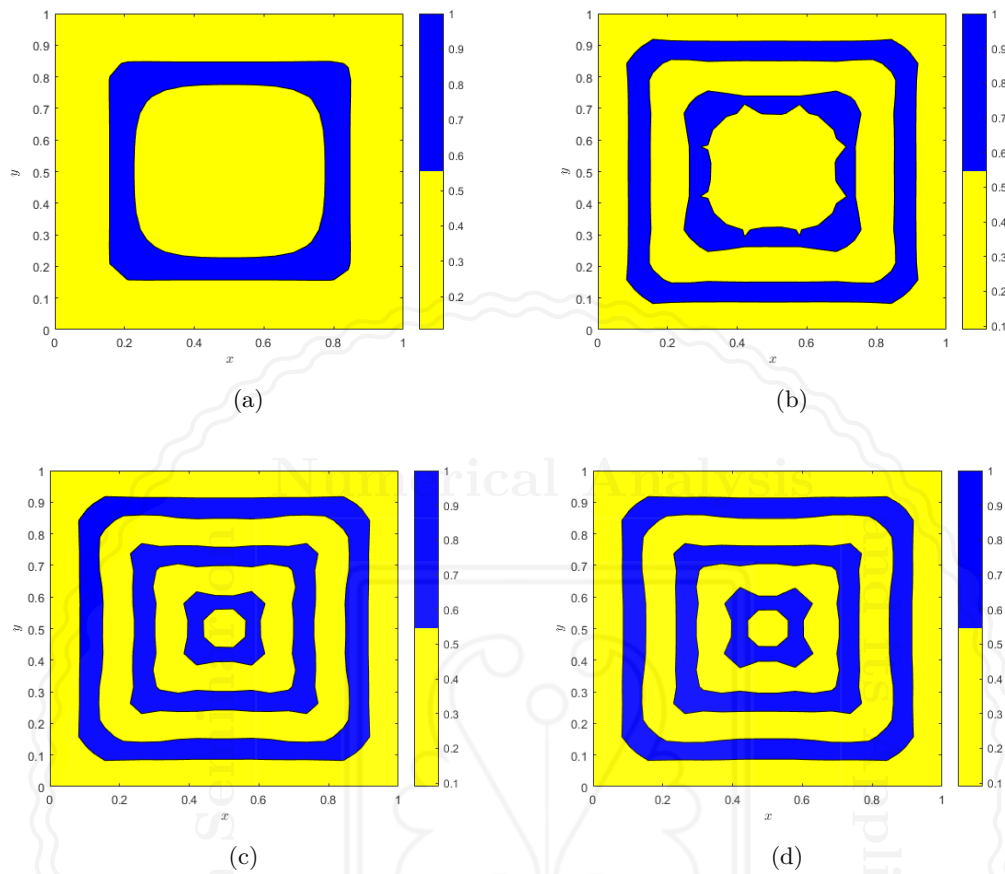


Figure 2: The contour plot of activator concentration u at different time levels (a) $t = 0$, (b) $T = 0.05$, (c) $T = 0.15$, and (d) $T = 0.25$.

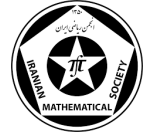
References

- [1] J. Schnakenberg, Simple chemical reaction systems with limit cycle behaviour, *Journal of Theoretical Biology*, 81 (1979) 389–400.
- [2] C. Tadjeran, M.M. Meerschaert, H.P. Scheffler, A second-order accurate numerical approximation for the fractional diffusion equation. *Journal of Computational Physics*, 213 (2006) 205–213.

Marzieh Raei
 Department of Mathematics, Malek Ashtar University of Technology, Esfahan, Iran
 email address: marzie.raei@gmail.com



University of Guilan



The direct meshless local Petrove Galerkin method for solving equations with distributed order derivatives

Ali Habibirad, Hadis Azin, Esmail Hesameddini

Abstract. In this work, a hybrid method is developed based on the generalized Moving least square (GMLS) approximation and finite difference method for solving this distributed order equation. To show the accuracy and efficiency of the proposed method we discuss an example.

1. Introduction

Equations with distributed order derivatives are a branch of fractional calculus that has recently expanded due to its applications. These equations have many applications for modeling complex systems. To see more properties and numerical solutions methods about equations with distributed order derivatives see [1] and references therein. In this work, we study the following distributed order time-fractional equation

$$\int_0^1 \varpi(\alpha) D_t^\alpha u(\mathbf{x}, t) d\alpha = \Delta u(\mathbf{x}, t) + u(\mathbf{x}, t) + f(\mathbf{x}, t), \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^2, \quad 0 \leq t \leq T, \quad (1.1)$$

subject to the initial and boundary conditions

$$\begin{cases} u(\mathbf{x}, 0) = u_0(\mathbf{x}), \\ u(\mathbf{x}, t) = \Psi(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega. \end{cases} \quad (1.2)$$

In which $D_t^\alpha u(\mathbf{x}, t)$ shows the Caputo fractional derivative of $u(\mathbf{x}, t)$ which is given by

$$D_t^\alpha u(\mathbf{x}, t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial u(\mathbf{x}, r)}{\partial r} \frac{dr}{(t-r)^\alpha}, \quad 0 < \alpha < 1, \quad (1.3)$$

Also, the weight function $\varpi(\alpha)$ has the following conditions

$$\varpi(\alpha) > 0, \quad 0 < \int_0^1 \varpi(\alpha) d\alpha < \infty. \quad (1.4)$$

One of the numerical methods that has been much considered by researchers in recent years is meshless methods. Due to their high flexibility, these methods have the ability to solve problems in regular and irregular domains. And unlike mesh-based methods, they often use nodes instead of meshing to solve problems. This saves less time for the procedure. The beginning of these

Keywords: Distributed order fractional derivative, Time-fractional reaction diffusion equation, Caputo fractional derivative.

AMS Mathematical Subject Classification [2010]: 65M12, 65M60, 34A45.

methods can be considered as the smooth particle hydrodynamics method. One of the meshless schemes is the meshless local Petrov-Galerkin method (MLPG), which was invented in 1998 by Atluri and Zhu [2]. The classic basis of this method is the moving least-squares approximation. By changing the test function in this method, six different types are known which are named MLPG1, MLPG2, MLPG3, MLPG4, MLPG5, and MLPG6. In these methods, numerical calculations are traditionally performed on MLS shape functions and their derivatives. In this case, the complexity of the shape functions and the lack of a closed-form of them require a large number of points (and shape functions) for them to obtain accurate results. Therefore, MLS subroutines must be recalled frequently, leading to high computational costs. To overcome this problem, Mirzaei and Schaback [3] used the generalized MLS method and thus invented the direct meshless local Petrov-Galerkin method (DMLPG) method. In the present paper, we use the DMLPG approach to discretization the Eq (1.1) in space variables. At first, we introduce the GMLS method in section 2, the proposed method is discussed in sec 3, to show the ability of the method one example is given in section 4, and finally, a short conclusion is given in section 5.

2. The GMLS method

Suppose function $u(x)$ is defined in the global bounded domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$. Also assume $\{x_i\}_{i=1}^n$ are randomly distributed in the global domain and Ω_k is a subdomain for point x_k . In the classical MLS method, the following expression $u^h(x)$ is used to approximate function $u(x)$ as

$$u^h(x) = \mathbf{p}^T(x)\mathbf{a}(x), \quad \forall x \in \Omega, \quad (2.1)$$

in which $\mathbf{p}^T(x) = [p_1(x), p_2(x), \dots, p_m(x)]$ is a vector matrix of complete monomial basis. As stated in [4], if we consider the sub-domains in the form of a circle and also assume that only N points are around the point x , in other words, the weight function is not zero in these N points, we get the following expression:

$$u^h(x) = \mathbf{a}^T(x)\mathbf{u} = \sum_{j=1}^N a_j(x)u(x_j), \quad (2.2)$$

in which $\mathbf{a}^T(x)$ is the shape functions vector of MLS that defined as follows

$$\mathbf{a}^T(x) = \mathbf{p}^T(x) (P^T W P)^{-1} P^T W. \quad (2.3)$$

Also, two matrices P and W are defined as

$$P = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \dots & p_m(x_1) \\ p_1(x_2) & p_2(x_2) & \dots & p_m(x_2) \\ \dots & \dots & \dots & \dots \\ p_1(x_N) & p_2(x_N) & \dots & p_m(x_N) \end{bmatrix}_{N \times m}, \quad W = \begin{bmatrix} w_1(x) & 0 & \dots & 0 \\ 0 & w_2(x) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & w_N(x) \end{bmatrix}_{N \times N}. \quad (2.4)$$

Let $\lambda(u)$ be a sufficiently smooth function. In the MLS scheme this function approximate by the following form of shape functions

$$\lambda(u) \cong \lambda(\hat{u}) = \sum_{j=1}^n \lambda(a_j)u(x_j). \quad (2.5)$$

Given this relationship, it is obvious that the act of $\lambda(u)$ goes back to the shape functions a_j and requires a lot of calculations to calculate the proper accuracy, and if function $\lambda(u)$ is complex, this

becomes more difficult. In GMLS approach, function $\lambda(u)$ is approximated directly by the nodal values $\{u(\mathbf{x}_i)\}_{i=1}^n$ without using the shape functions. This approximation is as follows

$$\lambda(\hat{u}) = \sum_{j=1}^n a_j(\lambda) u(\mathbf{x}_j). \quad (2.6)$$

The GMLS estimation $\lambda(\hat{u})$ can be calculated as $\lambda(\hat{u}) = \lambda(p^*)$ where $p^* \in \mathbb{P} = \text{span}\{p_1, p_2, \dots, p_m\}$, is the minimizer of the weighted least-squares

$$\sum_{j=1}^n w_j(\mathbf{x}) [p(\mathbf{x}_j) - u(\mathbf{x}_j)]^2, \quad (2.7)$$

across all $p \in \mathbb{P}$. The optimal solution $a^*(\lambda) \in \mathbb{R}^n$ can be demonstrated as

$$a^*(\lambda)^T = \lambda(\mathbf{p}^T) (P^T W P)^{-1} P^T W, \quad (2.8)$$

in which the matrices P and W are introduce in (2.4) and

$$\lambda(\mathbf{p}^T) = [\lambda(\mathbf{p}_1), \dots, \lambda(\mathbf{p}_m)] \in \mathbb{R}^m. \quad (2.9)$$

In this work, we use the following weight function as

$$w_i(\mathbf{x}) = \begin{cases} 1 - 6s_i^2 + 8s_i^3 - 3s_i^4, & s_i \leq 1, \\ 0, & s_i > 1, \end{cases} \quad (2.10)$$

in which $s_i = \frac{\|\mathbf{x} - \mathbf{x}_i\|}{r_i}$ and r_i is the size of support in the weight function. In the GMLS the functional λ doesn't act on shape functions, it indicates that integrations portray just over polynomials.

3. The proposed method

To construct the finite difference technique for the distributed order fractional in Eq. (1.1), let $\tau = \frac{T}{n_t}$ be the step size of time and define $t_n = n\tau$, $n = 0, 1, 2, \dots, n_t$. The Caputo fractional derivative $D_t^{\alpha_j} u(\mathbf{x}, t)$ in Eq. (1.1) is approximated by the $L2 - 1_\sigma$ method [1] as follows ($\sigma = 0.5$)

$$D_t^{\alpha_j} u(\mathbf{x}, t_{n+\sigma}) = \sum_{l=0}^n \frac{\tau^{-\alpha_j}}{\Gamma(2 - \alpha_j)} c_l^{(n+1, \alpha_j)} (u(\mathbf{x}, t_{n-l+1}) - u(\mathbf{x}, t_{n-l})) + O(\tau^{3-\alpha_{\max}}), \quad (3.1)$$

where $c_l^{(n+1, \alpha_j)}$ is introduced in [1]. Also, we use

$$u(\mathbf{x}, t) = \frac{1}{2} (u(\mathbf{x}, t_{n+1}) + u(\mathbf{x}, t_n)). \quad (3.2)$$

Substituting (3.1) and (3.2) in (1.1), results in

$$\sum_{l=0}^n d_l^{(n+1)} \left(\tilde{u}(\mathbf{x}, t_{n-l+1}) - \tilde{u}(\mathbf{x}, t_{n-l}) \right) - \frac{1}{2} \Delta (\tilde{u}_{n+1} + \tilde{u}_n) + \frac{1}{2} (\tilde{u}_{n+1} + \tilde{u}_n) = f_{n+\frac{1}{2}}, \quad (3.3)$$

where $d_l^{(n+1)} = \sum_{j=1}^M \frac{\tau^{-\alpha_j}}{2\Gamma(2 - \alpha_j)} \tilde{\omega}_j \varpi(\alpha_j) c_l^{(n+1, \alpha_j)}$ and $\tilde{u}_n = \tilde{u}(\mathbf{x}, t_n)$ is an approximation of the exact solution $u(\mathbf{x}, t_n)$. Now let $\{\mathbf{x}_i\}_{i=1}^n$ be the arbitrary nodes and scattered in global domain Ω . Also

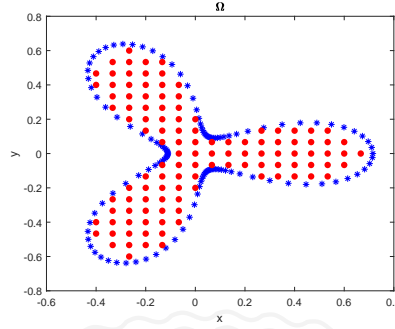


Figure 1: Global domain.

we assume a subdomain around every points. These subdomains can be any geometric shape and they usually assume a circle in the two-dimensional case. DMLPG2 is used to the strong form (3.3) directly by using the following approximations

$$\lambda_{1,k} = \tilde{u}(\mathbf{x}_k) \cong \hat{\lambda}_{1,k}(\tilde{u}) = \sum_{j=1}^n a_{1,j} \tilde{u}(\mathbf{x}_j) \quad \lambda_{2,k} = \Delta \tilde{u}(\mathbf{x}_k) \cong \hat{\lambda}_{2,k}(\tilde{u}) = \sum_{j=1}^n a_{2,j} \tilde{u}(\mathbf{x}_j) \quad (3.4)$$

applying Eq (2.8) we get

$$(a_i, :)^T = \lambda_{i,k}(\mathbf{p}^T) (P^T W P)^{-1} P^T W, \quad i = 1, 2, \quad (3.5)$$

in which

$$\lambda_{1,k}(\mathbf{p}) = [p_1(\mathbf{x}_k), \dots, p_m(\mathbf{x}_k)] \quad \lambda_{2,k}(\mathbf{p}) = [\Delta p_1(\mathbf{x}_k), \dots, \Delta p_m(\mathbf{x}_k)]. \quad (3.6)$$

To obtain the Dirichlet boundary conditions we used the relation (3.4).

4. Numerical results

In this section, a numerical example is studied to examine the accuracy of the presented method. To compute the accuracy of results are used by absolute error as

$$L_\infty = \max_{0 \leq j \leq n} |\tilde{u}_j - u_j|, \quad (4.1)$$

$$L_2 = \left(\int_{\Omega} (u(\mathbf{x}, T) - \tilde{u}_{n_t}(\mathbf{x}, T))^2 d\mathbf{x} \right)^{\frac{1}{2}}, \quad RMS = \sqrt{\frac{\sum_{i=1}^M \left(u(\mathbf{x}, T) - \tilde{u}_{n_t}(\mathbf{x}, T) \right)^2}{M}}.$$

where u and \tilde{u} are the exact solution and proposed numerical solution, respectively.

Example 4.1. We will check the following model

$$\int_0^1 \varpi(\alpha) D_t^\alpha u(\mathbf{x}, t) d\alpha - \Delta u(\mathbf{x}, t) + u(\mathbf{x}, t) = f(\mathbf{x}, t), \quad \mathbf{x} = (x, y) \in \Omega. \quad (4.2)$$

Here, the analytical solution is $u(x, y, t) = t^2 \sin(x) \sin(y)$ and $\varpi(\alpha) = \Gamma(3 - \alpha)$. We extract the initial condition, Dirichlet boundary conditions and the source term f from the exact solution. The

Table 1: The L_2 , L_∞ and RMS errors and related convergence orders for Example 4.1 over Ω with different final times.

T	L_2	L_∞	RMS
1	$8.8730E-06$	$1.0072E-06$	$4.1015E-07$
2	$3.9574E-06$	$4.4920E-07$	$1.8243E-07$
3	$2.2308E-06$	$2.5322E-07$	$1.0312E-07$
4	$1.4299E-06$	$1.6201E-07$	$6.6097E-08$
5	$9.9010E-07$	$1.1204E-07$	$4.5901E-08$

global domain is irregular shape Ω shown in Figure 1. The boundary of a global domain has the following parametric formula

$$\left\{ (x, y) \in \mathbb{R}^2 : x = r \cos(\theta), y = r \sin(\theta), \quad \theta \in [0, 2\pi], \quad r = 0.5 \sqrt{(\cos(3\theta) + \sqrt{1.1 - \sin(3\theta)^2})} \right\}. \quad (4.3)$$

The number of interiors and boundary points are 342 and 126 respectively. Table 1 demonstrates the numerical results for Example 4.1 have good accuracy in comparison with the analytical solution.

5. Conclusion

In this paper, a local collocation meshless scheme was used for the numerical solution of distributed order time-fractional reaction-diffusion equation. The GMLS approximation and $L2 - 1_\sigma$ method with the Gauss-Legendre numerical integration were employed to deal with this problem. One example was studied and it showed the accuracy and capability of the presented technique for solving such problems.

References

- [1] S. Guo, C. Yaping, M. Liquan, S. Yining, Finite difference/generalized Hermite spectral method for the distributed-order time-fractional reaction-diffusion equation on multi-dimensional unbounded domains, *Computers and Mathematics with Applications*, 93 (2021) 1–19.
- [2] S.N. Atluri, Z. Tulong, A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics, *Computational mechanics*, 22 (1998) 117–127.
- [3] D. Mirzaei, R. Schaback, Direct meshless local Petrov-Galerkin (DMLPG) method: a generalized MLS approximation. *Applied Numerical Mathematics*, 68 (2013) 73–82.
- [4] M. Ramezani, M. Mojtabaei, D. Mirzaei, DMLPG solution of the fractional advection-diffusion problem, *Engineering Analysis with Boundary Elements*, 59 (2015) 36–42 .

Ali Habibirad

Department of Mathematics, Shiraz University of Technology, Shiraz, Iran.

email address: a.habibirad@sutech.ac.ir

Hadis Azin

Department of Mathematics, University of Hormozgan, Bandar Abbas, Iran

email address: H.Azin1370@gmail.com

Esmail Hesameddini

Department of Mathematics, Shiraz University of Technology, Shiraz, Iran.

email address: hesameddini@sutech.ac.ir



University of Guilan



Exponential basis functions with three shape parameters

Jamshid Saeidian, Bahareh Nouri

Abstract. We study a modification of the model presented in [3] through adding a suitable shape parameter. The new curves have most of the features of the cubic Bézier curves with the advantage that the new parameter enables us to adjust the shape of the corresponding Bézier-like curve. We prove that the new curves have the monotony preservation property.

1. Introduction

As an important geometric modeling tool, the Bézier curve has been widely used in Computer Aided Geometric Design (CAGD) and Computer Graphics (CG). For a set of control points, a Bézier curve is defined based on Bernstein polynomials [2]. Once the control points are fixed, the shape of the Bézier curve cannot be changed. In order to overcome this deficiency, many researchers have tried to add shape parameters to the basis functions to create new curves whose structures are similar to the Bézier curve. In [3], Zhu and Han presented a new class of $\lambda\mu$ -Bernstein basis functions with two shape parameters. The $\lambda\mu$ -curves constructed by these basis have many basic properties of the cubic Bézier curves, besides by altering shape parameters the curve travels from cubic Bézier curve to control polygon.

In the present study, we modify the $\lambda\mu$ -Bernstein basis functions by adding a new shape parameter, which enables the corresponding family of $\lambda\mu$ -curve to travel between the control polygon and the straight line joining the first and last control points.

2. New Basis Function

Definition 2.1. For shape parameters $\lambda, \mu \in [0, +\infty]$ and $\nu \in [0, 1]$, the proposed blending Functions $b_i(t; \lambda, \mu, \nu)$, are defined for $t \in [0, 1]$ as

$$\begin{aligned} b_0(t; \lambda, \nu) &= (1 - \nu)(1 - t)^3 e^{-\lambda t} + \nu(1 - t)^2(1 + 2t) \\ b_1(t; \lambda, \nu) &= -(1 - \nu)(1 - t)^3 e^{-\lambda t} + (1 - \nu)(1 - t)^2(1 + 2t) \\ b_2(t; \mu, \nu) &= -(1 - \nu)t^3 e^{-\mu(1-t)} + (1 - \nu)t^2(3 - 2t) \\ b_3(t; \mu, \nu) &= (1 - \nu)t^3 e^{-\mu(1-t)} + \nu t^2(3 - 2t) \end{aligned} \quad (2.1)$$

The graphical behavior of proposed basis functions defined in Eq.(2.1) and the effect of shape parameter can be observed in Figure 1. For $\lambda = \mu = \nu = 0$, the blending Functions are the classical cubic Bernstein basis functions, see [2].

Keywords: Bernstein basis, Blending functions, Bézier curve, Shape parameter.

AMS Mathematical Subject Classification [2010]: 65D17, 65D18.

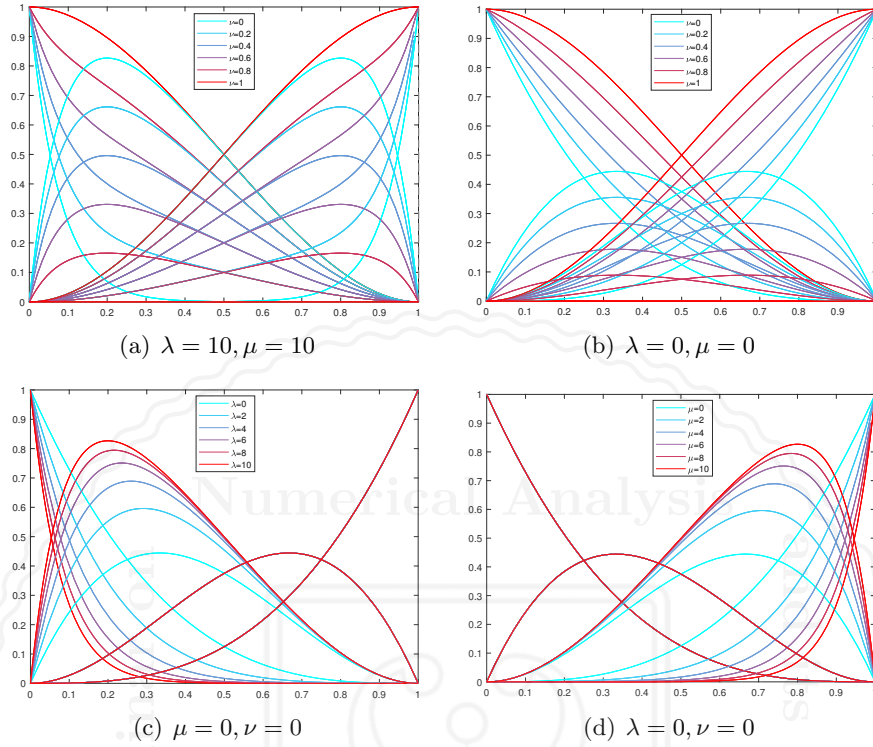


Figure 1: The plot of $\lambda\mu\nu$ - basis functions for various values of parameters

Theorem 2.2. For real numbers $\lambda, \mu \in [0, +\infty]$ and $\nu \in [0, 1]$, the blending functions defined in Definition 2.1 have the following properties:

- (a) *Linear independence:* The blending functions $b_i(t; \lambda, \mu, \nu)$ $i = 0, 1, 2, 3$ are linearly independent.
- (b) *Nonnegativity:* $b_i(t; \lambda, \mu, \nu) \geq 0$ ($i = 0, 1, 2, 3$).
- (c) *Partition of unity:* $\sum_{i=0}^3 b_i(t; \lambda, \mu, \nu) = 1$.
- (d) *Symmetry:* $b_i(t; \lambda, \mu, \nu) = b_{3-i}(1-t; \lambda, \mu, \nu)$ ($i = 0, 1, 2, 3$).
- (e) *Monotonicity:* For a given parameter $t \in (0, 1)$, $b_0(t; \lambda, \nu)$ and $b_3(t; \lambda, \nu)$ are monotonically decreasing for shape parameters λ and μ , respectively; $b_1(t; \lambda, \nu)$ and $b_2(t; \lambda, \nu)$ are monotonically increasing for shape parameters λ and μ , respectively. $b_1(t; \lambda, \nu)$ and $b_2(t; \lambda, \nu)$ are monotonically decreasing for shape parameters ν , $b_0(t; \lambda, \nu)$ and $b_3(t; \lambda, \nu)$ are monotonically increasing for shape parameters ν .
- (f) *Properties at the endpoints:*

$$b_i(0; \lambda, \mu, \nu) = \begin{cases} 1, & i = 0, \\ 0, & i \neq 0, \end{cases} \quad b_i(1; \lambda, \mu, \nu) = \begin{cases} 1, & i = 3, \\ 0, & i \neq 3, \end{cases}$$

$$b'_i(0; \lambda, \mu, \nu) = \begin{cases} -(1-\nu)(3+\lambda), & i = 0, \\ (1-\nu)(3+\lambda), & i = 1, \\ 0, & i = 2, 3, \end{cases} \quad b'_i(1; \lambda, \mu, \nu) = \begin{cases} (1-\nu)(3+\mu), & i = 3, \\ -(1-\nu)(3+\mu), & i = 2, \\ 0, & i = 0, 1, \end{cases}$$

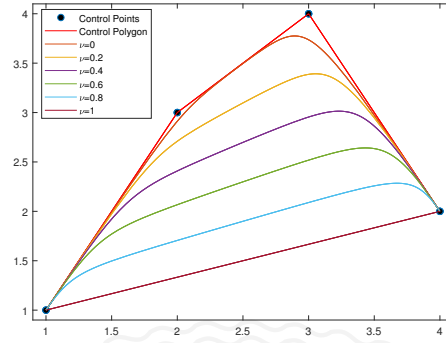


Figure 2: The shape of curves with $\lambda = 10, \mu = 10$ and different shape parameters ν .

Definition 2.3. Given control points $P_i (i = 0, 1, 2, 3)$ in \mathbb{R}^2 , for $\lambda, \mu \in [0, +\infty]$ and $\nu \in [0, 1]$,

$$r(t) = \sum_{i=0}^3 b_i(t; \lambda, \mu, \nu) P_i, \quad (2.2)$$

is called $\lambda\mu\nu$ -curve, where $b_i(t; \lambda, \mu, \nu) (i = 0, 1, 2, 3)$ are the blending functions expressed in (2.1).

Figure 2 shows the $\lambda\mu\nu$ -curve and the effect on the curves by altering the values of the shape parameters at the same time under keeping the control points unchanged.

For $\lambda = \mu = \nu = 0$, the $\lambda\mu\nu$ -curve is the classical cubic Bernstein-Bézier curve, see [2].

Definition 2.4. [1] A system of functions (u_0, \dots, u_n) is monotonicity preserving if for any $\alpha_0 \leq \alpha_1 \leq \dots \leq \alpha_n$ in \mathbb{R} , the function $\sum_{i=0}^n \alpha_i u_i$ is increasing.

Proposition 2.5. [1] Let (u_0, \dots, u_n) be a system of functions defined on an interval $[a, b]$. Let $v_i := \sum_{j=i}^n u_j$ for $i \in \{0, 1, \dots, n\}$. Then (u_0, \dots, u_n) is monotonicity preserving if and only if v_0 is a constant function and the functions v_i are increasing for $i = 1, \dots, n$.

Theorem 2.6. The blending functions $(b_0(t; \lambda, \nu), b_1(t; \lambda, \nu), b_2(t; \mu, \nu), b_3(t; \mu, \nu))$ defined in (2.1) are monotonicity preserving.

From Theorem 2.2, the $\lambda\mu\nu$ -curve has the following properties,

(a) Geometric property at the endpoints:

$$\begin{aligned} r(0) &= P_0, \quad r(1) = P_n, \\ r'(0) &= (1 - \nu)(3 + \lambda)(P_1 - P_0), \quad r'(1) = (1 - \nu)(3 + \lambda)(P_3 - P_2). \end{aligned}$$

(b) Symmetry: From the symmetry of the $\lambda\mu\nu$ -basis functions, one has

$$r(t; P_0, P_1, P_2, P_3) = r(1 - t; P_3, P_2, P_1, P_0).$$

(c) Geometric invariance: Because $r_n(t)$ is an affine combination of the control points, the shape of the $\lambda\mu\nu$ -curve is independent of the choice of coordinate system.

$$\begin{aligned} r(t; P_0 + Q, P_1 + Q, P_2 + Q, P_3 + Q) &= r(t; P_0, P_1, P_2, P_3) + Q, \\ r(t; MP_0, MP_1, MP_2, MP_3) &= Mr(t; P_0, P_1, P_2, P_3). \end{aligned}$$

(d) Convex hull property: Because the $\lambda\mu\nu$ -basis functions are nonnegative and sum to one, the $\lambda\mu\nu$ -curve must lie inside the convex hull of its control polygon.

References

- [1] J.M. Carnicer, M. Garcia-Esnaola, J.M. Peña, Convexity of rational curves and total positivity, *Journal of Computational and Applied Mathematics*, 71 (1996) 365–382.
- [2] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, Academic Press. New York, 1999.
- [3] Y. Zhu, X. Han, Curves and surfaces construction based on new basis with exponential functions, *Acta Applicandae Mathematicae*, 129 (2014) 183–203.

Jamshid Saeidian

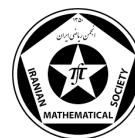
Faculty of Mathematical Sciences and Computer, Kharazmi University, Tehran, Iran

email address: j.saeidian@khu.ac.ir

Bahareh Nouri

Faculty of Mathematical Sciences and Computer, Kharazmi University, Tehran, Iran.

email address: std.nouri411@khu.ac.ir



Analysis of COVID-19 pandemic in Iran and Guilan based on SIRU model

Hossein Taheri, Nasrin Eghbali, Masoumeh Pourabd

Abstract. In this paper, we investigate the COVID-19 pandemic in Iran from a mathematical modeling perspective. By improving the well-known susceptible infected recovered (SIR) family of compartmental models and adding unreported cases obtain a local model for Iran. Also, Our proposed model was able to predict well based on the data reported. Moreover, Our calculations were run using MATLAB software. Since we only want infected cases, we have refused to add other classes that there are can be.

1. Introduction

A novel corona-virus (nCoV), has been causing the deadliest pandemic from the last months of 2019 up to now, defined as the corona-virus disease 2019 (COVID-19) by the World Health Organization (WHO). It is also known as severe acute respiratory syndrome 2 (SARS-CoV-2). The first cases that occurred in early December 2019, had been reported in Wuhan, China. To date, many countries and regions have been locked-down and applied strict social distancing measures to stop the virus propagation. From a strategic and health care management perspective, the propag

ation pattern of the disease and the prediction of its spread over time are of great importance, to save lives and minimize the s

ocial and economic consequences of the disease. There are five major types of models in the literature that can help us understand the transmissibility of the SARS-CoV-2 from its natural reservoirs to humans. In references [1–5], the authors gave the following models for Covid -19. But, finally by using reference [5], we believe that, the SIR model is the best mathematical model for the prediction of pandemic Covid-19. A first tentative mathematical model of this pandemic (see [2]), based on the Be-CoDiS model. For information of this model, see [3, 4].

2. The Model Description

Recently researchers identified the behavioral effects of the pandemic threat of Covid-19 which has not been described by a

ny of the existing analytically models [6]. In this scenario, an infected individual instead of being removed(recovery) contributes to the infection spreading upon the reinfection attempt. In this paper, we used a

new generalized SIR model, Susceptible-Infected(Reported and Unreported)- Susceptible- Reinfected (SIRUSI)to describe a

Keywords: Corona-virus pandemic globally; Mathematical modeling; SIRU-model; Parameter identification.

AMS Mathematical Subject Classification [2010]: Primary 26A33; Secondary 34D10, 45N05.

nd predict the epidemics of Covid-19 in Iran and the province of Guilan, By this model, we can find the daily number of unreported cases and we can estimate the number of symptomatic unreported infectious individuals too. The model variables and parameters are given by Table 1 Our system

Table 1: The model variables and parameters description

parameters and variables	Description
$S(t)$	Number of susceptible populations at time t
$I(t)$	Number of asymptomatic infectious at time t
$R(t)$	Number of reported infected cases at time t
$U(t)$	Number of unreported infected cases
t_0	Time at which the epidemic started
$S_0 = S(t_0)$	Number of susceptible at time t_0
$I_0 = I(t_0)$	Number of asymptomatic infectious at time t_0
$U_0 = U(t_0)$	Number of unreported cases at time t_0
τ	Contact transmission rate
$1/\nu$	Average time during which asymptomatic infectious are asymptomatic
$\nu_1 = f\nu$	Rate at which asymptomatic infectious become reported symptomatic
$\nu_2 = (1 - f)\nu$	Rate at which asymptomatic infectious become unreported symptomatic
$1/\mu$	Average time symptomatic infectious have symptoms

of equations was formulated based on the epidemic model (Fig.1) for COvid-19.

On the basis of the above-stated assumptions and the flow diagram of COVID-19 shown in Figure.1,

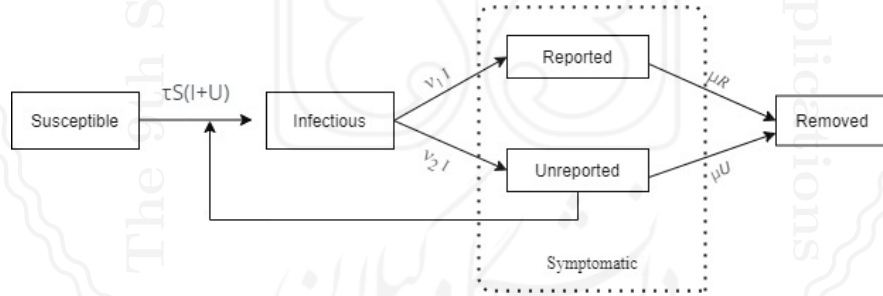


Figure 1: Modeling diagram for the transmission of Covid-19

we formulated a dynamical system consisting of four first-order differential equations shown below:

$$S'(t) = -\tau S(t)[I(t) + U(t)], \quad (2.1)$$

$$I'(t) = \tau S(t)[I(t) + U(t)] - \nu I(t),$$

$$R'(t) = \nu_1 I(t) - \mu R(t),$$

$$U'(t) = \nu_2 I(t) - \mu U(t).$$

Where the parameter f , the fraction of asymptomatic infections that become reported symptomatic infectious, this parameter plays an important role in our study. In short, we compare output data by solving the model and real data. Then we evaluate f such that this difference between real data and output data of the system should be minimal.

We consider these equations with initial conditions $S(t_0), I(t_0), U(t_0)$ and $R(t_0) = 0$, Also we note

that $R(t_0) = 0$. It should be noted that due to the nature of this disease hardly anyone was immune at the beginning of the epidemic, therefore, we assume that everyone in the understudy population is susceptible to infection.

3. Main results

To solve the system (2.1) it is necessary that the parameters μ, ν, N, f and s_0 be determined. Note that s_0 is considered as the total population. Also, we assume $1/\nu$ means that the average period of infectiousness of both unreported symptomatic infectious individuals and reported symptomatic infectious individuals is 14 days. We assume $1/\mu$, which means that the average period of infectiousness of asymptomatic infectious individuals is 7 days. These values can be modified as further epidemiological information becomes known. Here the parameter N is considered 5 days before the peak day. Note that the results are not very sensitive to the value of N and can be considered 5 to 15 days before the peak day. As mentioned earlier the important parameter is f , so we principally aim to gain the correct value of parameter f . To do this, we solve the system for different values of f by the Least Square method for finding the best parameters and by the Runge-Kutta method for solutions to the system of ODE. The results obtained from the system (2.1) for every value f are compared to the principle reported data from [7], the correct value f is chosen by observing more consistency between them. Also, to better compare the results and understand the importance of the value of parameter f , In what follows, we plotted the graph of $t \rightarrow R(t)$ for different f 's, as shown in figure 2 by changing the value of f the number of infected individuals is obtained by system (2.1) is changed.

It seems in the province of Guilan the reported infected cases are more real than in Iran's reported

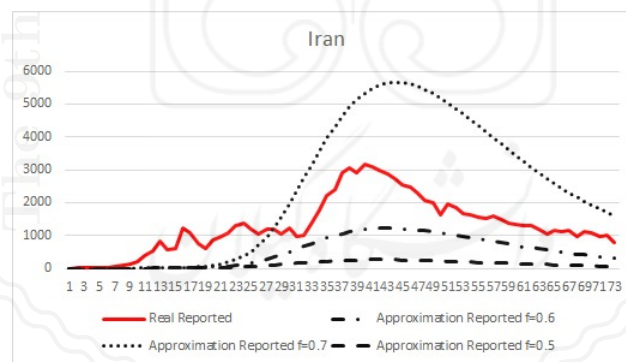


Figure 2: The values R (reported infected cases) of this diagram obtained from the solving of system (2.1) in Iran for $f = 0.5$, $f=0.6$ and $f=0.7$. We compare them with real reported cases.

cases, With $f = 0.9$ we can get an estimation of the daily unreported cases in the first outbreak period, the results are shown in figure3.

References

- [1] T.M. Chen, J. Rui, Q.P. Wang, Z.Y. Zhao, J.A. Cui, L. Yin, A mathematical model for simulating the phase-based transmissibility of a novel coronavirus, *Infectious Diseases of Poverty*, 9 (2020) no. 20.

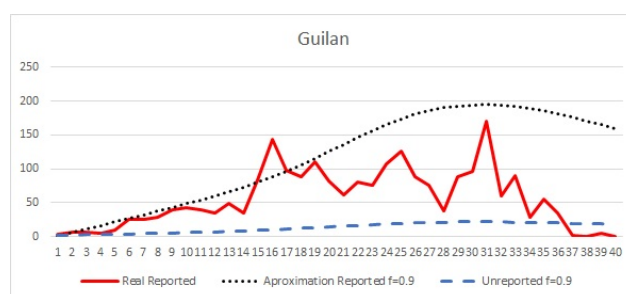


Figure 3: Diagram of Unreported and Reported Infected Cases from Solving of System 2.1 and Real Reported Cases for $f=0.9$ in Guilan

- [2] B. Ivorra, A.M. Ramos, Application of the Be-CoDis mathematical model to forecast the international spread of the 2019 Wuhan coronavirus outbreak, *Technical Report*, 9 February 2020, 1–13, DOI: [10.13140/RG.2.2.31460.94081](https://doi.org/10.13140/RG.2.2.31460.94081).
- [3] B. Ivorra, A.M. Ramos, Validation of the forecasts for the international spread of the coronavirus disease 2019 (COVID-19) done with the Be-CoDis mathematical model, *Technical Report*, 28 February 2020, 1–7, DOI: [10.13140/RG.2.2.34877.00485](https://doi.org/10.13140/RG.2.2.34877.00485).
- [4] B. Ivorra, A.M. Ramos, D. Ngom, Be-CoDiS: A mathematical model to predict the risk of human diseases spread between countries-validation and application to the 2014-2015 Ebola Virus Disease epidemic, *Bulletin of Mathematical Biology*, 77 (2015) 1668–1704.
- [5] Q. Lin, S. Zhao, D. Gao, Y. Lou, S. Yang, S.S. Musa, ... , D. He, A conceptual model for the outbreak of Coronavirus disease 2019 (COVID-19) in Wuhan, China with individual reaction and governmental action, *International Journal of Infectious Diseases*, 93 (2020) 211–216.
- [6] Z. Liu, P. Magal, O. Seydi, G. Webb, Understanding unreported cases in the COVID-19 epidemic outbreak in Wuhan, China, and the importance of major public health interventions, *Biology*, 9 (2020) no. 50.
- [7] *Worldometer website*: <https://www.worldometers.info/coronavirus/country/iran/>

Hossein Taheri

Department of Mathematics, Faculty of Mathematical Sciences, University of Mohaghegh Ardabili, 56199-11367, Ardabil, Iran.

email address: h.taheri.k@gmail.com

Nasrin Eghbali

Department of Mathematics, Faculty of mathematics, University of Guilan, Rasht, Iran

email address: eghbali@uma.ac.ir

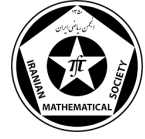
Masoumeh Pourabd

Department of Mathematics, Faculty of mathematics, University of Guilan, Rasht, Iran

email address: m.pourabd@gmail.com



University of Guilan



Computational aspects of stochastic partial differential equations using finite element method

Mahdieh Arezoomandan, Ali Reza Soheili

Abstract. This article describes the computational aspects of stochastic parabolic differential equations driven by additive noise. A fully discrete approximation of the stochastic problem is provided based on piecewise linear finite elements for the spacial discretization and the implicit Euler method for the temporal discretization. The computational aspects of the method are illustrated with a numerical test.

1. Introduction

Stochastic partial differential equations (SPDEs) are widely used models in applied sciences, engineering, and finance. Hence, the design of efficient computational methods for such problems is of great importance. In particular, the convergence analysis of numerical methods for approximating the solution of SPDEs is one of the most recently developed areas [1–3]. The aim of this work is to illustrate numerically the convergence properties of finite element method combined with implicit Euler method for a class parabolic semilinear SPDE, of the form

$$du(t) + Au(t)dt = F(u(t))dt + \sigma(t)dW_Q(t), \quad u(0) = u_0, \quad (1.1)$$

in a real separable Hilbert space \mathcal{H} with inner product (\cdot, \cdot) and norm $\|\cdot\| = (\cdot, \cdot)^{\frac{1}{2}}$. Here, A is assumed to be a linear, self adjoint, positive definite, not necessarily bounded operator with compact inverse. Moreover, $F : \mathcal{H} \rightarrow \mathcal{H}$ is a smooth nonlinearity and $\sigma : [0, T] \times \mathcal{H} \rightarrow \mathcal{H}$ is a deterministic mapping. $\{W_Q(t)\}_{t \geq 0}$ is considered to be a Q -Wiener process defined on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, \{\mathcal{F}_t\}_{t \geq 0})$. The following assumptions are standard in the literature on the numerical approximation of stochastic PDEs [4, 5]. Let $\{e_i\}_{i \in \mathbb{N}}$ be a complete orthonormal basis of the Hilbert space \mathcal{H} and the covariance operator Q be the linear, bounded, self adjoint operator on \mathcal{H} such that $Qv = \sum_{i=1}^{\infty} q_i \langle v, e_i \rangle e_i$, where $\{q_i\}_{i \in \mathbb{N}}$ is a sequence of non-negative real numbers. We assume $\{W_Q(t)\}_{t \geq 0}$ is a Q -Wiener process defined as follows:

$$W_Q(t) = \sum_{i=1}^{\infty} \sqrt{q_i} \beta_i(t) e_i, \quad (1.2)$$

where $\{\beta_i\}_{i \in \mathbb{N}}$ is a family of independent standard real valued Wiener processes. We assume that the nonlinear operator F in (1.1) is globally Lipschitz continuous. We also assume that the deterministic function $\sigma : [0, T] \times \mathcal{H} \rightarrow \mathcal{H}$ satisfies

$$\|A^{\frac{\beta-1}{2}} \sigma(t)\|_{L_2^0} \leq C, \quad \beta \in [0, 1]. \quad (1.3)$$

Keywords: stochastic partial differential equations, finite element methods, strong convergence rate, additive noise, implicit Euler method.

AMS Mathematical Subject Classification [2010]: 60H15, 60H35, 65C30, 65M65.

In this work, we are concerned with full discrete approximation of stochastic problem (1.1) based on the finite element spatial discretization combined with linear implicit Euler method for the temporal discretization. Let $\Delta t = \frac{T}{N}$ denote the time step size and $t_i = i\Delta t$, $i = 1, 2, \dots, N$. The full discrete method is defined by

$$u_h^{n+1} = E_{h,\Delta t}^n u_h^n + \Delta t E_{h,\Delta t}^n P_h F(u_h^n) + E_{h,\Delta t}^n P_h \sigma(t) \Delta W_Q^n, \quad (1.4)$$

for $n = 1, \dots, N$, where $E_{h,\Delta t}^n := (1 + \Delta t A_h)^{-1}$, with the initial condition $u_h^0 = P_h u_0$. In (1.4), the Wiener increments are denoted by $\Delta W_Q^n = W_Q((n+1)\Delta t) - W_Q(n\Delta t)$.

Theorem 1.1. [5] *Let $u(t)$ be the solution of (1.1) and let u_h^n be given by (1.4). Then, under the given assumptions, it holds that*

$$\|u(t_n) - u_h^n\|_{L_2(\Omega; \mathcal{H})} \leq C(h^\beta + \Delta t^{\frac{\beta}{2}}), \quad (1.5)$$

where C is a constant independent of h and Δt .

2. Numerical test

In this subsection, we present a numerical test to illustrate the convergence analysis. We consider the following stochastic problem

$$\begin{aligned} \frac{\partial u}{\partial t}(x, t) &= \frac{\partial^2 u}{\partial t^2}(x, t) + f(x, t) = \dot{W}, \\ u(x, 0) &= 10x^2(1-x)^2, & x \in [0, 1], \\ u(0, t) &= u(1, t) = 0, & t \in [0, 1]. \end{aligned} \quad (2.1)$$

where

$$f(x, t) = 15e^t x^2(1-x)^2 - 10e^t(2-12x+12x^2).$$

we use a piecewise linear finite element method for the spatial discretization and an implicit Euler method for the temporal discretization. Let u_h^n be the approximate solution of $u(t)$ in finite element space S_h at $t_n = n\Delta t$. The implicit Euler method is to find $u_h^n \in S_h$ such that, for all $\phi \in S_h$,

$$\left(\frac{u_h^{n+1} - u_h^n}{\Delta t}, \phi\right) + (A_h u_h^n, \phi) = \left(\frac{1}{\Delta t} P_h(W_Q(t_n) - W_Q(t_{n-1})), \phi\right) \quad (2.2)$$

$$= \frac{1}{\Delta t} \sum_{i=1}^{\infty} \sqrt{q_i} (\beta_i(t_n) - \beta_i(t_{n-1})) (e_i, \phi), \quad (2.3)$$

where $\frac{1}{\Delta t}(\beta_i(t_n) - \beta_i(t_{n-1})) = \mathcal{N}(0, 1)$. We choose two types of covariance operators, $Q = I$ and the other operator, $Qe_1 = 0$ and $Qe_i = \frac{1}{i \log i^2} e_i$ for $i \geq 2$. In Figure 1, we plot one realization of the stochastic problem (2.1) for the two types of the covariance operators. We also plot in Figure 2 the corresponding profiles at times $t = 0.25, 0.5, 0.75$ and final time $T = 1$. In Figure 3, we present the convergence curves for the strong error for the covariance operator $Q = I$. At first, we demonstrate the convergence rates for the temporal discretization. To do this, we compute the reference solution with the small timestep $\Delta t_{ref} = 2^{-11}$ and $h_{ref} = 2^{-7}$. We perform our numerical simulation with different time step sizes $\Delta t_{ref} = 2^{-i}$, $i = 3, \dots, 9$ and present the mean square errors in Figure 3 (left). As expected, we observe the convergence rate of order $\frac{1}{4}$, this is consistent with the strong convergence estimates of Theorem 1.1. Next, we turn to spatial error approximation. To this aim, we compute the reference solution using fixed small $h_{ref} = 2^{-10}$ and $\Delta t_{ref} = 2^{-6}$. We plot in Figure 3 (right) the mean square errors due to the spatial discretization using the step sizes $h = 2^{-i}$, $i = 2, \dots, 8$.

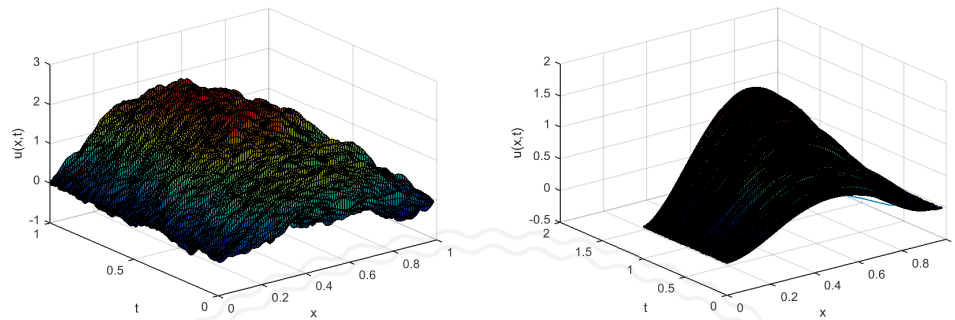


Figure 1: Samples of realization of SPDE (2.1) (left $Q = I, Tr(Q) < \infty$)

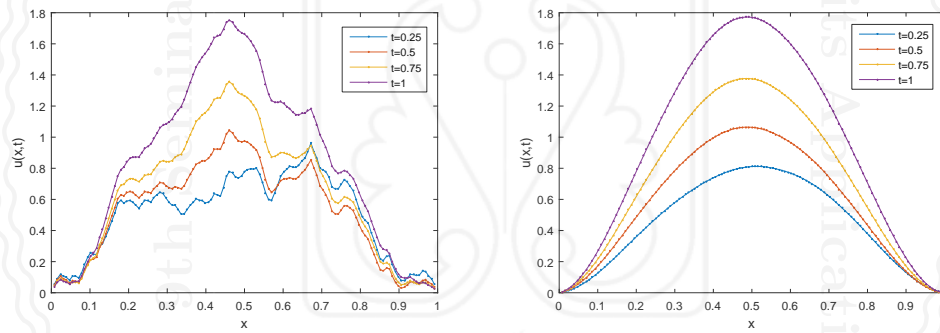


Figure 2: Solution profile at different times (left $Q = I, Tr(Q) < \infty$)

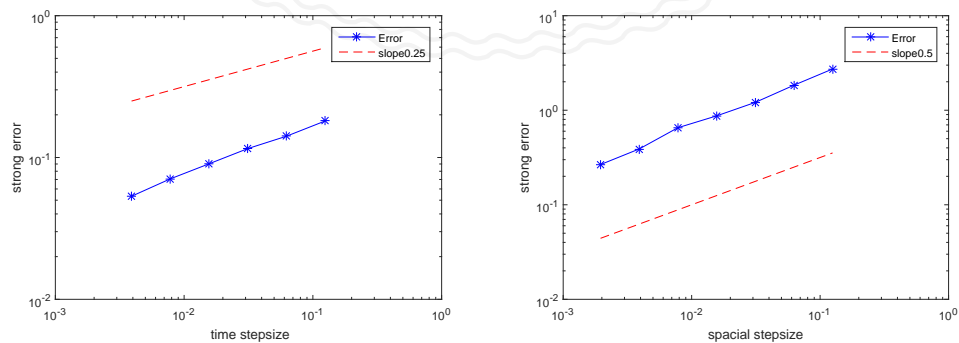


Figure 3: Error versus time stepsize (left) and space stepsize (right)

References

- [1] R. Kruse, Optimal error estimates of Galerkin finite element methods for stochastic partial differential equations with multiplicative noise, *IMA Journal of Numerical Analysis*, 34 (2014) 217–251.
- [2] A. Tambue, J.D. Mukam, Strong convergence of the linear implicit Euler method for the finite element discretization of semilinear SPDEs driven by multiplicative or additive noise, *Applied Mathematics and Computation*, 346 (2019) 23–40.
- [3] G.J. Lord, A. Tambue, Stochastic exponential integrators for the finite element discretization of SPDEs with multiplicative and additive noise, *IMA Journal of Numerical Analysis*, 2 (2013) 515–543.
- [4] X. Wang, Strong convergence rates of the linear implicit method for the finite element discretization of SPDEs with additive noise, *IMA Journal of Numerical Analysis*, 37 (2017) 965–984.
- [5] Y. Yan, Galerkin finite element methods for stochastic parabolic partial differential equations, *SIAM Journal on Numerical Analysis*, 43 (2005) 1363–1384.

Mahdieh Arezoomandan

Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad
email address: arezoomandan@mail.um.ac.ir

Ali Reza Soheili

Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran

email address: soheili@um.ac.ir

بررسی حالت تطبیقی روش‌های استیفنس‌گون با حافظه برای حل معادلات غیرخطی با بیشترین شاخص کارایی ممکن

محمدجواد لاله‌چینی

چکیده. در این مقاله به بررسی حالت تطبیقی روش‌های استیفنس‌گون با حافظه برای حل معادلات غیرخطی با بیشترین شاخص کارایی ممکن می‌پردازیم. هدف اصلی این کار معرفی روش تطبیقی برای دو روش از نوع استیفنس‌گون با حافظه با بالاترین شاخص‌های کارایی است. در روش‌های موجود، برای بهبود مرتبه همگرایی، تمرکز فقط بر تکرار فعلی و قبلی بوده است. با این حال، در این روش می‌توان از اولین تکرار تا تکرار فعلی، استفاده کرده و مرتبه همگرایی را بهبود بخشید. بنابراین، ما به روش‌هایی برای بالا بردن مرتبه همگرایی دست پیدا می‌کنیم و تا آنجا که ممکن است شاخص‌های کارایی بالایی به دست می‌آوریم.

۱. پیش‌گفتار

در روش‌های موجود برای حل معادلات غیرخطی، به منظور بهبود مرتبه همگرایی، تمرکز تنها بر تکرار فعلی و قبلی وجود دارد [۳]. و ممکن است با استفاده از پارامترهای شتاب دهنده مرتبه همگرایی را بهبود بخشند، اما در حالت تطبیقی با توجه به اولین تکرار تا تکرار کنونی تمامی داده‌ها را حفظ کرده و دستورالعمل‌های را برای بالا بردن مرتبه همگرایی با شاخص‌های کارایی بالا که هدف اصلی این کار است، بدست می‌آوریم. یکی از مهمترین موضوعات در توسعه الگوریتم‌های عددی، ایجاد الگوریتم‌های بهینه و مقرون به صرفه است. مثلاً، یکی از این موضوعات، توسعه یک روش تکراری برای بدست آوردن تقریبی از صفرهای یک معادله غیر خطی داده شده می‌باشد و مطالعات بسیاری به آن اختصاص یافته است [۲]. با توجه به این موضوع ما روش استیفنس‌گون را به صورت تطبیقی و با حافظه تبدیل می‌کنیم [۱، ۲].

۲. توسعه روش استیفنس

در ابتدا تراب روش استیفنس‌گون [۱] را به صورت با حافظه توسعه داد [۴].

$$\begin{cases} w_k = x_k + \gamma_k f(x_k), \\ x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \\ \gamma_{k+1} = -\frac{1}{N'(x_{k+1})}, \end{cases} \quad k = 0, 1, 2, \dots, \quad (1)$$

واژه‌های کلیدی: معادلات غیر خطی، روش استیفنس‌گون، روش با حافظه، روش تطبیقی، مرتبه همگرایی.
طبقه‌بندی موضوعی [۲۰۱۰]: 65H05; 49M15; 65J15.

که x_0 و γ_0 مقادیر داده شده اولیه هستند، و $N_1(t) = f(x_{k+1}) + (t - x_{k+1})f[x_{k+1}, x_k]$ یک چند جمله‌ای درونیاب نیوتن است. مرتبه همگرایی روش با حافظه (۱) برابر $2/\sqrt{2} \approx 1.414$ می‌باشد. علاوه بر این دزونیک با اضافه کردن یک پارامتر دیگر به روش استیفنس، روش با حافظه و با کارایی بیشتری را بدست آورد [۲]:

$$\begin{cases} x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k] + \lambda_k f(w_k)}, & k = 0, 1, 2, \dots, \\ \gamma_{k+1} = -\frac{1}{N'_2(x_{k+1})}, \\ w_{k+1} = x_{k+1} + \gamma_{k+1} f(x_{k+1}), \\ \lambda_{k+1} = \frac{-N'''_2(w_{k+1})}{2N'_2(w_{k+1})}, \end{cases} \quad (2)$$

که x_0 و γ_0 و λ_0 مقادیر داده شده اولیه هستند. این روش دارای مرتبه همگرایی برابر $3+\sqrt{17}/4 \approx 3.56$ می‌باشد. در ادامه تلاش خواهیم کرد روش‌های تطبیقی با حافظه منطبق با دو روش (۲) و (۱) صرف نظر از برتری آن‌ها ایجاد کنیم [۵]. برای این منظور ابتدا پارامتر شتاب دهنده γ_k را در هر تکرار نه تنها با استفاده از اطلاعات موجود در تکرارهای فعلی و قبلی، بلکه از همه تکرارهای قبلی آن به روز می‌کنیم. همچنین ثابت می‌کنیم این روش دارای مرتبه همگرایی $3/4$ که فراخوانی تابع در آن مشابه (۱) است. بنابراین شاخص کارایی آن بسیار بهتر است. به همین ترتیب، بار دیگر روش تطبیقی با حافظه را برای (۲) می‌گیریم، که مرتبه همگرایی $3/9$ با تعداد فراخوانی تابع مشابه در (۲) بدست می‌آید. بنابراین، این روش بهتر از تمام روش‌های با حافظه موجود با یک شتاب دهنده می‌باشد.

۳. توسعه حالت تطبیقی روش‌های با حافظه

این بخش با دو روش جدید تطبیقی با حافظه مورد بررسی قرار می‌گیرد. برای این منظور، روش‌های (۱) و (۲) را اصلاح و گسترش می‌دهیم به طوری که در آنها تمام اطلاعات قبلی را در نظر گرفته تا احتمالاً به حداکثر مرتبه همگرایی لازم برسد بدون آن که فراخوانی جدیدی داشته باشد. به این ترتیب، ما از یک ایده تطبیقی استفاده می‌کنیم.

۱.۳. یک شتاب دهنده برای روش تطبیقی با حافظه

در (۱) برای به روز رسانی شتاب دهنده γ_k در هر تکرار، ما فقط از اطلاعات تکرارهای فعلی و قبلی استفاده می‌کنیم و به مرتبه همگرایی ۳ می‌رسیم. با این حال، همانطور که روش پیش می‌رود، ممکن است نه تنها از اطلاعات قدیمی استفاده شود بلکه از اطلاعات ۳ مرحله فعلی و گامهای قبلی نیز استفاده شود. به عبارت دیگر، ما می‌خواهیم ایده تطبیقی را برای ساخت روش‌های جدید با حافظه بکار ببریم. بر این اساس، روش جدید تطبیقی با حافظه زیر را معرفی می‌کنیم:

$$\begin{cases} w_k = x_k + \gamma_k f(x_k), \\ x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k]}, & k = 0, 1, 2, \dots, \\ \gamma_{k+1} = -\frac{1}{N'_2(x_{k+1})}, \end{cases} \quad (3)$$

که x_0 و γ_0 مقادیر اولیه داده شده می‌باشند، و $N_{2k+2}(t)$ یک چند جمله‌ای درونیاب نیوتن از درجه $2k+2$ با استفاده از نقاط $x_0, w_0, x_1, w_1, x_2, w_2, \dots, x_k, w_k$ می‌باشد.

اگر در نظر بگیریم γ_k ثابت است، آنگاه مرتبه همگرایی روش بدون حافظه (۳) افزایش می‌یابد، از آنجا که α نامشخص است، نمی‌توانیم فرض کنیم $\gamma = -1/f'(\alpha)$. حتی اگر فرض کنیم که α شناخته شده بود، باز هم نمی‌توانستیم از آن برای ارزیابی $f'(\alpha)$ استفاده کنیم، زیرا فراخوانی تابع را افزایش داده و بهینه بودن روش را از بین می‌برد. فرض بر این است که دنباله $\{x_k\}$ به α همگرا می‌شود. علاوه بر این f' دارای پیوستگی است، بنابراین $\lim_{k \rightarrow \infty} f'(x_k) = f'(\alpha)$ وقتی $k \rightarrow \infty$. بنابراین، می‌توانیم $N'_{2k+2}(x_k)$ به جای $f'(x_k)$

استفاده کنیم. یعنی: $\gamma_k = -1/N'_{\gamma k+2}(x_k)$.

۲.۳. شتاب دهنده‌های روش تطبیقی با حافظه

حالا شتاب دهنده‌های روش تطبیقی با حافظه را معرفی می‌کنیم [۵]. از آنجایی که بیشتر جزئیات در (۳) آمده است، لذا از تکرار آنها خودداری کرده و دو شتاب دهنده جدید را برای روش تطبیقی با حافظه در نظر می‌گیریم:

$$\begin{cases} x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k] + \lambda_k f(w_k)}, & k = 0, 1, 2, \dots, \\ \gamma_{k+1} = -\frac{1}{N'_{\gamma k+2}(x_{k+1})}, \\ w_{k+1} = x_{k+1} + \gamma_{k+1} f(x_{k+1}), \\ \lambda_{k+1} = \frac{-N''_{\gamma k+2}(w_{k+1})}{2N'_{\gamma k+2}(w_{k+1})}, \end{cases} \quad (4)$$

که x_0 ، γ_0 و λ_0 به صورت مناسبی انتخاب شده‌اند. در آن صورت قضیه زیر را خواهیم داشت:

قضیه ۱. فرض می‌کنیم تقریب اولیه x_0 به اندازه کافی به ریشه α از f نزدیک باشد. همچنین R و p به ترتیب معرف مرتبه همگرایی دنباله $\{x_k\}$ و $\{w_k\}$ باشد که در روش تطبیقی با حافظه (۴) بدست می‌آید. در آن صورت خواهیم داشت:

$$\begin{cases} R^k p - R^k - (p+1) \sum_{i=0}^{k-1} R^i = 0 \\ R^{k+1} - 2R^k - 2(p+1) \sum_{i=0}^{k-1} R^i = 0 \end{cases} \quad (5)$$

۴. نتایج عددی

در این بخش، برای نشان دادن کارایی روش‌های تطبیقی با حافظه (۳) و (۴)، نتایج عددی آن‌ها را گزارش می‌دهیم. برای این منظور، در میان بسیاری از مسائل آزمایش شده، گزارش را به نتایج چهار تابع محدود می‌کنیم (نگاه کنید به جدول ۱). علاوه بر این، مقادیر اولیه در جدول ۱ داده شده و در جدول ۲ نتایج عددی روش‌های تطبیقی با حافظه (۴)، با توجه به مقادیر اولیه داده شده مشاهده می‌شود. لازم به ذکر است که $|x_k - \alpha|$ خطا را در هر تکرار نشان می‌دهد، $a(b)$ نشان دهنده a^b است.

جدول ۱: توابع آزمایشی $\gamma = 0.1$ ، $\lambda = 0.1$.

α	x_0	مثال
۲/۰۰	۲/۵۰	$f_1(t) = e^{(t^2-4)} + \sin(t-2) - t^4 + 15$
۱/۰۰	۲/۰۰	$f_2(t) = \frac{1}{t^4} - t^2 - \frac{1}{t} + 1$
۲/۰۰	۲/۴۰	$f_3(t) = (t-2)(t^{10} + t + 2)e^{-5t}$
۰/۰۰	۰/۴۵	$f_4(t) = e^{(t^2-3t)} + \sin(t) + \log(t^2 + 1)$

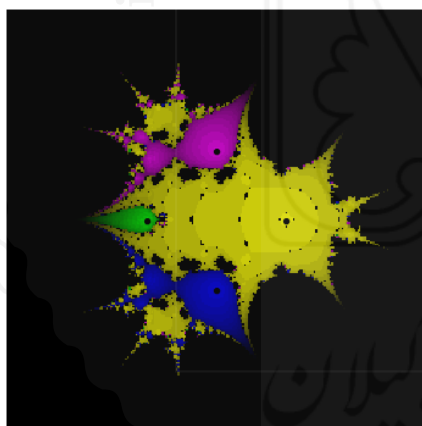
تا اینجا ثابت کرده‌ایم که روش تطبیقی جدید با حافظه (۴) تنها با دو فراخوانی تابع می‌تواند به یک شاخص کارایی مشابه برسد (نگاه کنید به قضیه (۱)). این بدان معنی است که این روش با هر روش چند گامی بهین بدون حافظه رقابت می‌کند.

جدول ۲: نتایج عددی برای رابطه (۴).

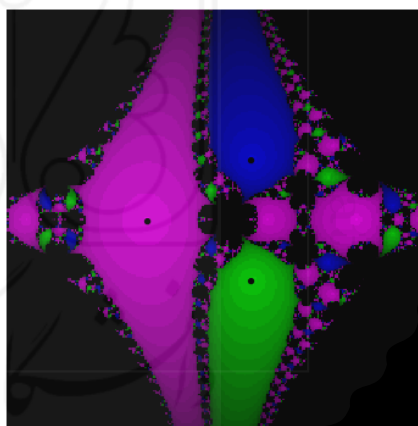
تابع.	$ x_1 - \alpha $	$ x_2 - \alpha $	$ x_3 - \alpha $	COC
f_1	$0/2902(0)$	$0/1028(-1)$	$0/1581(-7)$	$4/0078$
f_2	$0/9073(-1)$	$0/2554(-2)$	$0/2200(-8)$	$3/9115$
f_3	$0/1681(0)$	$0/1006(-2)$	$0/1582(-11)$	$3/9603$
f_4	$0/4153(-1)$	$0/4718(-3)$	$0/5529(-11)$	$4/0784$

۵. رفتار دینامیکی

در این جا ما روی رفتار پایدار روش‌های تطبیقی با حافظه (۳) و (۴) تمرکز می‌کنیم. برای این منظور از روش دینامیکی بصری [۵] استفاده می‌کنیم. اگر چه ما نمونه‌های بسیاری را آزمایش کرده‌ایم، اما به نتیجه گیری مشابهی دست یافته‌ایم. برای نشان دادن پایداری روش‌های یک پارامتری (۳) و دو پارامتری (۴)، خواص دینامیکی را بررسی کرده و بر روی صفحات دینامیکی مربوط به روش‌های تکراری تمرکز می‌کنیم. نتایج نشان می‌دهد هر چند روش‌های توسعه یافته با حافظه، از لحاظ پیچیدگی‌های محاسباتی دارای برتری هستند، اما به نوعی ناپایداری عددی در آن‌ها دیده می‌شود.



شکل (ب) ۲



شکل (ا) ۲

۶. نتیجه گیری

در این کار، دو روش جدید اما بسیار کارآمد با حافظه برای حل معادله غیر خطی ایجاد کردیم. نشان داده‌ایم که این روش‌ها می‌توانند به بالاترین شاخص‌های کارایی دست یابند و می‌توانند با هر روش با حافظه یا بدون حافظه در این مبحث رقابت کنند. و مانند هر روش دیگر در این مبحث، ممکن است روش با حافظه یک بی ثباتی را در عمل نشان دهد. بنابراین، آنها از کارایی محاسباتی سود می‌برند و از پایداری عددی رنج می‌برند. در نهایت، نتیجه گیری را با سوال تحقیق زیر مطرح می‌کنیم: چگونه می‌توان در یک روش با حافظه توسعه یافته نشان داد که در رفتار آن یک پایداری عددی وجود دارد؟

مراجع

- [1] I.F. Steffensen, Remarks on iteration, Skand. Aktuarietidskr. 16 (1933), 64-72.
- [2] J. Dzunic, M.S. Petkovic, A cubically convergent steffensen-like method for solving nonlinear equations, *Applied Mathematics Letters*, 25 (2012) 1881-1886.
- [3] M.J. Lalehchini, T. Lotfi, K. Mahdiani, On Developing an Adaptive Free-Derivative Kung and Traub's Method with Memory, *Accepted in Journal of Mathematical Extension*.
- [4] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice Hall, New York, 1964.
- [5] X. Wang, T. Zhang, Y. Qin, Efficient two-step derivative-free iterative methods with memory and their dynamics, *International Journal of Computer Mathematics*, 93 (2016) 1423-1446.

محمدجواد لاله‌چینی

گروه ریاضی، دانشگاه فرهنگیان، همدان، ایران.

آدرس ایمیل: lalehchini@gmail.com lalehchini@cfu.ac.ir

یک روش تکراری برای حل مسئله بهینه‌سازی با محدودیت PDE

حمید میرچی، داود خجسته سالکویه

چکیده. در این مقاله یک روش تکراری دو مرحله‌ای برای حل یک دستگاه معادلات خطی معرفی می‌کنیم که ماتریس ضرایب آن از گسسته‌سازی یک مسئله کنترل بهینه سهموی زمان-متناوب به کمک روش عناصر متناهی بدست می‌آید. در ادامه همگرایی روش ارائه شده و نتایج عددی حاصل از به کارگیری این روش ارائه می‌شود.

۱. پیش‌گفتار

مسئله کنترلی زیر را در نظر می‌گیریم [۲]:

$$\begin{aligned} \min_{y,u} \quad & \frac{1}{2} \int_0^T \int_{\Omega} |y(x,t) - y_d(x,t)|^2 dx dt + \frac{\nu}{2} \int_0^T \int_{\Omega} |u(x,t)|^2 dx dt, \\ \text{s.t.} \quad & \frac{\partial}{\partial t} y(x,t) - \Delta y(x,t) = u(x,t) \quad \text{in } Q_T, \\ & y(x,t) = 0 \quad \text{on } \Sigma_T, \\ & y(x,0) = y(x,T) \quad \text{on } \partial\Omega, \\ & u(x,0) = u(x,T) \quad \text{in } \Omega, \end{aligned} \quad (1)$$

که در آن Ω دامنهٔ باز و کراندار در \mathbb{R}^d برای $d = \{1, 2, 3\}$ و $\partial\Omega$ مرز Ω است. همچنین $Q_T = \Omega \times (0, T)$ استوانهٔ زمان مکان^۱ است و سطح جانبی آن $\Sigma_T = \partial\Omega \times (0, T)$ می‌باشد.

برای گسسته‌سازی این مسئله ابتدا تابع زمان-متناوب^۲ $y_d(x,t) = y_d(x)e^{i\omega t}$ را در نظر می‌گیریم، که در آن $\omega = \frac{2\pi m}{T}$ برای برخی m در \mathbb{Z} [۳]. جواب و تابع کنترل مسئله زمان-متناوب به صورت $y(x,t) = y(x)e^{i\omega t}$ و $u(x,t) = u(x)e^{i\omega t}$ است که $y(x)$ و $u(x)$ جواب مسئله کنترل بهینه زیر هستند،

$$\begin{aligned} \min_{y,u} \quad & \frac{1}{2} \int_{\Omega} |y(x) - y_d(x)|^2 dx + \frac{\nu}{2} \int_{\Omega} |u(x)|^2 dx, \\ \text{s.t.} \quad & i\omega y(x) - \Delta y(x) = u(x) \quad \text{in } \Omega, \\ & u(x) = 0 \quad \text{on } \partial\Omega. \end{aligned} \quad (2)$$

واژه‌های کلیدی: مسئله کنترلی توزیع شده، پیش شرط سازی، روش تکراری، دستگاه معادلات خطی، بهینه سازی. طبقه‌بندی موضوعی [۲۰۱۰]: 49M25, 49K20, 65F10.

¹space-time cylinder

²time-periodic

در ادامه فرض می‌کنیم $y_d(x)$ و $y(x)$ حقیقی هستند و تابع کنترل $u(x)$ مختلط در نظر گرفته می‌شود. در روند گسسته‌سازی به کمک روش عناصر متناهی، فرض کنیم $V_h \subset H_0^1(\Omega)$ یک فضای برداری m بعدی با توابع پایه‌ای $\{\phi_1, \phi_2, \dots, \phi_m\}$ باشد. برای محاسبه y و u از زیرفضای V_h استفاده می‌کنیم. همچنین با استفاده از ایده گسسته‌سازی و سپس بهینه‌سازی [۴]، مسئله (۲) به صورت زیر بازنویسی می‌شود

$$\min_{\bar{y}, \bar{u}} \quad \frac{1}{\nu} (\bar{y} - \bar{y}_d)^* M (\bar{y} - \bar{y}_d) + \frac{\nu}{\nu} \bar{u}^* M \bar{u}, \quad (3)$$

$$\text{s.t.} \quad i\omega M \bar{y} + K \bar{y} = M \bar{u},$$

که در آن ماتریس حقیقی M ماتریس جرم^۳ و K ماتریس سختی^۴ هستند (می‌توان دید که هر دو ماتریس معین مثبت متقارن هستند). همچنین $y(x)$ ، $y_d(x)$ و $u(x)$ در صورت گسسته‌سازی شده به ترتیب به بردارهای \bar{y} ، \bar{y}_d و \bar{u} که شامل ضرایب بسط‌های مربوط به توابع پایه‌ای روش عناصر متناهی هستند، تغییر یافته‌اند. تابع لاگرانژ مسئله (۳) به صورت

$$L(\bar{y}, \bar{u}, \bar{p}) = \frac{1}{\nu} (\bar{y} - \bar{y}_d)^* M (\bar{y} - \bar{y}_d) + \frac{\nu}{\nu} \bar{u}^* M \bar{u} + \bar{p}^* (i\omega M \bar{y} + K \bar{y} - M \bar{u}), \quad (4)$$

است که با در نظر گرفتن شرایط لازم مرتبه اول برای تابع لاگرانژ (۴) و با جای‌گذاری $\bar{q} = \frac{1}{\sqrt{\nu}} \bar{p}$ ، دستگاه زیر بدست می‌آید

$$\mathcal{A} \mathbf{x} = \begin{pmatrix} M & \sqrt{\nu}(K - i\omega M) \\ \sqrt{\nu}(K + i\omega M) & -M \end{pmatrix} \begin{pmatrix} \bar{y} \\ \bar{q} \end{pmatrix} = \begin{pmatrix} M \bar{y}_d \\ 0 \end{pmatrix} = \mathbf{b}. \quad (5)$$

حال در دستگاه (۵)، با توجه به این که \bar{y} حقیقی است و با در نظر گرفتن $\bar{q} = -z + it$ ، از سطر دوم دستگاه خواهیم داشت $t = \sqrt{\nu\omega} \bar{y}$. با جای‌گذاری $x = \sqrt{1 + \nu\omega} t$ و تعویض سطر اول و دوم، دستگاه زیر حاصل می‌شود

$$\mathcal{A}_1 \mathbf{x} = \begin{pmatrix} \sqrt{\nu}K & \tau M \\ -\tau M & \sqrt{\nu}K \end{pmatrix} \begin{pmatrix} z \\ x \end{pmatrix} = \begin{pmatrix} 0 \\ -M \bar{y}_d \end{pmatrix} = \mathbf{b}_1, \quad (6)$$

که در آن $\tau = \sqrt{1 + \nu\omega}$ است. برای حل دستگاه (۶) می‌توان از روش $MHSS$ ^۵ که توسط بای و همکاران در [۱] ارائه شده است، استفاده کرد. این روش تکراری دومرحله‌ای به صورت

$$\begin{cases} \begin{pmatrix} \alpha V + \sqrt{\nu}K & 0 \\ 0 & \alpha V + \sqrt{\nu}K \end{pmatrix} \begin{pmatrix} z^{(k+\frac{1}{\tau})} \\ x^{(k+\frac{1}{\tau})} \end{pmatrix} = \begin{pmatrix} \alpha V & -\tau M \\ \tau M & \alpha V \end{pmatrix} \begin{pmatrix} z^{(k)} \\ x^{(k)} \end{pmatrix} + \begin{pmatrix} 0 \\ -M \bar{y}_d \end{pmatrix} \\ \begin{pmatrix} \alpha V + \tau M & 0 \\ 0 & \alpha V + \tau M \end{pmatrix} \begin{pmatrix} z^{(k+1)} \\ x^{(k+1)} \end{pmatrix} = \begin{pmatrix} \alpha V & \sqrt{\nu}K \\ -\sqrt{\nu}K & \alpha V \end{pmatrix} \begin{pmatrix} z^{(k+\frac{1}{\tau})} \\ x^{(k+\frac{1}{\tau})} \end{pmatrix} + \begin{pmatrix} M \bar{y}_d \\ 0 \end{pmatrix} \end{cases}$$

می‌باشد، که در آن α یک مقدار حقیقی و مثبت و $V \in \mathbb{R}^{n \times n}$ یک ماتریس معین مثبت متقارن است. با توجه به مطالب مقاله [۱]، یک انتخاب برای V ماتریس $\sqrt{\nu}K$ می‌باشد. در هر تکرار از روش $MHSS$ باید چهار دستگاه حل شود که ماتریس ضرایب دو دستگاه $\alpha V + \sqrt{\nu}K$ و ماتریس ضرایب دو دستگاه دیگر $\alpha V + \tau M$ می‌باشند.

روشی که در ادامه ارائه می‌کنیم بدون پارامتر است و در هر تکرار این روش فقط با یک ماتریس ضرایب روبه‌رو هستیم.

³mass matrix

⁴stiffness matrix

⁵modified Hermitian and skew-Hermitian

۲. نتایج اصلی

برای دستگاه (۶) شکافت زیر را در نظر بگیرید

$$A_1 = \begin{pmatrix} \sqrt{\nu}K & \circ \\ \circ & \sqrt{\nu}K \end{pmatrix} + \begin{pmatrix} \circ & \tau M \\ -\tau M & \circ \end{pmatrix}.$$

به کمک این شکافت روش تکراری دو مرحله‌ای زیر را معرفی می‌کنیم

$$\begin{cases} \mathcal{K}\mathbf{x}^{(k+\frac{1}{2})} = \mathcal{J}\mathbf{x}^{(k)} + \mathbf{b}_1, \\ \mathcal{M}\mathbf{x}^{(k+1)} = \mathcal{N}\mathbf{x}^{(k+\frac{1}{2})} + \mathbf{b}_1, \end{cases} \quad (7)$$

که در آن

$$\mathcal{K} = \begin{pmatrix} \sqrt{\nu}K + \tau M & \circ \\ \circ & \sqrt{\nu}K + \tau M \end{pmatrix}, \quad \mathcal{J} = \begin{pmatrix} \tau M & -\tau M \\ \tau M & \tau M \end{pmatrix},$$

$$\mathcal{M} = \begin{pmatrix} \circ & \sqrt{\nu}K + \tau M \\ -(\sqrt{\nu}K + \tau M) & \circ \end{pmatrix}, \quad \mathcal{N} = \begin{pmatrix} -\sqrt{\nu}K & \sqrt{\nu}K \\ -\sqrt{\nu}K & -\sqrt{\nu}K \end{pmatrix}.$$

با محاسبه بردار $\mathbf{x}^{(k+\frac{1}{2})}$ از معادله اول (۷) و جای‌گذاری در معادله دوم روش تکراری یک مرحله‌ای

$$\mathbf{x}^{(k+1)} = \mathcal{M}^{-1}\mathcal{N}\mathcal{K}^{-1}\mathcal{J}\mathbf{x}^{(k)} + \mathcal{M}^{-1}(\mathcal{N}\mathcal{K}^{-1}\mathbf{b}_1 + \mathbf{b}_1), \quad (8)$$

بدست می‌آید.

قضیه ۱. مقادیر ویژه ماتریس تکرار \mathcal{H} به صورت $\mu = \frac{1}{\frac{1}{\sqrt{\nu}}(\frac{\sqrt{\nu}}{\tau}s + \frac{\tau}{\sqrt{\nu}}\frac{1}{s}) + 1}$ هستند، که در آن $s \in \sigma(M^{-1}K)$. همچنین مقادیر ویژه در بازه $[\frac{1}{\sqrt{\nu}}, \frac{\tau}{\sqrt{\nu}}]$ قرار دارند.

اثبات. ماتریس تکرار روش (۸)، یعنی $\mathcal{H} = \mathcal{M}^{-1}\mathcal{N}\mathcal{K}^{-1}\mathcal{J}$ ، در رابطه زیر صدق می‌کند

$$\mathcal{H} = \begin{pmatrix} \mathcal{F}^{-1} & \circ \\ \circ & \mathcal{F}^{-1} \end{pmatrix}, \quad (9)$$

که در آن $\mathcal{F} = \frac{\sqrt{\nu}}{\tau}M^{-1}K + \frac{\tau}{\sqrt{\nu}}K^{-1}M + I$ است. بنابراین، مقادیر ویژه ماتریس تکرار \mathcal{H} به صورت

$$\mu = \frac{1}{\frac{1}{\sqrt{\nu}}(\frac{\sqrt{\nu}}{\tau}s + \frac{\tau}{\sqrt{\nu}}\frac{1}{s}) + 1},$$

هستند، که در آن s یک مقدار ویژه ماتریس $P = M^{-1}K$ است. ماتریس P با ماتریس $M^{-\frac{1}{2}}KM^{-\frac{1}{2}}$ متشابه است که یک ماتریس معین مثبت متقارن است. در نتیجه، $s > 0$. بنابراین،

$$\frac{\sqrt{\nu}}{\tau}s + \frac{\tau}{\sqrt{\nu}}\frac{1}{s} \geq 2,$$

□

که نتیجه لازم به دست می‌آید.

جدول ۱: نتایج عددی مربوط به $h = 2^{-\nu}$.

		$\omega = 10^{-4}$		$\omega = 10^{-2}$		$\omega = 10^0$		$\omega = 10^2$		$\omega = 10^4$	
روش	ν	Its	CPU	Its	CPU	Its	CPU	Its	CPU	Its	CPU
New method	10^{-2}	۱۴	۰.۸۸	۱۴	۰.۸۹	۱۴	۰.۸۸	۱۹	۰.۹۶	۱۸	۰.۹۴
	10^{-4}	۱۹	۱.۱۱	۱۹	۱.۰۹	۱۹	۱.۱۳	۱۹	۱.۰۹	۱۸	۱.۰۴
	10^{-6}	۱۸	۱.۱۱	۱۹	۱.۱۰	۱۹	۱.۱۷	۱۹	۰.۹۲	۱۸	۱.۰۴
	10^{-8}	۱۸	۱.۱۵	۱۸	۱.۰۳	۱۸	۱.۰۱	۱۸	۰.۹۴	۱۸	۱.۰۹
MHSS method	10^{-2}	۳۹	۲.۳۰	۳۹	۲.۱۵	۳۹	۲.۱۳	۳۶	۱.۹۶	۳۸	۲.۰۵
	10^{-4}	۳۶	۱.۸۸	۳۶	۱.۹۹	۳۶	۱.۹۴	۳۶	۱.۹۲	۳۸	۲.۰۵
	10^{-6}	۳۸	۲.۱۱	۳۳	۱.۸۲	۳۳	۱.۸۹	۳۳	۰.۸۳	۳۸	۱.۹۷
	10^{-8}	۳۸	۲.۱۶	۳۸	۱.۸۲	۳۸	۲.۱۵	۳۸	۲.۰۴	۳۸	۲.۱۲

۳. نتایج عددی

مساله کنترل بهینه دو بعدی با دامنه $\Omega = (0, 1) \times (0, 1) \in \mathbb{R}^2$ و حالت مطلوب

$$y_d(x, y) = \begin{cases} (2x - 1)^2(2y - 1)^2, & \text{اگر } (x, y) \in (0, \frac{1}{2}) \times (0, \frac{1}{2}), \\ 0, & \text{در غیر این صورت} \end{cases} \quad (10)$$

را در نظر گرفته‌ایم. برای دستگاه حاصل از گسسته‌سازی، روش تکراری ارائه شده را پیاده‌سازی می‌کنیم. در به کارگیری این روش در هر تکرار دستگاه‌های درونی حاصل به کمک تجزیه چولسکی حل می‌شوند. نتایج عددی حاصل از مقایسه روش ارائه شده با روش MHSS برای $h = 2^{-\nu}$ و مقادیر مختلف ω و ν در جدول ۱ آمده است. در هر دو روش تکرار وقتی متوقف می‌شود که نرم مانده به اندازه 10^{-6} کاهش یابد و حدس اولیه همواره بردار صفر در نظر گرفته شده است. همچنین در روش MHSS مقدار α برابر به ۱ در نظر گرفته شده است. کلیه اجراها در برنامه متلب ۲۰۱۸ توسط یک لپ تاپ با پردازنده (Intel(R) core(TM) i7-4510) 2.60 GHz و هشت گیگابایت RAM و در ویندوز ۱۰ انجام شده‌اند. در جدول (۱) "Its" نشان دهنده تعداد تکرار و "CPU" نمایش زمان اجرا است. نتایج جدول به خوبی برتری روش ارائه شده بر روش MHSS را نشان می‌دهد.

مراجع

- [1] Z.Z. Bai, M. Benzi, F. Chen, Z.Q. Wang, Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems, *IMA Journal of Numerical Analysis*, 33 (2013) 343–369.
- [2] M. Kollmann, M. Kolmbauer, A preconditioned minres solver for time-periodic parabolic optimal control problems, *Numerical Linear Algebra with Applications*, 20 (2012) 761–784.
- [3] M. Kolmbauer, U. Langer, A robust preconditioned minres solver for distributed time-periodic eddy current optimal control problems, *SIAM Journal on Scientific Computing*, 34 (2012) 785–809.
- [4] T. Rees, H.S. Dollar, A.J. Wathen, Optimal solvers for PDE-constrained optimization, *SIAM Journal on Scientific Computing*, 32 (2010) 271–298.

حمید میرچی
دانشکده ریاضی، دانشگاه گیلان، رشت، ایران.
آدرس ایمیل: hamidmirchi@gmail.com

داود خجسته سالکویه
دانشکده ریاضی، دانشگاه گیلان، رشت، ایران.
آدرس ایمیل: khojasteh@guilan.ac.ir



روش Topsis در انتخاب و رتبه‌بندی سرویس‌های ابری

امیرحسین شاه‌بخش رضوی، مهدی زعفرانی

چکیده. آنتروپی شانون یکی از روش‌های تصمیم‌گیری چند معیاره است که فقط برای وزن‌دهی شاخص‌ها و گزینه‌ها که تصمیم‌گیرنده تعیین می‌کند در مسائل مختلف از جمله انتخاب سرویس‌های ابری مورد استفاده قرار می‌گیرد و با روش topsis رتبه‌بندی گزینه‌ها انجام می‌شود. در این پژوهش بر اساس ویژگی‌های سرویس‌های ابری مقایسه‌هایی بین گزینه‌های مختلف بر اساس هر شاخص صورت می‌پذیرد و با رتبه‌بندی بین گزینه‌های تصمیم‌بهترین سرویس ابری انتخاب می‌گردد. در این مقاله برای مقایسه سرویس‌های ابری مختلف از شاخص اندازه‌گیری خدمات که بر اساس ویژگی‌های مشترک سرویس‌های ابری تعیین می‌شود، استفاده شده است.

۱. پیش‌گفتار

شبکه ابری یک شبکه‌ی جهانی، از دیتاسنترهایی است که در سرتاسر زمین پخش شده‌اند. در حقیقت ردیفی گسترده از کامپیوترهایی است که به یکدیگر متصل شده‌اند. این شبکه‌ها سرویس‌هایی چون پردازش، ذخیره‌سازی، پشتیبان‌گیری فایل‌ها، فعالیت در شبکه‌های اجتماعی و حتی خرید آنلاین را در اختیار کاربران قرار می‌دهند تا کاربران بتوانند از راه دور و از طریق اینترنت به آنها دسترسی داشته باشند [۲]. سرویس‌های ابری کاملاً از پیش پیکربندی می‌شوند تا کاربران بتوانند بدون نیاز به اعمال هیچ‌گونه تنظیمات خاصی، به بهره‌بردن از آن بپردازد، مدل‌های مختلف خدمات ابری به سه صورت ذیل برای مشتریان ارائه می‌گردد که هر کدام به طور خلاصه در ذیل توضیح داده شده است [۳].

IaaS^۱: زیر ساختی به عنوان یک سرویس IaaS و یک مرکز داده^۲ جدا با ویژگی همچون ایجاد شبکه، قدرت پردازش و ظرفیت ذخیره‌سازی است که اجازه می‌دهد کاربران به طور مستقیم به این منابع دسترسی داشته و بتوانند از آنها استفاده کنند.

PaaS^۳: پلتفرمی به صورت یک سرویس است که به صورت ساختاری با ابزار و اجزای مناسب برای ساخت، ارزیابی و راه‌اندازی اپلیکیشن ارائه شده است.

SaaS^۴: نرم‌افزاری است که به عنوان یک سرویس و بر اساس نیازهای مختلف دنیای کسب و کار طراحی شده است.

در مقاله [۴] از روش تحلیل سلسله مراتبی و در مقاله [۱] از روش تحلیل سلسله مراتبی و topsis برای وزن‌دهی معیارها و رتبه‌بندی گزینه‌های سرویس‌های ابری استفاده شده است، ولی در این پژوهش جواب‌های بدست آمده از دقت بالاتری برخوردار می‌باشد.

واژه‌های کلیدی: هوش مصنوعی، اینترنت اشیاء، مساله بهینه‌سازی چند معیاره.
طبقه‌بندی موضوعی [۲۰۱۰]: 97Rxx, 90Bxx, 97R40.

¹Infrastructure as a service

²Data center

³Platform as a service

⁴Software as a service

۲. آنتروپی شانون و روش TOPSIS

در مسائل تصمیم گیری چند معیاره و بخصوص مسائل تصمیم گیری چند شاخصه، داشتن و دانستن وزن های نسبی شاخص های موجود، گام مؤثری در فرآیند حل مسئله بوده و مورد نیاز است. ایده اصلی آنتروپی شانون آن است که هر چه پراکندگی در مقادیر یک شاخص بیشتر باشد آن شاخص از اهمیت بیشتری برخوردار است روش آنتروپی یکی از روش های تصمیم گیری چند معیاره برای محاسبه وزن معیارها می باشد، که در پنج گام ذیل خلاصه می شود [۵].

- گام ۱: ابتدا ماتریس تصمیم را تشکیل می دهیم.
- گام ۲: ماتریس بدست آمده را نرمال سازی می کنیم و هر درایه نرمال شده را P_{ij} می نامیم. نرمال شدن به این صورت می باشد که درایه هر ستون را بر مجموع ستون تقسیم می کنیم.
- گام ۳: محاسبه آنتروپی هر شاخص: آنتروپی E_j به صورت ذیل محاسبه می گردد:

$$E_j = -k \sum_{i=1}^m P_{ij} * \ln P_{ij}, \quad i = 1, 2, \dots, m. \quad (1)$$

- گام ۴: مقدار d_j (درجه انحراف) محاسبه می شود که بیان می کند شاخص مربوطه (d_j) چه میزان اطلاعات مفید برای تصمیم گیری در اختیار تصمیم گیرنده قرار می دهد. هر چه مقادیر اندازه گیری شده شاخصی به هم نزدیک باشند نشان دهنده آن است که گزینه های رقیب از نظر آن شاخص تفاوت چندانی با یکدیگر ندارند:

$$d_j = 1 - E_j. \quad (2)$$

- گام ۵: مقدار وزن W_j محاسبه می گردد. در واقع وزن معیار برابر با هر d_j تقسیم بر مجموع d_j ها می باشد:

$$W_j = \frac{d_j}{\sum d_j}. \quad (3)$$

روش TOPSIS یکی از روش های تصمیم گیری چند شاخصه است که به رتبه بندی گزینه ها می پردازد. این روش در پنج گام ذیل خلاصه می شود [۶].

- گام ۱: نرمال سازی ماتریس تصمیم که با توجه به رابطه (۲) محاسبه می شود:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (4)$$

- گام ۲: در این گام وزن معیارها (بردار $W = [w_1, \dots, w_n]$) که از روش تحلیل سلسله مراتبی به دست آمده است، در ماتریس نرمال گام ۱ ضرب می شود تا ماتریس وزن دار W' محاسبه شود:

$$w'_{ij} = w_j r_{ij}. \quad (5)$$

- گام ۳: با توجه به جنبه‌های مثبت^۵ و منفی (مانند هزینه) معیارها، در این گام نوع جواب‌های ایده‌آل مثبت و منفی در حالتی که معیار مثبت باشد، بر اساس روابط (۶) و (۷) و در حالتی که معیار منفی باشد، بر اساس روابط (۸) و (۹) مشخص می‌شود:

$$s_j^+ = \max_{i=1, \dots, m} \{w'_{ij}\}, \quad (6)$$

$$s_j^- = \min_{i=1, \dots, m} \{w'_{ij}\}, \quad (7)$$

$$s_j^+ = \min_{i=1, \dots, m} \{w'_{ij}\}, \quad (8)$$

$$s_j^- = \max_{i=1, \dots, m} \{w'_{ij}\}. \quad (9)$$

- گام ۴: در این گام بر اساس رابطه (۶)، فاصله اقلیدسی هر گزینه از جواب ایده‌آل مثبت و منفی محاسبه می‌شود:

$$d_i^+ = \sqrt{\sum_{j=1}^n (w'_{ij} - s_j^+)^2} \quad d_i^- = \sqrt{\sum_{j=1}^n (w'_{ij} - s_j^-)^2}. \quad (10)$$

- گام ۵: در این گام شاخص شباهت که نشان‌دهنده امتیاز هر گزینه است، بر اساس رابطه (۷) محاسبه می‌شود. هرچقدر این شاخص به عدد یک نزدیکتر باشد، برتری آن گزینه را نشان می‌دهد:

$$C_i = \frac{d_i^-}{d_i^- + d_i^+}. \quad (11)$$

۳. رتبه‌بندی سرویس‌های ابری

از کاربردهای سرویس ابری، مدیریت ارتباط با مشتری به صورت آنلاین است. در این بخش و به عنوان یک مثال هدف این است که وزن هر معیار (امنیت، کیفیت خدمات، هزینه و اعتبار) با روش آنتروپی شانون محاسبه شود و از بین شرکت‌های ارائه دهنده خدمات ابری مانند Amazon، Hp، Microsoft، Rackspace، Google، CenturyLink، CityCloud، Linode، GoGrid، و SoftLayer، بهترین سرور ابری برای سیستم ارتباط با مشتری با روش TOPSIS، رتبه‌بندی و انتخاب شود. به دلیل در دسترس نبودن داده‌ها بنا به سیاست‌های شرکت‌های ارائه دهنده سرویس‌های ابری، تمامی داده‌های ماتریس تصمیم X به صورت تصادفی در جدول ۱ تولید شده‌اند و بر اساس داده‌های تولید شده و روابط (۴) و (۱۰) ماتریس وزن‌دار W' که در جدول ۲ آمده است، به دست می‌آید. وزن معیارهای امنیت، کیفیت خدمات، هزینه و اعتبار که با استفاده از روش آنتروپی شانون به دست آمده است برابر با است. با محاسبه شاخص شباهت بر اساس رابطه (۱۱)، گزینه‌ها رتبه‌بندی می‌شوند که نتایج در جدول ۳ آمده است.

نتیجه‌گیری

با توجه به نیاز و استفاده روزافزون کاربران از خدمات مختلف شرکت‌های ارائه‌دهنده سرویس‌های ابری، وجود معیارهای مختلف تصمیم‌گیری برای انتخاب گزینه ارجح از میان این شرکت‌ها، کاربران با چالش‌هایی در خصوص انتخاب شرکت مناسب با توجه به اهداف خود قرار دارند. در این پژوهش با استفاده از روش آنتروپی شانون، قبل از انتخاب سرویس ابری مناسب از سوی کاربران، معیارها وزن‌دهی و مورد مقایسه قرار می‌گیرند. سپس با به کارگیری روش TOPSIS سرویس‌های ابری رتبه‌بندی می‌گردند. از این رو با توجه به ویژگی‌ها و شاخص‌های مورد نظر کاربران، ترکیب روش آنتروپی شانون و TOPSIS دقت بالاتری نسبت به روش‌های دیگر دارد، بنابراین بهترین گزینه انتخاب را در اختیار کاربران استفاده کننده از سرویس‌های ابری قرار می‌دهد.

^۵ معیار مثبت، معیاری است که افزایش آن باعث بهبود در سیستم می‌شود.

جدول ۱: داده های بدست آمده اولیه

اعتبار	هزینه	کیفیت خدمات	امنیت	سرویس های ارائه دهنده ابری
۰/۲۳۳	۲	۰/۰۲۵	۰/۸۵۳	Amazon
۰/۲۳۰	۹	۰/۵۴۱	۰/۴۳۲	HP
۰/۲۱۸	۱	۰/۹۳۹	۰/۷۶۲	Microsoft
۰/۴۵۹	۴	۰/۳۸۱	۰/۱۳۷	Rackspace
۰/۲۸۹	۱	۰/۲۱۶	۰/۴۴۵	Google
۰/۲۱۱	۷	۰/۴۴۲	۰/۷۲۱	CenturyLink
۰/۸۳۷	۷	۰/۰۲۹	۰/۲۲۸	CityCloud
۰/۵۵۶	۷	۰/۲۲۱	۰/۹۴۵	Linode
۰/۶۴۲	۱۰	۰/۴۳۷	۰/۹۰۱	Gogrid
۰/۱۸۵	۶	۰/۴۹۵	۰/۳۰۵	SoftLayer

جدول ۲: ماتریس W' و معیارهای مثبت و منفی

d_i^-	d_i^+	اعتبار	هزینه	کیفیت خدمات	امنیت	سرویس های ارائه دهنده ابری
۰/۱۲۴۶	۰/۲۴۶۶	۰/۰۳۲۳	۰/۰۲۶۵	۰/۰۰۶۳	۰/۰۷۷۵	Amazon
۰/۱۳۴۱	۰/۱۷۵۱	۰/۰۳۱۹	۰/۱۱۹۳	۰/۱۳۶۹	۰/۰۳۹۳	HP
۰/۲۶۶۵	۰/۰۸۷۵	۰/۰۳۰۲	۰/۰۱۳۳	۰/۲۳۷۷	۰/۰۶۹۳	Microsoft
۰/۱۲۶۰	۰/۱۷۲۳	۰/۰۶۳۷	۰/۰۵۳۰	۰/۰۹۶۴	۰/۰۱۲۴	Rackspace
۰/۱۳۲۴	۰/۲۰۳۳	۰/۰۴۰۱	۰/۰۱۳۳	۰/۰۵۴۷	۰/۰۴۰۴	Google
۰/۱۲۴۷	۰/۱۷۳۵	۰/۰۲۹۳	۰/۰۹۲۸	۰/۱۱۱۹	۰/۰۶۵۵	CenturyLink
۰/۰۹۹۱	۰/۲۵۲۲	۰/۱۱۶۱	۰/۰۹۲۸	۰/۰۰۷۴	۰/۰۲۰۷	CityCloud
۰/۱۰۹۹	۰/۲۰۲۲	۰/۰۷۷۱	۰/۰۹۲۸	۰/۰۵۵۹	۰/۰۸۵۹	Linode
۰/۱۴۰۵	۰/۱۷۶۴	۰/۰۸۹۱	۰/۱۳۲۵	۰/۱۱۰۶	۰/۰۸۱۹	Gogrid
۰/۱۳۱۲	۰/۱۶۹۰	۰/۰۲۵۷	۰/۰۷۹۵	۰/۱۲۵۳	۰/۰۲۷۷	SoftLayer
		۰/۱۱۶۱	۰/۰۱۳۳	۰/۲۳۷۷	۰/۰۸۵۹	معیار مثبت
		۰/۰۲۵۷	۰/۱۳۲۵	۰/۰۰۶۳	۰/۰۱۲۴	معیار منفی

جدول ۳: رتبه نهایی سرویس های ارائه دهنده ابری

رتبه	رتبه نهایی گزینه ها	سرویس های ارائه دهنده ابری
۹	۰/۳۳۵۷	Amazon
۴	۰/۴۳۳۷	HP
۱	۰/۷۵۲۸	Microsoft
۵	۰/۴۲۲۴	Rackspace
۷	۰/۳۹۴۴	Google
۶	۰/۴۱۸۲	CenturyLink
۱۰	۰/۲۸۲۱	CityCloud
۸	۰/۳۵۲۱	Linode
۲	۰/۴۴۳۴	Gogrid
۳	۰/۴۳۷۰	SoftLayer

مراجع

- [۱] ا. شاه بخش رضوی، م. زعفرانی و م. جعفرزاده. استفاده از رویکرد بهینه سازی چند معیاره برای انتخاب سرویس های ابری، چهارمین سمینار ملی کنترل بهینه سازی (۱۴۰۰).

- [2] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, J.P. Jue, All one needs to know about fog computing and related edge computing paradigms, *A complete survey Journal of Systems Architecture*, 98 (2019) 289–330.
- [3] S. Shadroo, A.M. Rahmani, Systematic survey of big data and data mining in internet of things, *Computer Networks*, 139 (2018) 19–47.
- [4] G. Nie, Q. She, D. Chen, Evaluation index system of cloud service and the purchase decision-making process based on AHP. *In Proceedings of the 2011 International Conference on Informatics, Cybernetics, and Computer Engineering*, (211) 345–352.
- [5] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, *Acm computing surveys*, 38 (2006) 13-es.
- [6] A. E. Youssef, An integrated MCDM approach for cloud service selection based on TOPSIS and BWM, *IEEE Access*, 8 (2020) 71851–71865.

Numerical Analysis

امیرحسین شاه‌بخش رضوی

دانشکده ریاضی و کامپیوتر، حکیم سبزواری، سبزوار، ایران.

آدرس ایمیل: amirhossein.shrrzz@gmail.com

مهدی زعفرانیه

دانشکده ریاضی و کامپیوتر، حکیم سبزواری، سبزوار، ایران.

آدرس ایمیل: mehdi.zaferanieh@gmail.com

انتخاب ویژگی به کمک روش ترکیبی تحلیل مولفه اصلی و بهره‌ی اطلاعات

نگین علی‌خانی

چکیده. ابعاد بالا در مجموعه داده از اصلی‌ترین چالش هنگام طبقه‌بندی داده‌ها و داده‌کاوی محسوب می‌شود زیرا ویژگی‌های زائد تأثیر منفی بر نحوه عملکرد الگوریتم‌های طبقه‌بندی دارد. بنابراین در این پژوهش جهت کاهش ابعاد مدلی ترکیبی با استفاده از بهره اطلاعات و تحلیل مولفه اصلی پیشنهاد می‌شود. مدل‌سازی این پژوهش بر اساس درخت تصمیم و جنگل تصادفی صورت می‌گیرد و در نهایت مدل‌ها نسبت به چهار پارامتر مورد ارزیابی قرار می‌گیرند. نتایج مدل‌سازی حاکی از آن است که حذف ویژگی‌های زائد و کاهش ابعاد به صورت ترکیبی سبب بهبود کارایی مدل و کاهش زمان اجرای الگوریتم می‌شود.

۱. پیش‌گفتار

با افزایش حجم داده‌ها و گسترش داده‌ها با ابعاد بالا، کاهش ابعاد یک امر ضروری در علوم داده‌کاوی^۱ و یادگیری ماشین^۲ می‌باشد. یادگیری ماشین یک شاخه‌ای از هوش مصنوعی است و به سیستم‌ها توانایی یادگیری خودکار و بهبود تجربه بدون برنامه‌ریزی صریح را می‌دهد. به عبارتی می‌توانند به داده‌ها دسترسی داشته و از آن برای یادگیری خود استفاده کنند [۱]. روش‌های یادگیری ماشین به دو بخش تقسیم می‌شوند، یادگیری بانظر^۳ (داده‌های برچسب‌دار) و یادگیری بدون ناظر^۴ (داده‌های بدون برچسب) [۶].

داده‌کاوی یک حوزه میان رشته‌ای می‌باشد و شاخه‌های مختلفی چون آمار، یادگیری ماشین، پایگاه داده و سایر زمینه‌های مرتبط را با هم تلفیق کرده است تا دانش ارزشمند و روابط نهفته در حجم بزرگی از داده‌ها را استخراج نماید. هدف داده‌کاوی یافتن الگوها و مدل‌های ناشناخته و جدید موجود در پایگاه داده‌ها است که در میان حجم عظیمی از داده‌ها پنهان هستند [۳].

امروزه ابعاد بالای داده‌ها سبب مشکلاتی چون کاهش سرعت مدل‌ها، افزایش هزینه محاسباتی و کاهش معیارهای ارزیابی می‌شود. هدف از انجام این پژوهش، بررسی یک رویکرد ترکیبی نوین برای کاهش ابعاد بر اساس دو رویکرد انتخاب ویژگی و استخراج ویژگی است که منجر به انتخاب متغیرهای مهم و تاثیرگذار و حذف متغیرهای بی‌تاثیر خواهد شد. پژوهشگران داخلی و خارجی زیادی، به کاهش ابعاد و شناسایی ویژگی‌های تاثیرگذار و مهم پرداخته‌اند که به برخی از آن‌ها اشاره می‌شود.

لابردا و ریو^۵ (۲۰۲۱) به کمک روش‌های مختلف به انتخاب ویژگی پرداختند و مقایسه‌ای بین آن‌ها صورت گرفت. در این پژوهش از روش‌هایی چون ضریب همبستگی، درخت رگرسیون و دسته‌بندی و غیره استفاده شد. محققان این پژوهش، از روش‌های پیشرو جهت انتخاب ویژگی بهره گرفتند. نتایج این پژوهش نشان می‌دهد که مدل ترکیبی تحلیل مولفه‌های اصلی و بهره اطلاعات، ابعاد داده را کاهش

واژه‌های کلیدی: طبقه‌بندی، انتخاب ویژگی، بهره اطلاعات، تحلیل مولفه اصلی.
طبقه‌بندی موضوعی [۲۰۱۰]: 62H25, 62P05, 83A05

¹Data Mining

²Machine Learning

³Supervised Learning

⁴Unsupervised Learning

⁵Laborda and Ryoo

داده، ویژگی‌های مناسب را انتخاب کرده و سبب کاهش زمان آموزش می‌شود. در نتیجه عملکرد مدل با افزایش همراه بود و دقت‌های ۹۷/۷۰ و ۹۷/۶۲ حاصل شد [۴].

روچی و نظام‌آبادی (۱۳۹۶) یک روش ترکیبی با استفاده از رویکردهای خرد جمعی برای انتخاب ویژگی در داده‌های با بعد بالا، ارائه دادند. محققان ابتدا از یک روش فیلتر برای کاهش بعد داده استفاده نمودند و سپس دو الگوریتم روزآمد پیچشی با استفاده از رویکرد خرد جمعی بر روی ویژگی‌های کاهش یافته اعمال کردند و در نهایت به ترکیب نتایج پرداختند. نتایج به دست آمده از این پژوهش حاکی از برتری روش پیشنهادی نسبت به سایر روش‌های ارائه شده است [۱].

۲. مبانی نظری

درخت تصمیم: مدل درخت تصمیم در دسته الگوریتم‌های باناظر قرار می‌گیرد که می‌توان آن را روی داده‌های اسمی و عددی پیاده‌سازی کرد. چون مانند درخت، با گره ریشه شروع می‌شود که شاخه‌های بیشتری را گسترش می‌دهد و ساختار درخت مانند می‌سازد، به آن درخت تصمیم می‌گویند. در این مدل گره‌های داخلی ویژگی‌های مجموعه داده، شاخه‌ها قوانین تصمیم و برگ‌ها خروجی نتیجه را نشان می‌دهند. در یک درخت تصمیم، دو نوع گره، یعنی گره تقسیم و گره برگ وجود دارد. از گره‌های تقسیم برای شاخه‌زدن و تصمیم‌گیری استفاده می‌شود در حالی که گره‌های برگ، خروجی این شاخه‌ها و تصمیمات‌اند و شامل شاخه‌های بیشتری نخواهند بود که این تصمیمات بر مبنای ویژگی‌های مجموعه داده صورت می‌گیرند [۷].

جنگل تصادفی: الگوریتم جنگل تصادفی از ترکیب چندین درخت تصمیم به وجود می‌آید و از روش‌های باناظر است. در این مدل روش بوت‌استرپ^۶ بر روی گروهی از درختان تصمیم که تشکیل جنگل می‌دهند، اجرا می‌شود. در الگوریتم جنگل تصادفی ابتدا بر اساس داده‌ها و ایده بوت‌استرپ چندین درخت تشکیل می‌شود و به هر درخت بر اساس روش بوت‌استرپ داده تعلق می‌گیرد. سپس یک معیار ثابت برای شاخه‌زدن و تقسیم‌بندی داده‌ها برای تمامی درختان در نظر گرفته می‌شود. سپس درختان مختلفی آموزش داده می‌شوند و در نهایت بین درختان آموزش داده شده رای‌گیری می‌شود و به عنوان دسته‌بند نهایی انتخاب می‌شود.

انتخاب ویژگی: هنگام ایجاد یک مدل یادگیری ماشین، تقریباً نادر است که همه متغیرهای مجموعه داده برای ایجاد یک مدل مفید باشند. افزودن متغیرهای اضافی قابلیت تعمیم مدل و دقت کلی دسته‌بند را کاهش و پیچیدگی کلی مدل را افزایش می‌دهند. انتخاب ویژگی یک گام اساسی و مهم برای ساخت ماهرانه مدل‌ها در مرحله پیش‌پردازش است. برای افزایش قابلیت اطمینان، افزایش تعمیم‌پذیری و جلوگیری از آموزش بیش از حد، از تکنیک‌های انتخاب ویژگی در این پژوهش استفاده می‌شود.

استخراج ویژگی: تکنیک‌های استخراج ویژگی به یک نیاز آشکار در بسیاری از فرایندها تبدیل شده‌اند. این تکنیک به کاهش حجم داده‌های غیرضروری از مجموعه داده‌ها و ایجاد مدل با هزینه‌ی محاسباتی کمتر کمک می‌کند و همچنین سرعت یادگیری و گام‌های تعمیم در فرایند یادگیری ماشین را افزایش می‌دهد. در این رویکرد به کمک یک تبدیل خطی یا غیرخطی بر روی فضای ویژگی اصلی، ویژگی‌های جدید به وجود می‌آید.

تحلیل مولفه اصلی: تحلیل مولفه‌های اصلی الگوریتمی است که برای استخراج ویژگی‌ها به کار می‌رود و یک الگوریتم بدون ناظر است. در این الگوریتم هدف تعیین خطی می‌باشد که قرار است داده‌ها بر آن تصویر شوند و اطلاعات کمتری از دست برود، آن خط ویژه همان خط تحلیل مولفه‌های اصلی است که متناظر با بزرگ‌ترین مقدار ویژه‌ی ماتریس کواریانس است. این روش ویژگی‌های بی‌ارزش و بی‌ربط را حذف می‌کند، این کار به وسیله واریانس داده‌ها انجام می‌شود یعنی جهت‌هایی که واریانس و پراکندگی بیشتری دارد و از نظر اطلاعاتی مهم‌تر و با ارزش‌تر هستند را شناسایی می‌کند و بقیه را حذف می‌کند [۵].

بهره اطلاعات: بهره اطلاعات یکی از روش‌های انتخاب ویژگی است که مستقل از دسته‌بند عمل می‌کند. از ویژگی‌های این روش ساده بودن و سرعت عمل بالای آن است. بهره اطلاعات بین دو متغیر تصادفی محاسبه می‌شود و کاهش عدم قطعیت برای یک متغیر را با ارزش مشخص متغیر دیگر اندازه‌گیری می‌کند و بعلاوه نشان می‌دهد که چقدر اطلاعات می‌تواند از یک متغیر تصادفی با مشاهده یک متغیر تصادفی دیگر به دست آید. اگر بهره اطلاعات صفر باشد، به این معنی است که دو متغیر تصادفی مستقل هستند در این صورت

^۶Bootstrap

هیچ یک از متغیرهای درباره متغیر دیگر اطلاعاتی نمی دهد.

ابعاد بالای داده ها: امروزه بسیاری از پژوهشگران در مباحث یادگیری ماشین و داده کاوی با ابعاد بالا روبه رو هستند و به دنبال شناسایی ویژگی ها و متغیرهای متناسب با مدل خود می باشند. ابعاد بالای داده ها سبب می شود که الگوریتم ها قادر به پردازش سریع و مناسب داده ها نباشند. هدف الگوریتم های کاهش ابعاد شناسایی ویژگی های مناسب و حذف ویژگی های نامناسب و زائد است که این امر سبب افزایش سرعت و دقت یک مدل و کاهش زمان ساخت آن می شود [۲].

۳. نتایج اصلی

در این پژوهش برای پیش پردازش داده ها و کاهش ابعاد روش های تحلیل مولفه های اصلی و بهره اطلاعات به صورت ترکیبی و از روش های درخت تصمیم و جنگل تصادفی برای ساخت مدل استفاده می شود. به طور کلی مدل ها در چهار دیدگاه بررسی می شوند. در دیدگاه اول ابتدا الگوریتم جنگل تصادفی بر روی داده های خام اجرا می شود و کاهش ابعاد انجام نمی شود. در دیدگاه دوم ابتدا بر روی داده ها با روش تحلیل مولفه های اصلی استخراج ویژگی انجام می شود سپس جنگل تصادفی بر روی داده ها اجرا می شود. در دیدگاه سوم ابتدا با روش بهره ای اطلاعات انتخاب ویژگی انجام می شود سپس جنگل تصادفی بر روی داده ها پیاده سازی می شود. در دیدگاه چهارم و آخر ابتدا با روش تحلیل مولفه های اصلی استخراج ویژگی سپس با روش بهره ای اطلاعات انتخاب ویژگی انجام می شود سپس الگوریتم جنگل تصادفی بر روی داده ها پیاده سازی می شود. در نهایت مقایسه ای بین دیدگاه ها انجام می شود و عملکرد الگوریتم جنگل تصادفی بررسی می شود.

در این پژوهش رویکرد ارائه شده بر روی مجموعه داده بانک آلمانی که در سایت دانشگاه UCR^۷ در دسترس است، پیاده سازی می شود. در ابتدا مدل ها بر اساس کل داده ها و بدون کاهش ابعاد ساخته می شوند و سپس در چهار دیدگاه و نسبت به چهار پارامتر (ضریب همبستگی متیوز^۸ (MCC)، صحت^۹ (ACC)، خطای نوع اول^{۱۰} (EI) و خطای نوع دوم^{۱۱} (EII)) به کمک ماتریس در هم ریختگی مورد ارزیابی قرار می گیرند. از آن جایی که دیدگاه چهارم، که ارائه مدل ترکیبی است، نتایج قابل قبولی نسبت به سایر دیدگاه ها دارا می باشد، در جدول ۱ نتایج حاصل از مدل سازی، بر روی داده های خام (دیدگاه اول) و روش ترکیبی جهت کاهش ابعاد (دیدگاه چهارم) جهت مقایسه کارایی مدل، آورده شده است.

جدول ۱: نتایج حاصل از دیدگاه اول و چهارم

مدل های ارزیابی				مدل ها	دیدگاه ها
EII	EI	MCC	ACC		
۰/۵۰۸	۰/۱۰۶	۰/۴۲۴	۰/۷۷۵	جنگل تصادفی	اول
۰/۷۴۵	۰/۰۶۳	۰/۲۶۷	۰/۷۳۵	درخت تصمیم	
۰/۵۰۸	۰/۰۶۳	۰/۴۹۷	۰/۸۰۵	جنگل تصادفی	چهارم
۰/۴۹۱	۰/۱۲۰	۰/۴۱۷	۰/۷۷۰	درخت تصمیم	

۴. دست آوردهای پژوهش

طبق جدول ۱ در دیدگاه اول مشاهده می شود که عملکرد مدل جنگل تصادفی از درخت تصمیم بهتر بوده است و در دیدگاه چهارم نیز نتایج مشابهی دیده می شود. اما با مقایسه نتایج دو دیدگاه، یعنی مدل ساده و مدل ترکیبی دیده می شود که خطای نوع دوم در جنگل

^۷<https://library.ucr.edu/research-services/databases>

^۸Matthews correlation coefficient

^۹Accuracy

^{۱۰}Type I error

^{۱۱}Type II error

تصادفی ثابت بوده و در درخت تصمیم نیز خطای نوع دوم اندکی افزایش پیدا کرده است ولی در سایر پارامترها مدل‌های ارزیابی برای جنگل تصادفی و درخت تصمیم بهبود پیدا کرده‌اند. پس می‌توان گفت کاهش ابعاد و حذف ویژگی‌های بی‌ارزش به صورت ترکیبی تقریباً سبب کاهش خطای نوع اول و دوم و بهبود کارایی مدل‌ها می‌شود.

به طور کلی از آنجایی که تحلیل مولفه‌های اصلی جزو روش‌های قوی برای کاهش ابعاد و استخراج ویژگی است با ترکیب این روش با بهره اطلاعات یک الگوریتم ترکیبی قوی برای کاهش ابعاد به وجود می‌آید. با وجود پژوهش‌های متفاوت برای کاهش ابعاد به طور قطعی نمی‌توان گفت کدام روش‌ها از عملکرد مناسب‌تری برخوردار هستند اما در این پژوهش با استفاده از روش‌های ترکیبی که توانایی بسیار بالایی در انتخاب ویژگی در ابعاد بالا و همچنین تشخیص روابط خطی و غیرخطی دارند، روش قدرتمندی برای انتخاب ویژگی‌های مهم و تاثیرگذار به وجود آمده است که سبب شناسایی دقیق‌تر ویژگی‌های مناسب و کاهش زمان اجرای الگوریتم می‌شود.

سپاس‌گزاری

از اساتید و تمامی کسانی که در تهیه این مقاله من را یاری نموده‌اند کمال تشکر را دارم.

مراجع

- [۱] الف. روحی و ح. نظام‌آبادی‌پور، یک روش انتخاب ویژگی ترکیبی برای داده‌های با بعد بالا مبتنی بر خرد جمعی، نشریه مهندسی برق و مهندسی کامپیوتر ایران، دوره ۱۵، (۱۳۹۶) ۲۸۳-۲۹۴.
- [۲] ف. هلاکو، م. افتخاری و ع. اسماعیلی‌زاده کشکوئی، ترکیب ضریب همبستگی و رگرسیون ريج برای انتخاب ویژگی‌های مهم در بانک اطلاعاتی چند شکلی‌های تک نوکلئوتیدی SNP، نوزدهمین کنفرانس مهندسی برق ایران، (۱۳۹۰) ۳۱۹۲-۳۱۹۷.
- [3] J. Han, J. Pei, M. Kamber, *Data mining: concepts and techniques*, Elsevier. 2011.
- [4] J. Laborda, S. Ryoo, Feature Selection in a Credit Scoring Model. *Mathematics*, 9 (2021) 746.
- [5] E.O. Omuya, G.O. Okeyo, M.W. Kimwele, Feature selection for classification using principal component analysis and information gain. *Expert Systems with Applications*, 174 (2021) no. 114765.
- [6] I. Pan, R.M. Lachlan, O.K. Matar, Data-centric Engineering: integrating simulation, machine learning and statistics. Challenges and opportunities, *Chemical Engineering Science*, 249 (2022) no. 117271.
- [7] L. Zhou, Y.W. Si, H. Fujita, Predicting the listing statuses of Chinese-listed companies using decision trees combined with an improved filter feature selection method, *Knowledge-Based Systems*, 128 (2017) 93-101.

نگین علی‌خانی

دانشکده علوم پایه، دانشگاه آیت الله بروجردی، بروجرد، ایران

آدرس ایمیل: negin.alikhani750406@gmail.com

یک روش عددی برای حل معادله اول پینلوف با استفاده از پایه های موجکی شودر

مجید عرفانیان، حامد زیدآبادی

چکیده. در این مقاله یک روش عددی بر مبنای موجک های شودر برای حل معادله اول پینلوف با تبدیل آن به معادله انتگرال ولترا همرشین ارائه شده است. در ادامه کران بالا و همگرایی روش با استفاده از قضیه نقطه ثابت باناخ اثبات شده است. از آنجاییکه جواب تحلیلی و دقیق این معادله تاکنون نامعلوم است، جواب تقریبی بدست آمده را با چند روش مقایسه نمودیم و شکل تقریبی نیز در ادامه رسم شده است.

۱. مقدمه

معادلات انتگرال در شاخه های مختلف ریاضی و ریاضی فیزیک به طور گسترده ای ظاهر می شوند، و از آن جا که معادلات انتگرال نسبت به معادلات دیفرانسیل کارآمدتر می باشد، بنابراین بسیاری از مسائل مقدار اولیه و مقدار مرزی مربوط به معادلات دیفرانسیل معمولی و جزئی، را که می تواند به معادلات انتگرال تبدیل شوند از این طریق حل می نماییم.

۲. روش حل عددی

در این مقاله با استفاده از یک عملگر و پایه های شودر یک روش عددی برای حل معادله اول پینلوف زیر ارائه می دهیم.

$$\frac{\partial^2 u}{\partial x^2} = \epsilon u^2(x) + x, \quad u(0) = 0, \quad u'(0) = 1, \quad (1)$$

با دو بار انتگرال گیری از معادله (۱) نسبت به x و اعمال شرایط اولیه در مساله خواهیم داشت

$$u(x) = x + \frac{1}{\epsilon} x^3 + \epsilon \int_0^x (x-t) u^2(t) dt, \quad (2)$$

عملگر انتگرال $T : (X, \|\cdot\|_\infty) \rightarrow (X, \|\cdot\|_\infty)$ را در نظر می گیریم، در این صورت

$$(Tu)x = f(x) + \int_0^x K(x,t) W(x,t) dt, \quad (3)$$

پایه های شودر $\{b_n^*\}_{n \geq 1}$ در فضای باناخ $C([\alpha, \alpha + \beta])$ و پایه های شودر $\{B_n^*\}_{n \geq 1}$ در فضای باناخ $C([\alpha, \alpha + \beta]^2)$ هر دو شامل یک سوپرنرم مشخص می باشند. ما به دنبال یک دنباله متراکم $\{t_n\}_{n \geq 1}$ متشکل از نقاط مجزای $t_1 = \alpha, t_2 = \alpha + \beta$ در بازه $[\alpha, \alpha + \beta]$ می باشیم. در این مقاله از پایه های شودر در فضای باناخ $C[0, 1]$ استفاده می کنیم، یعنی $t_1 = 0, t_2 = 1$ انتخاب و

نقاط دیگر نیز از رابطه $t_{i+1} = \frac{2^k + 1}{2^{n+1}}$, $i = 2^n + k + 1$, که $0 \leq k < 2^n$ و k, n اعداد صحیح هستند. برای این منظور برای $t \in [\alpha, \alpha + \beta]$ که $n \geq 1$, b_n تابع تکه ای پیوسته $b_n(t) := 1$ را در نقاط مشخص $\{t_j : 1 \leq j \leq n\}$ است را بصورت زیر تعریف می کنیم

$$b_n(t_n) = 1, \quad b_n(t_k) = 0, \quad \forall k < n. \quad (4)$$

برای دنباله توابع دو متعامد $\{b_n^*\}_{n \geq 1}$ و عملگر متناظر با این دنباله $\{P_n\}_{n \geq 1}$ که برای هر $x \in C([\alpha, \alpha + \beta])$ دارای جواب یکسانی هستند. لذا بر اساس منبع ([1]) داریم

$$b_1^*(x) = x(t_1), \quad b_n^*(x) = x(t_n) - \sum_{k=1}^{n-1} b_k^*(x)b_k(t_n), \quad n \geq 2, \quad (5)$$

که $P_n(\sum_{n \geq 1} \Lambda_n x_n) = \sum_{k=1}^n \Lambda_k x_k$ برای ساخت پایه های شودر $\{b_n\}_{n \geq 0}$ در فضای باناخ $C([\alpha, \alpha + \beta]^2)$ یک نگاشت دوسویی بر اساس منبع ([1]) به صورت $\tau = (\tau_1, \tau_2) : N \rightarrow N * N$ تعریف می کنیم، که $[l]$ نشان دهنده جزء صحیح عدد l می باشد.

$$\tau(l) := \begin{cases} (\sqrt{l}, \sqrt{l}), & \text{if } [\sqrt{l}] = \sqrt{l}, \\ (l - [\sqrt{l}]^2, [\sqrt{l}] + 1), & \text{if } 0 < l - [\sqrt{l}]^2 \leq [\sqrt{l}], \\ ([\sqrt{l}] + 1, l - [\sqrt{l}]^2 - [\sqrt{l}]), & \text{if } [\sqrt{l}] < l - [\sqrt{l}]^2, \end{cases} \quad (6)$$

که $b_0(t) = 1$ برای $t \in [a, b]$ و $b_n(t_i) = 0$ و $n \geq 0$ ، اگر $b_n(t_i) = 1$ و $i < n$ ، بنابراین پایه های شودر $\{B_n\}_{n \geq 0}$ ، برای هر $n \in \mathbb{N}$ با $\tau(n) = (i, j)$ ، برای هر $t, s \in [\alpha, \alpha + \beta]$ بصورت $B_n(t, s) := b_i(t)b_j(s)$ تعریف می شوند. از آنجائیکه جواب صریح انتگرال وجود ندارد، تعیین یک دنباله $\{T^n(u_0)\}_{n \in \mathbb{N}}$ به صورت صریح امکان پذیر نیست. از این رو یک دنباله جدید از توابع مانند $\{u_i\}_{i \in \mathbb{N}}$ تعریف می کنیم که به صورت بازگشتی و با استفاده از پایه های شودر تولید می گردند. لذا خواهیم داشت

$$u_i(x) := f(x) + \int_0^x K(x, t)W(t, u_{i-1}(t))dt. \quad (7)$$

اگر قرار دهیم $\psi_{i-1}(x, t) := K(x, t)W(t, u_{i-1}(t))$ می توانیم $\psi(t, s)$ را بر حسب پایه های شودر به صورت زیر بسط دهیم $\psi(x, t) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} k_{ij}b_i(x)b_j(t)$ ، اگر Q_m یک تصویر متعامد باشد، در این صورت با استفاده از شرایط درونیابی داریم

$$Q_m(\psi(x, t)) = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} k_{ij}b_i(x)b_j(t). \quad (8)$$

بنابراین برای معادله انتگرال (2) خواهیم داشت

$$u_i(x) := f(x) + \int_0^x Q_m(\psi_{i-1}(x, t))dt. \quad (9)$$

۳. تحلیل خطا

در این بخش با استفاده از قضیه نقطه ثابت باناخ یک کران بالا برای روش ارائه شده به دست می آوریم.

لم ۱. فرض کنید $W \in C^1([0, 1] \times \mathbb{R})$ توابعی پیوسته و لیپشیتز با ثابت لیپشیتز L و $K : [0, 1]^2 \rightarrow \mathbb{R}$ تابعی کراندار با کران M باشد. در این صورت برای هر $u_0 \in C([0, 1])$ عملگر T تعریف شده توسط (۲) دارای یک نقطه ثابت مانند u است، و اگر $q := ML < 1$ آنگاه

$$\|u - T^i(u_0)\|_\infty \leq \|T(u_0) - u_0\|_\infty \times \sum_{j=i}^{\infty} q^j. \quad (۱۰)$$

قضیه ۱. فرض کنید $\psi_{i-1} \in C([0, 1]^2)$ و $\{u_i\}_{i \geq 1}$ یک زیر مجموعه از $C([0, 1])$ باشد و $W \in C([0, 1] \times \mathbb{R})$ تابعی لیپشیتز نسبت به متغیر دوم باشند. در این صورت ε_j برای $i, j = 1, 2, \dots$ اعداد مثبتی هستند و روش دارای مرتبه همگرایی ۲ می باشد و

$$\|u - u_i\|_\infty \leq \|T(u_0) - u_0\|_\infty \sum_{j=i}^{\infty} q^j + \sum_{j=1}^i q^{i-j} \varepsilon_j.$$

Numerical Analysis

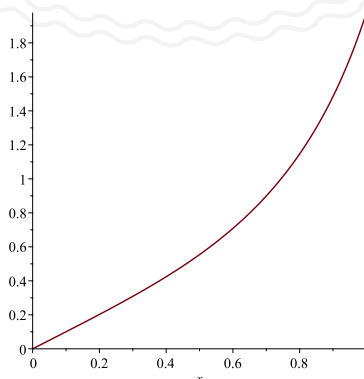
۴. مثالهای عددی

در این بخش معادله اول پنیلوف را با روش ارائه شده در بخش دوم به صورت عددی حل نموده و با چند روش دیگر مقایسه می نماییم. علاوه بر این شکل تقریبی معادله اول پنیلوف رسم شده است.

مثال ۱. معادله اول پنیلوف بیان شده در (۲) را با روش های چند جمله ای های چیشف، روش هم محلی، روش هموتویی، روش آدومین مقایسه نموده و در جدول زیر بیان شده است.

جدول ۱: جواب تقریبی بدست آمده برای معادله اول پنیلوف

x_i	[۳] VIM $n = 4$	[۲] MADM $n = 6$	[۲] ADM $n = 9$	Schauder $n = 6$
۱.۰	۱۰۰۲۱۶.۰	۱۰۰۲۶۰.۰	۱۰۰۲۱۵.۰	۱۰۰۲۱۶.۰
۲.۰	۲۰۲۱۳۹.۰	۲۰۲۱۲۸.۰	۲۰۲۱۱۷.۰	۲۰۲۱۳۹.۰
۳.۰	۳۰۸۶۳۰.۰	۳۰۸۶۳۰.۰	۳۰۸۶۳۰.۰	۳۰۸۶۳۰.۰
۴.۰	۴۲۳۹۸۶.۰	۴۲۳۹۸۵.۰	۴۲۳۹۸۶.۰	۴۲۳۹۸۶.۰
۵.۰	۵۵۴۳۳۹.۰	۵۵۴۳۳۷.۰	۵۵۴۳۳۵.۰	۵۵۴۳۴۰.۰
۷.۰	۸۹۹۲۲۹.۰	۸۹۹۲۱۹.۰	۸۹۹۲۱۷.۰	۸۹۹۲۸۹.۰
۹.۰	۴۸۱۷۷۸.۱	۴۸۱۴۸۸.۱	۴۸۱۲۰۱.۱	۴۸۴۷۸۷.۱



شکل ۱: جواب های تقریبی معادله پنیلوف.

مراجع

- [1] Z. Semadeni, Product Schauder bases and approximation with nodes in spaces of continuous functions, *Bulletin of the Polish Academy of Science*, 11 (1963) 387–391.
- [2] S.S. Behzadi, Convergence of iterative methods for solving Painleve equation, *Applied Mathematical Sciences*, 4 (2010) 1489–1507.
- [3] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral methods in Fluid Dynamic*, Englewood cliffs, N.J. Prentice-Hall, 1988.

مجید عرفانیان

دانشیار گروه ریاضی دانشکده علوم، دانشگاه زابل (نویسنده مسئول).

آدرس ایمیل: erfaniyan@uoz.ac.ir

حامد زیدآبادی

استادیار گروه ریاضی دانشگاه حکیم سبزواری

آدرس ایمیل: h.zeidabadi@yahoo.com

حل عددی معادله فیشر با استفاده از شبکه عصبی عمیق لژاندر

مریم بابائی، علیرضا افضل آقائی نائینی، کورش پرند

چکیده. در این مقاله به معرفی یک روش عددی کارا جهت حل انواع معادلات دیفرانسیل بر اساس الگوریتم های یادگیری عمیق می پردازیم. معماری شبکه عصبی عمیق لژاندر (LDNN) با بهره گیری از چند جمله ای های متعامد به عنوان توابع فعالیت، جهت افزایش قابلیت یادگیری شبکه عصبی برای حل معادلات دیفرانسیل ارائه شده است. نتایج حاصله بیانگر برتری مدل ارائه شده به نسبت شبکه های عصبی فیزیک آگاه (PINN) می باشد.

۱. پیش گفتار

شبکه عصبی مصنوعی زیرمجموعه ای از روش های یادگیری ماشین است که مکانیسم یادگیری در موجودات بیولوژیکی را شبیه سازی می کند. یک شبکه عصبی، گرافی جهت دار است که راس های آن (نورون ها) توسط یال های وزن دار به هم متصل شده اند. فرایند یادگیری در یک شبکه عصبی شامل یافتن وزن های مجهول گراف می باشد. این روال در هر تکرار شامل دو مرحله ی حرکت جلوسو^۱ و عملیات پس انتشار^۲ است. در مرحله ی نخست ورودی ها در شبکه به جریان می افتند و خروجی تولید می گردد. سپس مقدار بدست آمده توسط شبکه با داده های معتبر مقایسه می شوند و میزان خطای شبکه محاسبه می گردد. در مرحله پس انتشار خطا ابتدا به کمک مشتق جزئی تابع ضرر، سهم خطای هر وزن محاسبه شده و سپس به کمک یک الگوریتم بهینه ساز وزن های بهینه محاسبه و گراف به روزرسانی می گردد. یکی از گام های توسعه شبکه های عصبی، طراحی معماری شبکه ها است. در معماری شبکه های عصبی، عواملی مانند تعداد لایه های پنهان، تعداد نورون ها در هر لایه، توابع فعالیت و الگوریتم آموزش در نظر گرفته می شوند [۱]. توابع فعالیت $\tanh(x)$ و $\text{relu}(x) = \max(0, x)$ از رایج ترین توابع مورد استفاده در این معماری ها می باشد.

رییسی و همکاران [۴] این مدل ها را جهت تقریب جواب عددی معادلات دیفرانسیل با مشتقات جزئی تعمیم داده اند. این معماری که به نام شبکه های عصبی فیزیک آگاه^۳ شناخته می شود، بر اساس اکتشاف داده محور^۴ و راه حل داده محور^۵ به حل معادلات و همچنین یافتن پارامترهای مجهول مدل به کمک داده های واقعی می پردازد.

در این تحقیق با توسعه معماری شبکه عصبی فیزیک آگاه و استفاده از توابع متعامد لژاندر به عنوان توابع فعالیت جدید در مدل، معماری شبکه عصبی عمیق لژاندر را توسعه می دهیم. سپس صحت و دقت این معماری در حل معادله واکش-انتشار فیشر به نسبت معماری کلاسیک بررسی می شوند.

واژه های کلیدی: معادلات دیفرانسیل، کووید-۱۹، یادگیری عمیق.
طبقه بندی موضوعی [۲۰۱۰]: 92C60, 34A34, 68T20

¹Feedforward

²Backpropagation

³Physics-informed neural networks

⁴Data Driven Discovery

⁵Data Driven Solution

۲. چندجمله‌ای‌های متعامد لژاندر

چندجمله‌ای‌های لژاندر سیستم کاملی از چندجمله‌ای هاست که دو به دو بر یکدیگر عمودند. این توابع که در بازه $[-1, 1]$ تعریف می‌شوند، نسبت به تابع وزن $w(x) \equiv 1$ بر اساس تعریف

$$\langle f, g \rangle_w = \int_a^b f(x)g(x)w(x)dx,$$

خاصیت تعامد دارند. جهت تعریف این چندجمله‌ای‌ها می‌توان از معادلات اشتورم-لیوویل که به صورت

$$\frac{d}{dx}((1-x^2)\frac{d}{dx}P_n(x)) + \lambda_n P_n(x) = 0,$$

بیان می‌شوند، استفاده کرد [۲].

Numerical Analysis

۳. شبکه عصبی لژاندر

شبکه عصبی لژاندر با استفاده از ترکیب دو شبکه متوالی متصل به یکدیگر بدست می‌آید. در این معماری ابتدا از یک شبکه عصبی عمیق چندلایه جلوسو استفاده شده‌است که در لایه ابتدایی آن چند جمله‌ای‌های لژاندر به عنوان توابع فعال‌ساز به کار رفته‌اند. این چند جمله‌ای‌ها به صورت پویا برای ورودی‌های نرمال شده در بازه $[-1, 1]$ در شبکه تولید می‌شوند. و خروجی آن‌ها در لایه‌های دیگر به تابع فعال‌ساز $\tanh(x)$ داده می‌شود. ترکیب این لایه‌ها باعث می‌شود علاوه بر استفاده از روش قدرتمند طیفی، بتوانیم ویژگی جواب‌ها را نیز به ابعاد بالاتر ببریم و محدود به چند جمله‌ای‌ها نباشیم [۲]. در این صورت شبکه اول با n لایه به صورت زیر تعریف می‌شود:

$$\begin{aligned} \mathcal{H}_0 &= x, & x &\in \mathbb{R}^d, \\ \mathcal{H}_1 &= P(W^{(1)}\mathcal{H}_0 + b^{(1)}), \\ \mathcal{H}_i &= \tanh(W^{(i)}\mathcal{H}_{i-1} + b^{(i)}), \quad i = 2, 3, \dots, n-1, \\ \mathcal{H}_n &= W^{(n)}\mathcal{H}_{n-1} + b^{(n)}. \end{aligned}$$

که در آن \mathcal{H}_0 لایه ورودی با بعد d ، \mathcal{H}_1 لایه ابتدایی با تابع فعالیت‌های چند جمله‌ای‌های لژاندر و $2 \leq i \leq n-1$ ، \mathcal{H}_i پنهان با تابع فعالیت تانژانت هایپربولیک است.

شبکه‌های دوم این معماری با استفاده از گره‌های عملیاتی مشتق و ترکیب توابع آن‌ها تشکیل شده‌اند. این شبکه به محاسبه مقدار تابع هدف در گام اولیه با استفاده از روش تفاضل متناهی و خروجی شبکه قبل و همچنین مقدار تابع در شرایط مرزی در زمان ثانویه می‌پردازد. خروجی شبکه دوم با توجه به روش تفاضل متناهی به صورت زیر محاسبه می‌شود:

$$u_t + N(u) = 0, \quad (1)$$

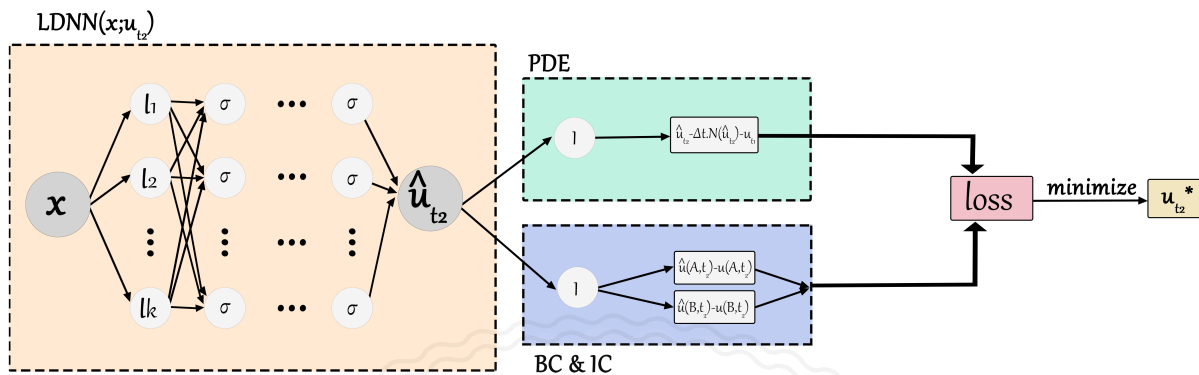
$$\frac{u_1 - u_0}{\Delta t} + N(u_1) = 0, \quad (2)$$

$$h := u_0 = u_1 + \Delta t N(u_1), \quad (3)$$

بنابراین تابع ضرر در این مدل به صورت زیر تعریف می‌شود:

$$Loss = MSE_h + MSE_{BC}.$$

در اینجا MSE_h میانگین مربعات خطا مقدار $h - u_0$ و MSE_{BC} میانگین مربعات خطا تابع باقیمانده برای شرایط اولیه و مرزی می‌باشد. جهت بدست آوردن یک تقریب مناسب از جواب معادله، تابع ضرر توسط الگوریتم مرتبه اول Adam و بهینه‌ساز شبه‌نیوتنی L-BFGS-B به سمت مقدار کمینه پیش می‌رود. شکل (۱) ساختار این شبکه را به صورت کلی نشان می‌دهد.



شکل ۱: معماری شبکه عصبی لژاندر برای حل معادلات فیشر

جدول ۱: جدول مقایسه نتایج در شبکه‌های LDNN و PINN.

شماره مثال	خطای مدل PINN	خطای مدل LDNN
۱	6×10^{-5}	$2/9 \times 10^{-5}$
۲	$7/5 \times 10^{-4}$	5×10^{-4}

۴. شبیه‌سازی و نتایج

معادله فیشر از معادلات پر کاربرد واکنش انتشار است که در پویایی‌شناسی جمعیت، فرایندهای تصادفی، تکثیر ژنتیکی و نیز مدل نمونه اولیه برای شعله در حال گسترش و ... کاربرد دارد. در این مقاله به حل معادله مرتبه ششم و یک حالت غیرخطی این گروه از معادلات می‌پردازیم.

مثال ۱. (معادله مرتبه ششم فیشر) با در نظر گرفتن $\alpha = 1$ و $\beta = 6$ معادله‌ی زیر بدست می‌آید

$$u_t = u_{xx} + u(1 - u^6), \quad u(x, 0) = \frac{1}{\sqrt[3]{1 + e^{(3/2)x}}}. \quad (4)$$

که این معادله دارای جواب دقیق قطعی به فرم $u(x, t) = \{1/2 \tanh[-3/4(x - 5/2t)] + 1/2\}^{1/3}$ می‌باشد. با توجه به فرم معادله و داشتن مقدار تابع در نقاط اولیه $t_1 = 0/1$ با بکارگیری شبکه عصبی پیشنهادی مقدار تابع هدف در زمان $t_2 = 0/9$ تخمین زده شده‌است.

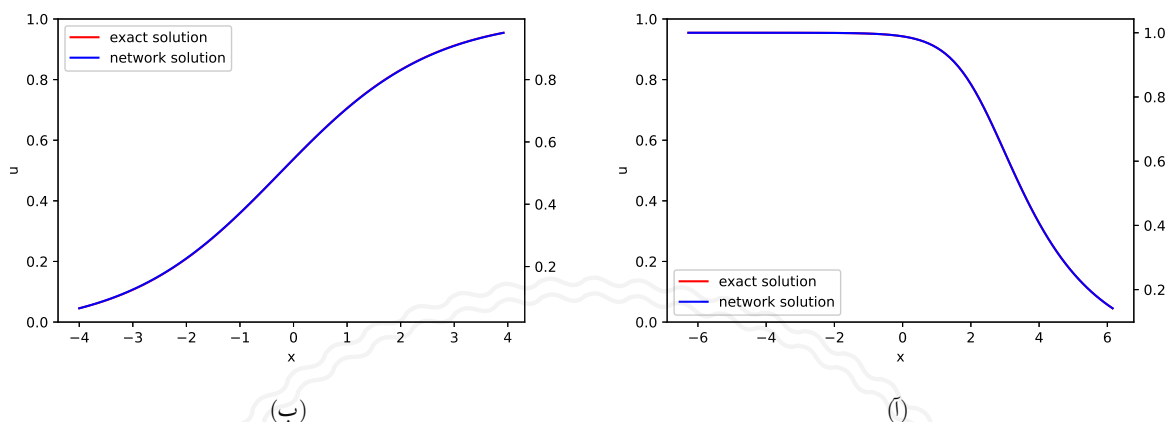
مثال ۲. (معادله غیرخطی فیشر) فرم کلی معادله غیرخطی فیشر به صورت زیر است

$$u_t = u_{xx} + u(1 - u)(u - a), \quad 0 < a < 1, \quad u(x, 0) = \frac{1}{1 + e^{(-1/\sqrt{2})x}}. \quad (5)$$

که این معادله دارای جواب دقیق قطعی به فرم

$$u(x, t) = \frac{1}{(1 + e^{-x/\sqrt{2} - (1/2-a)t})},$$

می‌باشد. با در نظر گرفتن $a = 0/3$ و مقدار تابع هدف در نقطه‌ی $t_1 = 0/1$ مقدار تابع هدف در نقطه‌ی $t_2 = 0/9$ تخمین زده شده است.



شکل ۲: مقایسه جواب تحلیلی و جواب تقریبی بدست آمده برای (آ) مثال ۱ و (ب) مثال ۲.

مراجع

- [1] I.J. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.
- [2] Z. Hajimohammadi, K. Parand, A. Ghodsi, Legendre Deep Neural Network (LDNN) and its application for approximation of nonlinear Volterra Fredholm Hammerstein integral equations, (2021), arXiv preprint arXiv:2106.14320.
- [3] K. Parand, A.A. Aghaei, M. Jani, A. Ghodsi, A new approach to the numerical solution of Fredholm integral equations using least squares-support vector regression, *Mathematics and Computers in Simulation*, 180 (2021) 114–128.
- [4] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational physics*, 378 (2019) 686–707.

مریم بابائی

گروه علوم داده‌ها و کامپیوتر، دانشکده ریاضی، دانشگاه شهید بهشتی، تهران، ایران
آدرس ایمیل: mar_babaei@sbu.ac.ir

علیرضا افضل آقائی نائینی

گروه علوم داده‌ها و کامپیوتر، دانشکده ریاضی، دانشگاه شهید بهشتی، تهران، ایران
آدرس ایمیل: alirezaafzalaghaei@gmail.com

کوروش پرند

گروه علوم داده‌ها و کامپیوتر، دانشکده ریاضی، دانشگاه شهید بهشتی، تهران، ایران
آدرس ایمیل: k_parand@sbu.ac.ir

کاربرد توابع متعامد لژاندر در تحلیل مولفه‌های اصلی جهت افزایش دقت روش‌های یادگیری ماشین

زهره بهروزه، علیرضا افضل آقائی نائینی، کوروش پرند، مسعود حجاریان

چکیده. در این پژوهش توابع متعامد لژاندر کسری در روش تحلیل مولفه‌های اصلی مبتنی بر هسته مورد استفاده قرار می‌گیرند. این الگوریتم پس از اعمال نگاشت ویژگی، به کمک تجزیه مقادیر ویژه فرایند کاهش ابعاد داده را انجام می‌دهد. دقت مدل ارائه شده توسط رگرسیون خطی و لاجستیک بر روی چند مجموعه داده بررسی شده و نتایج حاصله بیانگر دقت بالای روش ارائه شده در برابر توابع هسته رایج می‌باشد.

۱. پیش‌گفتار

افزایش توانایی یادگیری در مدل‌های یادگیری ماشین از اصلی‌ترین اهداف دانشمندان این حوزه است. از این رو دانشمندان مدل‌های مختلفی همچون رگرسیون خطی و لاجستیک، ماشین بردار پشتیبان و شبکه عصبی را توسعه داده‌اند. مدل‌های خطی، علی‌رغم سادگی آموزش، توانایی یادگیری الگوهای پیچیده غیرخطی در فضای داده‌ها را ندارند. روش‌های مبتنی بر هسته جهت غلبه بر این مشکل ارائه شده‌اند. در این پژوهش چندجمله‌ای‌های لژاندر کسری به عنوان توابع هسته روش تحلیل مولفه‌های اصلی ارائه می‌شوند. هنگام کار با داده‌های ابعاد بالا دچار مشکلات زیادی مانند: افزایش زمان محاسبه، درگیری بخش زیادی از حافظه و پدیده‌ای به نام مشقت بعدچندی خواهیم شد. این پدیده به این معنا می‌باشد که با بیشتر شدن حجم ابعاد، حجم فضا به گونه‌ای سریع افزایش می‌یابد که داده‌های موجود پراکنده و تُنک می‌شوند. دانشمندان برای کاهش اثر این مشکل روش‌های کاهش ابعاد داده با نام‌های استخراج ویژگی و انتخاب ویژگی را معرفی نمودند. تحلیل مولفه‌های اصلی از پرکاربردترین روش‌های استخراج ویژگی است. این الگوریتم، با استفاده از تجزیه مقادیر ویژه [۲]، پراهمیت‌ترین ویژگی‌های پنهان داده‌ها را استخراج می‌کند. در ادامه به توضیح این روش می‌پردازیم.

۱.۱. تحلیل مولفه‌های اصلی

در این رویکرد تلاش می‌شود به کمک ترکیب خطی ویژگی‌های موجود، ویژگی‌های جدیدی تولید شود. [۴] این روش با یافتن جهت‌های بیشترین واریانس در داده‌ها، یک دستگاه مختصات جدید ارائه می‌کند. در این دستگاه جدید اولین بُعد متناظر با جهت بیشترین واریانس، دومین بُعد متناظر با جهت دومین بیشترین واریانس و آخرین بُعد بیانگر جهت کمترین واریانس در میان داده‌ها است. از این رو با انتخاب k بُعد ابتدای این دستگاه می‌توان ویژگی‌های استخراج شده و مفیدی بدست آورد. در قضیه زیر به بیان دقیق این الگوریتم می‌پردازیم.

قضیه ۱. فرض کنید مجموعه داده X با ابعاد $d \times n$ با میانگین ستونی صفر داده شده است. در این صورت جهت‌های بیشترین واریانس بردارهای داده‌ها، توسط بردارهای ویژه ماتریس کواریانس ارائه می‌شود.

واژه‌های کلیدی: تحلیل مولفه‌های اصلی، توابع لژاندر کسری، کاهش ابعاد.
طبقه‌بندی موضوعی [۲۰۱۰]: 15A23, 68T10.

اثبات. با فرض صفر بودن میانگین ویژگی‌های مجموعه داده، ماتریس کواریانس به صورت

$$C = \frac{1}{n} \sum_{i=0}^n x_i x_i^T, \quad (1)$$

نمایش داده خواهد شد. فرض کنید بردار u_1 جهت بیشترین واریانس داده‌ها را نمایش دهد. برای یافتن این بردار مسئله بهینه‌سازی زیر را تشکیل می‌دهیم:

$$\begin{aligned} \max_{u_1} \quad & \frac{1}{n} u_1^T C u_1 \\ \text{s.t.} \quad & \|u_1\| = 1. \end{aligned} \quad (2)$$

محاسبه فرم دوگان ۲ مسئله مقدار ویژه‌ی زیر را نتیجه می‌دهد:

$$C \cdot u_1 = \lambda \cdot u_1. \quad (3)$$

بنابراین برای محاسبه جهت بیشترین واریانس، بزرگترین مقدار ویژه و بردارهای ویژه متناظر آن از ماتریس کواریانس انتخاب می‌شوند. به منظور یافتن دومین جهت بیشترین واریانس، دومین بزرگترین مقدار ویژه محاسبه می‌گردد. □

در روش تحلیل مولفه‌های اصلی مبتنی بر هسته ایده کلی به این صورت است که مجموعه داده را در فضای با ابعاد بالاتر بررسی می‌کنیم. فرض کنید نگاشت غیر خطی ϕ همان نگاشتی است که مجموعه داده را به فضای با ابعاد بالاتر انتقال می‌دهد و آن را نگاشت ویژگی می‌نامیم که به صورت $x \rightarrow \phi(x)$ تعریف می‌شود. حال هسته را به عنوان تابعی معرفی می‌کنیم که ضرب داخلی مجموعه داده را تحت نگاشت ϕ را محاسبه می‌نماید:

$$K(x_i, x_j) = \phi(x_i) \phi(x_j)^T.$$

کافی است در فرمول کواریانس، که در تحلیل مولفه‌های اصلی بیان شد، از ماتریس هسته استفاده کنیم. در این صورت تجزیه مقادیر ویژه، جهت‌های بیشترین واریانس در فضای جدید را محاسبه می‌کند.

$$C = \frac{1}{n} \sum_i^n \phi(x_i) \phi(x_i)^T = \frac{1}{n} \sum_i^n K(x_i, x_j).$$

لازم به ذکر است شرط صفر بودن میانگین داده‌ها در فضای ویژگی نیز به راحتی توسط تابع هسته قابل محاسبه است [۳].

۲.۱. چندجمله‌ای‌های متعامد لژاندر

چندجمله‌ای‌های لژاندر، سیستم کاملی از چندجمله‌ای‌های متعامدند که به علت ویژگی‌های جالبشان توسط دانشمندان در حل مسائل ریاضیاتی همچون حل معادلات دیفرانسیل استفاده می‌شوند [۱]. جهت تعریف این توابع می‌توان از فرمول زیر استفاده کرد:

$$P_n(x) = 2^n \sum_{k=0}^n x^k \binom{n}{k} \left(\frac{n+k-1}{n} \right).$$

لازم به ذکر است جهت تعریف این توابع در بازه $[a, b]$ از نگاشت $\tilde{P}_n(x) = P_n(\phi(x))$ استفاده می‌شود که در آن $\phi(x)$ برابر است با:

$$\phi(x) = \frac{(2x - a - b)}{b - a},$$

۳.۱. توابع متعامد لژاندر کسری

به علت محدودیت‌های پیاده‌سازی و سخت افزاری، نمی‌توان درجه چندجمله‌ای را به اندازه دلخواه افزایش داد. دانشمندان برای کم‌رنگ کردن این مشکل استفاده از توابع لژاندر کسری را پیشنهاد کرده‌اند. این توابع با استفاده از نگاشت

$$\phi(x) = \frac{(2x^\alpha - a - b)}{(b - a)},$$

جهت تعریف در بازه‌ی $[a, b]$ بیان می‌شوند.

یکی از کاربردهای این توابع، به عنوان هسته مدل‌های یادگیری ماشین می‌باشد [۵]. در این تحقیق چندجمله‌ای‌های متعامد لژاندر انتقال یافته کسری به عنوان هسته تابع مدل‌های یادگیری ماشین مورد استفاده قرار گرفته است. این توابع طبق تعریف هسته برای مجموعه داده با d ویژگی به صورت زیر فرموله‌بندی می‌شود:

$$K(x, t) = \prod_{i=1}^d K(x_i, t_i) = \prod_{i=1}^d \sum_{j=0}^q P_j(x_i) P_j(t_i). \quad (4)$$

۲. مدل پیشنهادی و نتایج

در این بخش با معرفی مجموعه داده و مدل پیشنهادی، به بررسی الگوریتم پیشنهادی بر روی چند مجموعه داده شناخته شده می‌پردازیم.

۱.۲. مجموعه داده‌ها

در جدول ۲ مجموعه داده‌های به‌کاربرده شده در این مقاله به همراه تعداد ویژگی‌ها و تعداد داده‌ها قرار داده شده است. مجموعه داده‌های Aids و Iris از نوع دسته‌بندی دو کلاسه و چندکلاسه می‌باشند. همچنین جهت بررسی الگوریتم ارائه شده در مسائل رگرسیون مجموعه داده‌های Boston و Triazines استفاده شده‌اند.

۲.۲. مدل پیشنهادی

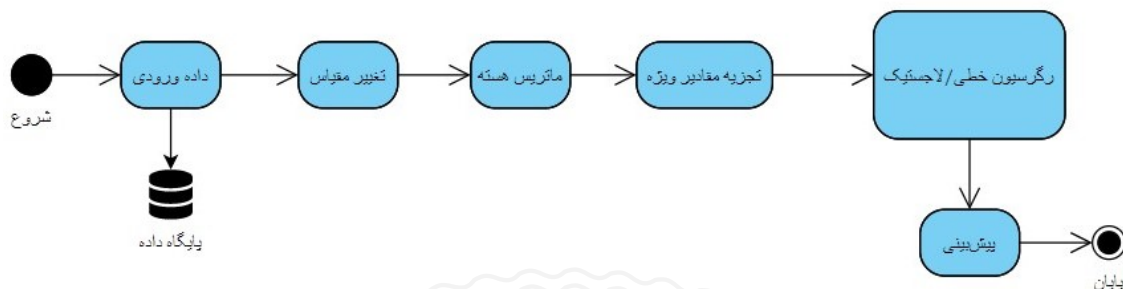
الگوریتم کلی روش ارائه شده در شکل ۱ آورده شده است. این الگوریتم پس از تغییر مقیاس داده‌ها به بازه $[0, 1]$ و محاسبه ماتریس هسته توسط توابع لژاندر کسری، از تحلیل مولفه‌های اصلی استفاده می‌کند. در نهایت نیز از رگرسیون خطی و لاجستیک جهت آموزش و یادگیری استفاده می‌نماید. رگرسیون خطی با مدل‌سازی جواب به صورت $y(x) = w^T x + b$ ، یک دستگاه معادلات خطی حل نموده و ضرایب مجهول w را محاسبه می‌کند. در مقابل رگرسیون لاجستیک برای حل یک مسئله دو کلاسه مسئله کمینه‌سازی زیر را در نظر می‌گیرد:

$$\min_{w, c} \frac{1}{2} w^T w + C \sum_{i=1}^n \log(\exp(-y_i(X_i^T w + c)) + 1). \quad (5)$$

۳.۲. نتایج

الگوریتم ارائه شده بر روی چهار مجموعه داده جدول ۲ مختلف تست و بررسی شده است. همچنین به جهت قابلیت مقایسه، از توابع هسته خطی و چندجمله‌ای معمولی نیز استفاده شده است. در جدول ۱ نتایج این تحقیق آورده شده است. این نتایج بر اساس اعتبارسنجی متقابل 3-Fold گزارش شده‌اند. معیارهای گزارش شده در جدول برای مسائل دسته‌بندی بر اساس دقت پیش‌بینی (Accuracy) و برای رگرسیون میانگین مربعات خطا (MSE) است که به صورت

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2,$$



شکل ۱: الگوریتم پیشنهادی

مجموعه داده	تعداد داده‌ها	تعداد ویژگی	هسته	پارامتر	iris	aids	triazines	boston
aids	۵۰	۵	خطی	-	۹۱%	۶۷%	۰.۰۳۳	۱۶۹
iris	۱۵۰	۴	چندجمله‌ای	$d = ۲$	۹۱%	۶۱%	۰.۰۳۲	۸۱۵۳۴۷
boston	۵۰۶	۱۳	لژاندر	$\alpha = ۰/۲۵$	۹۵%	۶۹%	۰.۰۳۱	۳۲۷۶
triazines	۱۸۶	۶۱	لژاندر	$\alpha = ۰/۵$	۹۶%	۶۵%	۰.۰۲۷	۲۰۸
			لژاندر	$\alpha = ۱$	۹۶%	۵۷%	۰.۰۲۹	۸۰
			لژاندر	$\alpha = ۱/۵$	۹۸%	۵۵%	۲.۵۳۸	۹۴

جدول ۲: جدول داده‌ها

جدول ۱: جدول نتایج

تعریف می‌گردد. مشاهده می‌شود نتایج حاصله به نسبت توابع کلاسیک برتری چشمگیری داشته‌اند.

مراجع

- [۱] ک. پرند، م. رزاقی، ج. امانی راد، م. دلخوش، م. معیری. رویکردهای نوین از روش‌های طیفی در محاسبات علمی: نظریه و کاربردها، دانشگاه شهید بهشتی، ۱۳۹۸.
- [۲] م. حجاریان، نخستین درس در جبر خطی عددی، انتشارات شهید بهشتی، تهران، ۱۳۹۷.
- [3] W. Wu, D.L. Massart, S.D. Jong, The kernel PCA algorithms for wide data. Part I" Theory and algorithms, *Chemometrics and Intelligent Laboratory System*, 36 (1997) 165–172.
- [4] S. Wold, K. Esbensen, P. Geladi, Chemometrics and Intelligent Laboratory Systems, *Principal component analysis*, 2 (1987) 37–52.
- [5] K.Parand, A.A. Aghaei, M. Jani, A. Ghodsi, A new approach to the numerical solution of Fredholm integral equations using least squares-support vector regression, *Mathematics and Computers in Simulation*, 180 (2021) 114–128.

ز. بهروزه، ع. آقائی، ک. پرند، م. حجاریان

گروه علوم داده‌ها و کامپیوتر، دانشکده علوم ریاضی، دانشگاه شهید بهشتی، تهران، ایران
آدرس ایمیل: zahrabehrouzeh@gmail.com

علیرضا افضل آقائی نائینی
گروه علوم داده‌ها و کامپیوتر، دانشکده علوم ریاضی، دانشگاه شهید بهشتی، تهران، ایران
آدرس ایمیل: alirezaafzalaghaei@gmail.com

کوروش پرند
گروه علوم داده‌ها و کامپیوتر، دانشکده علوم ریاضی، دانشگاه شهید بهشتی، تهران، ایران
آدرس ایمیل: k_parand@sbu.ac.ir

مسعود حجاریان
گروه علوم داده‌ها و کامپیوتر، دانشکده علوم ریاضی، دانشگاه شهید بهشتی، تهران، ایران
آدرس ایمیل: m_hajarian@sbu.ac.ir

The 9th Seminar on



دانشگاه کیدان

and its Applications

ساخت ماتریس متقارن غیر بدیهی برای طیف معین

سمیه زنگونی زاده، مسعود سبک خیز

چکیده. مسئله مقدار ویژه معکوس یکی از مسائل پرکاربرد در جبر خطی عددی می باشد که دارای کاربردهای فراوانی در زمینه علوم مهندسی، فیزیک و ... است. برای برخی از ماتریس های خاص با اندازه و شرایط محدود حل شده است. در این مقاله ابتدا روشی برای ساخت ماتریس های متقارن غیر بدیهی (ماتریس قطری با مجموعه مقادیر ویژه روی قطر اصلی یک جواب بدیهی است) با مرتبه دلخواه بیان و شرایطی برای نامنفی بودن ماتریس متقارن بدست آمده، مطرح می شود. در پایان با استفاده از این روش دو مثال عددی ارائه می شود.

۱. پیش گفتار

در سال ۱۹۷۴ فیدلر به مسئله مقدار ویژه ماتریس های متقارن پرداخت [۱] و جواب بسیار ساده زیر را برای حالت $n = 2$ و طیف $\sigma = \{\lambda_1, \lambda_2\}$ ارائه داد:

$$c = \frac{1}{2} \begin{bmatrix} \lambda_1 + \lambda_2 & \lambda_1 - \lambda_2 \\ \lambda_1 - \lambda_2 & \lambda_1 + \lambda_2 \end{bmatrix}. \quad (1)$$

در سال ۱۹۷۹ فریدلند و ملکمن از اندیشه فیدلر در حالت $n = 2$ استفاده و ثابت کردند مجموعه $\sigma = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ نیز تحت شرایطی می تواند طیف یک ماتریس سه قطری متقارن و نامنفی باشد [۲]. یکی از گام های مهم در حل مسئله در حالت متقارن، ارائه جواب های ترکیبی بود که توسط فیدلر پایه گذاری شد [۳]. او توانست برای دو طیف از دو ماتریس نامنفی یک ماتریس نامنفی بدست آورد که طیف آن از اجتماع دو طیف مذکور تشکیل شده بود. مقادیر ویژه یک ماتریس متقارن همگی حقیقی اند. از این رو برای یک طیف حقیقی یعنی داده شده، هدف ساخت یک ماتریس متقارن و حقیقی به گونه ای است که مجموعه داده شده مقادیر ویژه آن باشد. یکی از نتایج مهم این مقاله که مشابه قضیه نظری - شرافت می باشد [۴] و در بخش بعد با استفاده از آن نحوه ساخت ماتریس متقارن را بیان می کنیم، لم زیر است.

لم ۱. فرض کنید B ماتریس متقارن $m \times m$ و $\sigma_1 = \{\mu_1, \mu_2, \dots, \mu_n\}$ مجموعه مقادیر ویژه و μ_1 مقدار ویژه ماکسیمال B باشد. اگر A ماتریسی متقارن از مرتبه n به شکل

$$A = \begin{bmatrix} A_1 & a \\ b^T & \mu_1 \end{bmatrix} \quad (2)$$

واژه های کلیدی: (مسئله مقدار ویژه معکوس، طیف ماتریس، مقدار ویژه ماکسیمال، ماتریس متقارن).
طبقه بندی موضوعی [۲۰۱۰]: 65F30, 65F18.

باشد که در آن A_1 ماتریسی $(n-1) \times (n-1)$ ، a, b بردارهایی در C^{m-1} و $\sigma_2 = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ مجموعه مقادیر ویژه ماتریس A باشد، آنگاه ماتریس متقارن با بعد $(m+n-1) \times (m+n-1)$ به صورت

$$C = \begin{bmatrix} A_1 & as^T \\ sb^T & B \end{bmatrix} \quad (3)$$

وجود دارد که $\sigma = \{\mu_2, \dots, \mu_m, \lambda_1, \lambda_2, \dots, \lambda_n\}$ مجموعه مقادیر ویژه آن است. s بردار ویژه نرمال متناظر با مقدار ویژه μ_1 می باشد.

۲. ساخت ماتریس متقارن برای طیف های مشخص

در این بخش ابتدا یک قضیه را بیان کرده، سپس با استفاده از استقرا و به کمک لم (۱) قضیه را اثبات و روش ساختی برای ماتریس های متقارن ارائه می دهیم.

قضیه ۱. فرض کنید $\sigma = \{\lambda_1, \dots, \lambda_n\}$ مجموعه ای از اعداد حقیقی با شرط $\lambda_1 > \lambda_2 > \dots > \lambda_n$ باشد. در این صورت ماتریس متقارن غیر بدیهی مانند A از مرتبه n وجود دارد به طوری که σ مجموعه مقادیر ویژه آن است.

اثبات. ماتریس قطری با مقادیر ویژه روی قطر اصلی یک جواب بدیهی قضیه است. برای ماتریس غیر بدیهی A با استقرا قضیه را ثابت می کنیم: فرض کنیم $n = 2$ در این صورت ماتریس متقارن 2×2 به صورت

$$A = \begin{bmatrix} \frac{\lambda_1 + \lambda_2}{2} & \frac{\lambda_1 - \lambda_2}{2} \\ \frac{\lambda_1 - \lambda_2}{2} & \frac{\lambda_1 + \lambda_2}{2} \end{bmatrix} \quad (4)$$

جوابی از مسئله است. زیرا این ماتریس متقارن و دارای دو مقدار ویژه λ_1, λ_2 می باشد. فرض می کنیم $n = 3$ و قرار می دهیم $\sigma_1 = \{\lambda_1, \lambda_2\}$ که می تواند مجموعه مقادیر ویژه ماتریس (۴) باشد. حال قرار می دهیم $\sigma_2 = \{\frac{\lambda_1 + \lambda_2}{2}, \lambda_3\}$ که طبق حالت $n = 2$ مجموعه مقادیر ویژه ماتریس B به صورت

$$B = \begin{bmatrix} \frac{\lambda_1 + \lambda_2 + \lambda_3}{2} & \frac{\lambda_1 + \lambda_2 - \lambda_3}{2} \\ \frac{\lambda_1 + \lambda_2 - \lambda_3}{2} & \frac{\lambda_1 + \lambda_2 + \lambda_3}{2} \end{bmatrix} \quad (5)$$

است. چون A و B هر دو ماتریس متقارن می باشند و $\frac{\lambda_1 + \lambda_2}{2}$ مقادیر ویژه ماکسیمال ماتریس B است. لذا از لم (۱) می توان نتیجه گرفت که ماتریس

$$C = \begin{bmatrix} A_1 & as^T \\ sb^T & B \end{bmatrix}, \quad (6)$$

ماتریسی متقارن از مرتبه ۳ با مقادیر ویژه $\sigma = \{\lambda_1, \lambda_2, \lambda_3\}$ می باشد که $a = \frac{\lambda_1 - \lambda_2}{2}$ و $b = \frac{\lambda_1 + \lambda_2}{2}$ و A_1 حال فرض کنیم حکم قضیه برای $n-1$ برقرار باشد. برای ساختن ماتریس متقارن $n \times n$ با مقادیر ویژه $\sigma = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ قرار می دهیم $\sigma_1 = \{\lambda_1, \lambda_2, \dots, \lambda_{n-1}\}$. طبق فرض استقرا ماتریس متقارن A از مرتبه $n-1$ با مجموعه مقادیر ویژه σ_1 وجود دارد. همچنین تعریف می کنیم

$$\lambda = \frac{\lambda_1 + \lambda_2 + 2\lambda_3 + \dots + 2^{n-2}\lambda_{n-2}}{2^{n-2}}. \quad (7)$$

می‌توان نشان داد ماتریس متقارن A به صورت

$$A = \begin{bmatrix} A_1 & a \\ b^T & \frac{\lambda + \lambda_{n-1}}{2} \end{bmatrix}, \quad (8)$$

است که در آن A_1 ماتریسی متقارن $(n-2) \times (n-2)$ و a و b بردارهایی با بعد $1 \times (n-2)$ هستند به طوریکه تقارن نسبت به قطر اصلی در ماتریس A برقرار است. حال قرار می‌دهیم $\lambda' = \frac{\lambda + \lambda_{n-1}}{2}$. بنا به حالت $n = 2$ ماتریس متقارن B از مرتبه 2 با مجموعه مقادیر ویژه $\{\lambda', \lambda_n\}$ به صورت

$$B = \begin{bmatrix} \frac{\lambda' + \lambda_n}{2} & \frac{\lambda' - \lambda_n}{2} \\ \frac{\lambda' - \lambda_n}{2} & \frac{\lambda' + \lambda_n}{2} \end{bmatrix}, \quad (9)$$

می‌باشد. از آنجایی که تمام شرایط لم (1) برای ماتریس‌های A و B برقرار است. لذا طبق این لم ماتریس متقارن $C = \begin{bmatrix} A_1 & as^T \\ sb^T & B \end{bmatrix}$ با مجموعه مقادیر ویژه $\sigma = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ جوابی از مسئله است. \square

نکته 1. اگر شرایطی روی مقادیر ویژه لحاظ کنیم به یک ماتریس متقارن نامنفی خواهیم رسید. برای مثال اگر $n = 2$ کافی است $|\lambda_1| > |\lambda_2|$ همچنین اگر $n = 3$ ماتریس متقارن مذکور با شرط $\lambda_1 > |\lambda_2 + 2\lambda_3|$ نامنفی می‌شود. در حالت کلی، عنوان یک حدس برای نامنفی بودن ماتریس لازم است $\lambda_1 > |\lambda_2 + 2\lambda_3 + \dots + 2^{i-2}\lambda_i|$ و $i = 2, \dots, n-2$.

3. مثال عددی

مثال 1. فرض کنید $\sigma = \{5, -1, -1, -2\}$. قرار می‌دهیم $\sigma_1 = \{5, -1\}$. طبق حالت $n = 2$ ماتریس، A را به صورت زیر خواهیم داشت:

$$\begin{bmatrix} \frac{\lambda_1 + \lambda_2}{2} & \frac{\lambda_1 - \lambda_2}{2} \\ \frac{\lambda_1 - \lambda_2}{2} & \frac{\lambda_1 + \lambda_2}{2} \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix}, \quad (10)$$

که در آن $A_1 = 2$ و $a = 3$ و $b = 3$ هستند. در ادامه اگر $\sigma_2 = \{\frac{\lambda_1 + \lambda_2}{2}, \lambda_3\} = \{2, -1\}$ در نظر بگیریم، B را به صورت $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{3}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$ طیف σ_2 خواهیم داشت. چون $\mu_1 = 2$ بنابراین بردار ویژه نرمال متناظر با آن برابر است با $s = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ پس ماتریس C از مرتبه 3، با طیف $\sigma = \{5, -1, -1\}$ به صورت زیر بدست می‌آید

$$C = \begin{bmatrix} A_1 & as^T \\ sb^T & B \end{bmatrix} = \begin{bmatrix} 2 & \frac{3}{\sqrt{2}} & \frac{3}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} & \frac{1}{2} & \frac{3}{2} \\ \frac{3}{\sqrt{2}} & \frac{3}{2} & \frac{1}{2} \end{bmatrix}. \quad (11)$$

حال قرار می‌دهیم $A = \begin{bmatrix} 2 & \frac{3}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} & \frac{1}{2} \end{bmatrix}$ و $a = \begin{bmatrix} \frac{3}{\sqrt{2}} \\ \frac{3}{2} \end{bmatrix}$ و $b = \begin{bmatrix} \frac{3}{\sqrt{2}} \\ \frac{3}{2} \end{bmatrix}$ و $\sigma_3 = \{\frac{\lambda_1 + \lambda_2 + 2\lambda_3}{4}, \lambda_4\} = \{\frac{1}{4}, -2\}$ پس داریم $B = \begin{bmatrix} -\frac{3}{4} & \frac{5}{4} \\ \frac{5}{4} & -\frac{3}{4} \end{bmatrix}$ که $\mu = \frac{1}{4}$ مقدار ماکسیمال آن و بردار ویژه نرمال شده متناظر با M برابر است با $s = \begin{bmatrix} \frac{1}{\sqrt{4}} \\ \frac{1}{\sqrt{4}} \end{bmatrix}$ و در نهایت

ماتریس متقارن 4×4 ، C به صورت زیر بدست می‌آید

$$C = \begin{bmatrix} A_1 & as^T \\ sb^T & B \end{bmatrix} = \begin{bmatrix} 2 & \frac{3}{\sqrt{2}} & \frac{3}{2} & \frac{3}{2} \\ \frac{3}{\sqrt{2}} & \frac{1}{2} & \frac{3}{2\sqrt{2}} & \frac{2}{2\sqrt{2}} \\ \frac{3}{2} & \frac{3}{2\sqrt{2}} & -\frac{3}{4} & \frac{5}{4} \\ \frac{3}{2} & \frac{3}{2\sqrt{2}} & \frac{5}{4} & -\frac{3}{4} \end{bmatrix} \quad (12)$$

که می‌توان دید طیف آن $\sigma = \{5, -1, -1, -2\}$ است.

مثال ۲. اگر با داشتن طیف $\sigma = \{5, -1, -1, -2, \frac{9}{4}\}$ با استفاده از روند فوق، به ماتریس متقارن 5×5 زیر خواهیم رسید که مقادیر ویژه آن همان مجموعه σ می‌باشند:

$$C = \begin{bmatrix} 2 & \frac{3}{\sqrt{2}} & \frac{3}{2} & \frac{3}{2\sqrt{2}} & \frac{3}{2\sqrt{2}} \\ \frac{3}{\sqrt{2}} & \frac{1}{2} & \frac{3}{2\sqrt{2}} & \frac{3}{4} & \frac{3}{4} \\ \frac{3}{2} & \frac{3}{2\sqrt{2}} & -\frac{3}{4} & \frac{5}{4\sqrt{2}} & \frac{5}{4\sqrt{2}} \\ \frac{3}{2\sqrt{2}} & \frac{3}{4} & \frac{5}{4\sqrt{2}} & -\frac{3}{4} & \frac{3}{4} \\ \frac{3}{2\sqrt{2}} & \frac{3}{4} & \frac{5}{4\sqrt{2}} & -\frac{3}{4} & \frac{3}{4} \end{bmatrix}. \quad (13)$$

مراجع

- [1] M. Fiedler, Eigenvalues of nonnegative symmetric, *Linear Algebra and its Applications*, 9 (1974) 119–142.
- [2] S. Friedland, A.A. Melkman, On the eigenvalues of nonnegative Jacobi matrices, *Linear Algebra and its Applications*, 25 (1979) 239–254.
- [3] M. Fiedler, Eigenvalues of nonnegative symmetric, *Linear Algebra and its Applications*, 9 (1974) 119–142.
- [4] A.M. Nazari, F. Sherafat, On the inverse eigenvalue problem for nonnegative matrices of order two five, *Linear Algebra and its Applications*, 436 (2012) 1771–1790

سمیه زنگویی زاده

گروه ریاضی کاربردی، دانشکده ریاضی و رایانه، دانشگاه شهید باهنر کرمان، کرمان، ایران.
آدرس ایمیل: zangoei@uk.ac.ir

مسعود سبک خیز

گروه ریاضی کاربردی، دانشکده ریاضی و رایانه، دانشگاه شهید باهنر کرمان، کرمان، ایران.
آدرس ایمیل: masoudsabokkhiz75@gmail.com

تجزیه و تحلیل روش هم محلی در فضای چند جمله ایهای تکه ای پیوسته برای معادلات انتگرال ولترای غیر استاندارد با هسته منفرد ضعیف

افسانه حمزه لو، سعید پیش بین

چکیده. در این مقاله ابتدا معادله انتگرال غیر استاندارد ولترا و حالت خاصی از آن رامعرفی و قضیه وجود و یکتایی رابرای این نوع معادله مطرح می کنیم. در ادامه حالت منفرد ضعیف معادله انتگرال غیر استاندارد ولترا را در نظر گرفته و جواب این معادله را بر اساس چند جمله ای های تکه ای با پایه لاگرانژ در فضای چند جمله ای های تکه ای پیوسته تقریب می زنیم. انتخاب نقاط هم محلی و نوع افراز می تواند در همگرایی روش موثر باشد. در این مقاله با انتخاب مناسب نقاط هم محلی، همگرایی روش را در افراز یکنواخت و مدرج تضمین می کنیم. در پایان با ارائه نتایج عددی صحت و اعتبار این روش را تصدیق می نماییم.

۱. معرفی

معادله انتگرال غیر استاندارد ولترا به صورت زیر تعریف می شود:

$$u(t) = g(t) + \int_0^t K(t, s, u(t), u(s)) ds, t \in I := [0, T], \quad (1)$$

که در آن $u(t)$ تابع مجهول و توابع g و K معلوم هستند. تحلیل عددی و قضیه وجود و یکتایی این نوع از معادلات در مقاله [۱] بررسی شده است. به عنوان نوع خاصی از معادلات غیر استاندارد ولترا، معادله انتگرال اتو-کانولوشن تعمیم یافته را می توان به فرم زیر در نظر گرفت:

$$u(t) = g(t) + \int_0^t k(t, s) u(t-s) u(s) ds, t \in I := [0, T],$$

که در آن $D := \{(t, s) : 0 \leq s \leq t \leq T\}$ ، $K \in C(D)$ و $g \in C(I)$ توابعی معلوم و $u(t)$ تابعی مجهول است. این معادلات نقش مهمی را در توصیف بسیاری از پدیده های فیزیکی بازی می کنند، از جمله کاربردها می توان به مشخص کردن حافظه هسته در تئوری ویسکوالاستیک [۲] و محاسبه برخی توابع خاص مانند توابع میتاگ-لفلر [۴] اشاره کرد.

حال در این مقاله ما معادله انتگرال غیر استاندارد ولترا با هسته منفرد ضعیف را به صورت زیر در نظر می گیریم:

$$u(t) = g(t) + \int_0^t (t-s)^{-\alpha} K(t, s, u(t), u(s)) ds, t \in I := [0, T] (0 < \alpha < 1), \quad (2)$$

که در آن $u(t)$ تابع مجهول و توابع g و K توابعی معلوم اند. روش عددی هم محلی بر اساس چند جمله ای های تکه ای را برای تقریب جواب این معادله بکار می بریم و با انتخاب مناسب نقاط هم محلی، همگرایی روش را در افراز یکنواخت و مدرج تضمین می کنیم.

واژه های کلیدی: معادلات انتگرال ولترای غیر استاندارد، معادلات انتگرال ولترا با هسته منفرد ضعیف، روش هم محلی بر اساس چند جمله ای های تکه ای.

طبقه بندی موضوعی [۲۰۱۰]: 13D45, 39B42.

۲. وجود و یکتایی جواب معادلات انتگرال غیر استاندارد ولترا

قضیه ۱. [۱] فرض کنید که توابع داده شده در معادلات انتگرال غیر استاندارد ولترای (۱) در شرایط زیر صدق کنند:

$$۱. \quad K \in C^m(D \times \mathbb{R} \times \mathbb{R}) \text{ و } g \in C^m(I)$$

$$۲. \quad |K(t, s, u_1, v_1) - K(t, s, u_2, v_2)| \leq L(t, s)|u_1 - u_2| + M|v_1 - v_2|, \quad u_i, v_i \in \mathbb{R}, \quad i = 1, 2$$

و $D = \{(t, s) : 0 \leq s, t \leq T\}$ و $0 < L(t, s) \leq M$ و $\int_0^t L(t, s)ds \leq 1 - \delta$ که $\delta \in (0, 1)$ و $(t, s) \in D$ می باشند.

در این صورت یک جواب منحصر بفرد $u \in C^m(I)$ برای معادلات انتگرالی غیر استاندارد ولترا (۱) وجود دارد. توجه کنید برای $m = 0$ این قضیه را برای حالت منفرد ضعیف نیز می توان تعمیم داد ولی برای $m \geq 1$ می دانیم مشتقات معادلات منفرد ضعیف ممکن است در ابتدای بازه انتگرال کراندار نباشند.

۳. روش هم محلی در فضای چند جمله ای های تکه ای پیوسته

در این بخش، روش عددی هم محلی بر اساس چند جمله ای های تکه ای را برای تقریب جواب معادله (۲) در نظر می گیریم. برای یک عدد صحیح داده شده $N \geq 2$ و یک عدد حقیقی $r \geq 1$ ما یک افراز بر بازه داده شده I را به صورت زیر تعریف می کنیم:

$$I_h^r := \{t_n : (\frac{n}{N})^r T : n = 0, 1, \dots, N\}.$$

اگر $r > 1$ باشد در این صورت I_h^r یک افراز مدرج روی $[0, T]$ با نمای درجه بندی r می باشد و زمانی که $r = 1$ است، یک افراز یکنواخت داریم به طوری که از نماد $I_h := I_h^1$ نیز استفاده کنیم. هم چنین برای $n = 0, 1, \dots, N - 1$ نمادهای زیر را در نظر می گیریم:

$$h_n := t_{n+1} - t_n, \quad \sigma_n := [t_n, t_{n+1}], \quad h := \max_{(n)} \{h_n\} = h_{N-1}.$$

حال جواب u مربوط به معادله (۲) را با استفاده از نقاط هم محلی

$$X_h := \{t_n + c_i h_n : 0 < c_1 < \dots < c_m \leq 1, (0 \leq n \leq N - 1)\},$$

با u_h از فضای چند جمله ای پیوسته $S_m^{(0)}(I_h^r)$ که به صورت زیر تعریف می شود، تقریب می زنیم

$$S_m^{(0)}(I_h) := \{v \in C(I) : v|_{\sigma_n} \in P_m(\sigma_n) (0 \leq n \leq N - 1)\}.$$

بنابراین جواب تقریبی u_h توسط معادله هم محلی:

$$u_h(t) = g(t) + \int_0^t (t-s)^{-\alpha} K(t, s, u_h(t), u_h(s)) ds, \quad t \in X_h, \quad (۳)$$

به دست می آید که در آن $u_h(0) = g(0)$. در هریک از زیر بازه های σ_n نمایش موضعی جواب هم محلی u_h را می توان به صورت زیر در نظر گرفت:

$$u_h(t_n + sh_n) = \sum_{j=0}^m L_j(s) U_{n,j}, \quad s \in [0, 1], \quad (۴)$$

به طوری که $t_{n,j} = t_n + c_j h_n$ و $U_{n,j} = u_h(t_{n,j})$ و $U_{n,0} = u_h(t_n)$. همچنین چند جمله ای های:

$$L_0(s) := (-1)^m \prod_{k=1}^m \frac{s - c_k}{c_k},$$

$$L_j(s) := \frac{s}{c_j} \prod_{k \neq j}^m \frac{s - c_k}{c_j - c_k}, \quad j = 1, 2, \dots, m, s \in [0, 1],$$

شان دهنده توابع پایه لاگرانژ با توجه به نقاط متمایز $\{c_i\} \cup \{0\}$ هستند. حال با جایگذاری نقاط هم محلی و (۴) در (۳) و استفاده از تغییر متغیر مناسب داریم:

$$\begin{aligned} U_{n,i} &= g(t_{n,i}) + \int_0^{t_{n,i}} (t_{n,i} - s)^{-\alpha} K(t_{n,i}, s, U_{n,i}, u_h(s)) ds \\ &= g(t_{n,i}) + \int_0^{t_n} (t_{n,i} - s)^{-\alpha} K(t_{n,i}, s, U_{n,i}, u_h(s)) ds \\ &\quad + \int_{t_n}^{c_i h_n} (t_{n,i} - s)^{-\alpha} K(t_{n,i}, s, U_{n,i}, u_h(s)) ds \\ &= g(t_{n,i}) + \sum_{l=0}^{n-1} h_l^{-\alpha} \int_0^1 \left(\frac{t_{n,i} - t_l}{h_l} - s \right)^{-\alpha} K(t_{n,i}, t_l + s h_l, U_{n,i}, \sum_{j=0}^m L_j(s) U_{l,j}) ds \\ &\quad + h_n^{-\alpha} \int_0^{c_i} (c_i - s)^{-\alpha} K(t_{n,i}, t_n + s h_n, U_{n,i}, \sum_{j=0}^m L_j(s) U_{n,j}) ds, \quad (i = 1, 2, \dots, m) \end{aligned}$$

با حل دستگاه بالا نسبت به مجهولات $U_{n,i}$ جواب تقریبی معادله از رابطه (۴) حاصل می شود.

۴. نتایج عددی

در این بخش از مقاله مثالی از معادلات انتگرال غیراستاندارد ولترا با هسته منفرد ضعیف را در نظر می گیریم. لازم به ذکر است که تمامی محاسبات با استفاده از نرم افزار ممتیکا انجام شده است.

مثال ۱. معادله انتگرالی ولترا غیراستاندارد با هسته منفرد ضعیف بفرم:

$$u(t) = g(t) + \int_0^t \frac{1}{\sqrt{t-s}} \times (t \times s) \times (u(t) + u(s)) ds,$$

را با $K(t, s) = t \times s$ در نظر بگیرید. تابع $g(t)$ طوری تعیین می شود که جواب دقیق این معادله $u(t) = e^t$ باشد. با فرض $\alpha = \frac{1}{4}$ و $r = 1$ ماکزیمم خطا را برای $m = 2, 3$ و N مختلف در نقاط گره ای در جدول ۱ گزارش می دهیم.

جدول ۱: ماکزیمم خطا روش هم محلی برای $m = 2, 3$

N	$c_1 = \frac{1}{2}, c_2 = 1$	$c_1 = \frac{4}{5}, c_2 = \frac{9}{10}$	$c_1 = \frac{1}{3}, c_2 = \frac{1}{2}, c_3 = 1$	$c_1 = \frac{1}{3}, c_2 = \frac{1}{2}, c_3 = \frac{8}{9}$
۴	1.74×10^{-4}	7.88×10^{-4}	2.52×10^{-5}	1.49×10^{-5}
۸	9.49×10^{-5}	1.49×10^{-4}	9.21×10^{-6}	2.16×10^{-6}
۱۶	2.03×10^{-5}	1.03×10^{-4}	3.82×10^{-6}	9.15×10^{-7}

مراجع

- [1] Q. Guan, R. Zhang, Y. Zou, Analysis of collocation solutions for nonstandard Volterra integral equations, *IMA Journal of Numerical Analysis*, 32 (2012) 1755–1785.
- [2] H. Liang, H. Brunner, The convergence of collocation solutions in continuous pieewise polynomial spaces for weakly singular Volterra integral equations, *SIAM Journal on Numerical Analysis*, 57 (2019) 1875–1896.
- [3] L.V. Wolfersdorf, Einige Klassen quadratischer Integralgleichungen [Some classes of quadratic integral equations], *Sitz.ber. S'achs. Akad. Wiss. Leipz. Math.-Nat.wiss. Kl.*, 2000.
- [4] L.V. Wolfersdorf, Autoconvolution equations and special functions, *Integral Transforms and Special Functions*, 21 (2010) 295–306.

افسانه حمزه لو

دانشکده علوم، دانشگاه ارومیه، ارومیه، ایران.

آدرس ایمیل: afsaneh.hamzeloo@gmail.com

سعید پیش بین

دانشکده علوم، دانشگاه ارومیه، ارومیه، ایران.

آدرس ایمیل: s.pishbin@urmia.ac.ir

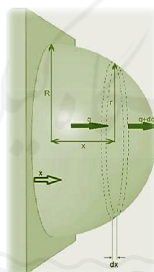
روش خطی سازی متوالی برای حل مسئله انتقال گرما روی پرک های نیم کره ای

صفیه باقری

چکیده. در این پژوهش قصد داریم تا با به کارگیری روش خطی سازی متوالی به حل مسئله انتقال گرما روی پرک های نیم کره ای بپردازیم. این پرک ها در بسیاری از ابزارهای صنعتی مثل یخچال ها، اتومبیل ها، تهویه کننده ها، تجهیزات پردازش شیمیایی و تراشه های الکترونیکی کاربرد دارند. برای بررسی دقت روش پیشنهادی مقایسه ای بین نتایج به دست آمده بر اساس روش خطی سازی متوالی و روش رانگ-کوتا انجام شده است.

۱. پیش گفتار و معادلات حاکم

پرک ها، کارآمدترین ابزار برای بالا بردن نرخ تبدیل گرما هستند. آن ها محیط تبادل گرما و سطح گرمای منتقل شده را افزایش می دهند. کراس و همکاران [۲] یک بررسی کامل روی این موضوع ارائه دادند. پرک ها می توانند دایره ای، کروی، مستطیلی و یا به شکل های دیگر باشند. از بین اشکال مختلف، پرک های مستطیل شکل به خاطر سهولت در پردازش و پروسه ساخت آسان بیش تر مورد استفاده قرار می گیرند. هم چنین این یک واقعیت است که تبدیل گرما در پرک ها تابعی از دما باشد. همان طور که در شکل ۱ نشان داده شده است، مقطع عمودی پرک را نیم کره ای با شعاع R ، در نظر می گیریم. سطح پرک، گرما را با روش تابشی-همرفتی منتقل می کند. باید توجه



شکل ۱: مسئله انتقال گرما روی پرک های نیم کره ای با روش همرفتی-تابشی

داشته باشیم که پرک های مبدل، گرمای درونی وابسته به دما دارند. دمای هوای اطراف را T_a ، دمای موثر چاه برای انتقال گرمای تابشی را T_s ، ضریب انتقال گرمای همرفتی ثابت را h و ظرفیت گرمای ویژه پرک را T_k ، مبدل گرمای داخلی را q^* و قابلیت تشعشع سطح را ϵ ، فرض می کنیم که همگی توابعی از دما هستند. روابط زیر را در نظر بگیرید [۴]:

$$k = k_0(1 + \alpha(T(x) - T_a)), \quad q^* = q_0(1 + \lambda(T(x) - T_s)), \quad \epsilon = \epsilon_0(1 + \beta(T(x) - T_s)), \quad (1)$$

واژه های کلیدی: روش خطی سازی متوالی، روش هم محلی، روش همرفتی-تابشی، پرک های نیم کره ای، انتقال گرما. طبقه بندی موضوعی [۲۰۱۰]: 13D45, 39B42.

که در آن k_0 ظرفیت گرمایی ویژه‌ی پرک در دمای همرفت چاه T_a ، ε_0 قابلیت تشعشع سطح در دمای چاه T_s و q_0 مبدل گرمای داخلی در دمای T_s می‌باشند. ثابت‌های α ، β و λ به ترتیب از تغییر ظرفیت گرمایی ویژه، مبدل گرمایی داخلی و قابلیت تشعشع سطح با دما اندازه‌گیری می‌شوند. رابطه‌ی تعادل انرژی برای این سیستم به صورت زیر نوشته می‌شود:

$$\begin{cases} q(x) - q(x + \Delta x) + q^* A(x) \Delta x - hp(x) \Delta x (T(x) - T_s), \\ -\alpha \varepsilon p(x) (T(x)^* - T_s^*) = 0. \end{cases} \quad (2)$$

هم چنین قانون فوریه‌ی رسانش به صورت

$$q(x) = k(T(x)) A(x) \frac{dT(x)}{dx}, \quad (3)$$

است. بعد از تقسیم بر Δx و گرفتن حد از آن و جایگذاری معادله‌ی (۱) و (۳) در (۲) خواهیم داشت [۱]:

$$\begin{cases} \frac{d}{dx} (A(x)) k_0 (1 + \alpha(T(x) - T_a)) \frac{dT(x)}{dx} + q_0 (1 + \lambda(T(x) - T_s)) A(x), \\ -hp(x) (T(x) - T_a) - \delta p(x) \varepsilon_0 (1 + \beta(T(x) - T_s)) (T(x)^* - T_s^*) = 0. \end{cases} \quad (4)$$

شرایط مرزی دمای ثابت پایه به شکل $T(0) = T_b$ ، $T'(R) = 0$ فرض می‌شود. با توجه به شکل ۱، رابطه‌ی بین دو پارامتر r و x به صورت $r^2 = R^2 - x^2$ بیان می‌شود. هم چنین داریم:

$$A(x) = \pi(R^2 - x^2), \quad p(x) = 2\pi(\sqrt{R^2 - x^2}).$$

پارامترهای بدون بعد زیر را در نظر می‌گیریم [۱]:

$$\begin{cases} \theta = \frac{T}{T_b}, \quad \theta_s = \frac{T_s}{T_b}, \quad \theta_a = \frac{T_a}{T_b}, \quad X = \frac{x}{R}, \quad A = \alpha T_b, \quad B = \lambda T_b, \\ c = \beta T_b, \quad Ng = \frac{R^2 q_0}{T_b k_0}, \quad Nc = \frac{2hR}{k_0}, \quad Nr = \frac{\delta \varepsilon T_b^* R}{k_0}. \end{cases}$$

که در آن Ng پارامتر رسانش-تبدیل است که به تأثیر مبدل گرمای درونی، Nc پارامتر رسانش-همرفتی است که به تأثیر همرفت سطح پرک و Nr یک پارامتر تابشی-رسانشی است که به تأثیر تابش پرک اشاره می‌کند. پارامترهای A ، B و C به ترتیب ظرفیت گرمایی، مبدل گرما و قابلیت تشعشع سطح می‌باشند. هم چنین θ_s و θ_a به ترتیب دمای همرفت بدون بعد و دمای همرفت تبدیل گرمای چاه می‌باشند. در نهایت می‌توان معادله‌ی (۴) را به صورت

$$\begin{aligned} (1 + A(\theta - \theta_a)) \frac{d^2 \theta}{dx^2} + A \left(\frac{d\theta}{dx} \right)^2 - \frac{2x}{1 - x^2} (1 + A(\theta - \theta_a)) \frac{d\theta}{dx} \\ + Ng(1 + B(\theta - \theta_s)) - Nc \frac{(\theta - \theta_s)}{\sqrt{1 - x^2}} - \frac{Nr}{\sqrt{1 - x^2}} (1 + C(\theta - \theta_s)) (\theta^* - \theta_s^*) = 0, \end{aligned} \quad (5)$$

و با شرایط مرزی $\theta(0) = 1$ ، $\theta'(1) = 0$ محاسبه کرد. در ادامه قصد داریم معادله‌ی غیرخطی (۵) با شرایط مرزی داده شده را با استفاده از روش خطی سازی متوالی SLM [۳] حل کنیم. جواب معادله را به صورت

$$\theta(x) = \theta_i(x) + \sum_{m=0}^{i-1} \theta_m(x),$$

در نظر می‌گیریم و در معادله‌ی (۵) جایگذاری نموده با حذف جملات غیرخطی شامل θ_i ، θ'_i و θ''_i به دسته معادلات خطی

$$a_{1,i-1}(x) \theta''_i(x) + a_{2,i-1}(x) \theta'_i(x) + a_{3,i-1}(x) \theta_i(x) = r_{i-1}(x), \quad (6)$$

با شرایط مرزی $\theta_i(0) = 0$, $\theta_i'(1) = 0$ می‌رسیم که در آن توابع $a_{k,i-1}(x)$, $k = 1, 2, 3$ و $r_{i-1}(x)$ به صورت زیر تعریف می‌شوند:

$$\begin{aligned} a_{1,i-1}(x) &= 1 + A \sum_{m=0}^{i-1} \theta_m(x) - A\theta_a, \\ a_{2,i-1}(x) &= 2A \sum_{m=0}^{i-1} \theta'_m(x) - \frac{2x}{1-x^2} \left(1 + A \sum_{m=0}^{i-1} \theta_m(x) - A\theta_a \right), \\ a_{3,i-1}(x) &= A \sum_{m=0}^{i-1} \theta''_m(x) - \frac{2x}{1-x^2} A \sum_{m=0}^{i-1} \theta'_m(x) + NgB - \frac{Nc}{\sqrt{1-x^2}} - \frac{Nr}{\sqrt{1-x^2}} \\ &\quad + \left[2 \left(\sum_{m=0}^{i-1} \theta_m(x) \right)^2 + \Delta C \left(\sum_{m=0}^{i-1} \theta_m(x) \right)^2 - C\theta_s^* - 2C\theta_s \left(\sum_{m=0}^{i-1} \theta_m(x) \right) \right], \\ r_{i-1}(x) &= - \sum_{m=0}^{i-1} \theta'''_m(x) - A \sum_{m=0}^{i-1} \theta_m(x) \sum_{m=0}^{i-1} \theta''_m(x) + A\theta_a \sum_{m=0}^{i-1} \theta''_m(x) \\ &\quad - A \left(\sum_{m=0}^{i-1} \theta'_m(x) \right)^2 + NgB\theta_s + \frac{NC}{\sqrt{1-x^2}} \left(\sum_{m=0}^{i-1} \theta_m(x) - \theta_s \right) \\ &\quad + \frac{2x}{1-x^2} \left(\sum_{m=0}^{i-1} \theta'_m(x) + A \sum_{m=0}^{i-1} \theta_m(x) \sum_{m=0}^{i-1} \theta'_m(x) - A\theta_a \sum_{m=0}^{i-1} \theta'_m(x) \right) \\ &\quad - Ng \left(1 + B \sum_{m=0}^{i-1} \theta_m(x) \right) - \frac{Nr}{\sqrt{1-x^2}} \left[C\theta_s^* + C\theta_s \left(\sum_{m=0}^{i-1} \theta_m(x) \right) \right] \\ &\quad + \frac{Nr}{\sqrt{1-x^2}} \left[\left(\sum_{m=0}^{i-1} \theta_m(x) \right)^2 - \theta_s^* + C \left(\sum_{m=0}^{i-1} \theta_m(x) \right)^2 - C\theta_s^* \sum_{m=0}^{i-1} \theta_m(x) \right]. \end{aligned}$$

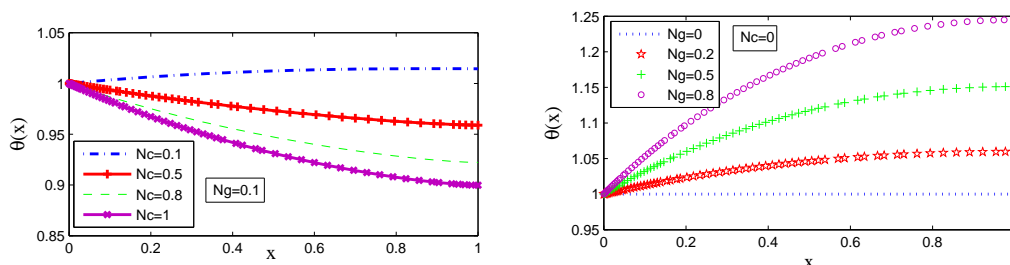
الگوریتم SLM را با توجه به شرایط مرزی، با تقریب اولیه $\theta_0(x) = 1$ آغاز می‌کنیم. مجموعه معادلات خطی سازی شده (۶) الگوریتم SLM را با روش هم‌محلی و حل دستگاه ادامه می‌دهیم. ابتدا برای یک n طبیعی دلخواه، جواب معادله‌ی (۶) را به شکل سری قطع شده‌ی زیر بر اساس چندجمله‌ای‌های چبیشف نوع اول تقریب می‌زنیم:

$$\theta_i(x) = \sum_{j=0}^n \theta_{ij} T_j(2x-1). \quad (7)$$

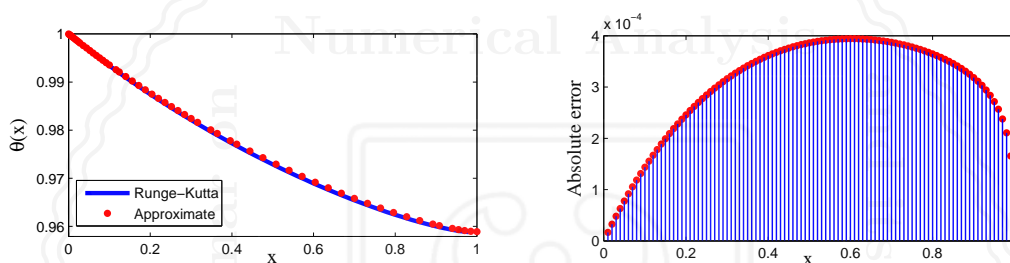
سری قطع شده‌ی (۷) را در معادلات دیفرانسیل خطی (۶) جای‌گذاری نموده، تابع باقی‌مانده را تشکیل می‌دهیم. هدف محاسبه‌ی $(n+1)$ ضریب بسط (۷) می‌باشد. با توجه به شرایط مرزی، دو معادله از بسط (۷) و اعمال شرایط بوجود می‌آید. علاوه بر آن، تابع باقی‌مانده باید در $x_j = \cos\left(\frac{\pi j}{N}\right)$, $j = 1, \dots, n-1$ صفر گردد. در نتیجه $(n-1)$ معادله حاصل می‌شود که با دو معادله‌ی قبلی، یک دستگاه $(n+1)$ معادله با $(n+1)$ مجهول تشکیل خواهند داد. جواب دستگاه همان ضرایب بسط (۷) یعنی θ_{ij} , $k = 1, 2, \dots, n$ را تولید می‌کند. در نتیجه تقریب جواب معادله دیفرانسیل خطی (۶) در هر تکرار محاسبه می‌شود و در پایان، جواب معادله دیفرانسیل غیرخطی با بسط $\theta(x) \simeq \sum_{i=0}^M \theta_i(x)$ تقریب زده می‌شود که در آن M ، تعداد تکرار الگوریتم SLM است.

۱.۱. نتایج عددی

این مسئله را با در نظر گرفتن تعداد پایه $n = 30$ ، مقادیر مختلف پارامتر Nc, Ng و با شش تکرار SLM حل کرده و نتایج در نمودار ۲ نمایش داده شده است. در شکل ۳، نمودار تابع تقریبی به ازای مقادیر $Nc = 0.8$, $Ng = 0.5$, $A = B = c = 0.5$, $\theta_a = \theta_s = 0$ ، $Nr = 0$ و $Ng = 0.1$ و $Nc = 0.5$ با مقادیر عددی روش رانگ-کوتا مقایسه شده است. مشاهده می‌شود که روش پیشنهادی از دقت مطلوبی برخوردار است.



شکل ۲: جواب تقریبی مسئله انتقال گرما از روش SLM مرتبه شش با تغییر پارامترهای Nc و Ng



شکل ۳: مقایسه بین جواب تقریبی مسئله انتقال گرما از روش SLM مرتبه شش و روش رانگ-کوتا

مراجع

- [1] M. Hatami, A. Hasanpour, D.D. Ganji, Heat transfer study through porous fins (Si3N4 and AL) with temperature-dependent heat generation, *Energy Conversion and Management*, 74 (2013) 9–16.
- [2] A.D. Kraus, A. Aziz, J.R. Welty, *Extended Surface Heat Transfer*, John Wiley, New York, 2002.
- [3] Z. Makukula, S. Motsa, P. Sibanda, On a new solution for the viscoelastic squeezing flow between two parallel plates, *Journal of Advanced Research in Applied Mathematics*, 4 (2010) 31–38.
- [4] S. Sabbaghi, A. Rezaii, Gh.R. Shahri, M.S. Baktash, Mathematical analysis for the efficiency of a semi-spherical fin with simultaneous heat and mass transfer, *International Journal of Refrigeration*, 34 (2011) 1877–1882.

صفیه باقری

دانشکده ریاضی، دانشگاه یزد، یزد، ایران.

آدرس ایمیل: sbagheri1354@stu.yazd.ac.ir

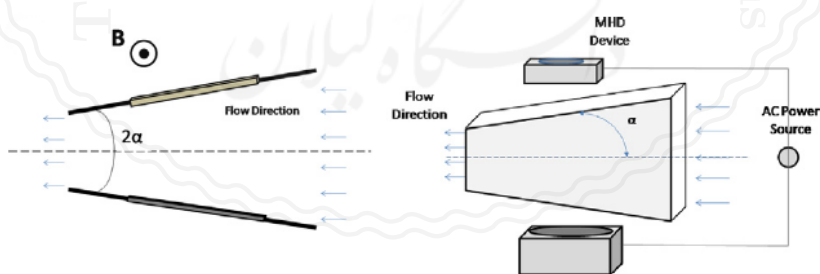
حل عددی معادله جریان جفري-هامل با استفاده از روش تفاضل متناهی تکراری

زهراسادات یزدانی، محمد حیدری، اعظم سادات هاشمی

چکیده. در این پژوهش به حل عددی مسئله جریان جفري-هامل به عنوان یکی از مسائل کاربردی در مکانیک سیالات خواهیم پرداخت. ابتدا معادلات غیرخطی حاکم، با استفاده از روش شبه خطی سازی به دنباله ای از معادلات دیفرانسیل خطی تبدیل شده و سپس از روش تفاضل متناهی برای حل آن ها استفاده می کنیم. در پایان برای نشان دادن کارایی روش پیشنهادی، مقایسه ای با نتایج حاصل از روش رانگ-کوتا مرتبه چهار ترکیب شده با روش پرتابی ارائه می شود.

۱. پیشگفتار

طیف وسیعی از پدیده های فیزیکی توسط انواع مختلفی از معادلات دیفرانسیل توصیف می شوند. مسئله جریان جفري-هامل^۱ یکی از پرکاربردترین و مهم ترین مسائل در مکانیک سیالات است که جریان سیال لزج تراکم ناپذیر بین دو دیواره غیر موازی را مدل سازی می کند [۳]. در مسئله جریان جفري-هامل دو پارامتر عددی رینولدز^۲ مغناطیسی و عدد هارتمن^۳، چگونگی حرکت سیال و سرعت آن را در کانال تعیین می کنند. جریان دوبعدی ثابت از یک سیال لزج تراکم ناپذیر بین دو صفحه سخت با زاویه 2α را مطابق شکل ۱ در نظر بگیرید. برای زاویه $\alpha > 0^\circ$ مسئله را کانال واگرا و اگر $\alpha < 0^\circ$ مسئله را کانال همگرا می نامند. فرض کنید که سرعت به طور



شکل ۱: جریان جفري-هامل در یک کانال واگرا با زاویه 2α .

کامل شعاعی و به صورت $v = (u(r, \theta), 0)$ در نظر گرفته شود که در آن r و θ مولفه های دستگاه مختصات قطبی، u مولفه سرعت و

واژه های کلیدی: معادله جریان جفري-هامل، روش شبه خطی سازی و روش تفاضل متناهی.
طبقه بندی موضوعی [۲۰۱۰]: 65M06, 80M20.

¹Jeffery-Hamel flow problem

²Reynolds

³Hartmann

ν سرعت شعاعی هستند. برای مسئله جریان جفری-هامل، معادله پیوستگی و معادلات ناویر-استوکس^۴ در مختصات قطبی به صورت زیر به دست می آیند [۳]:

$$\frac{\partial}{\partial r}(ru(r, \theta)) = 0, \quad (۱)$$

$$u(r, \theta) \frac{\partial u(r, \theta)}{\partial r} = \frac{-1}{\rho} \frac{\partial \rho}{\partial r} + \nu \left(\nabla^2 u(r, \theta) - \frac{u(r, \theta)}{r^2} \right), \quad (۲)$$

$$\frac{-1}{\rho} \frac{\partial P}{\partial \theta} + \frac{2\nu}{r} \frac{\partial u(r, \theta)}{\partial \theta} = 0, \quad (۳)$$

که در آن $\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}$ ، فشار، ρ چگالی سیال و ν ضریب لزجت جنبشی هستند. حال با تعریف پارامترهای بدون بعد به صورت $F(x) = \frac{f(\theta)}{f_{max}}$ و $x = \frac{\theta}{\alpha}$ که در آن θ زاویه جهتی $(-\alpha < \theta < \alpha)$ و α نصف زاویه کانال گوه شکل هستند و جایگذاری در (۲) و (۳) و حذف جمله فشار، معادله دیفرانسیل معمولی مرتبه سوم غیرخطی زیر را داریم:

$$\begin{cases} F'''(x) + 2\alpha Re F(x)F'(x) + 4\alpha^2 F'(x) = 0, & x \in [0, 1], \\ F(0) = 1, & F'(0) = 0, & F(1) = 0, \end{cases} \quad (۴)$$

که در آن $Re = \frac{\alpha f_{max}}{\nu}$ عدد رینولدز است.

۲. حل عددی مسئله

در این بخش سعی داریم با ترکیب روش های شبه خطی سازی و تفاضل منتهای [۱] یک روند تکراری برای حل مسئله (۴) ارائه دهیم. برای این هدف، ابتدا با انتگرال گیری از معادله (۴) روی بازه $[0, x]$ و اعمال شرط اولیه $F(0) = 1$ داریم:

$$\begin{cases} F''(x) + \alpha Re F^2(x) + 4\alpha^2 F(x) = F''(0) + \alpha Re + 4\alpha^2, \\ F'(0) = 0, & F(1) = 0, \end{cases} \quad (۵)$$

که در آن $F''(0)$ مقداری مجهول است. حال با استفاده از روش شبه خطی سازی [۲] مسئله (۵) را می توان به یک دنباله از مسائل مقدار مرزی خطی به صورت زیر تبدیل کرد:

$$\begin{cases} F''_{n+1}(x) + (2\alpha Re F_n(x) + 4\alpha^2) F_{n+1}(x) = F''_{n+1}(0) + \alpha Re F_n^2(x) + L, \\ F'_{n+1}(0) = 0, & F_{n+1}(1) = 0, & n = 0, 1, \dots, \end{cases} \quad (۶)$$

که در آن $L = \alpha Re + 4\alpha^2$. توجه داریم که در رابطه (۶) تابع $F_0(x)$ به عنوان شروع اولیه در نظر گرفته می شود و معمولاً به گونه ای آن را انتخاب می کنیم که حداقل شرایط اولیه و مرزی مسئله را برآورده سازد. در اینجا تابع $F_0(x) = 1 - x^2$ را در نظر می گیریم. حال به حل معادله (۶) در هر تکرار با استفاده از روش تفاضل منتهای می پردازیم. برای این منظور ابتدا بازه $[0, 1]$ را با طول گام $h = \frac{1}{M}$ ، $M \in \mathbb{N}$ افزایش می دهیم و نقاط $x_i = ih$ را برای $i = 0, 1, \dots, M$ در نظر می گیریم. با جایگذاری x_i در معادله (۶) و در نظر گرفتن فرمول های تفاضلی [۱]

$$F''_{n+1}(x_i) = \frac{1}{h^2} [F_{n+1}(x_{i+1}) - 2F_{n+1}(x_i) + F_{n+1}(x_{i-1})] + O(h^2), \quad (۷)$$

$$F''_{n+1}(0) = F''_{n+1}(x_0) = \frac{1}{h^2} [2F_{n+1}(x_0) - 5F_{n+1}(x_1) + 4F_{n+1}(x_2) - F_{n+1}(x_3)] + O(h^2), \quad (۸)$$

^۴Navier-Stokes

شکل گسسته زیر به دست می آید:

$$\begin{aligned} & \frac{1}{h^2} F_{n+1}^{[i+1]} + \left(-\frac{2}{h^2} + 2\alpha Re F_n^{[i]} + 2\alpha^2 \right) F_{n+1}^{[i]} + \frac{1}{h^2} F_{n+1}^{[i-1]} + \frac{5}{h^2} F_{n+1}^{[1]} \\ & - \frac{4}{h^2} F_{n+1}^{[2]} + \frac{1}{h^2} F_{n+1}^{[3]} = \alpha Re (F_n^{[i]})^2 + L + \frac{2}{h^2}, \end{aligned} \quad (9)$$

که در آن $F_n^{[i]} = F_n(x_i)$ و $n = 0, 1, \dots, i = 2, 3, \dots, M-1$ با توجه به این که رابطه (9) یک دستگاه معادلات خطی شامل $M-2$ معادله و $M-1$ مجهول $F_{n+1}^{[i]}$ ، $i = 1, 2, \dots, M-1$ است، لذا برای محاسبه مجهولات به یک معادله دیگر نیاز داریم. این معادله را با استفاده از شرط اولیه $F_{n+1}'(x_0) = 0$ و فرمول تفاضل پیشرو

$$F_{n+1}'(0) = F_{n+1}'(x_0) = \frac{1}{h} \left[-\frac{3}{2} F_{n+1}(x_0) + 2 F_{n+1}(x_1) - \frac{1}{2} F_{n+1}(x_2) \right] + O(h^2),$$

به صورت زیر به دست می آوریم:

$$2 F_{n+1}^{[1]} - \frac{1}{2} F_{n+1}^{[2]} = \frac{3}{2}. \quad (10)$$

توجه داریم که بر اساس شرط اولیه $F(0) = 1$ در تمام محاسبات انجام شده $F_{n+1}(x_0) = F_{n+1}^{[0]} = 1$ اکنون می توان دستگاه معادلات (9) و (10) را به شکل ماتریسی $A_n F_{n+1} = b_n$ بازنویسی کرد که در آن

$$A_n = \begin{bmatrix} \frac{5}{h^2} & -\frac{4}{h^2} & \frac{1}{h^2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{5}{h^2} & -\frac{4}{h^2} & \frac{1}{h^2} & 0 & \dots & 0 \\ \frac{5}{h^2} & -\frac{4}{h^2} & \frac{1}{h^2} & 0 & \dots & 0 \\ \frac{5}{h^2} & -\frac{4}{h^2} & \frac{1}{h^2} & 0 & \dots & 0 \\ 2 & -\frac{1}{2} & 0 & 0 & \dots & 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{h^2} & \gamma_2 & \frac{1}{h^2} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \frac{1}{h^2} & \gamma_{M-3} & \frac{1}{h^2} & 0 \\ 0 & \dots & 0 & \frac{1}{h^2} & \gamma_{M-2} & \frac{1}{h^2} \\ 0 & \dots & 0 & 0 & \frac{1}{h^2} & \gamma_{M-1} \\ 0 & \dots & 0 & 0 & 0 & 0 \end{bmatrix},$$

و $\gamma_i = \frac{2}{h^2} + 2\alpha Re F_n^{[i]} + 2\alpha^2$ همچنین

$$b_n = \left[\delta_2 \quad \delta_3 \quad \delta_4 \quad \dots \quad \delta_{M-1} \quad \frac{3}{2} \right]^T, \quad \delta_i = \alpha Re (F_n^{[i]})^2 + L + \frac{2}{h^2}.$$

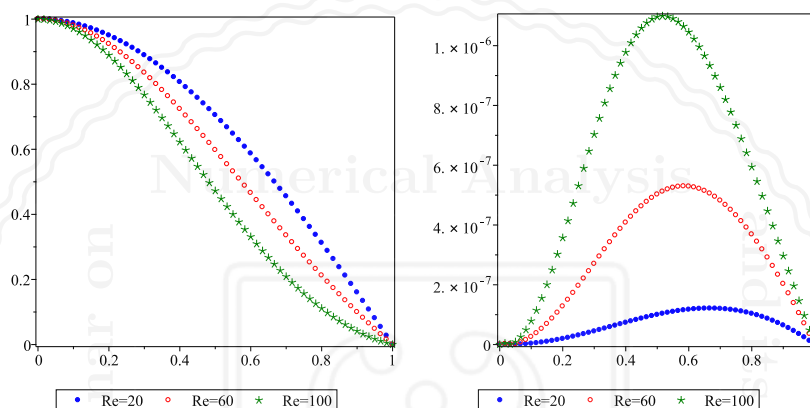
۳. نتایج عددی

در این بخش به بیان نتایج عددی حاصل از روش مطرح شده می پردازیم. لازم به ذکر است که تمام محاسبات با استفاده از نرم افزار میپل، برای $M = 600$ و در پنج تکرار انجام شده است. با توجه به اهمیت مقدار $F''(0)$ در محاسبات فیزیکی در جدول ۱ مقادیر به دست آمده از روش مطرح شده و روش رانگ-کوتا مرتبه چهار ترکیب شده با روش پرتابی^۵ را مقایسه کرده ایم. در شکل ۲ نمودارهای مربوط به جواب تقریبی به دست آمده به ازای مقادیر مختلف عدد رینولدز و خطای مطلق نسبت به روش رانگ-کوتا مرتبه چهار ارائه شده است. نتایج عددی حاصل در جدول ۱ و شکل ۲ دقت قابل قبول و کارایی مناسب روش پیشنهادی را نشان می دهد.

⁵Shooting method

جدول ۱: مقایسه مقادیر عددی $F'''(0)$ حاصل از روش پیشنهادی و روش رانگ-کوتا برای $\alpha = 5^\circ$.

رانگ-کوتا مرتبه چهارم [۳]	روش پیشنهادی	Re
-۲/۵۲۷۱۹۲۲۵	-۲/۵۲۷۱۹۱۶۲	۲۰
-۳/۱۶۹۷۱۲۲۰	-۳/۱۶۹۷۱۱۳۷	۴۰
-۳/۹۴۲۱۴۰۲۸	-۳/۹۴۲۱۳۹۸۳	۶۰
-۴/۸۴۵۰۷۱۸۲	-۴/۸۴۵۰۷۲۴۲	۸۰
-۵/۸۶۹۱۶۵۱۱	-۵/۸۶۹۱۶۷۳۵	۱۰۰



شکل ۲: نمودار جواب‌های تقریبی حاصل از روش پیشنهادی و خطای مطلق آن نسبت به روش رانگ-کوتا به‌ازای مقادیر مختلف Re .

مراجع

- [1] R.L. Burden, J.D. Faires, *Numerical Analysis*, 9th ed. Boston: Brooks/Cole. Butt R, 2010.
- [2] V.B. Mandelzweig, F. Tabakin, Quasilinearization approach to nonlinear problems in physics with application to nonlinear ODEs. *Computer Physics Communications*, 141 (2001) 268-281.
- [3] S.S. Motsa, P. Sibanda, G.T. Marewo, On a new analytical method for flow between two inclined walls. *Numerical algorithms*, 61 (2012) 499-514.

زهرا سادات یزدانی

دانشکده ریاضی، دانشگاه یزد، یزد، ایران.

آدرس ایمیل: z.yazdani@stu.yazd.ac.ir

محمد حیدری

دانشکده ریاضی، دانشگاه یزد، یزد، ایران.

آدرس ایمیل: m.heydari@yazd.ac.ir

اعظم سادات هاشمی

دانشکده ریاضی، دانشگاه یزد، یزد، ایران.

آدرس ایمیل: as.hashemi@stu.yazd.ac.ir

ارائه رویکردی مبتنی بر ماشین بردار پشتیبان ترکیبی برای کشف کلاهبرداری در سرویس‌های پرداخت مالی

مجید ابتیاع، معراج ابتیاع

چکیده. امروزه با رشد روزافزون کارت‌های اعتباری در صنعت بانکی، حجم تراکنش با این کارت‌ها افزایش یافته است. با افزایش تعداد کاربران کارت‌های الکترونیکی و حجم تراکنش‌ها، زمینه تقلب در این حوزه به یکی از مشکلات بزرگ تبدیل شده است. با وجود حجم بسیار زیاد تعداد داده‌ها، شناسایی و کشف تقلب یک امر زمان‌بر و طولانی است و هزینه سنگینی را در پی خواهد داشت. لذا برای شناسایی تقلب و بی‌نظمی مالی استفاده از روش‌های یادگیری ماشین، ضروری به نظر می‌رسد. با وجود حجم انبوه تراکنش‌ها و تعداد کم تراکنش‌های غیرقانونی استفاده از روش‌های ترکیبی که مدل‌های قوی‌تری هستند توصیه می‌شود. در این پژوهش از مدل ماشین بردار پشتیبان به صورت ترکیبی که یک رویکرد نوین است برای کشف و طبقه‌بندی تقلب در کارت‌های اعتباری بهره گرفته می‌شود و سپس مقایسه‌ای بین عملکرد هسته‌های مختلف این مدل صورت می‌گیرد. به منظور نمایش عملکرد مدل، از داده‌های یک بانک اروپایی استفاده شده است. نتایج پژوهش حاکی از دقت و کارایی بالاتر مدل ماشین بردار پشتیبان ترکیبی با هسته گاوسی در مقایسه با هسته‌های دیگر است.

۱. پیش‌گفتار

امروزه یکی از پرکاربردترین روش‌های پرداخت استفاده از کارت‌های اعتباری است که از سرعت بالایی برای انجام تراکنش‌های بانکی برخوردار می‌باشند و یکی از مهم‌ترین ابزارهای توسعه تجارت الکترونیک محسوب می‌شوند. با این حال در زمینه استفاده از کارت‌های بانکی چالش‌های گوناگونی وجود دارد که یکی از این چالش‌های مهم تقلب از کارت‌های بانکی است. کارت‌های بانکی یک هدف بسیار مهم و با ارزش برای سودجویی متقلبان هستند زیرا آن‌ها می‌توانند در زمان بسیار کوتاهی مقدار قابل توجهی پول برداشت کنند در حالی که اغلب این برداشت‌ها در روزهای بعد، کشف و شناسایی می‌شوند. با پیشرفت تکنولوژی مدل‌های مبتنی بر داده‌کاوی^۱ و یادگیری ماشین^۲ توجه بسیاری از پژوهش‌گران را در زمینه‌های مختلف به خود جلب کرده است [۲، ۴]. در این پژوهش، شناسایی تراکنش‌های تقلبی در کارت‌های اعتباری مورد مطالعه قرار گرفته است و به منظور شناسایی این تراکنش‌ها و کشف بی‌نظمی مالی از روش‌های ترکیبی مبتنی بر ماشین بردار پشتیبان^۳ (SVM) استفاده شده است. از آنجایی که پیاده‌سازی ماشین بردار پشتیبان ساده می‌باشد و به‌خوبی می‌تواند با داده‌های کم آموزش ببیند، اما با افزایش حجم داده‌ها این روش از عملکرد مناسبی برخوردار نیست. برای حل این مشکل از ایده ترکیبی بوت استرپ^۴ (تقسیم‌بندی داده‌ها به‌طور تصادفی با جایگذاری با اندازه یکسان) بهره گرفته می‌شود که سبب تشکیل مدل‌های قوی‌تر خواهد شد که این امر موجب افزایش دقت و کارایی مدل می‌شود [۵].

واژه‌های کلیدی: کشف تقلب، ماشین بردار پشتیبان، طبقه‌بندی.
طبقه‌بندی موضوعی [۲۰۱۰]: 68T05, 05C85.

¹Data Mining

²Machine Learning

³Support Vector Machine

⁴Bootstrap

۲. مبانی نظری

مدل ماشین بردار پشتیبان: SVM یکی از پرطرفدارترین الگوریتم های یادگیری ماشین است که جداسازی بهتری روی داده ها نسبت به سایر روش ها در مسائل دسته بندی دارد. دسته بند SVM به دو صورت خطی و غیرخطی وجود دارد. SVM خطی برای داده های قابل تفکیک خطی به کار می رود و از طرفی با توجه به ماهیت بسیاری از داده ها، جهت طبقه بندی آن ها نمی توان از مدل خطی استفاده نمود. در چنین مواقعی از مدل SVM غیرخطی می توان بهره گرفت. توابع هسته^۵ در مسائل جداسازی غیرخطی مورد استفاده قرار می گیرند. از جمله مزایای مدل SVM می توان به آموزش ساده، نظریه ی قوی، عملکرد خوب برای مدل های ساده یا پیچیده و همچنین عملکرد خوب با تعداد داده های آموزشی کم، اشاره نمود. هدف این روش حل مسئله بهینه سازی (۱) است.

$$\begin{cases} \min_{\gamma, w, b} \frac{1}{\gamma} \|w\| + C \sum_{i=1}^m \xi_i, \\ y^{(i)} (w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \dots, m, \\ \xi_i \geq 0, \quad i = 1, \dots, m. \end{cases} \quad (1)$$

به شکل دیگر، می توان از فرم دوگان آن طبق رابطه (۲) استفاده نمود [۱]:

$$\begin{cases} \max_{\alpha} W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{\gamma} \sum_{i,j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle, \\ 0 \leq \alpha_i \leq C, \\ \sum_{i=1}^m \alpha_i y^{(i)} = 0, \end{cases} \quad i = 1, \dots, m, \quad (2)$$

۳. نتایج اصلی

فرآیند مدل پیشنهادی در این پژوهش به دو مرحله تقسیم می شود.

گام اول، نرمال کردن و متوازن سازی داده ها: تعداد نمونه های مجموعه داده بانک اروپایی شامل ۲۸۴۸۰۷ تراکنش است که از این تعداد ۴۹۲ تراکنش برحسب غیرقانونی دارند که نشان از عدم توازن شدید در داده ها است. مجموعه داده X به صورت $X = \{(x_i, y_i) | x_i \in \mathbb{R}^d, y_i = 1, 0 \quad i = 1, \dots, n\}$ که در آن y_i تراکنش قانونی و $y_i = 1$ تراکنش غیرقانونی می باشد را در نظر بگیرید. جهت مقیاس بندی و نرمال سازی داده ها می توان هر ویژگی را در محدوده $[-1, 1]$ یا $[0, 1]$ مقیاس بندی کرد. سپس برای متوازن کردن داده ها از روش نمونه برداری بهره گرفته می شود. به این صورت که از بین ۲۸۴۳۱۵ تراکنش قانونی، به طور تصادفی ۴۹۲ تراکنش را انتخاب و با تراکنش های غیرقانونی ترکیب کرده تا مجموعه داده متوازن تشکیل شود.

گام دوم، بهینه سازی پارامترها و آموزش: مجموعه X به دو بخش داده های آموزشی و آزمایشی تقسیم می شود و سپس داده های آموزشی به N زیرمجموعه $A_j = \{A_1, A_2, A_3, \dots, A_N\}$, $j = 1, 2, \dots, N$ ، با استفاده از روش بوت استرپ افرابندی خواهند شد. سپس برای ساخت مدل ابتدا به روش اعتبارسنجی، پارامترهای مدل ماشین بردار پشتیبان تنظیم و بهترین آن ها انتخاب می شود. پارامترهای بهینه ی دسته بندی به این شکل محاسبه می شوند که ابتدا یک شبکه از مقادیر پارامترها انتخاب می شود و به ازای هر نقطه از این شبکه، پارامترهای این مدل مقداردهی خواهند شد. سپس از روش اعتبارسنجی متقابل برای ارزیابی عملکرد پارامتر در نقطه ی مذکور بهره گرفته می شود. در روش اعتبارسنجی متقابل، داده های آموزشی به طور تصادفی و با جایگذاری، به K قسمت مساوی تقسیم می شوند. از زیرنمونه ها، یکی به عنوان داده های اعتبارسنجی برای آزمایش مدل حفظ می شود و بقیه ی زیرنمونه ها مدل را می سازند. بعد از ساخت مدل، زیرنمونه ی آزمایشی برای میزان اعتبار مدل ساخته شده به کار گرفته می شود. به این ترتیب به ازای هر نقطه ی شبکه، فرآیند اعتبارسنجی متقابل، K بار تکرار می شود و در هر بار معیار عملکرد آن نقطه سنجیده می شود. در انتها از میانگین معیارهای به دست

⁵Kernel functions

آمده برای سنجش میزان اعتبار آن نقطه بهره برده می شود. در میان نقاط مختلف شبکه‌ی مذکور، نقطه‌ای که بهترین اعتبار را تولید کرده است به عنوان پارامترهای مدل اصلی استفاده می شود. در نهایت بعد از انتخاب بهترین پارامترها، الگوریتم ماشین بردار پشتیبان روی مجموعه‌ی A_z اعمال و مدل نهایی دسته‌بند ساخته می شود. در انتها N مدل دسته‌بند به دست می آید که هر کدام بخشی از داده‌های آموزشی را مشاهده کرده‌اند. اکنون برای ارزیابی روش، بین این مدل‌ها روی داده‌های آزمایشی رأی‌گیری انجام می شود تا کلاس هر کدام از داده‌های آزمایشی پیش‌بینی شود. به این معنا که هر یک از داده‌های آزمایشی به N مدل ارائه می شوند و کلاس آن‌ها توسط هر کدام از دسته‌بندها پیش‌بینی می شود. در نهایت بین این پیش‌بینی‌ها رأی‌گیری صورت گرفته و پیش‌بینی نهایی آن داده مشخص می شود. با انجام این کار می توان بین پیش‌بینی کلاس و کلاس واقعی داده‌های آزمایشی مقایسه انجام داد و عملکرد روش را سنجید. در این پژوهش مدل پیشنهادی بر روی مجموعه داده بانک اروپایی [۲] پیاده‌سازی شده و سپس مدل ماشین بردار پشتیبان ساخته شده به کمک ماتریس درهم‌ریختگی^۶ و پارامترهای زیر ارزیابی خواهد شد.

۱. معیار دقت^۷ (ACC)، نشان‌دهنده دقت کلی مدل و توانایی مدل در دسته‌بندی کلی تراکنش‌های قانونی و غیرقانونی می باشد:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}.$$

۲. معیار ارزش پیش‌گویی کننده مثبت^۸ (PPV)، نشان‌دهنده پیش‌بینی‌های صحیح انجام شده برای تراکنش‌های غیرقانونی می باشد:

$$PPV = \frac{TP}{TP + FP}.$$

۳. معیار نرخ مثبت صحیح^۹ (TPR) یا پوشش تراکنش‌های غیرقانونی صحیح، مشخص می کند که به چه نسبتی پیش‌بینی صحیح صورت گرفته است:

$$TPR = \frac{TP}{TP + FN}.$$

۴. چون معیارهای (۲) و (۳) عکس یکدیگر هستند و افزایش یکی باعث کاهش دیگری و بالعکس می شود، پس از معیار دیگری به نام معیار F_1 (F1-Score) استفاده می شود که میانگین معیار (۲) و (۳) است تا عملکرد کلی مدل ارزیابی شود:

$$F_1 - Score = \frac{2TP}{2TP + FP + FN}.$$

مقیاس تمامی معیارهای فوق، بین صفر و یک می باشد که هر چه مقادیر به یک نزدیک تر باشند، نشان از عملکرد خوب مدل دسته‌بند است. نتایج حاصل از مدل‌سازی در جدول (۱) مشاهده می شود.

۴. دست‌آوردهای پژوهش

با توجه به جدول (۱)، SVM ساده با هسته گاوسی^{۱۰} در تمام معیارها عملکرد بهتری نسبت به هسته سیگموئید^{۱۱} و هسته چندجمله‌ای^{۱۲} دارد. در عملکرد SVM ترکیبی نیز نتایج مشابهی دیده می شود و این مدل با هسته گاوسی در همه پارامترها عملکرد بهتری را نسبت به دیگر هسته‌ها دارد. لذا می توان گفت، SVM با هسته گاوسی از دقت بالاتر و تعمیم‌پذیری مناسب‌تری برخوردار است که باعث

^۶Confusion Matrix

^۷Accuracy

^۸Positive Predictive Value

^۹True Positive Rate

^{۱۰}Gaussian Kernel

^{۱۱}Sigmoidal Kernel

^{۱۲}Polynomial Kernel

جدول ۱: نتایج حاصل از ماشین بردار پشتیبان ساده و ترکیبی

مدل های ارزیابی				هسته	مدل
ACC	PPV	TPR	F1-Score		
۰/۵	۰/۶	۰/۹۴	۰/۷۷	چندجمله ای	ماشین بردار پشتیبان ساده
۰/۵۵	۰/۶۴	۰/۹۴۳	۰/۷۳	سیگموئید	
۰/۷	۰/۷۱	۰/۹۵	۰/۸۵	گوسی	
۰/۵۴	۰/۶۱	۰/۹۳	۰/۷۹	چندجمله ای	ماشین بردار پشتیبان ترکیبی
۰/۵۷	۰/۷۰۵	۰/۸۷	۰/۷۶	سیگموئید	
۰/۷۶۵	۰/۷۳۳	۰/۹۶	۰/۸۸	گوسی	

طبقه بندی دقیق تر و پیش بینی درست تر تراکنش ها خواهد شد. همچنین SVM ترکیبی با هسته گوسی نیز نسبت به SVM ساده با هسته گوسی نتایج بهتری را دارا می باشد که این نشان دهنده برتری مدل ترکیبی پیشنهادی نسبت به مدل ساده است. در این پژوهش تلاش شده است تا با استفاده از رویکرد SVM ترکیبی که از روش های نوین یادگیری ماشین و برگرفته از روش جنگل تصادفی^{۱۳} (RF) است، مدلی کارا جهت طبقه بندی تراکنش های مالی ارائه شود. از آن جایی که SVM جزو دسته بندی های قوی می باشد با کمک روش ترکیبی بر پایه ی مدل SVM می توان مدل هایی با دقت بالاتر و تعمیم پذیری بیشتری به وجود آورد. نتایج این رویکرد نشان دهنده کارایی بیشتر دسته بندی های ترکیبی است.

سپاس گذاری

از اساتیدی که در تهیه این مقاله ما را یاری نموده اند کمال تشکر را داریم.

مراجع

- [1] C. Cortes, V. Vapnik, Support vector machine, *Machine learning*, 20 (1995) 273–297.
- [2] J. Han, J. Pei, and M. Kamber, *Data mining: Concepts and Techniques*, Elsevier, 2011.
- [3] <https://datahub.io/machinelearning/creditcard>. Checked on 20 Jan, 2022.
- [4] O. Kolodiziev, A. Mints, P. Sidelov, I. Pleskun, O. Lozynska, Automatic Machine Learning Algorithms for Fraud Detection in Digital Payment Systems, 5 (2020) 14–26.
- [5] D. Meenakshi, Janani, Gayathri, MRS. IndiraN, Credit card fraud detection using random forest. *International Research Journal of Engineering and Technology*, 6 (2019) 6662–6666.

مجید ابتیاع

دانشکده علوم پایه، دانشگاه آیت الله بروجردی، بروجرد، ایران

آدرس ایمیل: majid.ebtia@gmail.com

معراج ابتیاع

دانشکده فنی و مهندسی، دانشگاه آیت الله بروجردی، بروجرد، ایران

آدرس ایمیل: ebtia.meraj@gmail.com

¹³Random Forest

مروری بر حل مسأله تخمین پارامتر معادله دیفرانسیل جزئی واکنش-انتشار

یاسمن رزاقی، خدیجه ندایی اصل، سلمان خدایی فر

چکیده. پارامترهای معادلات دیفرانسیل جزئی تفسیرهای علمی جالبی دارند، اما اغلب ناشناخته هستند و باید از داده‌های تجربی تخمین زده شوند. در این مقاله مسائل تخمین پارامتر در معادلات دیفرانسیل جزئی را شرح داده و سپس برای تخمین پارامتر معادله دیفرانسیل جزئی واکنش-انتشار از روش عناصر منتهای برای گسسته‌سازی معادله دیفرانسیل جزئی و از روش تندترین کاهش برای حل مسأله بهینه‌سازی استفاده می‌کنیم.

۱. پیش‌گفتار

معادلات دیفرانسیل جزئی کاربردهای فراوانی در توصیف پدیده‌های فیزیکی، شیمیایی و زیستی دارند. پارامترهای بسیاری از پدیده‌های فیزیکی مشخص نیستند و نیاز به ارائه تقریبی از آن‌ها است. ارائه تقریب از این پارامترها منجر به مسائل تخمین پارامتر می‌شود که در حالت کلی این مسائل با عنوان مسائل بهینه‌سازی با قیود معادلات دیفرانسیل جزئی شناخته می‌شوند [۴]. باتوجه به این که اغلب مسائل راه‌حل تحلیلی ندارند، پیاده‌سازی روش‌های عددی در پیش‌برد حل این مسائل موثر است. مطالعه‌ی علمی سیستم‌های فیزیکی، به سه بخش تقسیم می‌شود:

الف. پارامتری کردن سیستم: یافتن مجموعه‌ی کمینه به طوری که بتواند سیستم را به طور کامل مدل‌بندی کند،

ب. مدل‌بندی مستقیم: یافتن قوانین فیزیکی به طوری که بتواند برای مقادیر داده شده مدل برای پیش‌بینی یا نتیجه برای بعضی از پارامترهای قابل مشاهده ارائه داد،

ج. مدل‌بندی وارون: استفاده کردن از مقادیر بعضی از پارامترها برای استنباط مقادیر واقعی پارامترهای مدل [۵].

مسائل تخمین پارامتر معادلات دیفرانسیل جزئی در رده‌بندی مسائل وارون قرار می‌گیرند که به عنوان بازیابی یا تخمین ضرایب بر پایه یک سری داده‌های اندازه‌گیری شده از معادله دیفرانسیل جزئی بیان می‌شوند. رهیافت استفاده از روش‌های بهینه‌سازی برای حل مسائل تخمین پارامتر است [۱]. معادله دیفرانسیل جزئی واکنش-انتشار با شرط مرزی دیریکله در ناحیه $\Omega = [0, 1]^2$ را به صورت

$$\begin{cases} -\nabla \cdot (m \nabla u)(x) + \vec{v} \cdot \nabla u(x) = f(x), & x \in \Omega, \\ u(x) = 0, & x \in \Gamma, \end{cases} \quad (1)$$

در نظر می‌گیریم. این مسأله که به عنوان مسأله رسانای گرما یا جریان آب زیرزمینی شناخته می‌شود، هدف بازیابی $m \in L^\infty(\Omega)$ است، داده‌های اندازه‌گیری شده را با u_d مشخص می‌کنیم. حال اگر تابع $f = 1$ و m واقعی برابر با

$$m_t(x, y) = \begin{cases} 4, & (x, y) \in \sqrt{(x - \frac{1}{4})^2 + (y - \frac{1}{4})^2} \leq \frac{1}{5}, \\ 8, & \text{سایر جاها}, \end{cases}$$

واژه‌های کلیدی: معادلات دیفرانسیل جزئی، مسائل وارون، بهینه‌سازی، تخمین پارامتر.
طبقه‌بندی موضوعی [۲۰۱۰]: 35R30, 49N45.

باشد، روند حل مدل (۱) را با استفاده از [۷] بررسی می‌کنیم. رهیافت انتخاب شده برای حل مسأله تخمین پارامتر (۱) استفاده از روش‌های بهینه‌سازی است، روش انتخابی برای گسسته‌سازی و حل معادله دیفرانسیل جزئی، عناصر متناهی است [۶] که با استفاده از کتابخانه فینکس [۳] در پایتون انجام می‌گردد. با توجه به نامشخص بودن ضریب m در مدل (۱)، نیاز به یک مقدار اولیه مناسب m_i وجود دارد تا گسسته‌سازی معادله دیفرانسیل جزئی انجام گردد. با توجه به این‌که در این مدل m_t مشخص شده است، می‌توان معادله دیفرانسیل جزئی را ابتدا با m_t حل کرد و سپس به داده‌ها با استفاده از توزیع‌های آماری اختلال وارد کرد و در این مدل، داده‌های اندازه‌گیری شده u_d را ساخت. رهیافت حل مسأله تخمین پارامتر، روش‌های بهینه‌سازی است، بنابراین با استفاده از روش‌های برپایه گرادین، روش تندترین کاهش که نیاز به محاسبه مشتق تابع هدف و تبدیل مسأله مقید بهینه‌سازی به مسأله نامقید است از روش الحاق (تابع لاگرانژین) استفاده می‌کنیم [۲]. گام‌های لازم برای حل مدل (۱) را بررسی می‌کنیم:

۱. نوشتن مسأله بهینه‌سازی: هدف یافتن m به‌گونه‌ایست که فاصله بین u و u_d (مشاهدات) کمترین مقدار باشد:

$$\begin{cases} \min_m J(m) = \frac{1}{\gamma} \int_{\Omega} (u - u_d)^2 dx + \frac{\gamma}{\gamma} \int_{\Omega} |\nabla m|^2 dx, \\ \text{s.t. } \nabla \cdot (m \nabla u)(x) + \vec{v} \cdot \nabla u(x) = f(x), \quad x \in \Omega, \\ u(x) = 0, \quad x \in \Gamma. \end{cases} \quad (2)$$

۲. نوشتن فرم ضعیف معادله دیفرانسیل جزئی: جواب معادله دیفرانسیل جزئی (۱) را با استفاده از روش عناصر متناهی با انتخاب $\phi \in H_0^1(\Omega)$ ، تقریب می‌زنیم. بنابراین فرم ضعیف معادله را می‌نویسیم:

$$\begin{cases} \langle -\nabla \cdot (m \nabla u) + \vec{v} \cdot \nabla u, \phi \rangle = \langle f, \phi \rangle, \\ \langle -\nabla \cdot (m \nabla u), \phi \rangle + \langle \vec{v} \cdot \nabla u, \phi \rangle = \langle f, \phi \rangle, \\ -\langle m \nabla u, \phi \rangle + \langle \vec{v} \cdot \nabla u, \phi \rangle = \langle f, \phi \rangle. \end{cases} \quad (3)$$

۳. تابع لاگرانژین: طبق تعریف مدل (۱)، $m \in L^\infty(\Omega)$ است، بنابراین تابع لاگرانژین به‌صورت

$$L : L^\infty(\Omega) \times H_0^1(\Omega) \times H_0^1(\Omega) \rightarrow \mathbb{R},$$

$$L(m, u, p) := \frac{1}{\gamma} \langle u - u_d, u - u_d \rangle + \frac{\gamma}{\gamma} \langle \nabla m, \nabla m \rangle + \langle m \nabla u, \nabla p \rangle + \langle \vec{v} \nabla u, p \rangle - \langle f, p \rangle,$$

تعریف می‌شود،

$$\begin{cases} L_p(m, u, p)(\tilde{p}) = \langle m \nabla u, \tilde{p} \rangle + \langle \vec{v} \nabla u, \tilde{p} \rangle - \langle f, \tilde{p} \rangle = 0, \\ L_u(m, u, p)(\tilde{u}) = \langle m \nabla u, \nabla \tilde{u} \rangle + \langle u - u_d, \tilde{u} \rangle + \langle \vec{v} \nabla \tilde{u}, p \rangle = 0, \\ L_m(m, u, p)(\tilde{m}) = \gamma \langle \nabla m, \nabla \tilde{m} \rangle + \langle \tilde{m} \nabla u, p \rangle = 0. \end{cases}$$

۴. محاسبه گرادین تابع هدف: متغیر مسأله بهینه‌سازی (۲)، تابع m است، لذا گرادین تابع هدف $J(m)$ را نسبت به m محاسبه می‌کنیم که برابر با

$$\langle g, \tilde{m} \rangle := G(m)(\tilde{m}) = \gamma \langle \nabla m, \nabla \tilde{m} \rangle + \langle \tilde{m} \nabla u, p \rangle,$$

روش تندترین کاهش با جستجوی خطی آرمیجو^۱ با ثابت c که در پی‌یافتن β است، را به کار می‌بریم

$$J(m - \beta g) \leq J(m) - \beta c \langle g, g \rangle,$$

که در این الگوریتم شرط توقف $\frac{|g_m|}{|g|} \leq \tau$ است.

¹ Armijo

۵. داده‌های اندازه‌گیری شده (مشاهدات): در این مسئله داده‌های اندازه‌گیری شده را با اعمال اختلال بر داده‌های اولیه حاصل از حل مسئله می‌سازیم. اگر مسئله را با m_t حل کنیم و داده‌های حاصل را u_t بنامیم در این صورت

$$u_d = u_t + \eta, \quad \eta \sim N(0, \sigma^2).$$

در حل این مسئله اختلال بر داده‌ها را با استفاده از توزیع لاپلاس انجام می‌دهیم.

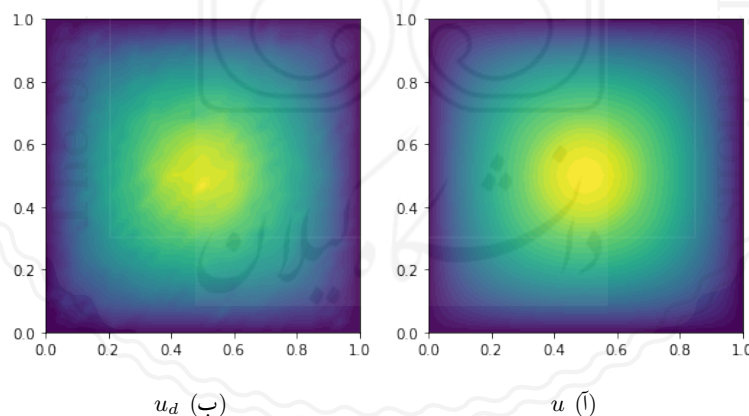
۶. باتوجه به این که m در مسئله مجهول است، با تعریف مقدار اولیه برای m اولین گام حل معادله دیفرانسیل جزئی و محاسبه تابع هدف انجام می‌شود.

۲. نتایج عددی

در این بخش به حل مسئله تخمین پارامتر

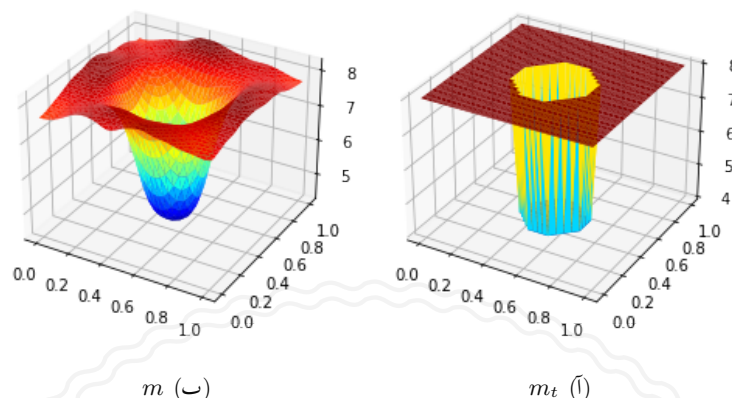
$$\begin{cases} \min_m J(m) = \frac{1}{2} \int_{\Omega} (u - u_d)^2 dx + \frac{1 \times 10^{-5}}{2} \int_{\Omega} |\nabla m|^2 dx, \\ \text{s.t.} \quad \nabla \cdot (m \nabla u)(x) = 1, \quad x \in \Omega, \\ u(x) = 0, \quad x \in \Gamma, \end{cases} \quad (4)$$

با $m_i = 4$ و اختلال $\alpha = 0.01$ می‌پردازیم. ابتدا با m_t قید مسئله (۴) را حل کرده، سپس اختلال $\alpha = 0.01$ را با استفاده از توزیع لاپلاس به داده‌ها وارد می‌کنیم و آن را u_d می‌نامیم (شکل ۱).



شکل ۱: u و u_d مسئله (۴).

در گام بعدی با استفاده از مقدار اولیه $m_i = 4$ ، تابع u را به دست می‌آوریم. پس از ۲۸۵ تکرار، مساله در شرط همگرایی صدق کرده و اجرای الگوریتم به پایان می‌رسد، در پایان m_t و m نهایی بازیابی شده را رسم می‌کنیم (شکل ۲).



شکل ۲: m و m_t نهایی بازیابی شده

مراجع

- [1] E. Haber, L. Hanson, *Model problems in PDE-constrained optimization*, Report, Emory University, 2007.
- [2] M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, *Optimization with PDE constraints*, Springer Science Business Media, 2008.
- [3] H.P. Langtangen, A. Logg, *Solving PDEs in python: the FEniCS tutorial I*. Springer Nature, 2017.
- [4] F. Troltzsch, *Optimal control of partial differential equations: theory, methods, and applications*, American Mathematical Society, 2010.
- [5] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*, Society for Industrial and Applied Mathematics, USA, 2004.
- [6] J.N. Reddy, *An introduction to the finite element method*, McGraw-Hill New York, 2004.
- [7] G. Omar, *Computational and Variational Inverse Problems*, Fall 2015.

یاسمن رزاقی

دانشکده ریاضی، دانشگاه تحصیلات تکمیلی علوم پایه، زنجان، ایران.

آدرس ایمیل: y.razzaghi@iasbs.ac.ir

خدیجه ندایی اصل

دانشکده ریاضی، دانشگاه تحصیلات تکمیلی علوم پایه، زنجان، ایران.

آدرس ایمیل: nedaiasl@iasbs.ac.ir

سلمان خدایفر

دانشکده ریاضی، دانشگاه تحصیلات تکمیلی علوم پایه، زنجان، ایران.

آدرس ایمیل: s.khodayifar@iasbs.ac.ir

حل عددی دستگاه معادلات دیفرانسیل حرکت پرتابگر با رویکرد فازی

سعید سرآبادان، محمد مقدس، علی آخوندزاده

چکیده. در این مقاله قصد داریم حل عددی دستگاه معادلات دیفرانسیل حرکت پرتابگرها را با رویکرد فازی مورد بحث قرار دهیم. بعد از مدل‌بندی مسئله و تعیین پارامترهای نوسانی پرتابگرها باید بتوان آنها را به صورت مناسبی تقریب زد. در این مقاله، روش درونیابی نوینی با استفاده از تعمیم شبکه تابع پایه شعاعی و با در نظر گرفتن مختصات و ساختار فضایی داده‌ها ارائه شده است. در این روش، ماتریس تعمیم‌یافته توابع انتقال متشکل از توابع انتقال تصحیح، مختصات گره‌های واحد پنهان، برای حل ماتریس ضرایب اوزان استفاده و در پایان درونیابی در هر نقطه از شبکه انجام می‌شود. در ادامه با استفاده از روش حل معادلات دیفرانسیل نسبی فازی، دستگاه معادلات نسبی حرکت پرتابگرها را حل خواهیم کرد که قبلاً با روش‌های سنتی حل شده است. در پایان نتایج روش پیشنهادی را با نتایج دیگر روش‌های مرسوم مقایسه خواهیم کرد.

۱. مقدمه

یکی از ابزارهای دفاعی پرکاربرد، پرتاب موشک از پرتابگر است که شامل شاسی خودرو، سکوی کج و کانتینر است. تعیین نوسانات و تأثیرات آنها بر شرایط اولیه مسیر موشک، یکی از مهمترین مسائل در پرتاب راکت شیب دار است. این پدیده، پایداری وسیله پرتابگر و دقت شلیک را تحت تأثیر قرار می‌دهد. محققان در منابع [۴، ۵] با بررسی تمام نیروها و گشتاورهای وارده بر پرتابگر توانسته‌اند آنالیز دقیقی از مسئله داشته باشند و آن را فرمول‌بندی کنند. شبکه‌های مبتنی بر تابع پایه شعاعی^۱ (RBF) یکی از ابزار کاربردی برای تقریب توابع چندمتغیره است. توابع پایه شعاعی برای اولین بار برای حل مسئله درونیابی چندمتغیره مورد استفاده قرار گرفت [۳]. در حال حاضر، مبحث یادشده یکی از زمینه‌های اصلی پژوهش در تجزیه و تحلیل عددی است. در زمینه شبکه‌های عصبی، توابع پایه شعاعی برای اولین بار توسط برومهد و لو^۲ [۱] به کار گرفته شد. واحدهای پنهان در شبکه‌های عصبی، به صورت مجموعه‌ای از توابع به منظور ترکیب بردارهای ورودی بر پایه الگوهای تصادفی بیان می‌شوند؛ توابع مورد استفاده در این نوع شبکه‌ها، توابع پایه شعاعی هستند. در بسیاری از سیستم‌های خطی، برخی از پارامترهای سیستم مبهم یا نادقیق هستند و ریاضیات فازی برای مدل‌سازی این مسائل ابزار مناسب‌تری است و از این رو حل یک سیستم خطی فازی یا یک معادله دیفرانسیل فازی اهمیت بیشتری پیدا می‌کند. در این مقاله بعد از مدل‌بندی مسئله و تعیین پارامترهای نوسانی پرتابگرها، آنها را با استفاده از روش RBF تقریب می‌زنیم و در ادامه دستگاه معادلات نسبی حرکت پرتابگرها با پارامترهای فازی را حل خواهیم کرد.

واژه‌های کلیدی: دستگاه معادلات دیفرانسیل حرکت پرتابگر، توابع پایه شعاعی، معادلات دیفرانسیل فازی، دستگاه معادلات فازی. طبقه‌بندی موضوعی [۲۰۱۰]: 65P99, 35R13.

¹Radial Basis Function

²Broomhead and Lowe

۲. شبکه توابع پایه شعاعی

تعریف ۱. تابع $\Phi: \mathbb{R}^d \rightarrow \mathbb{R}$ را یک تابع شعاعی^۳ گوئیم هرگاه تابع یک متغیره‌ای مانند $\phi: [0, \infty) \rightarrow \mathbb{R}$ موجود باشد به طوری که

$$\Phi(x) = \varphi(\|x\|_2, \varepsilon),$$

که $x = (x_1, x_2, \dots, x_d)$ ، $\|\cdot\|_2$ نرم اقلیدسی و $\varepsilon \in \mathbb{R}$ پارامتر شکل است.

تعریف ۲. مجموعه‌ی توابع $B = \{\Phi_1, \Phi_2, \dots, \Phi_n\}$ را توابع پایه‌ای شعاعی گوئیم هرگاه B یک پایه و هر یک از Φ_i ها یک تابع شعاعی باشد.

۱.۲. ساختار شبکه توابع پایه شعاعی

شبکه RBF، به عنوان یک شبکه عصبی، شامل سه لایه ورودی، میانی و خروجی است. لایه ورودی با وجود گره‌های زیاد، دارای ابعاد بالا است. لایه میانی دارای n گره، با مرکز C_n است و فرایندی که در لایه میانی صورت می‌گیرد، این است که ابتدا فاصله بردار ورودی از مرکز متناظر محاسبه، تابع انتقال φ اعمال و سپس مقدار عددی حاصل از لایه خروجی y عبور داده می‌شود. آخرین لایه دارای تبدیل خطی است که یک ترکیب خطی از مقادیر لایه میانی است.

۳. معادلات دیفرانسیل فازی

تعریف ۳. عدد فازی یک مجموعه فازی مانند $u: \mathbb{R} \rightarrow [0, 1]$ است که در شرایط زیر صدق کنند:

• u یک مجموعه فازی نرمال باشد.

• u یک مجموعه فازی محدب باشد.

• u نیم پیوسته بالایی باشد.

• $\text{Supp}(u)$ فشرده باشد.

تعریف ۴. شکل پارامتری عدد فازی u به صورت یک زوج مانند (\underline{u}, \bar{u}) از توابع $\underline{u}(r)$ و $\bar{u}(r)$ ، $0 \leq r \leq 1$ که در خواص زیر صدق می‌کنند:

• $\underline{u}(r)$ یک تابع کراندار اکیدا نزولی از چپ پیوسته است.

• $\bar{u}(r)$ یک تابع کراندار اکیدا صعودی از راست پیوسته است.

• $0 \leq r \leq 1$ ، $\underline{u}(r) \leq \bar{u}(r)$.

۵. دستگاه ماتریسی

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{pmatrix} * \begin{pmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \cdots & \tilde{x}_{1n} \\ \tilde{x}_{21} & \tilde{x}_{22} & \cdots & \tilde{x}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{x}_{m1} & \tilde{x}_{m2} & \cdots & \tilde{x}_{mn} \end{pmatrix} = \begin{pmatrix} \tilde{b}_{11} & \tilde{b}_{12} & \cdots & \tilde{b}_{1n} \\ \tilde{b}_{21} & \tilde{b}_{22} & \cdots & \tilde{b}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{b}_{m1} & \tilde{b}_{m2} & \cdots & \tilde{b}_{mn} \end{pmatrix}, \quad (1)$$

³Radial Function

به طوری که a_{ij} ، $1 \leq i, j \leq m$ اعداد قطعی و \tilde{b}_{ij} ، $1 \leq i \leq m$ ، $1 \leq j \leq n$ اعداد فازی هستند، یک دستگاه معادلات ماتریسی فازی ^۴ (FMES) گفته می‌شود. با استفاده از نمادگذاری ماتریسی‌ها داریم:

$$A\tilde{X} = \tilde{B} \quad (۲)$$

و همچنین ماتریس اعداد فازی

$$\tilde{X} = (\tilde{x}_{ij}) = (\bar{x}_{ij}(r), \underline{x}_{ij}(r)), \quad 1 \leq i \leq m, \quad 1 \leq j \leq n, \quad 0 \leq r \leq 1,$$

یک جواب از معادله ماتریسی فازی معادله (۲) گفته می‌شود اگر \tilde{X} در معادله (۲) صدق کند.

۱.۳. حل دستگاه معادلات ماتریسی فازی

اکنون جواب‌های دستگاه معادلات فازی (۱) را بررسی خواهیم کرد. برای این منظور ابتدا دستگاه را به یک دستگاه ماتریسی قطعی تبدیل کرده و سپس دستگاه فازی را با استفاده از آن حل کرده و سه تقریب برای آن ارائه خواهیم داد. ابتدا، دستگاه ماتریسی (۱) را به یک دستگاه ماتریسی بر مبنای ضرب کرونگر ماتریسی‌ها تبدیل می‌کنیم.

قضیه ۱ ([۲]). فرض کنیم $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ و $\tilde{X} = (\tilde{x}_{ij}) = (\underline{x}_{ij}(r), \bar{x}_{ij}(r)) \in \mathbb{R}^{m \times n}$ و همچنین $B = (b_{ij}) \in \mathbb{R}^{m \times n}$ در این صورت

$$\text{vec}(A\tilde{X}B) = (B^T \otimes A)\text{vec}(\tilde{X}).$$

که $\text{vec}(\cdot)$ توسعه یک ماتریس، روی ستون‌هایش و \otimes ضرب کرونگر است.

قضیه ۲ ([۲]). ماتریس $\tilde{X} \in E^{m \times n}$ یک جواب دستگاه ماتریسی فازی (۱) است اگر و تنها اگر $\tilde{x} = \text{vec}(\tilde{X}) \in E^{mn}$ یک جواب دستگاه ماتریسی زیر باشند:

$$G\tilde{x} = \tilde{y}, \quad (۳)$$

که در آن $G = I_n \otimes A$ یک ماتریس $mn \times mn$ و $\tilde{y} = \text{vec}(\tilde{B})$ یک بردار mn بعدی از اعداد فازی است.

نکته ۱. دستگاه خطی فازی (۳) می‌تواند به یک دستگاه از معادلات قطعی به صورت زیر توسعه داده شود:

$$\begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1p} \\ g_{21} & g_{22} & \cdots & g_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ g_{p1} & g_{p2} & \cdots & g_{pp} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix} = \begin{pmatrix} \tilde{y}_1(1) \\ \tilde{y}_2(1) \\ \vdots \\ \tilde{y}_p(1) \end{pmatrix},$$

که دستگاه فوق معادل است با

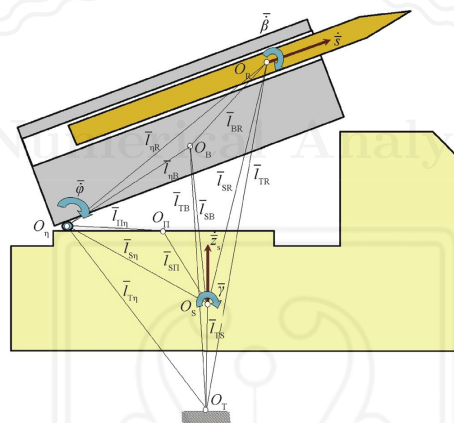
$$\sum_{i=1}^p \sum_{j=1}^p g_{ij} (x_i - \alpha_j(r), x_i + \alpha_j(r)) = (\underline{y}_i(r), \bar{y}_i(r)),$$

به طوری که $\bar{y}_i(1), \underline{y}_i(1) \in \mathbb{R}$ و $i = 1, \dots, p$ ، $\alpha_i(r)$ و $i = 1, \dots, p$ توابع مشخصی هستند.

⁴Fuzzy Matrix Equation System

۴. معادلات دینامیکی کلی پرتابگر راکت

در این مقاله، دستگاه نوسان‌داری بررسی شده است که از اتصال راکت و پرتابگر بر بدنه‌ای سخت مانند شکل ۱ تشکیل شده است، همچنین اجزای این دستگاه به وسیله عناصر کشسان به هم وصل شده‌اند. به طور خلاصه، وسیله پرتابگر دارای دو جزء اصلی، شاسی خودرو (این امکان را فراهم می‌آورد تا بتوان پرتابگر و راکت را از مکانی به مکان دیگر انتقال داد) و سکوی کج (معمولاً دارای یک کانتینر برای راکت است) است. در سیستم پرتابه (راکت در حال پرتاب)، شش نقطه مشخصه وجود دارد (شکل ۱): O_T (نقطه ثابت وابسته به زمین)، O_s (مرکز جرم شاسی و وسیله نقلیه)، O_{pi} (مرکز جرم محافظ چرخان)، O_η (نقطه تلاقی میله چرخان سکوی کج و صفحه عمودی وسیله در حال پرتاب)، O_B (مرکز جرم سکوی کج)، O_R (مرکز جرم راکت در حال پرتاب).



شکل ۱: نمودار محاسبات پرتابگر راکت.

۱.۴. فرم ماتریسی معادلات حرکت سیستم پرتابه

اکنون فرم ماتریسی معادلات دینامیکی حرکت سیستم پرتابه را ارائه می‌دهیم. متغیرهای دینامیکی مستقل حرکت پرتابگر را می‌توان با بردار ستونی زیر نشان داد [۵]:

$$X_{6 \times 1} = [s \quad \varphi_y \quad \varphi_z \quad z_s \quad \gamma_x \quad \gamma_y]^T,$$

که در آن z_s میزان انتقال شاسی، γ_y میزان دوران شاسی (حرکت قوسی شاسی)، γ_x میزان دوران شاسی (حرکت چرخشی شاسی)، φ_z دوران سکوی کج (دوران حول محور عمودی)، φ_y دوران سکوی کج (حرکت قوسی سکوی) و s میزان انتقال راکت است. در این صورت می‌توان شکل ماتریسی دستگاه معادلات دیفرانسیلی مرتبه دوم شامل مؤلفه‌های حرکتی پرتابگر را به صورت زیر داشت:

$$\ddot{X}_{6 \times 1} = B_{6 \times 6} \cdot \dot{X}_{6 \times 1} + C_{6 \times 6} \cdot X_{6 \times 1} + N_{6 \times 15} \cdot \xi_{15 \times 1} + F_{6 \times 3} \cdot \Phi_{3 \times 1}, \quad (4)$$

به طوری که در آن $B_{6 \times 6}$ ماتریس ضرایب سرعت اولیه $\dot{X}_{6 \times 1}$ ، $C_{6 \times 6} = (c_{i,j})_{i,j=1,6}$ ماتریس ضرایب متغیرهای مجهول X ، $N_{6 \times 15} = (n_{i,j})_{i,j=1,15}$ ماتریس ضرایب ترکیب غیرخطی مجهولات متغیر زیر است:

$$\xi_{15 \times 1} = \left[\dot{\gamma}_x^2 \quad \dot{\gamma}_y^2 \quad \dot{\gamma}_x \dot{\gamma}_y \quad \dot{\varphi}_y^2 \quad \dot{\varphi}_z^2 \quad \dot{\varphi}_y \dot{\varphi}_z \quad \dot{\gamma}_x \dot{\varphi}_y \quad \dot{\gamma}_x \dot{\varphi}_z \quad \dot{\gamma}_y \dot{\varphi}_y \quad \dot{\gamma}_y \dot{\varphi}_z \quad \dot{s} \dot{\gamma}_x \quad \dot{s} \dot{\gamma}_y \quad \dot{s} \dot{\varphi}_y \quad \dot{s} \dot{\varphi}_z \quad \mu \sqrt{F_{Ry}^2 + F_{Rz}^2} \right]^T. \quad (5)$$

همچنین، $F_{6 \times 3} = (f_{i,j})_{i=1,6, j=1,3}$ ماتریس نیروهای خارجی است که بر دستگاه زیر اثر می‌کنند:

$$\Phi_{3 \times 1} = \begin{bmatrix} g & T & F_{jet} \end{bmatrix}^T. \quad (6)$$

بردار (۶) برای بیان اثر نیروهای خارجی بر حرکت دستگاه استفاده می‌شود. در این بردار اولین عنصر متناظر با نیروی وزن، دومین عنصر متناظر با فشار راکت و عنصر آخر متناظر با نیروی جت راکت است.

مدل ریاضی بیان شده می‌تواند برای هر وسیله پرتابگری شبیه پرتابگر توصیف شده استفاده شود. برای حل دستگاه مورد بحث، در ابتدا باید دستگاه شامل معادلات دیفرانسیلی مرتبه دوم (۵) را به یک دستگاه از معادلات دیفرانسیلی مرتبه اول کاهش داد. برای این منظور متغیرهای زیر را تعریف می‌کنیم:

$$v_s = \dot{s}, \quad v_{zs} = \dot{z}_s, \quad \omega_{\varphi_y} = \dot{\varphi}_y, \quad \omega_{\varphi_z} = \dot{\varphi}_z, \quad \omega_{\gamma_x} = \dot{\gamma}_x, \quad \omega_{\gamma_y} = \dot{\gamma}_y. \quad (7)$$

با استفاده از متغیرهای جدید تعریف شده با روابط (۷)، بردار متغیرهای مجهول را می‌توان به صورت زیر نمایش داد:

$$X_{12 \times 1} = [v_s \ \omega_{\varphi_y} \ \omega_{\varphi_z} \ v_{zs} \ \omega_{\gamma_x} \ \omega_{\gamma_y} \ s \ \varphi_y \ \varphi_z \ z_s \ \gamma_x \ \gamma_y]^T. \quad (8)$$

حال با استفاده از متغیرهای جدید و بردار (۸) و همچنین رابطه (۴) می‌توان شکل ماتریسی جدید شامل معادلات دیفرانسیلی مرتبه اول را به صورت زیر نوشت:

$$\dot{X}_{12 \times 1} = P_{12 \times 12} \cdot X_{12 \times 1} + Q_{12 \times 15} \cdot \xi_{15 \times 1} + R_{12 \times 3} \cdot \Phi_{3 \times 1}, \quad (9)$$

که در آن

$$P_{12 \times 12} = \begin{bmatrix} B_{6 \times 6} & C_{6 \times 6} \\ I_{6 \times 6} & O_{6 \times 6} \end{bmatrix}, \quad Q_{12 \times 15} = \begin{bmatrix} N_{6 \times 15} \\ O_{6 \times 15} \end{bmatrix}, \quad R_{12 \times 3} = \begin{bmatrix} F_{6 \times 3} \\ O_{6 \times 3} \end{bmatrix},$$

به طوری که $O_{6 \times 6}$ ، $O_{6 \times 15}$ و $O_{6 \times 3}$ ماتریس‌های صفر و همانطور که قبل از این بیان شد باقی بلوک‌ها ماتریس‌هایی تصادفی هستند که درایه‌های آنها مقادیری تصادفی هستند که حین پرتاب بر پرتابگر وارد می‌شوند.

۵. نتایج عددی

اکنون دستگاه (۹) را به وسیله روش توابع پایه شعاعی و روش ارائه شده در بخش ۳ حل می‌کنیم. در این بخش روش حل فازی دستگاه (۹) را با روش‌های شبکه و بدون شبکه مقایسه می‌کنیم. تمام نتایج عددی ارائه شده توسط نرم افزار متمتیکا محاسبه شده‌اند. همچنین در تمامی روش‌ها ماتریس‌های $P_{12 \times 12}$ ، $Q_{12 \times 15}$ و $R_{12 \times 3}$ توسط دستور $\text{RandomReal}[0,0.5]$ تولید شده‌اند.

جدول ۱: مقایسه نتایج با استفاده از روش‌های مختلف برای $t = 0.5$.

FMES	RBF-FD	RBF	FDM	X_i
-۱/۴۷۸۰	-۰/۷۶۵۰	-۰/۹۵۶۳	-۲/۰۰۰۱	X_1
۰/۱۳۲۵	۰/۱۱۹۸	۰/۱۵۶۷	۰/۳۹۹۹	X_2
۰/۰۰۰۰	-۰/۰۰۵۴	-۰/۰۰۶۵	-۰/۰۱۹۰	X_3
-۰/۱۰۲۷	-۰/۰۷۷۶	-۰/۱۰۱۱	-۰/۱۶۶۶	X_4
۰/۰۱۲۹	۰/۰۴۰۰	۰/۰۶۵۵	۰/۰۷۹۵	X_5
-۰/۰۳۷۴	-۰/۰۲۰۱	-۰/۰۳۴۰	-۰/۰۵۶۱	X_6
-۰/۰۱۹۵	-۰/۰۰۹۸	-۰/۰۱۲۱	-۰/۰۲۳۱	X_7
-۰/۱۵۴۱	-۰/۰۳۴۱	-۰/۰۶۰۲	-۰/۰۹۹۰	X_8
۰/۰۷۲۵	۰/۰۶۱۴	۰/۰۷۰۰	۰/۰۸۰۸	X_9
۰/۰۶۰۸	۰/۰۴۴۰	۰/۰۵۰۲	۰/۰۸۶۱	X_{10}
۰/۱۰۲۵	۰/۰۳۸۴	۰/۰۴۰۷	۰/۰۹۰۲	X_{11}
۰/۰۳۶۵	۰/۰۱۸۹	۰/۰۲۵۵	۰/۰۳۵۷	X_{12}

همچنین تابع پایه شعاعی را چند مکعبی و برای هر $i = 1, \dots, n$ با پارامتر ثابت $\varepsilon_i = 1$ در نظر گرفته شده‌اند. همچنین بعد فضای توابع پایه شعاعی در همه‌ی الگوریتم‌ها برابر ۶ فرض شده است. جدول ۱ نتایج روش پیشنهادی FMES را در مقایسه با روش‌های تفاضل محدود (FDM)، روش توابع پایه شعاعی و رویکرد تعمیم یافته تابع پایه شعاعی - تفاضل متناهی (RBF-FD) نمایش می‌دهد.

مراجع

- [1] D. Broomhead, D. Lowe, Multivariable Functional interpolation and adaptive networks, *Complex System* 2 (1988), 321–355.
- [2] A.A.A. Elsayed, N. Ahmad, G. Malkawi, On the solution of fully fuzzy Sylvester matrix equation with trapezoidal fuzzy numbers, *Journal of Computational and Applied Mathematics*, 39 (2020) 278.
- [3] S. Garg, P. Mohit, Meshfree Methods: A comprehensive review of applications, *International Journal of Computational Methods*, 15 (2018) 1830001.
- [4] P. Somoia, C.-E. Moldoveanu, Numerical research on the stability of launching devices during firing, *Defence Technology*, 9 (2013) 242–248.
- [5] P. Şomoia, F. Moraru, D. Safta, C.-E. Moldoveanu, A mathematical model for the motion of a rocket-launching device system on a heavy vehicle, *WSEAS Transactions on Applied and Theoretical Mechanics*, 2 (2007) 95–101.

سعید سرآبادان

دانشکده و پژوهشکده علوم پایه، دانشگاه جامع امام حسین (ع)، تهران، ایران.
آدرس ایمیل: sarabadan@ihu.ac.ir

محمد مقدس

دانشکده و پژوهشکده علوم پایه، دانشگاه جامع امام حسین (ع)، تهران، ایران.
آدرس ایمیل: moh.moghaddas.sci@iauctb.ac.ir

علی آخوندزاده

دانشکده و پژوهشکده علوم پایه، دانشگاه جامع امام حسین (ع)، تهران، ایران.
آدرس ایمیل: ali.akhondzadeh75@gmail.com

روش هم مکانی فیوناتچی برای تقریب جواب دستگاه معادلات دیفرانسیل جزئی کسری دو بعدی

شاهد مشهودی، اسمعیل بابلیان

چکیده. در این مقاله، یک روش هم مکانی بر اساس چند جمله‌ای‌های فیوناتچی برای حل دستگاه معادلات دیفرانسیل جزئی کسری (FPDEs) ارائه شده است. در این روش، تابع مجهول بر حسب چند جمله‌ای‌های فیوناتچی بسط داده می‌شود و سپس با استفاده از روش هم مکانی، یک دستگاه معادلات دیفرانسیل جزئی کسری به مجموعه‌ای از معادلات جبری خطی تبدیل می‌شود. در آخر، با حل دستگاه حاصل، جواب تقریبی به دست می‌آید. همچنین کارایی روش پیشنهادی در یک مثال عددی بررسی شده است.

۱. پیش‌گفتار

حساب کسری از شاخه‌های ریاضیات نظری است که در زمینه‌های مختلف علوم و مهندسی نقش مهمی دارد (مثلاً منبع [۱]). به طور کلی نمی‌توان به راحتی جواب‌های تحلیلی FPDEs را یافت، بنابراین به دست آوردن جواب‌های عددی این معادلات دارای اهمیت است. این مقاله می‌پردازد به معرفی روش عددی هم مکانی فیوناتچی مبتنی بر روابط ماتریسی برای حل دستگاه FPDEs دو بعدی زیر

$$\begin{cases} {}^c D_x^\alpha u + u_x + f_1(u, v) = g_1(x, t) \\ {}^c D_t^\beta v + v_t + f_2(u, v) = g_2(x, t) \end{cases}; \quad (x, t) \in [0, 1] \times [0, 1], \quad 1 < \alpha, \beta \leq 2. \quad (1)$$

با شرایط اولیه $u(0, t) = u(x, 0) = v(0, t) = v(x, 0) = 0$ که در آن ${}^c D_x^\alpha u$ و ${}^c D_t^\beta v$ مشتق‌های کپوتو هستند. روش هم مکانی فیوناتچی مبتنی بر روابط ماتریسی، برای حل عددی انواع معادلات دیفرانسیل به کار می‌رود (مثلاً منبع [۲]).

۲. روش هم مکانی فیوناتچی مبتنی بر روابط ماتریسی

در این بخش، پس از ارائه برخی روابط ماتریسی اساسی برای تبدیل دستگاه (۱) به معادلات ماتریسی، روش هم مکانی را به کار می‌بریم. بدین منظور، ابتدا جواب تقریبی معادله را بر حسب سری فیوناتچی برش یافته به صورت زیر بسط می‌دهیم

$$u_N(x, t) = \sum_{m=1}^N \sum_{n=1}^N a_{mn} F_m(x) F_n(t), \quad v_N(x, t) = \sum_{m=1}^N \sum_{n=1}^N b_{mn} F_m(x) F_n(t), \quad (2)$$

که در آن a_{mn} و b_{mn} به ازای $m, n = 1, \dots, N$ ضرایب فیوناتچی مجهول و F_m و F_n به ازای $m, n = 1, \dots, N$ چند جمله‌ای‌های فیوناتچی هستند که به صورت زیر تعریف می‌شوند

$$F_n(x) = \sum_{j=0}^{\lfloor \frac{(n-1)}{2} \rfloor} \binom{n-j-1}{j} x^{n-2j-1}, \quad [(n-1)/2] = \begin{cases} \frac{n-1}{2}, n \text{ زوج} \\ \frac{n-1}{2}, n \text{ فرد} \end{cases}$$

واژه‌های کلیدی: روش هم مکانی، چند جمله‌ای‌های فیوناتچی، دستگاه معادلات دیفرانسیل جزئی کسری (FPDEs).
طبقه‌بندی موضوعی [۲۰۱۰]: 35R11, 65L60, 11B39

برای تعیین ضرایب فیوناتچی مجهول، می‌توان جواب تقریبی (۲) را به شکل ماتریسی زیر نوشت

$$u(x, t) = X(x)C^T \bar{X}(t)\bar{C}^T A, \quad v(x, t) = X(x)C^T \bar{X}(t)\bar{C}^T B, \quad (3)$$

که در آن $B = [B_1 \ B_2 \ \dots \ B_N]^T_{1 \times N^2}$ و $A = [A_1 \ A_2 \ \dots \ A_N]^T_{1 \times N^2}$ به طوری که

$$A_i = [a_{i1} \ a_{i2} \ \dots \ a_{iN}]^T_{1 \times N}, \quad B_i = [b_{i1} \ b_{i2} \ \dots \ b_{iN}]^T_{1 \times N}, \quad i = 1, 2, \dots, N,$$

$$\bar{C}^T = \begin{bmatrix} C^T & 0 & \dots & 0 \\ 0 & C^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & C^T \end{bmatrix}, \quad \bar{X}(t) = \begin{bmatrix} X(t) & 0 & \dots & 0 \\ 0 & X(t) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & X(t) \end{bmatrix}, \quad X(x) = [1 \ x \ \dots \ x^{N-1}].$$

در صورتی که N زوج باشد، برای C داریم

$$C = \begin{bmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & 0 & 0 & \dots & 0 \\ 0 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & 0 & 0 & \dots & 0 \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} & 0 & \begin{pmatrix} 2 \\ 0 \end{pmatrix} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \begin{pmatrix} (n-2)/2 \\ (n-2)/2 \end{pmatrix} & 0 & \begin{pmatrix} n/2 \\ (n-4)/2 \end{pmatrix} & 0 & \dots & 0 \\ 0 & \begin{pmatrix} n/2 \\ (n-2)/2 \end{pmatrix} & 0 & \begin{pmatrix} (n+2)/2 \\ (n-4)/2 \end{pmatrix} & \dots & \begin{pmatrix} n-1 \\ 0 \end{pmatrix} \end{bmatrix},$$

و اگر N فرد باشد داریم

$$C = \begin{bmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & 0 & 0 & \dots & 0 \\ 0 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & 0 & 0 & \dots & 0 \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} & 0 & \begin{pmatrix} 2 \\ 0 \end{pmatrix} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \begin{pmatrix} (n-1)/2 \\ (n-3)/2 \end{pmatrix} & 0 & \begin{pmatrix} (n+1)/2 \\ (n-5)/2 \end{pmatrix} & \dots & 0 \\ \begin{pmatrix} (n-1)/2 \\ (n-1)/2 \end{pmatrix} & 0 & \begin{pmatrix} (n+1)/2 \\ (n-3)/2 \end{pmatrix} & 0 & \dots & \begin{pmatrix} n-1 \\ 0 \end{pmatrix} \end{bmatrix},$$

C را ماتریس مشخصه روابط ماتریسی می‌نامیم (منبع [۲]). از طرفی ما می‌توانیم شکل ماتریسی مشتقات جزئی را برای هر متغیر مستقل توابع $u(x, t)$ و $v(x, t)$ به صورت زیر تعریف کنیم

$$u_x(x, t) = X(x)DC^T \bar{X}(t)\bar{C}^T A, \quad v_x(x, t) = X(x)DC^T \bar{X}(t)\bar{C}^T B, \quad (۴)$$

$$u_t(x, t) = X(x)C^T \bar{X}(t)\bar{D}\bar{C}^T A, \quad v_t(x, t) = X(x)C^T \bar{X}(t)\bar{D}\bar{C}^T B, \quad (۵)$$

که در آن

$$\bar{D} = \begin{bmatrix} D & \circ & \dots & \circ \\ \circ & D & \dots & \circ \\ \vdots & \vdots & \ddots & \vdots \\ \circ & \circ & \dots & D \end{bmatrix}, \quad D = \begin{bmatrix} \circ & ۱ & \circ & \dots & \circ \\ \circ & \circ & ۲ & \dots & \circ \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \circ & \circ & \circ & \dots & N-۱ \\ \circ & \circ & \circ & \dots & \circ \end{bmatrix}.$$

در نهایت می‌توانیم $\frac{\partial^\alpha u(x, t)}{\partial x^\alpha}$ یعنی مشتق کسری تابع $u(x, t)$ را نسبت به متغیر x به شکل زیر بنویسیم

$$\left[\frac{\partial^\alpha u(x, t)}{\partial x^\alpha} \right] = M(x)C^T \bar{X}(t)\bar{C}^T A, \quad (۶)$$

که در آن

$$M(x) = \frac{\partial^\alpha X(x)}{\partial x^\alpha} = \begin{bmatrix} \circ & D^\alpha x & D^\alpha x^2 & \dots & D^\alpha x^{N-1} \end{bmatrix}_{1 \times N},$$

$$D^\alpha x^i = \frac{\Gamma(i+1)}{\Gamma(i+1-\alpha)} x^{i-\alpha}, \quad i = 1, \dots, N-1.$$

همچنین می‌توانیم $\frac{\partial^\beta v(x, t)}{\partial t^\beta}$ یعنی مشتق کسری تابع $v(x, t)$ نسبت به متغیر t را به شکل زیر بنویسیم

$$\left[\frac{\partial^\beta v(x, t)}{\partial t^\beta} \right] = N(x)C^T \bar{X}(t)\bar{C}^T B, \quad (۷)$$

که در آن

$$N(t) = \frac{\partial^\beta X(x)}{\partial t^\beta} = \begin{bmatrix} \circ & D^\beta t & D^\beta t^2 & \dots & D^\beta t^{N-1} \end{bmatrix}_{1 \times N}$$

به طوری که $D^\beta t^i = \frac{\Gamma(i+1)}{\Gamma(i+1-\beta)} t^{i-\beta}$, $i = 1, \dots, N-1$. با به کارگیری روابط (۴)، (۵)، (۶) و (۷) در معادله (۱)، می‌توانیم شکل بسته ماتریسی زیر را داشته باشیم

$$\begin{aligned} M(x)C^T \bar{X}(t)\bar{C}^T A + X(x)DC^T \bar{X}(t)\bar{C}^T A + f_1(F(x)\bar{F}(t)A, F(x)\bar{F}(t)B) &= g_1(x, t) \\ N(x)C^T \bar{X}(t)\bar{C}^T B + X(x)DC^T \bar{X}(t)\bar{C}^T B + f_2(F(x)\bar{F}(t)A, F(x)\bar{F}(t)B) &= g_2(x, t) \end{aligned} \quad (۸)$$

که در آن $\bar{F}(t) = \bar{X}(t)\bar{C}^T$. شکل ماتریسی شرایط اولیه دستگاه معادلات (۱) با استفاده از رابطه (۳) به شرح زیر است

$$\begin{aligned} u(x, \circ) &= X(x)C^T \bar{X}(\circ)\bar{C}^T A = \circ, & v(x, \circ) &= X(x)C^T \bar{X}(\circ)\bar{C}^T B = \circ \\ u(\circ, t) &= X(\circ)C^T \bar{X}(t)\bar{C}^T A = \circ, & v(\circ, t) &= X(\circ)C^T \bar{X}(t)\bar{C}^T B = \circ. \end{aligned} \quad (۹)$$

با مختصرنویسی معادلات ماتریسی (۸) و جایگذاری نقاط هم‌مکانی در آن، دستگاه معادلات ماتریسی زیر را خواهیم داشت

$$\begin{cases} W_1(x_i, t_j) [A] + f_1(x_i, t_j) [A, B] = g_1(x_i, t_j) \\ W_2(x_i, t_j) [B] + f_2(x_i, t_j) [A, B] = g_2(x_i, t_j) \end{cases}, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, N,$$

$$x_i = \frac{l}{N-1}(i-1), \quad i = 1, 2, \dots, N,$$

$$t_j = \frac{\tau}{N-1}(j-1), \quad j = 1, 2, \dots, N,$$

که یک دستگاه معادلات ماتریسی است. بنابراین معادله ماتریسی اساسی $W[A, B] = G$ به دست می‌آید که با لحاظ شرایط اولیه همگن رابطه (۹) و حل دستگاه حاصل، ضرایب مجهول فیوناتچی به صورت $[A, B] = W^{-1}G$ تعیین شده و در نهایت جواب تقریبی از رابطه (۳) به دست می‌آید. تحلیل خطا و همگرایی، مشابه منابع [۲، ۳] است که برای ایجاز، از تکرار آن پرهیز شده است.

Numerical Analysis

۳. نتایج عددی

به عنوان یک مثال برای روش هم‌مکانی فیوناتچی، دستگاه معادلات دیفرانسیل کسری زیر را در نظر بگیرید (منبع [۳]).

$$\begin{cases} D_x^\alpha u(x, t) + u_x + 2v - u = \lambda x^{\frac{\tau}{2}} \sinh(t)/\sqrt{\pi} + 3x^2 \sinh(t) + 2t^3 \sinh(x) - x^3 \sinh(t) \\ D_t^\beta v(x, t) + v_t + 2u + v = \lambda x^{\frac{\tau}{2}} \sinh(x)/\sqrt{\pi} + 3t^2 \sinh(x) + 2x^3 \sinh(t) + t^3 \sinh(x) \end{cases}, \quad (10)$$

با شرایط اولیه $u(0, t) = u(x, 0) = v(0, t) = v(x, 0) = 0$ با شرایط $\alpha = \beta = \frac{3}{4}$ ازای $n = m = 8$ و 5 در $t = 0/3, 0/6, 0/9$ آورده شده است. در این جدول، $Error_u$ نشان دهنده خطای روش برای $u(x, t) = x^3 \sinh(t)$ ، $v(x, t) = t^3 \sinh(x)$ خطای دقیق روش پیشنهادی برای حل دستگاه معادلات (۱۰) به ازای $n = m = 8$ و 5 و $Error_v$ نشان دهنده خطای روش برای تابع $v(x, t)$ است. نتایج نشان می‌دهند روش از دقت خوبی برخوردار است.

t	x	Error _u		Error _v	
		m = n = 5	m = n = 8	m = n = 5	m = n = 8
0/3	0/2	7/05 × 10 ⁻⁸	3/66 × 10 ⁻¹³	3/30 × 10 ⁻⁷	7/31 × 10 ⁻¹²
	0/5	1/10 × 10 ⁻⁶	5/85 × 10 ⁻¹²	9/92 × 10 ⁻¹⁶	2/60 × 10 ⁻¹²
	0/8	4/51 × 10 ⁻⁶	2/40 × 10 ⁻¹¹	3/45 × 10 ⁻⁷	8/44 × 10 ⁻¹²
0/6	0/2	9/61 × 10 ⁻⁸	5/23 × 10 ⁻¹³	2/64 × 10 ⁻⁶	5/85 × 10 ⁻¹¹
	0/5	1/50 × 10 ⁻⁶	8/24 × 10 ⁻¹²	8/60 × 10 ⁻¹⁶	2/09 × 10 ⁻¹¹
	0/8	6/15 × 10 ⁻⁶	3/37 × 10 ⁻¹¹	2/76 × 10 ⁻⁶	6/76 × 10 ⁻¹¹
0/9	0/2	2/75 × 10 ⁻⁷	5/88 × 10 ⁻¹²	8/91 × 10 ⁻⁶	1/97 × 10 ⁻¹⁰
	0/5	4/29 × 10 ⁻⁶	9/19 × 10 ⁻¹¹	8/32 × 10 ⁻¹⁶	7/06 × 10 ⁻¹¹
	0/8	1/76 × 10 ⁻⁵	3/76 × 10 ⁻¹⁰	9/33 × 10 ⁻⁶	2/28 × 10 ⁻¹⁰

جدول ۱. خطای دقیق روش پیشنهادی برای دستگاه معادلات دیفرانسیل کسری (۱۰) به ازای پارامترهای مورد نظر.

۴. نتیجه‌گیری

در این مقاله یک روش عددی کارآمد و دقیق بر اساس روش چند جمله‌ای فیوناتچی برای به دست آوردن جواب‌های عددی با دقت بالا برای دستگاه‌های دو بعدی FPDEs پیشنهاد شد. برای بررسی کارایی روش یک مثال مطرح شد و سپس به کمک نرم‌افزار MATLAB،

جوابهای عددی محاسبه شدند. نتایج حاکی از آن است که روش از دقت خوبی برخوردار است.

مراجع

- [1] O.A. Arqub, A. EI-Ajor, S. Momani, Constructing and predicting solitary pattern solutions for non-linear time-fractional dispersive PDEs, *Journal of Computational Physics*, 293 (2015) 385–389.
- [2] A.Kurt, S.Yalcinbas, Numerical solutions and error estimations for the space fractional diffusion equation with variable coefficient via Fibonacci collocation method, *Springer Plus*, 5 (2016) no. 1375.
- [3] F. Zhao, Q. Huang, J. Xie, Y.Li, L. Ma, J. Wang, Chebyshev polynomials approach for numerically solving system of two-dimensional fractional PDEs and convergence analysis, *Applied Mathematics and Computation*, 313 (2017) 321–330.

شاهد مشهودی

گروه ریاضی، دانشکده علوم پایه، دانشگاه آزاد اسلامی واحد رشت، رشت، ایران

آدرس ایمیل: shahed.mashhoodi@gmail.com

اسمعیل بابلیان

گروه کامپیوتر، دانشکده علوم ریاضی و کامپیوتر، دانشگاه خوارزمی، تهران، ایران.

آدرس ایمیل: babolian@khu.ac.ir

یک روش مبتنی بر هسته‌های بازگشتی در حل نامساوی‌های تغییراتی پدید آمده از اختیارهای معامله آمریکایی

مجتبی مرادی پور، ناصر عباسی

چکیده. در این مقاله مساله قیمت‌گذاری اختیار معامله آمریکایی را به صورت یک نامساوی تغییراتی نوشته و به حل عددی آن می‌پردازیم. ابتدا نامساوی تغییراتی مورد نظر را در بعد زمان گسسته سازی کرده و به دنباله‌ای از نامساوی‌های تغییراتی در بعد مکان دست می‌یابیم، سپس برای حل نامساوی‌های تغییراتی در بعد مکان، هسته‌های بازگشتی را به همراه روش هم‌مکانی به کار می‌بریم. استفاده از هسته‌های بازگشتی دارای این مزیت مهم است که شرایط مرزی مساله به طور خودکار برقرار می‌شود. روش به کار رفته شده مستقل از شبکه بندی نقاط هم‌مکانی بوده و منجر به تولید نتایج دقیقی می‌شود. این روش کارایی بالایی در حل نامساوی‌های تغییراتی دارد و همچنین قابل تعمیم به مسایل با ابعاد بالاتر خواهد بود.

۱. پیش‌گفتار

مساله یافتن ارزش قرارداد اختیار معامله آمریکایی یک معادله دیفرانسیل جزئی با شرایط مرزی آزاد است که به صورت نامساوی تغییراتی زیر نیز فرمول‌بندی می‌شود.

$$\begin{cases} u_t - u_{xx} \geq 0, & -\infty < x < \infty, \quad 0 \leq t \leq t_{max}, \\ u \geq g, \\ (u_t - u_{xx})(u - g) = 0, \\ u(x, 0) = g(x, 0), \\ \lim_{x \rightarrow \pm\infty} u(x, t) = g(x, t), \end{cases} \quad (1)$$

که در آن

$$g(x, t) = e^{\frac{t}{\tau}(q+1)^+} \max\{e^{\frac{x}{\tau}(q-1)} - e^{\frac{x}{\tau}(q+1)}, 0\}$$

بیانگر شرط اولیه مساله و پارامتر q عددی ثابت است. جزئیات دقیق‌تر را در [۳]، [۴] و [۶] ببینید. همچنین برای یافتن شرح جامع و کامل نامساوی‌های تغییراتی خواننده را به [۲] ارجاع می‌دهیم. در [۳] روش عناصر طیفی و در [۴] یک روش بر اساس توابع پایه‌ای شعاعی برای حل (۱) به کار رفته شده‌اند. در این مقاله از روش هم‌مکانی مبتنی بر هسته‌های بازگشتی معرفی شده توسط [۵] برای حل مساله (۱) بهره می‌گیریم.

۲. گسسته سازی و حل نامساوی تغییراتی با استفاده از هسته‌های بازگشتی

ابتدا با استفاده از روش تفاضل متناهی پیشرو اوایلر مساله (۱) را به صورت زیر در بعد زمان گسسته‌سازی می‌کنیم

$$u_t(x, t_n) = \frac{u(x, t_n) - u(x, t_{n-1})}{\delta}, \quad (2)$$

که در آن δ طول گام زمانی است. با نمادگذاری $u^n = u(x, t_n)$ و جایگذاری (۲) در معادله (۱) به دنباله زیر از نامساوی‌های تغییراتی در بعد مکان دست پیدا می‌کنیم:

$$\begin{cases} u^n - \delta u^{n''} - u^{n-1} \geq 0, \\ u^n \geq g^n, \\ (u^n - \delta u^{n''} - u^{n-1})(u^n - g^n) = 0, \\ u^\circ = g(x, 0), \\ \lim_{x \rightarrow \pm\infty} u^n = g(x, t_n). \end{cases} \quad (3)$$

نامساوی‌های تغییراتی موجود در (۳) همگی بر بازه نامتناهی $(-\infty, +\infty)$ تعریف شده‌اند و همانند مساله اصلی (۱) در هر دومرز به صفر میل می‌کنند.

حال دامنه نامتناهی مساله نیم‌گسسته (۳) را برش زده و آن را با یک دامنه متناهی چون بازه $[a, b]$ جایگزین می‌کنیم. فرض کنیم $a < x_1 < x_2 < \dots < x_m < b$ شبکه‌ای از نقاط دلخواه در بازه $[a, b]$ باشد. برای ساختن یک هسته مثبت معین که شرایط مرزی صفر را برآورده کند به صورت زیر عمل می‌کنیم. تعریف می‌کنیم

$$\begin{aligned} K_1(x, y) &= K(x, y) - \frac{K(x, a)K(y, a)}{K(a, a)}, \\ K_2(x, y) &= K_1(x, y) - \frac{K_1(x, b)K_1(y, b)}{K_1(b, b)}. \end{aligned} \quad (4)$$

که در آن $K(x, y)$ می‌تواند هر کدام از هسته‌های Gaussian (GA), Inverse Quadratic (IQ), Multiquadric (MQ), Inverse Multiquadric (IMQ) و Thin Plate Splines باشد. تابع $K_2(x, y)$ یک هسته مثبت معین با این خاصیت مهم است که شرایط مرزی همگن را به صورت خودکار برآورده می‌کنند. در [۱] این هسته‌ها به همراه روش هم‌مکانی برای حل معادلات دیفرانسیل تاخیری به کار برده شده‌اند. حال توابع پایه‌ای روش هم‌مکانی را به صورت زیر تعریف می‌کنیم

$$\psi_j(x) = K_2(x, x_j), \quad j = 1, 2, \dots, m.$$

همگی این توابع در نقاط مرزی $x = a$ و $x = b$ برابر صفر می‌باشند. جواب معادله نیم‌گسسته (۳) را به صورت ترکیب خطی زیر از توابع پایه‌ای می‌نویسیم:

$$u^n(x) \simeq \tilde{u}^n(x) = \sum_{j=1}^m \tilde{u}^n(x_j) \phi_j(x) \quad (5)$$

با جایگذاری (۵) در معادله نیم‌گسسته (۳) و مقداردهی آن در نقاط هم‌مکانی به یک دستگاه از مسائل مکمل خطی به صورت:

$$\begin{cases} \mathbf{u}^n - \mathbf{g}^n \geq 0, \\ \mathbf{M}\mathbf{u}^n - \mathbf{B}\mathbf{u}^{n-1} \geq 0, \\ (\mathbf{u}^n - \mathbf{g}^n)^\top (\mathbf{M}\mathbf{u}^n - \mathbf{B}\mathbf{u}^{n-1}) = 0. \end{cases} \quad (6)$$

بباید $\mathbf{u}^n \in \mathbb{R}^m$ را به طوری که

دست پیدا می‌کنیم. مسائل مکمل خطی (۶) را همان‌طور که در [۳] شرح داده شده است با استفاده از روش فوق تخفیف متوالی تصویر شده (Projected Successive Over Relaxation) حل می‌کنیم تا جواب نامساوی‌های تغییراتی در هر سطر زمانی به دست آید. محاسبات عملی نشان می‌دهد روش ارائه شده بسیار کارا بوده و نتایج دقیقی تولید می‌کند.

۳. نتایج عددی

در این بخش مساله یافتن قیمت یک اختیار معامله آمریکایی با نرخ بهره $r = 0.05$ ، تغییرپذیری $\sigma = 0.25$ ، قیمت اعمال $K = 10$ و زمان انقضای $T = 0.5$ را مورد بررسی قرار می‌دهیم. پارامترهای q و t_{max} در معادله (۱) به صورت زیر می‌باشند:

$$q = \frac{\gamma r}{\sigma^2} = 1/6, \quad t_{max} = \frac{\sigma^2}{\gamma} T = 0.01562.$$

مساله را در بازه زمانی $[0, t_{max}]$ توسط $n = 100$ نقطه هم‌فاصله گسسته‌سازی می‌کنیم. سپس دامنه نامتناهی مساله نیم‌گسسته را با دامنه محاسباتی $[a, b] = [-2, 2]$ جایگزین کرده، تعداد $m = 50$ نقطه هم‌مکانی هم‌فاصله را در این بازه در نظر می‌گیریم. از توابع پایه‌ای شعاعی (MQ) استفاده کرده، هسته بازگشتی متناظر را طبق (۴) تعریف می‌کنیم. با اعمال روش هم‌مکانی در نقاط ذکر شده، مساله به صورت کامل گسسته‌سازی شده و یک دنباله از مسائل مکمل خطی به فرم (۶) حاصل می‌شود. مسائل مکمل خطی پدید آمده را برای هر سطر زمانی با استفاده از روش PSOR حل کرده و جواب را در هر سطر زمانی به دست می‌آوریم. در جدول ۱ جواب‌های به دست آمده توسط روش ارائه شده در این مقاله را با جواب‌های به دست آمده از روش هم‌مکانی ارائه شده توسط [۴] مقایسه کرده‌ایم. جواب‌های دقیق ارائه شده در جدول ۱ توسط روش دوجمله‌ای و با تعداد ۱۰۰۰۰ نقطه به دست آمده‌اند. در مرجع [۴] از تعداد $n = 100$ مرحله زمانی به همراه $m = 120$ نقطه هم‌مکانی استفاده شده است. نتایج به دست آمده نشان می‌دهند روش ارائه شده با وجود مرتبه پیچیدگی کمتر، منجر به تولید جواب‌های با دقت بهتر می‌شود.

۱۳	۱۲	۱۱	۱۰	۹	۸	۷	قیمت نقدی دارایی پایه
۰/۰۴۴۵	۰/۱۱۶۳	۰/۲۷۸۵	۰/۶۰۲۱	۱/۱۶۴۳	۲/۰۰۱۸	۳/۰۰۰۲	جواب به دست آمده از روش هسته‌های بازگشتی
۰/۰۴۴۹	۰/۱۱۶۹	۰/۲۷۹۱	۰/۶۰۲۷	۱/۱۶۴۹	۲/۰۰۲۲	۳/۰۰۰۷	جواب‌های ارائه شده توسط مرجع [۴]
۰/۰۴۴۳	۰/۱۱۵۹	۰/۲۷۸۱	۰/۶۰۲۱	۱/۱۶۴۱	۲/۰۰۱۴	۳	جواب دقیق

جدول ۱: مقایسه جواب‌های به دست آمده از روش هسته‌های بازگشتی با روش ارائه شده در مرجع [۴]. جواب‌ها برای مساله (۱) در زمان $t = t_{max}$ و به ازای مقادیر مختلف قیمت نقدی دارایی پایه به دست آمده‌اند.

مراجع

- [1] H. Azari, M. Moradipour, Using kernel-based collocation methods to solve a delay partial differential equation with application to finance, *International journal of computing science and mathematics*, 10 (2019) 105–114.
- [2] D. Kinderlehrer, G. Stampacchia, *An introduction to variational inequalities and their applications*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2000.
- [3] M. Moradipour, S. Yousefi, Using spectral element method to solve variational inequalities with applications in finance, *Chaos, Solitons and Fractals*, 81 (2015) 208–217.
- [4] M. Moradipour, S. Yousefi, Using a meshless kernel-based method to solve the Black–Scholes variational inequality of American options, *Computational and Applied Mathematics*, 37 (2018) 627–639.
- [5] M. Mouattamid, R. Schaback, *Recursive kernels, Analysis in Theory and Applications*, 25 (2009) 301–316.

- [6] R. U. Seydel, *Tools for computational finance*, Universitext Springer-Verlag, Berlin, fourth edition, 2009.

مجتبی مرادی‌پور

دانشکده علوم پایه، گروه ریاضی، دانشگاه لرستان، خرم‌آباد، ایران.

آدرس ایمیل: moradipour.mo@lu.ac.ir

ناصر عباسی

دانشکده علوم پایه، گروه ریاضی، دانشگاه لرستان، خرم‌آباد، ایران.

آدرس ایمیل: abasi.n@lu.ac.ir



روش ماتریس عملگر ژاکوبی انتقال یافته نوین برای معادلات دیفرانسیل تأخیری با مشتق از مرتبه کسری

حمیدرضا خداپنده لو، الیاس شیوانیان

چکیده. این پژوهش، معادلات دیفرانسیل تأخیری با مشتقات کسری از مرتبه متغیر را مورد بررسی قرار می‌دهد. در این کار، یک تکنیک ماتریس عملگر ژاکوبی انتقال یافته نوین برای حل یک دسته معادلات دیفرانسیل تأخیری چند عبارتی با مشتقات کسری از مرتبه متغیر معرفی می‌شود که با کاهش مساله اصلی به یک دستگاه معادلات جبری، آن دستگاه را به روش عددی حل می‌نماییم. تکنیک پیشنهادی را بطور موفقیت آمیز برای مساله فوق الذکر، توسعه داده و آزمایشات عددی جامعی را برای نشان دادن اثربخشی، عمومیت، دقت الگوی پیشنهادی و انعطاف این روش، ارائه کرده‌ایم. تکنیک حاضر را با روش‌های موجود دیگر از قبیل روش آدامز کسری ($NSJOM$)، روش پیشگو-اصلاحگر جدید ($NPCM$)، الگوریتم آدامز-بشفورت-مولتون و روش پیشگو اصلاحگر $L_1 - PCM$ مقایسه کرده و از مقایسه نتایج عددی این روش‌ها، همچنین مقایسه روش جاری با جواب دقیق، اثربخشی و کارآمدی این روش را نمایان ساخته‌ایم. توجه داریم که پیاده سازی روش مذکور بسیار آسان بوده و این تکنیک می‌تواند به عنوان یک تعمیمی از بسیاری از الگوهای عددی در نظر گرفته شود.

۱. پیش‌گفتار

تجزیه، تحلیل و کاربردهای حسابان کسری یک موضوع فعال و دارای رشد سریع برای تحقیق در سه دهه اخیر محسوب می‌شود. در حال حاضر به دلیل کاربردهای گسترده آن در رشته‌های مختلف علمی مانند فیزیک، تغییرات منظم در ترمودینامیک، داده‌های تجربی و غیره به ابزار مهمی تبدیل شده است [۶، ۷].

گنجانیدن تأخیر در معادلات دیفرانسیل مرتبه کسری باعث ایجاد دیدگاه‌های جدیدی مخصوصاً در زمینه مهندسی زیستی می‌شود زیرا در مهندسی زیستی، درک پویایی‌هایی که در بافت‌های بیولوژیکی رخ می‌دهد، توسط مشتقات کسری بهبود می‌یابد [۱، ۸]. در ریاضیات، معادلات دیفرانسیل تأخیری (DDEs)، نوعی معادله دیفرانسیل هستند که در آن مشتق تابع مجهول در زمان معینی بر حسب مقادیر تابع در زمان‌های قبلی داده می‌شود.

معادلات دیفرانسیل تأخیری کسری با نوع معمولی آن از این نظر که مشتق در هر زمان به جواب (و در مورد معادلات خنثی روی مشتق) در زمان‌های قبلی بستگی دارد، متفاوت است. بسیاری از رویدادها در دنیای طبیعی را می‌توان به شکل معادلات دیفرانسیل تأخیری از مرتبه کسری مدل سازی کرد. معادلات دیفرانسیل با مرتبه کسری با مدل سازی مسائل مختلف، کاربردهای زیادی در رشته‌های مختلف علوم مانند اقتصاد، الکتروپدینامیک، زیست‌شناسی، کنترل و ... دارد. در سال‌های اخیر، محققانی همچون مارگادو^۱ و

واژه‌های کلیدی: معادلات دیفرانسیل تأخیری با مشتقات کسری از مرتبه متغیر، چندجمله‌ای‌های ژاکوبی، تکنیک ماتریس عملگر. طبقه‌بندی موضوعی [۲۰۱۰]: (۱ تا ۳ مورد) 13D45, 39B42.

¹Margado

دفتردار- گججی^۲، عباس زاده^۳ و دهقان^۴، خداپنده لو^۵ و همکاران و... تحقیقات متعددی در رابطه با معادلات دیفرانسیل تأخیری با مشتق کسری به انجام رسانیده‌اند که برای جزئیات بیشتر، علاقمندان می‌توانند به منابع [۱، ۴، ۷، ۹، ۱۰] مراجعه نمایند. علاوه بر این، از روش‌های طیفی که اساساً به مجموعه‌ای از چندجمله‌ای متعامد بستگی دارد، برای حل معادلات دیفرانسیل مرتبه کسری استفاده می‌شود. یکی از معروف‌ترین آنها چند جمله‌ای‌های ژاکوبی کلاسیک است که به صورت زیر نشان داده شده‌اند:

$$P_n^{(\alpha, \beta)}(x) \quad (n \geq 0, \alpha > -1, \beta > -1).$$

این چند جمله‌ای‌ها به‌طور گسترده در تحلیل‌های ریاضی و کاربردهای عملی مورد استفاده قرار گرفته‌اند، زیرا مزیت‌های بدست آوردن جواب‌های عددی در پارامترهای α و β را دارند. بنابراین، انجام یک مطالعه سیستماتیک از طریق چندجمله‌ای ژاکوبی با شاخص‌های کلی α و β مفید خواهد بود و این امر به وضوح یکی از اهداف و بدیع بودن بازه زمانی $t \in [0, T]$ در نظر گرفته می‌شود [۳]. اکنون هدف این پژوهش، تعمیم چندجمله‌ای‌های متعامد در پایه جواب است. در واقع، ما یک ماتریس عملگر ژاکوبی انتقال یافته نوین را برای حل معادلات دیفرانسیل تأخیری چند عبارتی با مشتقات کسری از مرتبه متغیر بدست می‌آوریم که مساله آن بصورت زیر است:

$$\sum_{j=1}^n \alpha_j D^{\eta_j(t)} w(t) + \alpha_{n+1} w(t - \tau) = F(t, w(t), D^{\eta_1(t)} w(t), \dots, D^{\eta_n(t)} w(t), w(t - \tau)), \quad 0 \leq t \leq T, \quad (1)$$

$$w(t) = g(t), \quad t \in [-\tau, 0], \quad w(0) = w_0,$$

که $0 < T$ و $\alpha_{n+1} \neq 0$ و به ازای $\alpha_j \in \mathbb{R} (j = 1, 2, \dots, n+1)$ و $\eta_j(t) \in \mathbb{R} (j = 1, 2, \dots, n)$ مشتق کسری متغیر با تعریف کاپوتو می‌باشند.

۲. ساختار کلی روش NSJOM و مبانی و مقدمات

در اولین قسمت این بخش، برخی از ویژگی‌های اساسی و مهم نظریه حسابان کسری را بررسی می‌کنیم. سپس برخی از ویژگی‌های مهم چند جمله‌ای‌های ژاکوبی را که به ما در توسعه تکنیک پیشنهادی کمک می‌کند، یادآوری می‌نماییم.

تعریف ۱. مشتق کسری کاپوتو از مرتبه متغیر $\eta(t)$ برای $w(t) \in C^m[0, T]$ بصورت زیر می‌باشد [۴]:

$$D^{\eta(t)} w(t) = \frac{1}{\Gamma(1 - \eta(t))} \int_{0+}^t \frac{w'(\tau)}{(t - \tau)^{\eta(t)}} d\tau + \frac{w(0^+) - w(0^-)}{\Gamma(1 - \eta(t))} t^{-\eta(t)}. \quad (2)$$

در زمان اولیه و برای $0 < \eta(t) < 1$ ، خواهیم داشت:

$$D^{\eta(t)} w(t) = \frac{1}{\Gamma(1 - \eta(t))} \int_{0+}^t \frac{w'(\tau)}{(t - \tau)^{\eta(t)}} d\tau. \quad (3)$$

طبق معادله (۲)، خواهیم داشت:

$$D^{\eta(t)} C = 0. \quad (4)$$

که C عددی ثابت می‌باشد. از طرف دیگر داریم:

$$D^{\eta(t)} t^k = \begin{cases} 0, & \text{for } k = 0, \\ \frac{\Gamma(k+1)}{\Gamma(k+1 - \eta(t))} t^{k-\eta(t)} & \text{for } k = 1, 2, \dots \end{cases} \quad (5)$$

²Daftardar-Gejji

³Abbaszadeh

⁴Dehghan

⁵Khodabandehlo

تعریف ۲. $P_{T,i}^{(\alpha,\beta)}(t)$ ، i -مین مرتبه از چندجمله‌ایهای ژاکوبی انتقال یافته در بازه $[0, T]$ دارای فرم تحلیلی زیر می‌باشد [۳]:

$$P_{T,i}^{(\alpha,\beta)}(t) = \sum_{k=0}^i (-1)^{i-k} \frac{\Gamma(\alpha+i+1)\Gamma(\alpha+\beta+k+i+1)}{\Gamma(\alpha+\beta+i+1)\Gamma(\alpha+1+k)\Gamma(k+1)\Gamma(i-k+1)} T^k t^k, \quad (6)$$

$$= \sum_{k=0}^i \frac{\Gamma(\beta+i+1)\Gamma(\alpha+\beta+k+i+1)}{\Gamma(\alpha+\beta+i+1)\Gamma(\beta+1+k)\Gamma(k+1)\Gamma(i-k+1)} T^k (T-t)^k. \quad (7)$$

۱.۲. تقریب تابع بوسیله چند جمله‌ایهای ژاکوبی انتقال یافته

فرض کنیم که تابع $w(t)$ با ملاحظه تابع وزن $\omega_T^{(\alpha,\beta)}(t)$ در $[0, T]$ ، انتگرال پذیر مربعی^۶ باشد، پس می‌تواند به شکل عبارت زیر بسط پیدا کند [۳، ۵]:

$$w(t) = \sum_{i=0}^{\infty} a_i P_{T,i}^{(\alpha,\beta)}(t), \quad a_i = \frac{1}{h_{T,j}^{(\alpha,\beta)}} \int_0^T \omega_T^{(\alpha,\beta)} P_{T,i}^{(\alpha,\beta)}(t) w(t) dt, \quad i = 0, 1, \dots \quad (8)$$

بنابراین، ما می‌توانیم جواب تقریبی را با در نظر گرفتن $(N+1)$ جمله سری در معادله (۸)، تخمین بزنیم:

$$w(t) \simeq w_N(t) = \sum_{i=0}^N a_i P_{T,i}^{(\alpha,\beta)}(t) = A^T \Phi_{T,N}(t), \quad (9)$$

که $A = [a_0, a_1, \dots, a_N]^T$ و $\Phi_{T,N}(t) = [P_{T,0}^{(\alpha,\beta)}(t), P_{T,1}^{(\alpha,\beta)}(t), \dots, P_{T,N}^{(\alpha,\beta)}(t)]^T$ در اینجا فرض شده است که $S(t) = [1, t, t^2, t^3, \dots, t^N]^T$ با معادله (۹)، بردار $\Phi_{T,N}(t)$ بصورت $\Phi_{T,N}(t) = B_{(\alpha,\beta)} S(t)$ ارائه می‌شود که $B_{(\alpha,\beta)}$ یک ماتریس مربعی از مرتبه $(N+1) \times (N+1)$ می‌باشد و داریم:

$$b_{i+1,j+1} = \begin{cases} (-1)^{i-j} \frac{(\alpha+i)!(\alpha+\beta+j+i)!}{(\alpha+\beta+i)!(\alpha+j)!(j)!(i-j)! T^j}, & i \geq j, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

برای $0 \leq i, j \leq N$ از اینرو داریم:

$$S(t) = B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t). \quad (11)$$

۳. ماتریس عملگر ژاکوبی انتقال یافته (SJOM)

در این قسمت، ما (SJOM) مرتبه متغیر کسری را برای پشتیبانی از حل عددی معادله (۱) بررسی می‌کنیم. بنابراین، مساله را به سیستم جبری از معادلاتی که به صورت عددی در نقاط کالوکیشن^۷ حل می‌شوند تبدیل می‌کنیم. در ابتدا، $(i = D^{\eta_i(t)} \Phi_{T,N}(t))$ می‌تواند بصورت زیر نتیجه شود: بخاطر داریم که $\Phi_{T,N}(t) = B_{(\alpha,\beta)} S(t)$ ، پس:

$$D^{\eta_i(t)} \Phi_{T,N}(t) = D^{\eta_i(t)} (B_{(\alpha,\beta)} S(t)) = B_{(\alpha,\beta)} D^{\eta_i(t)} [1, t, \dots, t^N]^T, \quad i = 1, 2, \dots, n. \quad (12)$$

^۶Square integrable

^۷Collocation points

که با ترکیب معادلات (۵) و (۱۲) داریم:

$$\begin{aligned}
 D^{\eta_i(t)} \Phi_{T,N}(t) &= B_{(\alpha,\beta)} D^{\eta_i(t)}(S(t)) \\
 &= B_{(\alpha,\beta)} \left[\circ, \frac{\Gamma(\mathfrak{z})t^{(1-\eta_i(t))}}{\Gamma(\mathfrak{z}-\eta_i(t))}, \dots, \frac{\Gamma(N+1)t^{(N-\eta_i(t))}}{\Gamma(N+1-\eta_i(t))} \right]^T \\
 &= B_{(\alpha,\beta)} \begin{bmatrix} \circ & \circ & \circ & \dots & \circ \\ \circ & \frac{\Gamma(\mathfrak{z})t^{-\eta_i(t)}}{\Gamma(\mathfrak{z}-\eta_i(t))} & \circ & \dots & \circ \\ \circ & \circ & \frac{\Gamma(\mathfrak{z})t^{-\eta_i(t)}}{\Gamma(\mathfrak{z}-\eta_i(t))} & \dots & \circ \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \circ & \circ & \circ & \dots & \frac{\Gamma(N)t^{-\eta_i(t)}}{\Gamma(N+1-\eta_i(t))} \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ \vdots \\ t^N \end{bmatrix} \quad (13) \\
 &= B_{(\alpha,\beta)} Q_i(t) S(t), \quad i = 1, 2, \dots, n.
 \end{aligned}$$

با استفاده از معادله (۱۱) خواهیم داشت:

$$D^{\eta_i(t)} \Phi_{T,N}(t) = B_{(\alpha,\beta)} Q_i(t) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t), \quad i = 1, 2, \dots, n.$$

ماتریس عملگر $D^{\eta_i(t)} \Phi_{T,N}(t)$, $(i = 1, 2, \dots, n)$ ، مساوی $B_{(\alpha,\beta)} Q_i(t) B_{(\alpha,\beta)}^{-1}$ خواهد بود. اکنون می‌توانیم مشتق کسری مرتبه متغیر تابع تقریبی را که در معادله (۹) به دست می‌آید، تخمین بزنیم. سرانجام ما از t_j ($j = 0, 1, 2, \dots, m$) که ریشه‌های $P_{T,m+1}^{(\alpha,\beta)}(t)$ می‌باشد استفاده می‌کنیم. ازاینرو معادله (۱) به دستگاه جبری زیر تبدیل می‌گردد که با روش‌های عددی شناخته شده قابل حل است تا بردار مجهول A معین گردد، به عبارت دیگر جواب عددی ارائه شده در معادله (۹) حاصل می‌شود.

$$\begin{aligned}
 \sum_{i=1}^n \alpha_i (A^T B_{(\alpha,\beta)} Q_i(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)) + \alpha_{n+1} A^T \Phi_{T,N}(t_j - \tau) = \\
 F(t_j, A^T \Phi_{T,N}(t_j), (A^T B_{(\alpha,\beta)} Q_1(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)), (A^T B_{(\alpha,\beta)} Q_2(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)), \dots, \\
 (A^T B_{(\alpha,\beta)} Q_n(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)), A^T \Phi_{T,N}(t_j - \tau)), \quad j = 0, 1, 2, \dots, m. \quad (14)
 \end{aligned}$$

۴. دست‌آوردهای پژوهش

در این بخش، بر اساس بحث قبلی، یک مثال عددی برای نشان دادن دقت، کارایی، کاربرد، عمومیت و اعتبار تکنیک پیشنهادی آورده شده است. نتایج روش حاضر توسط نرم افزار Mathematica ۱۰ محاسبه شده است. به منظور آزمایش الگوی ارائه شده، ما آن را با سایر روش‌های شناخته شده از نظر خطاهای مطلق $|w_{exact}(t) - w_n(t)|$ و مقدار زمان CPU مورد نیاز برای حل مثال مقایسه کرده‌ایم. مقایسه نتایج به دست آمده از این تکنیک با جواب دقیق مثال، نشان می‌دهد که این تکنیک جدید تطابق بهتری نسبت به روش‌های دیگر دارد. پایداری، ثبات و اجرای آسان این تکنیک باعث می‌شود تا این روش کاربردی تر و قابل اعتمادتر باشد.

مثال ۱. معادله دیفرانسیل تأخیری کسری برای $0 < \eta \leq 1, \tau > 0$ را در نظر بگیرید:

$$D^\eta w(t) = \frac{\mathfrak{z} w(t)^{1-\frac{\eta}{\mathfrak{z}}}}{\Gamma(\mathfrak{z}-\eta)} + w(t-\tau) - w(t) + \mathfrak{z} \tau \sqrt{w(t)} - \tau^2,$$

$$w(t) = 0, \quad t \leq 0.$$

(۱۵)

جدول ۱: مقایسه خطای مطلق روش حاضر با روش‌های موجود در [۲، ۱] با $\alpha = 0, \beta = 0$ و $T = 2$ برای مثال ۱.

$t \in [0, T]$	روش حاضر، $N = 2$	روش حاضر، $N = 7$	روش [۱]، $N = 2000$	روش [۲]، $N = 2000$
$0/2$	$5/19525 \times 10^{-7}$	$4/23272 \times 10^{-16}$	$7/81197 \times 10^{-2}$	$7/81550 \times 10^{-2}$
$0/4$	$1/00211 \times 10^{-6}$	$1/38778 \times 10^{-17}$	$1/29928 \times 10^{-1}$	$1/29978 \times 10^{-1}$
$0/6$	$1/44777 \times 10^{-6}$	$1/66533 \times 10^{-16}$	$1/90687 \times 10^{-1}$	$1/90760 \times 10^{-1}$
$0/8$	$1/85649 \times 10^{-6}$	0	$2/48601 \times 10^{-1}$	$2/48994 \times 10^{-1}$
$1/0$	$2/22827 \times 10^{-6}$	0	$3/07649 \times 10^{-1}$	$3/07763 \times 10^{-1}$
$1/2$	$2/56312 \times 10^{-6}$	0	$3/66427 \times 10^{-1}$	$3/66563 \times 10^{-1}$
$1/4$	$2/86104 \times 10^{-6}$	0	$4/25322 \times 10^{-1}$	$4/25479 \times 10^{-1}$
$1/6$	$3/12202 \times 10^{-6}$	0	$4/84208 \times 10^{-1}$	$4/84387 \times 10^{-1}$
$1/8$	$3/34607 \times 10^{-6}$	0	$5/43110 \times 10^{-1}$	$5/43312 \times 10^{-1}$
$2/0$	$3/53318 \times 10^{-6}$	0	$6/02019 \times 10^{-1}$	$6/02243 \times 10^{-1}$
time CPU	$0/01560 s$	$0/63960 s$	$104/343750 s$	$215/031250 s$

توجه داریم که جواب دقیق برابر $w(t) = t^2$ و $\eta = 0/6, \tau = 0/3, T = 2, 0 \leq t \leq T$ می باشد.

مراجع

- [1] V. Daftardar-Gejji, Y. Sukale, S. Bhalekar, Solving fractional delay differential equations: a new approach, *An International Journal for Theory and Applications*, 2015.
- [2] S. Bhalekar, V. Daftardar-Gejji, A predictor-corrector scheme for solving non-linear delay differential equations of fractional order, *Journal of Fractional Calculus and Applications*, 2011 no. 5.
- [3] A.A. El-Sayed, D. Baleanu, P. Agarwal, A novel Jacobi operational matrix for numerical solution of multi-term variable-order fractional differential equations, *Journal of Taibah University for Science*, 14 (2020) 963–974.
- [4] H.R. Khodabandehlo, E. Shivanian, S. Abbasbandy, Numerical solution of nonlinear delay differential equations of fractional variable-order using a novel shifted Jacobi operational matrix, *Engineering with Computers*, 18 May 2021.
- [5] A.H. Bhrawy, M.A. Zaky, A method based on the Jacobi tau approximation for solving multi-term time-space fractional partial differential equations, *Journal of Computational Physics*, 281 (2015) 876–895.
- [6] D. Baleanu, R.L. Magin, S. Bhalekar, V. Daftardar-Gejji, Chaos in the fractional order nonlinear Bloch equation with delay, *Communications in Nonlinear Science and Numerical Simulation*, 25 (2015) 41–49.

- [7] K. Diethelm, N.J. Ford, A. D. Freed, Detailed error analysis for a fractional Adams method, *Numer Algorithms*, 36 (2004) 31–52, .
- [8] A. Jhinga, V. Daftardar-Gejji, A new numerical method for solving fractional delay differential equations, *Computational and Applied Mathematics*, 38 (2019) no. 4.
- [9] M.L. Morgado, N.J. Ford, P. Lima, Analysis and numerical methods for fractional differential equations with delay, *Journal of Computational and Applied Mathematics*, 252 (2013) 159–168.
- [10] M. Abbaszadeh, M. Dehghan, Numerical and analytical investigations for neutral delay fractional damped diffusion-wave equation based on the stabilized interpolating element free Galerkin (IEFG) method, *Applied Numerical Mathematics*, 145 (2019) 488–506.

Numerical Analysis

حمیدرضا خدابنده لو

گروه ریاضی کاربردی، دانشکده علوم پایه، دانشگاه بین المللی امام خمینی (ره)، قزوین، ایران.

آدرس ایمیل: khodabandelo.hamidreza@yahoo.com

الیا س شیان

گروه ریاضی کاربردی، دانشکده علوم پایه، دانشگاه بین المللی امام خمینی (ره)، قزوین، ایران.

آدرس ایمیل: eshivanian@gmail.com

ابزارهای نوظهور در بازارهای مالی (مدل بندی و شبیه سازی اوراق قرضه وقایع فاجعه آمیز)

ساغر حیدری، فرزانه میر

چکیده. در این مقاله به مطالعه اوراق قرضه وقایع فاجعه آمیز می پردازیم. در چند دهه گذشته افزایش وقایع فاجعه آمیز و در نتیجه مشکلات پوشش ریسک ناشی از تعهدات بیمه ای این وقایع برای شرکت ها و موسسات بیمه ای از یک سو و کمبود منابع و ظرفیت ها برای جبران خسارت های ناشی از این وقایع از سوی دیگر، زمینه بهره مندی از ظرفیت ها و پتانسیل های موجود در بازارهای سرمایه و ابزارهای مالی را فراهم نموده است. در این راستا اوراق قرضه وقایع فاجعه آمیز یکی از ابزارهای نوظهور در بازارهای مالی هستند که در جهت تامین منابع مالی و مدیریت ریسک مورد استفاده قرار می گیرند. با توجه به اهمیت وجود این اوراق قرضه بیمه ای در هر دو حوزه بازار سرمایه و صنعت بیمه، لزوم قیمت گذاری منصفانه این ابزارها پیش از پیش احساس می شود. در این پژوهش به مدل سازی و قیمت گذاری اوراق قرضه وقایع فاجعه آمیز می پردازیم. برای این منظور از مدل تصادفی مرتون توسعه یافته (هو-لی) برای دینامیک نرخ بهره به دلیل سادگی این مدل در مدل های تک عاملی استفاده کرده و سپس برای توابع بازپرداخت مختلف، قیمت منصفانه این اوراق را به کمک شبیه سازی مونت کارلو بدست می آوریم.

۱. پیش گفتار

وقوع رو به رشد وقایع فاجعه آمیز و بلایای طبیعی مانند سیل، طوفان و زلزله و به دنبال آن افزایش خسارات و زیان های جانی و مالی ناشی از این حوادث یکی از مشکلات اخیر در جوامع و کشورها به خصوص کشورها با موقعیت جغرافیایی حادثه خیز است. یک رویداد فاجعه آمیز طبیعی می تواند بیش از صدها میلیارد دلار خسارت به همراه داشته باشد. بنابراین جبران خسارت و پوشش ریسک ناشی از این وقایع یکی از موضوعات مورد توجه شرکت های بیمه در کشورهای حادثه خیز است. از جنبه دیگر، افزایش خسارات ناشی از بلایای طبیعی از یک طرف و کمبود ظرفیت های شرکت های بیمه ای برای جبران خسارات وارده از طرف دیگر، باعث بی میلی و عدم تمایل بیمه گران برای ادامه فعالیت در این حوزه شده است. مطالعات و بررسی ها در بازارهای مالی و سرمایه گذاری حاکی از این واقعیت است که حجم معاملات روزانه بازارهای مالی جهانی ممکن است تا ده ها میلیارد دلار باشد که این حجم از ظرفیت مالی در بازار سرمایه قابل توجه است. با این اوصاف می توان از ظرفیت بازارهای مالی در جهت پوشش ریسک های بزرگ در جامعه استفاده نمود. اولین بار ایده انتقال ریسک های بیمه ای به بازار سرمایه در جهت تامین منابع مالی و در نتیجه افزایش ظرفیت شرکت های بیمه و بیمه های اتکایی در سال ۱۹۹۳ در بورس شیکاگو مطرح و باعث شکل گیری ابزار نوینی در صنعت بیمه و در بازارهای مالی با عنوان اوراق بهادار بیمه ای شد. این ابزار نوظهور مالی دارای دو مشخصه عمده است: امکان انتقال ریسک ها به خصوص ریسک های ناشی از خسارات بلایای طبیعی از شرکت های بیمه و بیمه اتکایی را به سرمایه گذاران بازار سرمایه فراهم می کند و ثانیاً راهی برای تأمین مالی شرکت های بیمه و جبران کمبود ذخیره سرمایه آنها ایجاد می کند. از طرفی ابزار بسیار مناسبی برای متنوع سازی سبد مالی سرمایه گذاران در بازار سرمایه می باشد. دارایی پایه در اوراق قرضه بیمه ای، ریسک فاجعه آمیز ناشی از بیمه نامه های صادر شده توسط شرکت بیمه است. لازم به ذکر است که، این اوراق

واژه های کلیدی: ابزارهای نوین بازار مالی، اوراق قرضه وقایع فاجعه آمیز، مدل تصادفی نرخ بهره، شبیه سازی مونت کارلو.
طبقه بندی موضوعی [۲۰۱۰]: 13D45, 39B42.

می‌توانند علاوه بر پوشش ریسک حوادث فاجعه‌آمیز، برای ریسک‌های ناشی از مرگ و میر و یا ریسک از بین رفتن محصولات کشاورزی نیز مورد استفاده قرار گیرند. اما بیشترین حجم اوراق بهادار بیمه‌ای منتشر شده، مربوط به اوراق حوادث فاجعه‌آمیز است. در صورت وقوع حادثه که اوراق به خاطر آنها طراحی شده‌اند، بخش یا تمام اصل اوراق به زیان دارنده اوراق که همان سرمایه‌گذار است بازپرداخت نمی‌شود. با توجه به اهمیت وجود اوراق قرضه بیمه‌ای در هر دو حوزه بازار سرمایه و صنعت بیمه، لزوم قیمت‌گذاری این محصول نوین بیش از پیش احساس می‌شود. در این راستا در این پژوهش برانیم تا به مدل‌بندی و قیمت‌گذاری اوراق قرضه بیمه‌ای بپردازیم. برای این منظور مدل نرخ بهره هو-لی را در نظر گرفته و برای توابع بازپرداخت مختلف، قیمت منصفانه این اوراق را با این فرض که وقایع فاجعه‌آمیز مستقل از رفتار بازارهای مالی هستند، را می‌یابیم.

۲. قیمت گذاری اوراق فاجعه آمیز

در این بخش به منظور مدل‌بندی و قیمت‌گذاری اوراق قرضه بیمه‌ای نمادهای زیر را تعریف می‌کنیم:

r_t (نرخ لحظه‌ای سود بدون ریسک در زمان t)، W_t (حرکت براونی)، τ (زمان توقف)، ϕ_i (تابع توزیع τ_i)، F_t (فیلتریشن تا لحظه t)، N_t (تعداد رویدادهای فاجعه‌آمیز تا لحظه t)، U_i (مقدار زیان رویداد فاجعه‌آمیز i ام)، $F\nu$ (ارزش اسمی اوراق قرضه بدون کوپن)، T (زمان سررسید)، $B(t, T)$ (قیمت اوراق قرضه بدون کوپن در لحظه t با سررسید T)، $B(\circ, T)$ (قیمت اوراق قرضه بدون کوپن در لحظه صفر با سررسید T)، $\nu_{IB_p(T, F\nu)}$ (تابع بازپرداخت)، $P^M(\circ, T)$ (منحنی اوراق قرضه بدون کوپن)، $f^M(\circ, \circ)$ (نرخ پیش فروش لحظه‌ای بازار در زمان صفر برای اوراق قرضه بدون کوپن)، $IB_s(\circ)$ (قیمت اوراق قرضه تحت تابع بازپرداخت پله‌ای در لحظه صفر)، $IB_p(\circ)$ (قیمت اوراق قرضه تحت تابع بازپرداخت تکه‌ای در لحظه صفر)، E^Q (مقدار مورد انتظار تحت اندازه Q)، E^P (مقدار مورد انتظار تحت اندازه P).

حال با معرفی نمادهای بالا به معرفی تابع بازپرداخت اوراق قرضه می‌پردازیم. در این مقاله ما دو نوع تابع بازپرداخت تحت عنوان تابع بازپرداخت پله‌ای و تکه‌ای را در نظر می‌گیریم. فرمول این دو تابع بازپرداخت به ترتیب برابر:

$$F\nu(1 - \sum_{i=1}^n w_i 1_{\tilde{N} > K_i}),$$

و

$$F\nu(1 - \sum_{i=1}^n \frac{\tilde{N} \wedge K_i - \tilde{N} \wedge K_{i-1}}{K_i - K_{i-1}}),$$

است. که در آن \tilde{N}_t مجموع زیان‌ها تا لحظه t ، K_i و w_i مقادیر ثابت مربوط به قرارداد بیمه است.

۳. مدل هو-لی

فرض می‌کنیم دینامیک نرخ بهره بدون ریسک r_t توسط مدل هو-لی (مرتون توسعه یافته) توصیف شده است. در این صورت ساختار آن به صورت زیر است:

$$dr_t = \vartheta(t)dt + \sigma dW_t,$$

که در آن $\sigma > 0$ و برای $\vartheta(t)$ داریم:

$$\vartheta(t) = \frac{\partial f^M(\circ, t)}{\partial t} + \sigma^2 t + \lambda \sigma,$$

که در آن λ ریسک قیمت بازار است.

قضیه ۱. اگر دینامیک نرخ بهره از مدل هو-لی پیروی کند و $\phi(T)$ را به صورت زیر در نظر بگیریم:

$$\phi(T) = \sum_{i=1}^n w_i \phi_i$$

قیمت اوراق قرضه در زمان صفر برای دو تابع بازپرداخت پله‌ای و تکه‌ای به ترتیب برابر

$$IB_s(\circ) = P^M(\circ, T) \exp(Tf^M(\circ, \circ) - Tr_\circ)F\nu(1 - \phi(T))$$

و

$$IB_p(\circ) = P^M(\circ, T) \exp(Tf^M(\circ, \circ) - Tr_\circ)F\nu E^P \nu_{IB_p(T, F\nu)}$$

است.

Numerical Analysis

۴. نتایج عددی

در این بخش برای قیمت‌گذاری اوراق قرضه فاجعه‌آمیز و تجزیه و تحلیل فرمول‌های قیمت‌گذاری ارائه شده، از شبیه سازی مونت کارلو استفاده می‌کنیم. هدف اصلی ما نشان دادن امکان قیمت‌گذاری اوراق قرضه فاجعه‌آمیز از طریق محاسبات عددی، علیرغم ماهیت پیچیده معادلات معرفی شده است. برای این منظور N_t را به صورت یک فرایند پواسن همگن با پارامتر $31/7143$ مدل سازی می‌کنیم و فرض می‌کنیم U_i دارای توزیع وایبل با پارامترهای $0/0187$ و $0/2656$ است. ارزش اسمی اوراق برابر یک و زمان سررسید را یک سال فرض می‌کنیم. در ادامه σ ، r_\circ و $r(1)$ را پارامترهایی با مقادیر $0/014$ ، $0/095$ و $0/1$ در نظر می‌گیریم. نقاط شروع K_0 ، K_1 و K_2 را به ترتیب برابر $Q_{HPP-W}(0/75)$ ، $Q_{HPP-W}(0/85)$ و $Q_{HPP-W}(0/95)$ قرار می‌دهیم و با قیمت‌گذاری اوراق قرضه فاجعه‌آمیز برای مجموعه پارامترهای داده شده شروع می‌کنیم. سپس یک یا دو پارامتر را تغییر می‌دهیم و دیگری را ثابت در نظر می‌گیریم. در هر آزمایش ما $N = 1000000$ شبیه سازی ایجاد می‌کنیم.

جدول ۱: نتایج بدست آمده با ثابت قراردادن K_2 و K_1

$K_2(Q_{HPP-W})$	0/75	0/8	0/85	0/9	0/95
قیمت	0/665199	0/675093	0/684722	0/69736	0/713708
$K_1(Q_{HPP-W})$	0/5	0/55	0/6	0/65	0/7
قیمت	0/713708	0/724512	0/73516	0/746877	0/757744

۵. بحث و نتیجه‌گیری

با توجه به رشد روز افزون وقایع و بلایای طبیعی به ویژه برای کشورها با پتانسیل وقایع حادثه خیز مانند کشور ایران، خسارات ناشی از این بلایا برای شرکت‌های بیمه و بیمه‌های اتکایی جبران‌ناپذیر است. بنابراین استفاده از منابع مالی بازارهای سرمایه به کمک ابزارهای مالی شامل اوراق قرضه بیمه‌ای از یک سو باعث مدیریت ریسک و افزایش ظرفیت شرکت‌های بیمه و در نتیجه باعث رشد و توسعه صنعت بیمه می‌شود و از سوی دیگر به عنوان یک نوآوری در بازار مالی، ابزار جدیدی را به سبد مالی سرمایه‌گذاران بازار سرمایه ایران که از عدم تنوع ابزارهای سرمایه‌گذاری در دسترس رنج می‌برند، اضافه کرده و در نتیجه باعث توسعه بازارهای سرمایه می‌شود. با توجه به اهمیت موضوع در این مقاله ما اوراق قرضه را ابتدا با استفاده از مدل نرخ بهره بدون ریسک قیمت‌گذاری و سپس مدل هولی را بررسی نمودیم. در این پژوهش دو ساختار پیچیده از توابع بازپرداخت اوراق قرضه شامل توابع تکه‌ای خطی را در نظر گرفته و با استفاده از رویکرد قیمت‌گذاری مارتینگل، فرمول‌های ارزش گذاری برای اوراق قرضه فاجعه‌آمیز را یافتیم. در نهایت از روش‌های عددی از جمله، شبیه‌سازی مونت کارلو برای تجزیه و تحلیل فرمول‌های قیمت‌گذاری بدست آمده استفاده نمودیم.

مراجع

- [1] P. Nowak, M. Romaniuk, T. Ermolieva, *Evaluation of portfolio of financial and insurance instrumentssimulation of uncertainty*. In: Ermoliev Y, Makowski M, Marti K (eds) *Managing safety of heterogeneous systems: decisions under uncertainties and risks, Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Berlin Heidelberg, 2012, 351–366.
- [2] P. Nowak, M. Romaniuk, Pricing and simulations of catastrophe bonds. *Insurance: Mathematics and Economics*, 52 (2013) 18–28.
- [3] P. Nowak, M. Romaniuk, *Pricing of catastrophe bond in fuzzy framework*. In: Borgelt C, Gil M, Sousa JMC, Verleysen M (eds) *Towards advanced data analysis by combining soft computing and statistics. Studies in fuzziness and soft computing*, Springer-Verlag, Berlin Heidelberg, 2013, 137–150.
- [4] P. Nowak, M. Romaniuk, Valuing catastrophe bonds involving correlation and CIR interest rate model, *Computational and Applied Mathematics*, 37 (2018) 365–394.
- [5] V.E. Vaugirard, Pricing catastrophe bonds by an arbitrage approach, *The Quarterly Review of Economics and Finance*, 43 (2003) 119–132.

ساغر حیدری

دانشکده ریاضی، دانشگاه شهید بهشتی، تهران، ایران.

آدرس ایمیل: s_heidari@sbu.ac.ir

فرزانه میر

دانشکده ریاضی، دانشگاه شهید بهشتی، تهران، ایران.

آدرس ایمیل: farzanehmir25@gmail.com