

Research Article

Study on Banded Implicit Runge–Kutta Methods for Solving Stiff Differential Equations

M. Y. Liu , L. Zhang , and C. F. Zhang

School of Physics & Electronics, Henan University, 475004 Kaifeng, China

Correspondence should be addressed to L. Zhang; zhanglei@vip.henu.edu.cn

Received 13 June 2019; Revised 12 August 2019; Accepted 29 August 2019; Published 10 October 2019

Academic Editor: Luigi Rodino

Copyright © 2019 M. Y. Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The implicit Runge–Kutta method with A-stability is suitable for solving stiff differential equations. However, the fully implicit Runge–Kutta method is very expensive in solving large system problems. Although some implicit Runge–Kutta methods can reduce the cost of computation, their accuracy and stability are also adversely affected. Therefore, an effective banded implicit Runge–Kutta method with high accuracy and high stability is proposed, which reduces the computation cost by changing the Jacobian matrix from a full coefficient matrix to a banded matrix. Numerical solutions and results of stiff equations obtained by the methods involved are compared, and the results show that the banded implicit Runge–Kutta method is advantageous to solve large stiff problems and conducive to the development of simulation.

1. Introduction

A stiff equation is a differential equation for which certain numerical methods for solving the equation are numerically unstable, unless the step size is taken to be extremely small. Differential equations in chemical reactions, automatic control, electronic networks, and biology are often expressed as stiff equations [1]. For general numerical methods, stiff equations must take small steps to obtain satisfactory results, which increases the computational complexity and is unfavourable to simulation. In the meantime, with the increase of step size, the round-off error becomes larger and larger, which affects the accuracy of the simulation. Therefore, for the stiff problem, the general numerical method is not competent, and it is necessary to use a method with better stability to solve stiff differential equations.

The Runge–Kutta method is a widely used numerical method because of its high accuracy and high stability [2–6]. The Runge–Kutta method is divided into two types: explicit Runge–Kutta method and implicit Runge–Kutta (IRK) method. The IRK method has wider stable region and higher accuracy compared with explicit Runge–Kutta methods [7].

Therefore, the IRK method is suitable for solving stiff differential equations, and abundant results have been mentioned in the literature [8–11].

However, unlike explicit Runge–Kutta methods, the IRK method needs to solve a system of equations in each step of calculation, which increases the cost of calculation. The aim of reducing the computational cost for IRK methods leads to the singly diagonally implicit Runge–Kutta (SDIRK) method, whose coefficient matrix A is lower triangular with same diagonal elements λ instead of using a full coefficient matrix [12–16]. However, it also has some drawbacks that its stability and accuracy are adversely affected by the simplification of the coefficient matrix A . Another classic transformed method whose Runge–Kutta matrix has just one real s fold eigenvalue is called a singly implicit Runge–Kutta (SIRK) method [17, 18]. Although this method has obvious computational advantages for large systems over fully IRK method, it has a complex transformation [19]. Apart from transformation, the stability and accuracy are also reduced compared to traditional IRK methods. Therefore, the search for high accuracy and high stability while retaining computational advantages leads to the banded implicit Runge–Kutta (BIRK) method.

2. The Problem of Implementing IRK Methods

For initial value problems of an ordinary differential equation with dimension N ,

$$y' = f(y(x)), y(x_0) = y_0. \quad (1)$$

The approximate solution obtained by the s stage implicit Runge–Kutta method is of the form

$$Y_i = y_n + h \sum_{j=1}^s a_{ij} f(x_n + c_j h, Y_j), \quad i = 1, \dots, s, \quad (2)$$

$$y_{n+1} = y_n + h \sum_{j=1}^s b_j f(x_n + c_j h, Y_j), \quad (3)$$

where $x_n = x_0 + nh$ ($n = 0, 1, \dots, h > 0$) and h is the step size of each integration. c_i is the node and b_i is the weight coefficient. The coefficient matrix $\mathbf{A} = (a_{ij})$ ($i, j = 1, 2, \dots, s$) is in the form of $s \times s$ and $c_i = \sum_{j=1}^s a_{ij}$.

A complicated iteration scheme for the solutions of the internal stages is necessary for IRK methods. In order to solve (2), let

$$z_i = Y_i - y_n. \quad (4)$$

If the coefficient matrix of the IRK method is non-singular, then (4) can be written as

$$\begin{bmatrix} z_1 \\ \vdots \\ z_s \end{bmatrix} = \mathbf{A} \begin{bmatrix} hf(t_n + c_1 h, y_n + z_1) \\ \vdots \\ hf(t_n + c_s h, y_n + z_s) \end{bmatrix}. \quad (5)$$

To solve (5), the Newton–Raphson iteration method can be used. The system of the Newton method can be written as

$$\mathbf{M} \Delta \mathbf{Z}^{(k)} = -\mathbf{Z}^{(k)} + h(\mathbf{A} \otimes \mathbf{I}_N) \mathbf{F}(\mathbf{Z}^{(k)}), \quad (6)$$

$$\mathbf{M} = (\mathbf{I}_s \otimes \mathbf{I}_N - h\mathbf{A} \otimes \mathbf{J}), \quad (7)$$

$$\mathbf{Z}^{(k+1)} = \mathbf{Z}^{(k)} + \Delta \mathbf{Z}^{(k)}, \quad (8)$$

where \otimes is the Kronecker product, \mathbf{I}_s is the $s \times s$ identity matrix, and \mathbf{I}_N is the $N \times N$ identity matrix. Let $\mathbf{J} = (\partial f / \partial y)$, and \mathbf{M} is the $sN \times sN$ matrix.

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_N - ha_{11}\mathbf{J} & -ha_{12}\mathbf{J} & \cdots & -ha_{1s}\mathbf{J} \\ -ha_{21}\mathbf{J} & \mathbf{I}_N - ha_{22}\mathbf{J} & \cdots & -ha_{2s}\mathbf{J} \\ \vdots & \vdots & \ddots & \vdots \\ -ha_{s1}\mathbf{J} & -ha_{s2}\mathbf{J} & \cdots & \mathbf{I}_N - ha_{s2}\mathbf{J} \end{bmatrix}. \quad (9)$$

For the IRK method, the coefficient matrix \mathbf{M} is required for each iteration.

Let $\mathbf{Z}^{(k+1)} = (z_1^{(k+1)}, \dots, z_s^{(k+1)})^T$ be the modified approximation after a single iteration, $\Delta \mathbf{Z}^{(k)} = (\Delta(z_1^{(k)}), \dots, \Delta(z_s^{(k)}))^T$ be the Newton update, and $\mathbf{F}(\mathbf{Z}^{(k)}) = (f(x_n + c_1 h, y_n + z_1^{(k)}), \dots, f(x_n + c_s h, y_n + z_s^{(k)}))^T$. Then, \mathbf{F} and \mathbf{Z} are the $s \times N$ matrix because the dimensions of Y_i and $f(x_n + c_j h, Y_j)$ are N .

We can see that the computation of the stage values is typically the most expensive component in their implementation because of the iteration. For each iteration, we need to do calculations about evaluation of \mathbf{F} , evaluation of $-\mathbf{Z}^{(k)} + h(\mathbf{A} \otimes \mathbf{I}_N) \mathbf{F}(\mathbf{Z}^{(k)})$, LU factorization of matrix \mathbf{M} , and back substitution to the Newton update $\Delta \mathbf{Z}^{(k)}$. In addition, the computation cost grows rapidly as the stage of methods s or the dimension of the system N increases. Especially, for simulation of large system problems the cost is increasingly expensive. Therefore, it is needed to reduce the cost of calculation by the following method.

3. Methods

3.1. SDIRK and SIRK. In order to reduce the computation cost of the IRK method with a full coefficient matrix \mathbf{A} , the SDIRK method is introduced. SDIRK is a method whose coefficient matrix \mathbf{A} is lower triangular with same diagonal elements λ . For this method, \mathbf{M} will be changed from (9) to a matrix of this form:

$$\mathbf{M}_D = \begin{bmatrix} \mathbf{I}_N - h\lambda \mathbf{J} & 0 & \cdots & 0 \\ -ha_{12}\mathbf{I}_N & \mathbf{I}_N - h\lambda \mathbf{J} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -ha_{s1}\mathbf{I}_N & a_{s2}\mathbf{I}_N & \cdots & \mathbf{I}_N - h\lambda \mathbf{J} \end{bmatrix}. \quad (10)$$

In this case, it may only be necessary to use repeatedly the LU factorization of $\mathbf{I}_N - h\lambda \mathbf{J}$ to obtain the upper triangular matrix of the LU factorization of the matrix \mathbf{M}_D . So, the cost of LU factorization of the matrix \mathbf{M}_D is decreased greatly, and the calculation cost of back substitution to the Newton update is reduced.

Although the SDIRK method has obvious computational advantages over the fully IRK method, the method has some drawbacks that its stability and accuracy are affected by the simplification of the coefficient matrix \mathbf{A} . The maximum attainable order for s stage with the A-stable IRK method is $2s$, whereas the s stage SDIRK method has the maximum attainable order $s + 1$ only when $s = 1, 2, 3, 5$ [20].

Simultaneously, the SIRK method can achieve the same effect of reducing the cost of calculation as the SDIRK method through transformation. Let \mathbf{T} be a nonsingular matrix such that $\bar{\mathbf{A}} = \mathbf{T}^{-1} \mathbf{A} \mathbf{T}$ is the Jordan canonical form of \mathbf{A} with the same diagonal elements.

$$\bar{\mathbf{M}} \Delta \bar{\mathbf{Z}}^{(k)} = -\bar{\mathbf{Z}}^{(k)} + h(\bar{\mathbf{A}} \otimes \mathbf{I}_N) \bar{\mathbf{F}}(\mathbf{Z}^{(k)}), \quad (11)$$

where

$$\bar{\mathbf{M}} = (\mathbf{I}_s \otimes \mathbf{I}_N - h\bar{\mathbf{A}} \otimes \mathbf{J}), \quad (12)$$

$$\bar{\mathbf{Z}}^{(k+1)} = (\mathbf{T}^{-1} \otimes \mathbf{I}_N) \mathbf{Z}^{(k)}, \quad (13)$$

$$\bar{\mathbf{F}}(\mathbf{Z}^{(k)}) = (\mathbf{T}^{-1} \otimes \mathbf{I}_N) \mathbf{F}(\mathbf{Z}^{(k)}), \quad (14)$$

$$\Delta \mathbf{Z}^{(k)} = (\mathbf{T} \otimes \mathbf{I}_N) \Delta \bar{\mathbf{Z}}^{(k)}. \quad (15)$$

By using some transformation matrices, we can still have the iteration matrix $\mathbf{I}_N - h\lambda \mathbf{J}$ for each stage iteration.

The other fact is that it needs a complex transformation, and it is obvious from (12)–(15). It is required to find a transformation matrix \mathbf{T} which can make a Jordan canonical $\mathbf{T}^{-1}\mathbf{A}\mathbf{T}$ form of the coefficient matrix \mathbf{A} and then transform \mathbf{M} , $\mathbf{Z}^{(k)}$, $\Delta\mathbf{Z}^{(k)}$, and $\mathbf{F}(\mathbf{Z}^{(k)})$ into $\overline{\mathbf{M}}$, $\overline{\mathbf{Z}}^{(k+1)}$, $\Delta\overline{\mathbf{Z}}^{(k)}$, and $\overline{\mathbf{F}}(\mathbf{Z}^{(k)})$, respectively. This transformation cost is relatively high, which is even poor for low-dimensional systems. In addition, not all SIRK methods are A-stable, and the maximum attainable order is reduced. The SIRK method has the maximum attainable order $s+1$ [21]. The effective method is to reduce the computation cost on the basis of maintaining high accuracy and good stability. Therefore, our BIRK method is proposed as follows.

3.2. Construction of BIRK Methods. The construction of IRK methods relies on the simplifying assumptions

$$B(p) : \sum_{i=1}^s b_i c_i^{q-1} = \frac{1}{q}, \quad q = 1, \dots, p; \quad (16)$$

$$C(\eta) : \sum_{j=1}^s a_{ij} c_j^{q-1} = \frac{c_i^q}{q}, \quad i = 1, \dots, s, q = 1, \dots, \eta; \quad (17)$$

$$D(\zeta) : \sum_{i=1}^s b_i c_i^{q-1} a_{ij} = \frac{b_j}{q} (1 - c_j^q), \quad j = 1, \dots, s, q = 1, \dots, \zeta. \quad (18)$$

Theorem 1 (see [22]). *If the coefficient b_i , c_i , and a_{ij} of a Runge–Kutta method satisfy $B(p)$, $C(\eta)$, and $D(\zeta)$ with $p \leq \eta + \zeta + 1$ and $p \leq 2\eta + 2$, then the method is of order p .*

According to Theorem 1, the modified coefficients b_i , c_i , and a_{ij} will affect the order of the method. Because of the modified order conditions which have to be solved, it is difficult to construct the SDIRK method or SIRK method with high stages and high order. Different from the SDIRK method and SIRK method, we do not change the coefficient matrix \mathbf{A} , node coefficient c_i , and weight coefficient b_i of IRK methods, which will save a lot of work. We just need to find a mature IRK method with good stability and high accuracy and then try to reduce the computation cost of the method. The method we consider is the Gauss implicit Runge–Kutta (Gauss-IRK) method which was introduced by Butcher [22].

Assume the numerical solution y_n is equal to the exact solution $y(x_n)$ at the point x_n . Then,

$$l_{n+1} = y(x_{n+1}) - y_{n+1} = o(h^{p+1}), \quad (19)$$

where l_{n+1} is the local truncation error. The accuracy of a numerical solution can be interpreted in terms of the local truncation error. A numerical solution method is said to be of order p as in (19), where p is an integer and $p > 0$.

The Gauss-IRK method is a collocation method based on the Gaussian quadrature formulas. Since the Gaussian quadrature formulas for point s have the algebraic accuracy of $2s-1$, the truncation error is $l_{n+1} = y(x_{n+1}) - y_{n+1} = o(h^{2s+1})$, and it satisfies the order conditions in Theorem 1,

so the order the implicit method can be obtained is $2s$. The order of numerical methods is an important indicator to measure the accuracy of the method. Generally, the higher the order, the higher the accuracy of the numerical method. The s stage Gauss-IRK method can be obtained is $2s$, and the SDIRK and SIRK methods have the maximum attainable order $s+1$. It is clear that the Gauss-IRK method has higher order and higher accuracy, which is the main reason why we choose this method. Even so, when this method is applied to a large system simulation, it is hesitant because it is fully implicit with a large computational cost. This is one disadvantage that contributes to the expensive computation cost by the Gauss-IRK method with a full coefficient matrix \mathbf{A} , and the computation cost grows increasingly expensive for high stage methods and high dimensional systems. So, it is necessary to reduce its computational cost. Since the transformation method of SIRK is relatively complex, we proposed a simple method to reduce the computation cost in this paper.

Firstly, the method introduced in this paper retains the coefficients b_i , c_i , and a_{ij} of the Gauss-IRK method. This avoids the same result as the SIRK method, which changes the order condition and reduces the order of the method due to the change in the method coefficients. Then, we let the two sides of equation (6) be pre-multiplied by $\mathbf{A}^* \otimes \mathbf{I}_N$ in the process of the method implementation, and this operation has no effect on the equation results. Equation (6) becomes

$$\mathbf{M}_B \Delta\mathbf{Z}^{(k)} = -(\mathbf{A}^* \otimes \mathbf{I}_N) \mathbf{Z}^{(k)} + h(\mathbf{A}^* \mathbf{A} \otimes \mathbf{I}_N) \mathbf{F}(\mathbf{Z}^{(k)}), \quad (20)$$

where

$$\mathbf{M}_B = (\mathbf{A}^* \otimes \mathbf{I}_N - h\mathbf{A}^* \mathbf{A} \otimes \mathbf{J}), \quad (21)$$

where $\mathbf{A}^* = (\mathbf{A}_{ij})$ ($i, j = 1, 2, \dots, s$) is the adjugate matrix of the coefficient matrix \mathbf{A} , and $\mathbf{A}^* \mathbf{A} = |\mathbf{A}| \mathbf{I}_s$.

This is similar in form to the transformation method of SIRK, and we can see that a simple transformation method is not as complicated as the transformation of SIRK. \mathbf{A}^* can be easily found by \mathbf{A} , and there is no need to transform vector $\mathbf{F}(\mathbf{Z}^{(k)})$. Newton iterative update $\Delta\mathbf{Z}^{(k)}$ also can be obtained directly from equation (8) without repeated transform. Furthermore, \mathbf{M}_B will be a matrix as the following form:

$$\begin{bmatrix} \mathbf{A}_{11} \mathbf{I}_N - h|\mathbf{A}| \mathbf{J} & \mathbf{A}_{21} \mathbf{I}_N & \cdots & \mathbf{A}_{s1} \mathbf{I}_N \\ \mathbf{A}_{12} \mathbf{I}_N & \mathbf{A}_{22} \mathbf{I}_N - h|\mathbf{A}| \mathbf{J} & \cdots & \mathbf{A}_{s2} \mathbf{I}_N \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{1s} \mathbf{I}_N & \mathbf{A}_{2s} \mathbf{I}_N & \cdots & \mathbf{A}_{ss} \mathbf{I}_N - h|\mathbf{A}| \mathbf{J} \end{bmatrix}. \quad (22)$$

$\mathbf{A}_{ij} \mathbf{I}_N$ is

$$\begin{bmatrix} \mathbf{A}_{ij} & 0 & \cdots & 0 \\ 0 & \mathbf{A}_{ij} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_{ij} \end{bmatrix}. \quad (23)$$

The main achievement is that the matrix \mathbf{M}_B is a banded diagonal matrix with bandwidth $N(s-1)$ rather than a full

coefficient matrix of $sN \times sN$. So we call it the banded implicit Runge–Kutta (BIRK) method.

Theorem 2 (see [23, 24]). *If the n order banded matrix \mathbf{M} with the upper bandwidth q and the lower bandwidth p has a LU factorization $\mathbf{M} = \mathbf{L}\mathbf{U}$, then \mathbf{L} is the lower triangular matrix with lower bandwidth p , and \mathbf{U} is the upper triangular matrix with upper bandwidth q .*

According to Theorem 2, the matrix \mathbf{M}_B is decomposed into an upper triangular matrix with bandwidth $N(s-1)$ and lower triangular matrix with bandwidth $N(s-1)$ by making full use of the characteristic of banded matrix LU factorization. Therefore, compared with the traditional Gauss-IRK method, the BIRK method reduces the computational complexity of the LU factorization and back substitution to the Newton update in iteration. The larger the system dimension N , the more effective the method is. In spite of this, the s stage BIRK method can obtain the maximum order $2s$ because the coefficients of the Gauss-IRK method are retained. Therefore, the accuracy of the BIRK method are not reduced compared with the Gauss-IRK method. Furthermore, the BIRK method is easier to implement programmatically than SDIRK and SIRK.

3.3. Stability of BIRK Methods. When each IRK method is applied to the test equation,

$$y' = \lambda y. \quad (24)$$

A straightforward computation

$$y_1 = R(h\lambda)y_0. \quad (25)$$

Solving this and inserting equations (2) and (3) leads to

$$R(z) = 1 + zb^T(\mathbf{I} - z\mathbf{A})^{-1}e, \quad (26)$$

where $z = h\lambda$, $b^T = (b_1, \dots, b_s)$, and $e = (1, \dots, 1)^T$. $R(z)$ is called the stability function of the IRK method.

The equation (26) shows that the stability function of the IRK method is related to the coefficients of the method, and the transformation method in the process of implementation has no effect on its stability function. Because the BIRK method retains the coefficients of the Gauss-IRK method, and the stability function of the BIRK method is the same as that of the Gauss-IRK method. Ehle shows that the stability function of the s stage $2s$ order Gauss-IRK method is (s, s) -Padé approximation and the method is A-stable [25]. Therefore, the s stage BIRK method of order $2s$ is A-stable.

4. Experiments and Results

This section describes the implementation of some IRK methods on the MATLAB R2017b in the computer with a CPU i5-4590, 3.3 GHz, and RAM 6 GB. Take stiff systems (27)–(29) as three simulation problems:

$$\begin{cases} y_1' = -0.1y_1 + 49.9y_2, y_1(0) = 2, \\ y_2' = -40y_2, y_2(0) = 1, \\ y_3' = 70y_2 - 300y_3, y_3(0) = 2, \end{cases} \quad t \in [0, 5], \quad (27)$$

$$\begin{cases} y_1' = -y_1 + y_2, y_1(0) = 1, \\ y_2' = -100y_1 - y_2, y_2(0) = 0, \\ y_3' = -100y_3 + y_4, y_3(0) = 1, \\ y_4' = -10000y_3 - 100y_4, y_4(0) = 0, \end{cases} \quad t \in [0, 5], \quad (28)$$

$$\begin{cases} y_1' = -10y_1 + 100y_2, y_1(0) = 1, \\ y_2' = -100y_1 - 10y_2, y_2(0) = 1, \\ y_3' = -4y_3, y_3(0) = 1, \\ y_4' = -y_4, y_4(0) = 1, \\ y_5' = -0.5y_5, y_5(0) = 1, \\ y_6' = -0.1y_6, y_6(0) = 1, \end{cases} \quad t \in [0, 5]. \quad (29)$$

Several IRK methods with stage size $s = 2$ and A-stable are used to solve the stiff systems (27)–(29). These three stiff systems have different dimensions. The test results are shown in Tables 1–3.

In the Tables 1–2, the test results of systems (27) and (28) are presented. When the step size $h = 0.01, 0.005, 0.001$, the maximum absolute error of the BIRK method and Gauss-IRK method is the same and the maximum absolute error of the SDIRK method and SIRK method is the same, but the maximum absolute error of the BIRK method and Gauss-IRK method is smaller than that of the SDIRK method and SIRK method. When step size $h = 0.01$, the maximum absolute error of the BIRK method and GIRK method is similar to that of the SDIRK method and SIRK method in the case of $h = 0.005$. When step size $h = 0.005$, the maximum absolute error of the BIRK method and Gauss-IRK method is similar to that of the SDIRK method and SIRK method in the case of $h = 0.001$. In addition, the SDIRK method has the shortest computation time, although it does not have the least number of Newton iterations among the four IRK methods. This is because the SDIRK method has the smallest computation cost in LU factorization of the matrix and back substitution to the Newton update. Although the SIRK method has the same computation cost in LU factorization of the matrix and back substitution to the Newton update, it needs a complex transformation process. It has no advantage for three-dimensional systems (27) and four-dimensional systems (28), and the SIRK method has the largest number of Newton iterations, so it always needs the longest computation time. The Newton iteration times of the BIRK method and Gauss-IRK method differ little or sometimes the same, but they are less than the number of Newton iterations of the SDIRK method and the SIRK method. The calculation time of the BIRK method is slightly longer than that of the Gauss-IRK method, but shorter than that of the SIRK method. This is because the BIRK method also needs a simple transformation. For the three-dimensional system (27) and the four-dimensional system (28), the reduction of computing in the LU factorization of the matrix and back substitution to

TABLE 1: The test results for system (27).

		Maximum absolute error	Number of Newton iterations	Time
System (24)	Step size $h = 0.01$			
	BIRK	4.696875980934exp-02	999	0.0818
	SDIRK	2.951335592737exp-01	1002	0.0771
	SIRK	2.951335592737exp-01	1098	0.1158
	Gauss-IRK	4.696875980934exp-02	999	0.0801
	Step size $h = 0.005$			
	BIRK	4.632141120340exp-03	1366	0.1015
	SDIRK	8.059127347433exp-02	1996	0.0941
	SIRK	8.059127347433exp-02	2100	0.1867
	Gauss-IRK	4.632141120340exp-03	1364	0.0949
	Step size $h = 0.001$			
	BIRK	7.163047590741exp-06	5120	0.2795
	SDIRK	1.163100888497exp-03	5139	0.2143
SIRK	1.163100888497exp-03	5276	0.4993	
Gauss-IRK	7.163047590741exp-06	5120	0.2769	

TABLE 2: The test results for system (28).

		Maximum absolute error	Number of Newton iterations	Time
System (25)	Step size $h = 0.01$			
	BIRK	1.899433882193exp-01	2499	0.1194
	SDIRK	5.402682507296	2933	0.0929
	SIRK	5.402682507296	4076	0.2534
	Gauss-IRK	1.899433882193exp-01	2493	0.1068
	Step size $h = 0.005$			
	BIRK	1.231643373468exp-02	4254	0.1899
	SDIRK	9.714581002096exp-01	4666	0.1325
	SIRK	9.714581002096exp-01	5188	0.3267
	Gauss-IRK	1.231643373468exp-02	4174	0.1725
	Step size $h = 0.001$			
	BIRK	2.003231104197exp-05	15604	0.3766
	SDIRK	1.124295167250exp-02	16546	0.3154
SIRK	1.124295167250exp-02	18473	1.0362	
Gauss-IRK	2.003231104197exp-05	15526	0.3651	

TABLE 3: The test results for system (29).

		Maximum absolute error	Number of Newton iterations	Time
System (26)	Step size $h = 0.01$			
	BIRK	6.745545867356exp-03	509	0.0700
	SDIRK	2.431772336420exp-01	704	0.0692
	SIRK	2.431772336420exp-01	837	0.1206
	Gauss-IRK	6.745545867356exp-03	702	0.0829
	Step size $h = 0.005$			
	BIRK	4.451608892082exp-04	1003	0.0873
	SDIRK	4.724422976707exp-02	1004	0.0834
	SIRK	4.724422976707exp-02	1430	0.1654
	Gauss-IRK	4.451608892082exp-04	1005	0.0990
	Step size $h = 0.001$			
	BIRK	7.399547229947exp-07	5000	0.2770
	SDIRK	4.679447423339exp-04	5000	0.2294
SIRK	4.679447423341exp-04	5009	0.5155	
Gauss-IRK	7.399547229947exp-07	5000	0.3121	

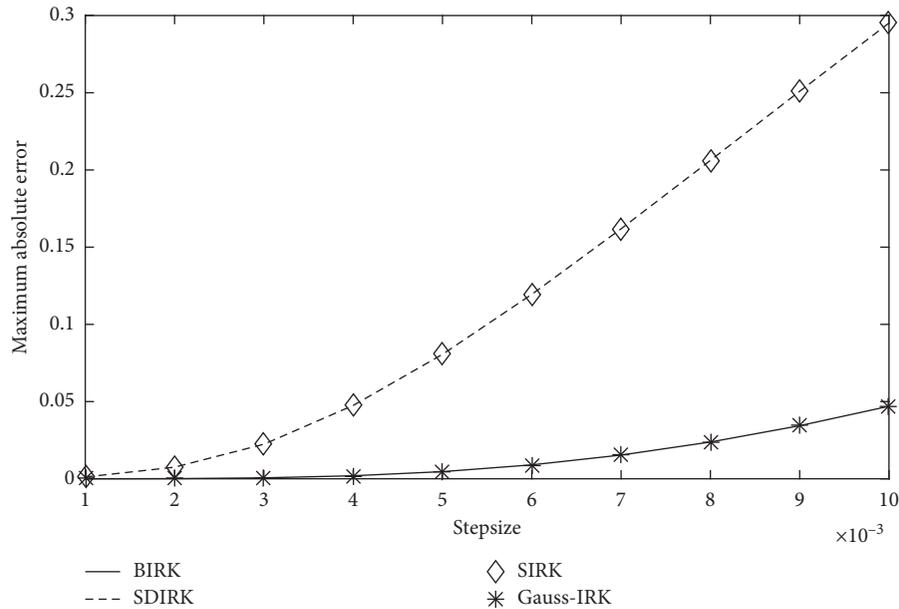


FIGURE 1: Relation curve of maximum absolute error of stiff system (27) with step size. Step size is from 0.001 to 0.1.

the Newton update is not obvious, so the computation time is slightly more than that of the Gauss-IRK method.

Table 3 shows the test results of system (29). The same situation as Tables 1–2 is that the maximum absolute error of the BIRK method and Gauss-IRK method is the same and the maximum absolute error of the SDIRK method and SIRK method is the same, and the maximum absolute error of the BIRK method and Gauss-IRK method is smaller than that of the SDIRK method and SIRK method as the step size $h = 0.01, 0.005, 0.001$. When step size $h = 0.005$, the maximum absolute error of the BIRK method and Gauss-IRK method is similar to that of the SDIRK method and SIRK method in the case of $h = 0.001$. In Table 3, the SDIRK method still has the minimum computation time. Due to the complexity of the transformation process of the SIRK method, there is no obvious computing advantage for the 6-dimensional system (29), and the number of Newton iterations is the largest, so the SIRK method also needs the most computing time compared with the other three methods. It is possible that the SIRK method needs to be presented in a system of larger dimensions to show that it has the advantage of reducing the cost of computation. However, the BIRK method has the least number of Newton iterations compared with the other three methods, and the BIRK method also takes less computing time than the Gauss-IRK method in the case of step size $h = 0.01, 0.005$. When $h = 0.001$, only one Newton iteration is needed for each step, so the BIRK method and Gauss-IRK method have the same number of iterations. Although the number of Newton iterations is the same, the calculation time of the BIRK method is still shorter than that of the Gauss-IRK method, which indicates that the BIRK method has the advantage of reducing the cost of computation.

To compare the accuracy of the four IRK methods, we tested the maximum absolute error of the three stiff systems

(27)–(29) as the step size increased from 0.001 to 0.01. The results are shown in Figures 1–3.

As can be seen from Figures 1–3, the maximum absolute error of the stiff systems (27)–(29) also gradually increases with the increase in integration step size. When the integral step size is the same, the maximum absolute error of the BIRK method is the same as the maximum absolute error of the GIRK method, and the maximum absolute error of the SDIRK method is the same as the maximum absolute error of the SIRK method, but the maximum absolute error of the BIRK method and the GIRK method is always smaller than the maximum absolute error of the SDIRK method and the SIRK method. Therefore, the BIRK method and the Gauss-IRK method have a higher accuracy when the step size is fixed.

We know that the order of the BIRK method and the Gauss-IRK method for 2 stage and A-stable method are 4, but the SIRK method and the SDIRK are 3. The accuracy of the numerical method with higher order should be better for the same stage IRK methods. Since the highest order obtainable by SIRK and SDIRK methods is $p = s + 1$ and the highest order obtainable by BIRK and Gauss-IRK is $p = 2s$, if we add the same stages to these numerical methods, the order of SIRK and SDIRK methods will be much lower than that of BIRK and Gauss-IRK methods, which means that the accuracy of SIRK and SDIRK methods will be much lower than that of BIRK and Gauss-IRK methods. In the other words, for the same stage SDIRK method and the SIRK method, if they want to achieve the accuracy similar to the BIRK method and the Gauss-IRK method, they need to greatly reduce the step size, which will increase the calculation cost or may cost more computation time than the BIRK method and the Gauss-IRK method. For another, compared with the Gauss-IRK method, the computation cost in the LU factorization and back substitution to the Newton update in iteration of the BIRK method are less as

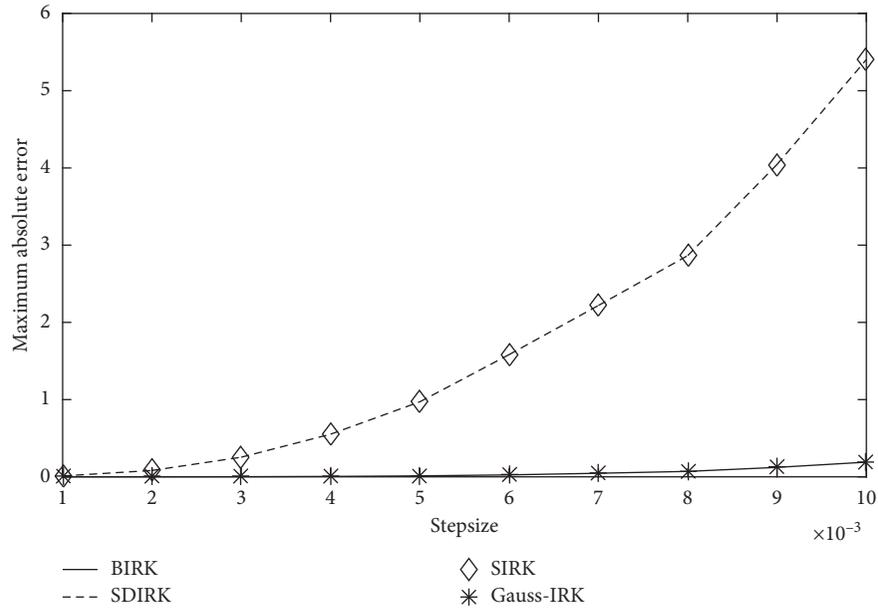


FIGURE 2: Relation curve of maximum absolute error of stiff system (28) with step size. Step size is from 0.001 to 0.1.

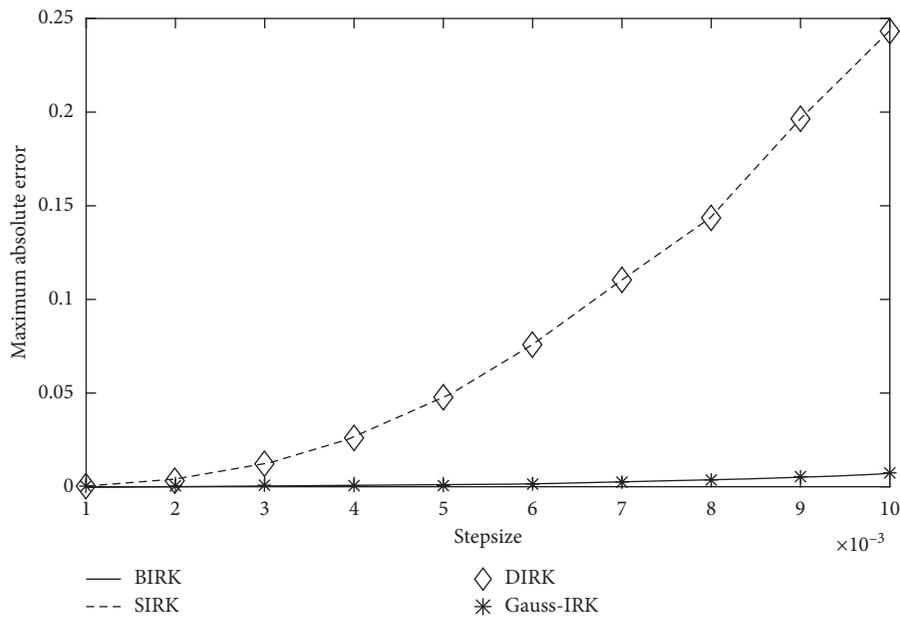


FIGURE 3: Relation curve of maximum absolute error of stiff system (29) with step size. Step size is from 0.001 to 0.1.

mentioned in the previous section. Therefore, the BIRK method is more effective than the other three methods, with the advantages of high accuracy, high stability, and low computational cost.

5. Discussion and Conclusions

Although the SDIRK method and the SIRK method have lower accuracy, the cost of LU factorization and back substitution in the Newton iteration is greatly reduced. So, the SDIRK method and the SIRK method are suitable for the simulation of large systems with low precision requirements

when the number of Newton iterations is similar. The traditional Gauss-IRK method is suitable for lower dimensional systems. Because the computational cost for low-dimensional systems is small, the Gauss-IRK method has higher accuracy than the SDIRK method and SIRK, and the transformation process of BIRK and SIRK is omitted. However, in some practical systems such as chemical reactions, electronic networks, and especially automatic control systems, systems are often large and complex with many factors. The BIRK method has higher accuracy than SDIRK and SIRK and less cost of computation than the Gauss-IRK method. Therefore, the BIRK method not only

satisfies the requirements of high accuracy and high stability for solving the stiff differential equations in large system simulation but also has lower computation cost. It becomes more effective while the dimension of the system is increased, and to some extent, unnecessary costs are avoided. So, the BIRK method has broad application prospects.

In practical applications, we certainly hope to combine the characteristics of specific problems, make full use of the differences between different numerical methods, choose a more appropriate method, and strive to get the best possible results.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant no. 61773008.

References

- [1] E. Hairer, G. Wanner, and S. P. Nørsett, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, pp. 2–11, Springer-Verlag, Berlin, China, 1996.
- [2] V. Nirmala, V. Parimala, and P. Rajarajeswari, “Application of Runge-Kutta method for finding multiple numerical solutions to intuitionistic fuzzy differential equations,” *Journal of Physics: Conference Series*, vol. 1139, no. 1, Article ID 012012, 2018.
- [3] B. Su, X. Tuo, and L. Xu, “Two modified symplectic partitioned Runge-Kutta methods for solving the elastic wave equation,” *Journal of Geophysics and Engineering*, vol. 14, no. 4, pp. 811–821, 2017.
- [4] I. E. Athanasakis, E. P. Papadopoulou, and Y. G. Saridakis, “Runge-Kutta and Hermite Collocation for a biological invasion problem modeled by a generalized Fisher equation,” *Journal of Physics: Conference Series*, vol. 490, no. 1, Article ID 012133, 2014.
- [5] J. H. Tian, J. Zou, Y. S. Wang, J. Liu, J. Yuan, and Y. Zhou, “Simulation of bipolar charge transport with trapping and recombination in polymeric insulators using Runge-Kutta discontinuous Galerkin method,” *Journal of Physics D: Applied Physics*, vol. 41, no. 19, Article ID 195416, 2008.
- [6] S. Chowdhury and P. K. G. Das, *Numerical Solutions of Initial Value Problems Using Mathematica*, pp. 1–11, Morgan & Claypool Publishers, Saint-Raphaël, France, 2018.
- [7] A. Samsudin, N. Rosli, and A. N. Ariffin, “Stability analysis of explicit and implicit stochastic Runge-Kutta methods for stochastic differential equations,” *Journal of Physics: Conference Series*, vol. 890, no. 1, Article ID 012084, 2017.
- [8] V. Kazemi-Kamyab, A. H. Van Zuijlen, and H. Bijl, “Analysis and application of high order implicit Runge-Kutta schemes to collocated finite volume discretization of the incompressible Navier-Stokes equations,” *Computers & Fluids*, vol. 108, no. 15, pp. 107–115, 2015.
- [9] A. Jameson, “Evaluation of fully implicit Runge Kutta schemes for unsteady flow calculations,” *Journal of Scientific Computing*, vol. 73, no. 2-3, pp. 819–852, 2017.
- [10] K. Liu, X. Liao, and Y. Li, “Parallel simulation of power systems transient stability based on implicit Runge-Kutta methods and W-transformation,” *Electric Power Components and Systems*, vol. 45, no. 20, pp. 2246–2256, 2017.
- [11] V. Antohe and I. Gladwell, “Performance of Gauss implicit Runge-Kutta methods on separable Hamiltonian systems,” *Computers & Mathematics with Applications*, vol. 45, no. 1–3, pp. 481–501, 2003.
- [12] J. M. Franco, I. Gómez, and L. Rández, “SDIRK methods for stiff ODEs with oscillating solutions,” *Journal of Computational and Applied Mathematics*, vol. 81, no. 2, pp. 197–209, 1997.
- [13] J. M. Franco and I. Gomez, “Two three-parallel and three-processor SDIRK methods for stiff initial-value problems,” *Journal of Computational and Applied Mathematics*, vol. 87, no. 1, pp. 119–134, 1997.
- [14] W. Liao and Y. Yan, “Singly diagonally implicit Runge-Kutta method for time-dependent reaction-diffusion equation,” *Numerical Methods for Partial Differential Equations*, vol. 27, no. 6, pp. 1423–1441, 2011.
- [15] M. K. Wing, N. Senu, and M. Suleiman, “A five-stage singly diagonally implicit Runge-Kutta-Nyström method with reduced phase-lag,” *AIP Conference Proceedings*, vol. 1482, no. 1, pp. 315–320, 2012.
- [16] O. Y. Ababneh and R. Ahmad, “Construction of third-order diagonal implicit Runge-Kutta methods for stiff problems,” *Chinese Physics Letters*, vol. 26, no. 8, Article ID 080503, 2009.
- [17] J. C. Butcher and P. Chartier, “A generalization of singly-implicit Runge-Kutta methods,” *Applied Numerical Mathematics*, vol. 24, no. 2-3, pp. 343–350, 1997.
- [18] K. Burrage, J. C. Butcher, and F. H. Chipman, “An implementation of singly-implicit Runge-Kutta methods,” *BIT Numerical Mathematics*, vol. 20, no. 3, pp. 326–340, 1980.
- [19] K. Burrage, “A special family of Runge-Kutta methods for solving stiff differential equations,” *BIT Numerical Mathematics*, vol. 18, no. 1, pp. 22–41, 1978.
- [20] R. Alexander, “Diagonally implicit Runge-Kutta methods for stiff O. D. E.’s,” *SIAM Journal on Numerical Analysis*, vol. 14, no. 6, pp. 1006–1021, 1977.
- [21] D. J. L. Chen, “The efficiency of Singly-implicit Runge-Kutta methods for stiff differential equations,” *Numerical Algorithms*, vol. 65, no. 3, pp. 533–554, 2014.
- [22] J. C. Butcher, “Implicit Runge-Kutta processes,” *Mathematics of Computation*, vol. 18, no. 85, pp. 50–64, 1964.
- [23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, London, UK, vol. 3, pp. 152–156, 2012.
- [24] J. W. Demmel, *Applied numerical linear algebra*, SIAM, Philadelphia, PA, USA, vol. 56, pp. 79–82, 1997.
- [25] B. L. Ehle, “On padé approximations to the exponential function and a-stable methods for the numerical solution of initial value problems,” Doctoral dissertation, University of Waterloo, Waterloo, Canada, 1969.



Hindawi

Submit your manuscripts at
www.hindawi.com

