Journal of Progressive Research in Mathematics
Vol 20,1 - 2023 — E-ISSN: 2395-0218
http://www.scitecresearch.com/journals/index.php/jprm

Numerical Methods for Convex Quadratic Programming with Nonnegative Constraints

Qi Wang

School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou, China
Email: bmwzwq@126.com

Received: January 23, 2023; Accepted: February 16, 2023; Published: March 6, 2023

Abstract

This paper deals with some problems in numerical simulation for convex quadratic programming with nonnegative constraints. For systems of ordinary differential equations which derived from the above mentioned problem, we construct a kind of new numerical method: the modified implicit Euler method. Under some restrictions for step-size, we obtained the numerical solution which satisfied with the termination condition. Compared with the classical Matlab command ODE23, the new method has ideal computation cost.

Keywords

Quadratic Programming; Ordinary Differential Equations; the Modified Implicit Euler Method

1. Introduction

In the past few years, quadratic programming problems play an important role in all kinds of research. Generally speaking, there are three types of quadratic programming in according to the limits on constraint conditions. That is, nonnegativity constrains, box constraints and equality constraints are included. In particular, nonnegativity constrains quadratic programming problems always appears in science, engineering and business, and they may fall into nonnegativity constrained least-squares problems. Moreover, in support vector machines, computing the maximum margin hyperplane also produces a nonnegativity constrained quadratic programming [1].

Usually, solving a quadratic programming problem can be converted into solving a system of ordinary differential equation (ODE). There are many numerical methods for solving the system of ODE. Such as one step methods [2, 3], Runge-Kutta methods [4, 5], pseudo-spectral method [6], waveform relaxation methods [7, 8], finite transfer method [9], general linear methods [10, 11]. The interested readers can see the most recent articles [12, 13, 14]. Different from the above mentioned papers, in this paper we introduce a easy-

How to cite this paper: Q. Wang (2023) Numerical Methods for Convex Quadratic Programming with Nonnegative Constraints. Journal of Progressive Research in Mathematics, 20(1), 39-48. Retrieved from http://scitecresearch.com/journals/index.php/jprm/article/view/2194

to-use and effective method to solve the system of ODE which corresponding to a quadratic programming problem with nonnegativity constrains. The experimental results show that the new method is a simple and efficient method for solving the system of ODE.

Consider the following nonnegativity constrained quadratic programming

$$\begin{array}{ll} \min & q(x) = \frac{1}{2}x^TQx + c^Tx, \\ s.t. & x \geq 0, \end{array} \tag{1}$$

where $x, c \in \mathbb{R}^n$ and $Q = (q_{ij}) \in \mathbb{R}^{n\times n}$. The superscript "T" means the transpose. Throughout this paper, we always assume that Q is symmetric and positive semi-definite.

For convex quadratic programming problem, a kind of continuous method [15] often be applied. The main idea in this approach is to formulate an ODE for each optimization problem such that the limiting equilibrium point of the ODE corresponds to an optimal solution of the corresponding optimization problem. Usually, this ODE is an equation with high dimension, it will result in some difficulty when using Matlab ODE solver ODE23 in numerical treatment. Thus, seeking a type of efficient and fast numerical method is very urgent and meaningful. In this paper, we construct a new method: the modified implicit Euler method which can be seen as a effective method to simulate the systems of ODE.

The rest of this paper is organized as follows. In Section 2, some corresponding theoretical results for original problem will be presented. In Section 3, the modified implicit Euler method will be constructed for systems of ODE corresponding to the above mentioned nonnegativity constrained quadratic programming. From the view point of nonnegativity and monotonicity, we find the scope of step-size. Some examples are given in Section 4. Finally, conclusions are drawn in Section 5.

2. Preliminaries

The following ODE for Problem (1) can be constructed

$$\begin{array}{l} \frac{dx(t)}{dt} = -X(Qx + c), \\ x(0) = x^0, \end{array} \tag{2}$$

where $X = diag(x) \in \mathbb{R}^{n\times n}$, Q and c are defined in (1), $x^0 \geq 0$ is an initial vector.

Lemma 1. *[16] Let $x(t)$ be a solution of (2), then $x(t)$ is unique, well defined and $x(t) > 0$ for all $t \geq 0$.*

Lemma 2. *(Weak Convergence)[16] Let $x(t)$ be the solution of (2), then $\lim_{t\to+\infty} X(Qx + c) = 0$.*

Denote

$$\mathbb{R}^n_+ = \{x \in \mathbb{R}^n | x \geq 0\},$$

and

$$\Omega_1(x^0) = \left\{y \in \mathbb{R}^n_+ | y \quad \text{is a cluster point of the solution} \quad x(t) \quad \text{of} \quad (2)\right\},$$

thus

Lemma 3. *[16] Any point $x \in \Omega_1(x^0)$ is an optimal solution of (1).*

Lemma 4. *(Strong Convergence)[16] $x \in \Omega_1(x^0)$ only contains a single point.*

Lemmas 3 and 4 guarantee the limit point is an optimal solution for Problem (1).

3. The modified implicit Euler method

In this section, we construct a new method: the modified implicit Euler method to solve (2). Furthermore, proper restrictions on step-size are established.

3.1. Construction a numerical method

Firstly, we apply the explicit Euler method and the implicit Euler method to (2) gives

$$x_{k+1} = x_k - hX_k(Qx_k + c), \tag{3}$$

and

$$x_{k+1} = x_k - hX_{k+1}(Qx_{k+1} + c), \tag{4}$$

respectively, where h is step-size, $x_k \in \mathbb{R}^{n\times 1}$ is an approximation to $x(t)$ at t_k.

Through practical test we find that the explicit Euler method is slow when n is large. Meanwhile, the implicit Euler method is also infeasible because we have to handle high-dimensional and non-linear systems of algebra equation (4) which will occupy too much of time. How to cope with this difficulty? Noticing the structure of (3) and (4), we separate the terms X and $Qx + c$ (one for implicit and the other for explicit) gives the following new difference form

$$x_{k+1} = x_k - hX_{k+1}(Qx_k + c), \tag{5}$$

we call Formula (5) as the modified implicit Euler method.

For simplicity, we take $[x_k]_i$ as i-th component, so from (5) we have

$$[x_{k+1}]_i = \frac{[x_k]_i}{1 + h[Qx_k + c]_i}, \tag{6}$$

where $i = 1, 2, \cdots, n$.

3.2. Selection of step-size

In original quadratic programming problem, the nonnegativity for x and the monotonicity for the objective function must be satisfied. In other words, $x \geq 0$ and $q(x_{j+1}) < q(x_j)$. Naturally, we must require $x_k \geq 0$ and $q(x_{k+1}) < q(x_k)$ for numerical case.

Theorem 1. *(Preserve the nonnegativity) The step-size can be selected as*

$$h_{n+1} = \begin{cases} 2h_n, \mathit{when} \min(Qx_k + c) \geq 0, \\ r, \mathit{when} -1 < \min(Qx_k + c) < 0, \\ -1/\min(Qx_k + c), \mathit{when} \min(Qx_k + c) \leq -1, \end{cases} \tag{7}$$

where $0 < r < 1$.

Proof. According to the above analysis, in this case, we require $x_k \geq 0$ for all $k \geq 1$. Thus, from (6) we have $1 + h[Qx_k + c]_i > 0$, that is $h \min[Qx_k + c]_i > -1$. The results can be obtained by simple derivation. □

Theorem 2. *(Preserve the monotonicity) The step-size can be selected as*

$$\begin{cases} \mathit{no\ bound}, \mathit{when} \quad \lambda - \gamma \leq 0, \\ h < 2/((\lambda - \gamma)\max[x_k]_i), \mathit{when} \quad \lambda - \gamma > 0, \end{cases} \tag{8}$$

where λ is the maximum eigenvalue of Q and $\gamma = \min(2[Qx_k + c]_i/[x_k]_i)$.

Proof. In view of $q(x) = 1/2 \times x^T Qx + c^T x$ and (5), we have

$$\begin{aligned} q(x_{k+1}) &= \tfrac{1}{2} x_{k+1}^T Qx_{k+1} + c^T x_{k+1} \\ &= \tfrac{1}{2} \left(x_k - hX_{k+1}(Qx_k + c)\right)^T Q \left(x_k - hX_{k+1}(Qx_k + c)\right) \\ &\quad + c^T \left(x_k - hX_{k+1}(Qx_k + c)\right) \\ &= q(x_k) - hx_k^T QX_{k+1}(Qx_k + c) - hc^T X_{k+1}(Qx_k + c) \\ &\quad + \tfrac{h^2}{2}(Qx_k + c)^T X_{k+1}QX_{k+1}(Qx_k + c) \\ &= q(x_k) - h(Qx_k + c)^T X_{k+1}(Qx_k + c) \\ &\quad + \tfrac{h^2}{2}(Qx_k + c)^T X_{k+1}QX_{k+1}(Qx_k + c) \\ &= q(x_k) - \tfrac{h}{2}(Qx_k + c)^T X_{k+1}(2X_{k+1}^{-1} - hQ)X_{k+1}(Qx_k + c). \end{aligned} \tag{9}$$

To satisfy $q(x_{k+1}) < q(x_k)$ we only require $2X_{k+1}^{-1} - hQ$ is positive definite. So we have

$$\min \frac{2}{h[x_{k+1}]_i} > \lambda,$$

by (6) we obtain

$$\min \frac{2(1 + h[Qx_k + c]_i)}{h[x_k]_i} > \lambda,$$

further

$$\min \frac{2}{h[x_k]_i} + \min \frac{2[Qx_k + c]_i}{[x_k]_i} > \lambda,$$

that is

$$\min \frac{2}{h[x_k]_i} > \lambda - \gamma.$$

The proof is complete. □

Corollary 1. *The step-size can be selected as*

$$\begin{cases} \textit{no bound, when} \quad \gamma_i - \lambda \geq 0, \\ h < 2/\max\{(\lambda - \gamma_i)[x_k]_i\}, \textit{when} \quad \gamma_i - \lambda < 0, \end{cases} \tag{10}$$

where λ is the maximum eigenvalue of Q and $\gamma_i = 2[Qx_k + c]_i/[x_k]_i$.

Combine Theorem 1 with Theorem 2 or Corollary 1, we can get the range of h which is helpful to obtain the desired numerical solution.

4. Numerical experiments

In this section, some numerical examples are addressed to test the effectiveness of the new method. Moreover, this method is compared with ODE23 in [16].

The initial point for the randomly generated (Q, c) is set to $x^0 = (1, 1, \cdots, 1)^T$. Let $r = 0.1$ and the initial step-size $h_0 = 0.1$. The termination condition is $|dx/dt|_\infty \leq 10^{-4}$.

In Table 1, we compare the Cpu-time consumed by Matlab command ODE23 and the modified implicit Euler method (MIE). From this table we can see that our method need less time than ODE23. In Table 2, we list the termination condition (Error) and the absolute error (AE) which is the difference between exact solution and numerical solution for the objective function. Together with Figures 1-4 we can see that the numerical solution is approximate to the exact solution and the objective function also decreases monotonically. In Table 3, we give the steps of iteration and the source of step-size h. The symbols C_1 and C_2 stand for h comes form Theorem 1 and Theorem 2, respectively. From Table 3 and Figures 5-8 we easily see that the choice of h is largely depend on criterion $q(x_{k+1}) < q(x_k)$.

5. Conclusions

In this paper we give a new method: the modified implicit Euler method in numerical simulation for convex quadratic programming with nonnegative constraints. To satisfy theoretical requirements, we present two kinds of selection criterion for step-size. Numerical experiments illustrate that the modified implicit Euler method with these selection criterions for step-size is effective. We will consider the convex quadratic programming with equality constraints in our future work.

Conflicts of interest

There are no conflicts to declare.

n	ODE23	MIE
200	2.4919e-01	2.3274e-02
400	4.7448e-01	4.9198e-02
600	7.8367e-01	9.6974e-02
800	1.2884e+00	1.3462e-01
1000	2.7535e+00	1.9811e-01
1200	3.3639e+00	6.1610e-01
1400	4.0642e+00	1.6912e+00
1600	6.0213e+00	2.1896e+00
1800	6.5972e+00	3.6472e+00
2000	1.0262e+01	4.0912e+00
4000	3.1919e+01	2.1877e+01

Table 1: Comparison Cpu-time between ODE23 and MIE

n	Error	AE
200	9.9685e-05	9.3917e-05
400	9.9798e-05	7.7618e-05
600	9.9781e-05	2.0374e-04
800	9.9984e-05	2.2021e-04
1000	9.9968e-05	3.4164e-04
1200	9.9949e-05	3.7970e-04
1400	9.9846e-05	5.3768e-04
1600	9.9743e-05	3.8696e-04
1800	9.9751e-05	5.8864e-04
2000	9.9891e-05	5.6978e-04
4000	9.9808e-05	8.9196e-04

Table 2: Error and AE

n	steps	C_1	C_2
200	367	9	358
400	401	9	392
600	484	12	472
800	507	10	497
1000	522	10	512
1200	544	10	534
1400	571	11	560
1600	500	10	490
1800	705	12	693
2000	484	10	474
4000	764	12	752

Table 3: The source of step-size

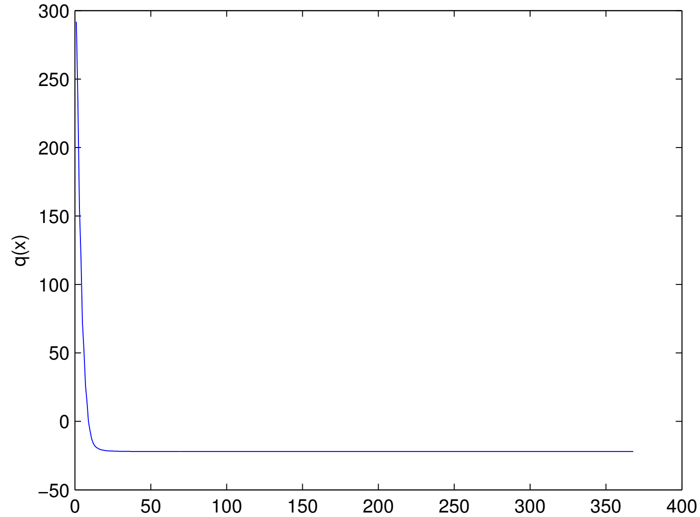

Figure 1: The curve of $q(x)$ with $n = 200$.

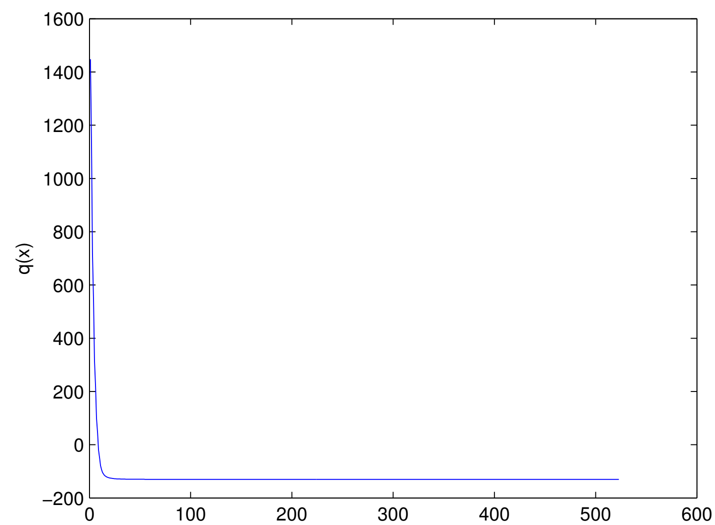

Figure 2: The curve of $q(x)$ with $n = 1000$.

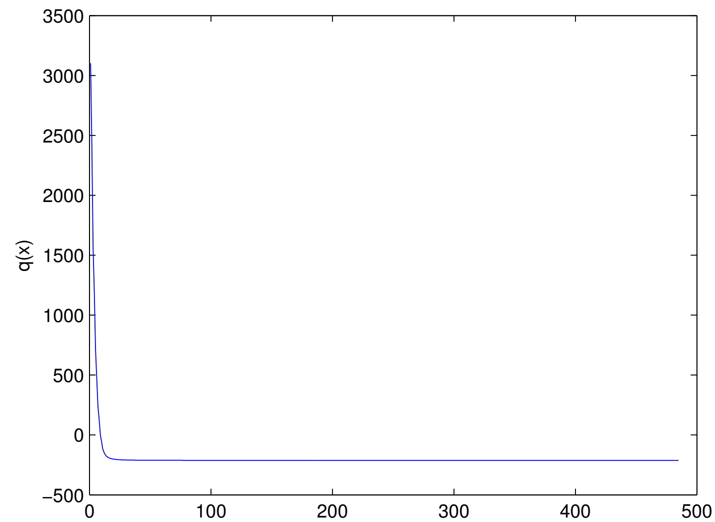

Figure 3: The curve of $q(x)$ with $n = 2000$.

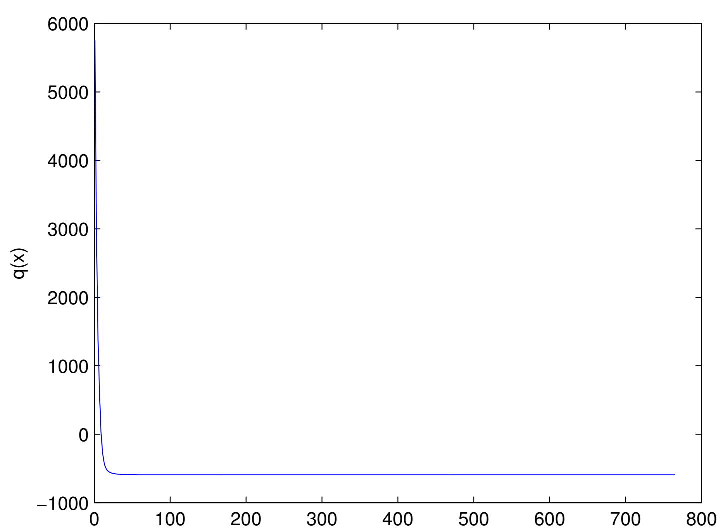

Figure 4: The curve of $q(x)$ with $n = 4000$.

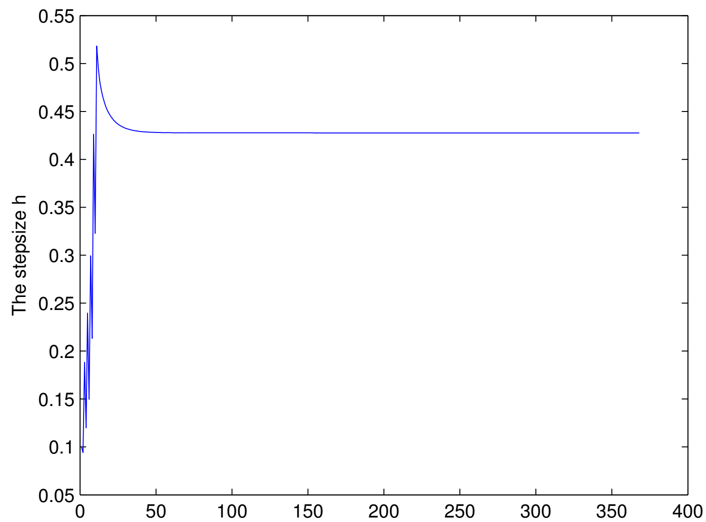

Figure 5: The curve of h with $n = 200$.

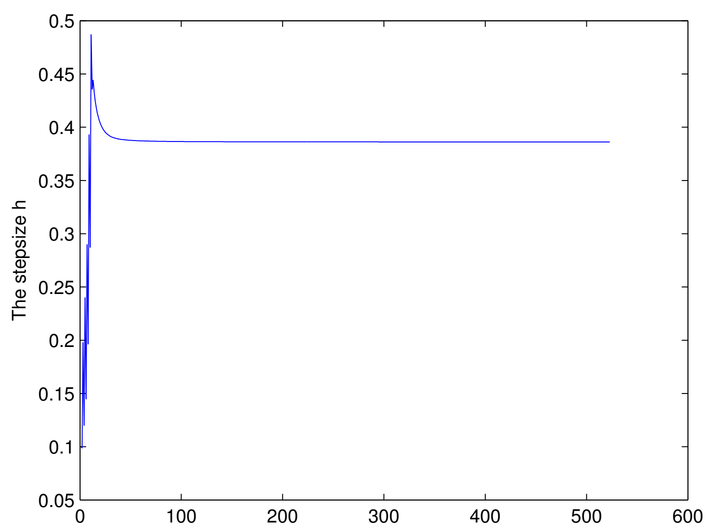

Figure 6: The curve of h with $n = 1000$.

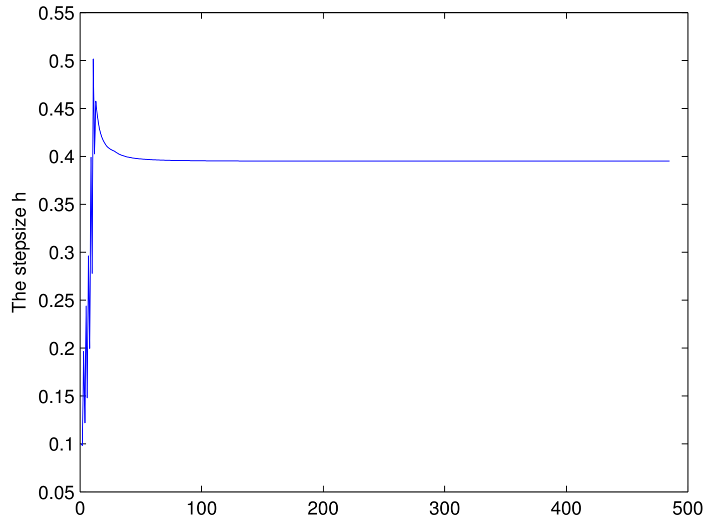

Figure 7: The curve of h with $n = 2000$.

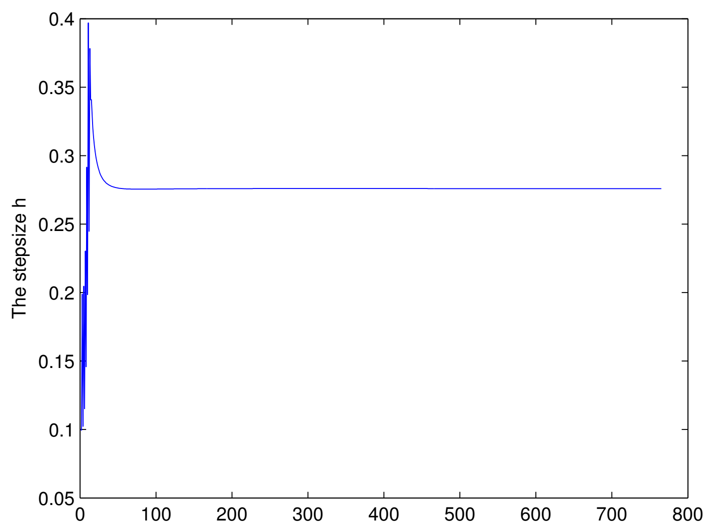

Figure 8: The curve of h with $n = 4000$.

Acknowledgments

The author thanks the Editor and the referees for their comments. The author also would like to thank Prof. Zhanwen Yang for his valuable suggestions and sharing his insights with me.

References

[1] F. Sha, L.K. Saul, D.D. Lee. Multiplicative updates for nonnegative programming in support vector machines. Advances in Neural Information Processing Systems, 15:1041-1048, 2003.

[2] M.V. Bulatov, A.V. Tygliyan, S.S. Filippov. A class of one-step one-stage methods for stiff systems of ordinary differential equations. Computational Mathematics and Mathematical Physics, 51(7): 1167-1180, 2011.

[3] H. Zheng, W. Han. On some discretization methods for solving a linear matrix ordinary differential equation. J. Math. Chem., 49: 1026-1041, 2011.

[4] V.M. Abdullaev, K.R. Aida-Zade. Numerical method of solution to loaded nonlocal boundary value problems for ordinary differential equations. Computational Mathematics and Mathematical Physics, 54(7): 1096-1109, 2014.

[5] H. Liang, M.Z. Liu, Z.W. Yang. Stability analysis of Runge-Kutta methods for systems $u'(t) = Lu(t) + Mu([t])$. Applied Mathematics and Computation, 228: 463-476, 2014.

[6] M. Saravi, E. Babolian, R. England, et al. System of linear ordinary differential and differential-algebraic equations and pseudo-spectral method. Computers and Mathematics with Applications, 59: 1524-1531, 2010.

[7] M. Khanamiryan. Quadrature methods for highly oscillatory linear and non-linear systems of ordinary differential equations: part I. BIT Numerical Mathematics, 48: 743-761, 2008.

[8] M. Khanamiryan. Quadrature methods for highly oscillatory linear and non-linear systems of ordinary differential equations: part II. BIT Numerical Mathematics, 52: 383-405, 2012.

[9] L. Gimena, P. Gonzaga, F.N. Gimena. Boundary equations in the finite transfer method for solving differential equation systems. Applied Mathematical Modelling, 38: 2648-2660, 2014.

[10] Z. Jackiewicz. Construction and implementation of general linear methods for ordinary differential equations: A Review. Journal of Scientific Computing, 25(1/2): 29-49, 2005.

[11] H. Zhang, A. Sandu, S. Blaise. Partitioned and implicit-explicit general linear methods for ordinary differential equations. Journal of Scientific Computing, 61: 119-144, 2014.

[12] M. Qayyum, Q. Fatima. Solutions of stiff systems of ordinary differential equations using residual power series method. Journal of Mathematics, 2022: 1-7, 2022.

[13] F.M. Burrell, P.E. Warwick, I.W. Croudace, W.S. Walters. Development of a numerical simulation method for modelling column breakthrough from extraction chromatography resins. Analyst, 146: 4049-4065, 2021.

[14] R. Abu-Gdairi, S. Hasan, S. Al-Omari, et al. Attractive multistep reproducing kernel approach for solving stiffness differential systems of ordinary differential equations and some error analysis. Computer Modeling in Engineering and Sciences, 130(1): 299-313, 2022.

[15] L.Z. Liao, H.D. Qi, L.Q. Qi. Neurodynamical optimization. JOGO, 28:175-195, 2004.

[16] H.W. Yue. First-order affine scaling continuous method for convex quadratic programming. Hong Kong Baptist University, 2014.