Numer Algor (2014) 65:533–554
DOI 10.1007/s11075-013-9797-5

ORIGINAL PAPER

The efficiency of Singly-implicit Runge-Kutta methods for stiff differential equations

D. J. L. Chen

Received: 18 April 2013 / Accepted: 1 November 2013 / Published online: 26 November 2013

Abstract Singly-implicit Runge-Kutta methods are considered to be good candidates for stiff problems because of their good stability and high accuracy. The existing methods, SIRK (Singly-implicit Runge-Kutta), DESI (Diagonally Extendable Singly-implicit Runge-Kutta), ESIRK (Effective order Singly-implicit Rung-Kutta) and DESIRE (Diagonally Extended Singly-implicit Runge-Kutta Effective order) methods have been shown to be efficient for stiff differential equations, especially for high dimensional stiff problems. In this paper, we measure the efficiency for the family of singly-implicit Runge-Kutta methods using the local truncation error produced within one single step and the count of number of operations. Verification of the error and the computational costs for these methods using variable stepsize scheme are presented. We show how the numerical results are effected by the designed factors: additional diagonal-implicit stages and effective order.

Keywords Singly-implicit Runge-Kutta methods · Diagonally-implicit stages · Effective order

1 Introduction

Consider the standard initial value problem in autonomous form with dimension N

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

The author is also an adjunct professor at the Department of Finance and Risk Management, Ling Tung University and the author's work is partly supported by the National Science Council of Taiwan

D. J. L. Chen (✉)
Department of Information Technology, Ling Tung University, Taichung, Taiwan
e-mail: dchen@mail.ltu.edu.tw

If h is the stepsize, then for each integration step, an s-stage implicit Runge Kutta (IRK) method is of the form

$$Y_i = y_0 + h\sum_{j=1}^{s} a_{ij} f(Y_j), \qquad i = 1, \ldots, s, \tag{2}$$
$$y_1 = y_0 + h\sum_{i=1}^{s} b_i f(Y_i),$$

or, in a more compact notation,

$$Y = e \otimes y_0 + h(A \otimes I_s)F, \tag{3}$$
$$y_1 = y_0 + h(b^T \otimes I_s)F,$$

where $\otimes$ is the Kronecker product and I_s is the $s \times s$ identity matrix, $e = [1, 1, \ldots, 1]^T$ with dimension s, $b = [b_1, b_2, \ldots, b_s]^T$,$Y = [Y_1, Y_2, \ldots, Y_s]^T$ and

$$F = [f(Y_1), f(Y_2), \ldots, f(Y_s)].^T$$

We note that the dimensions of Y and F is $s \times N$ because the dimensions of Y_i and $f(Y_i)$ is N. Although implicit Runge-Kutta (IRK) methods have good stability and accuracy for stiff problems, unlike explicit Runge-Kutta methods, a complicated iteration scheme for the solutions of the internal stages is necessary. The computation of the stage values is typically the most expensive component in their implementation. Since the dimension of the initial value system (1) is N, for an s-stage Runge-Kutta method, the nonlinear stage value system (3) can be written as

$$Z = h(A \otimes I_N)F, \tag{4}$$

where Z is the sN dimensional vector made up from subvectors $Y - ey_0$. Because implicit Runge-Kutta methods are designed to solve stiff problems, functional iteration is not feasible for solving (4) and the "Newton-Raphson" algorithm is used instead. In this case, the linear system of the Newton method can be written as

$$M(\triangle Z) = -Z + h(A \otimes I_N)F, \tag{5}$$

where M is the Jacobian matrix, $\triangle Z$ is the Newton update. $Z + \triangle Z$ becomes the next approximation. Hence, in an iteration of a full Newton-Raphson scheme, for each iteration, the total computation cost will include: evaluation of F, evaluation of $-Z + h(A \otimes I_N)F$, LU factorization of the Jacobian matrix M and back substitution to yield the Newton update.

The cost of each iteration grows rapidly as s or N increases. The LU factorizations and the back substitutions need approximately s^3N^3 and s^2N^2 operations respectively. For high order methods and high dimensional systems, the cost is increasingly expensive. In practice, the cost of the LU factorization can be lowered because the Jacobian matrix,

$$M = I_s \otimes I_N - hA \otimes J, \tag{6}$$

where $J = \partial f/\partial y$, can be kept unchanged over several steps as the Jacobian J, required for each stage, can also be kept constant for all stages or even for several steps.

The aim of reducing the computational cost for IRK methods leads to the so-called "DIRK" methods [1]. Instead of using a full coefficient matrix A, one can use a lower triangular matrix A for the method. In this case, the nonlinear system (2) can be solved stage by stage sequentially. The total operations can be reduced from $s^3N^3 + s^2N^2$ to $sN^3 + sN^2$. This is a considerable saving, especially for large dimension systems. For $i = 1, 2, \ldots, s$, the N-dimensional iteration matrix is of the form

$$I_N - ha_{ii}\frac{\partial f}{\partial y}.$$

If all diagonal elements, a_{ii}, of A are equal(SDIRK), then the Jacobian $\partial f/\partial y$ and so the iteration matrix (if h is constant or is only changed moderately), is constant over all stages. The cost can be reduced further to $N^3 + sN^2$.

The computational cost has been cut down for IRK methods by using DIRK methods, but the accuracy and stability are effected because of the simplification of the coefficient matrix A. The fact that the maximum attainable order for an A-stable s-stage IRK method is $2s$, whereas there are only four cases: $s = 1, 2, 3, 5$, for an s-stage SDIRK method to have the maximum attainable order $s + 1$ [15]. This is one of the main reasons why the existing codes based on SDIRK methods cannot compete with the codes based on IRK methods for many stiff problems. The low stage order is a drawback of many IRK methods. Another important disadvantage for DIRK methods is the lack of accuracy of the stage approximations. Unlike multistep methods, the multistage methods have to calculate several stages before they advance the initial values to the numerical solutions for each integration step. For IRK methods, the output values are obtained from the combination of the input values and the derivatives of the internal stages, the accuracy of the stage values effect the accuracy of the numerical solution. Some investigations have shown that for some stiff problems, the accuracy of the numerical solutions obtained is not the expected accuracy, and is closely related to the accuracy of the internal stage approximations (order reduction phenomenon). The very poor accuracy of the stage value for DIRK methods is therefore a serious handicap. In fact, many existing famous IRK methods also have similar drawbacks. Therefore, the search for high accuracy (for both output value and stage value) and high stability while retaining computational advantages lead to the family of "Singly-Implicit Runge-Kutta" (SIRK) methods ([2],[5]) which are our main focus in this paper.

The accuracy of a numerical solution can be interpreted in terms of the difference between the exact solution and the numerical solution. That is, a numerical solution of the nth step y_n is said to be of order p at the integrated point x_n if

$$y(x_n) - y_n = O(h^{p+1}),$$

where $y(x_n)$ is the exact solution of the nth step. Due to the implementation and accuracy considerations, the design of an s-stage SIRK method is based on the following two assumptions:

(1) the coefficient matrix A has a one point spectrum property, and is written as $\sigma(A) = \{\lambda\}$;
(2) the order of the stage values and the order of the output value each equals s.

The reason for designing SIRK methods with the property $\sigma(A) = \{\lambda\}$ is that the A matrix can be transformed to a Jordan canonical form with the same diagonal elements and the same bi-diagonal elements. By using some transformation matrixes, we can still have the iteration matrix $I_N - h\lambda\partial f/\partial y$ for each stage iteration. Hence, the overall computational cost is similar to that of SDIRK methods but also includes the extra transformation costs $3s^2N$ operations. It turns out that A-stable s-stage SIRK methods can be obtained for $s = 1, 2, \ldots, 6, 8$ and have highly accurate stage values (same order as the output value) without too much extra computational cost. The improvement achieved is more apparent for large dimensional systems. Furthermore, for order higher than 8, almost A-stable SIRK methods are available. The existing variable order code *STRIDE* [3] based on SIRK methods from order 1 up to order 14 has been shown to be efficient for large problems, especially for hyperbolic partial differential equations. Because the order of the stage approximations is equal to the stage number for SIRK methods, the choices of the abscissae for SIRK methods are limited. The abscissae $c_1, c_2, \ldots, c_s$ of an s-stage SIRK method must satisfy

$$c_i = \lambda\xi_i, \qquad \xi_i \quad \text{is zero number } i \text{ of the } s \text{ degree Laguerre polynomial} \quad L_s(x),$$

for all $i = 1, 2, \ldots, s$. For $s > 2$, some of the abscissae are greater than 1. It causes some difficulties in the numerical behaviour of algorithms based on these methods, especially for nonlinear problems. This difficulty was partly solved by a combination of the SDIRK and SIRK methods. Using SIRK methods to retain high stage order and using the SDIRK methods to obtain some free parameters. These methods are the so-called "DESI" (Diagonally Extended Singly Implicit Rung-Kutta) methods [7]. The coefficient matrix A of a DESI method has a singly-implicit block and a diagonally-implicit block. The method still has a single-implicitness property with the order and stage order equal to the stage number. Because of these additional stages, DESI methods have more freedom in choosing their coefficients. In particular, the abscissae of the diagonal stages can be chosen to reduce the local truncation errors of DESI methods. The magnitude of this reduction depends on the number of the additional diagonal stages. It has been shown that the existing variable order *DESI* code [14], based on the DESI formulae with three additional stages are more efficient than SIRK methods and competitive with the BDF methods even for many small problems. Because of high accuracy, good stability and the implementation advantage, DESI methods have potential to also solve PDEs and DAEs efficiently.

Another approach to removing the restriction on the abscissae for SIRK methods is the use of "effective order" [4] in which the difficulty that no s-stage explicit Runge Kutta method can attain order s when $s \geq 5$ is overcome(so-called Butcher's barrier).

The effective order of SIRK methods is defined in terms of the ability of the method to conform to some perturbations of the exact solution, $\psi(y(x))$. The mapping ψ can be a notional one-step method. Thus, if ϕ denotes the specified method, for ϕ to have effective order p, it is necessary that $||\phi(\psi(y(x_0))) - \psi(y(x_0 + h))|| = O(h^{p+1})$. These perturbations introduce some free parameters, so the limitation on the abscissae is removed. The abscissae for the effective order SIRK methods (ESIRK) ([8], [9]) can be chosen arbitrarily as long as they are distinct. As a matter of fact, ESIRK methods are first introduced to overcome the Butcher's barrier for the ERK methods. It is noted that ESIRK methods need a starting procedure to produce the perturbed initial value. This is usually done by using the associated classical SIRK methods. Because all the stage approximations for an s-stage ESIRK method are designed to be of the classical order s, an approximation to the exact solution can be obtained at any time. In addition to removing the restriction on abscissae, a remarkable outcome for ESIRK methods is the reduction of the local error when these methods are applied using a special stepsize changing scheme [10].

Adding additional stages or applying effective order to SIRK methods seems to be a successful generalization of the classical SIRK methods. However, the existing DESI formulae suffer from the fact that some of their abscissae are greater than one if $p > 4$. On the other hand, the truncation error for the ESIRK methods is considerly higher than for DESI methods. The combination of these ideas leads to DESIRE (Diagonally Extended Singly Implicit Runge-Kutta Effective order) methods. In section 2, overviews for SIRK methods and DESI methods are presented. ESIRK methods and DESIRE methods are discussed in sections 3 and 4 respectively. The efficiency measurement and numerical comparison among the family of singly-implicit methods: SIRK, DESI, ESIRK and DESIRE methods are given in the final section.

2 SIRK methods, DESI methods

2.1 SIRK methods

Let T be a non-singular matrix such that $T^{-1}AT = \bar{A}$ is the Jordan canonical form of A. In [5], it was shown how to use T to transfer (5), (6) to the following system,

$$[(I_s \otimes I_N) - h\bar{A} \otimes J]\triangle\bar{Z} = -\bar{Z} + h(\bar{A} \otimes I_N)\bar{F},$$

where

$$\begin{aligned} \bar{Z} &= (T^{-1} \otimes I_N)Z, \\ \bar{F} &= (T^{-1} \otimes I_N)F, \\ \triangle Z &= (T \otimes I_N)\triangle\bar{Z}. \end{aligned}$$

Because of this transformation, the operations for the LU factorizations will become proportional to dN^3, d is the number of distinct eigenvalues of A, and the operations for the back substitutions are reduced to sN^2. For large systems, these are reasonable savings. But there are three transformation costs (s^2N) for each $\triangle\bar{Z}$, $\bar{Z}$

and the update $\triangle Z$ need to take into account. From this approach, one can see the biggest reduction in the computational cost occurs when the coefficient matrix A has a one point spectrum. The factorization cost will then reduce to N^3. Methods with a one-point spectrum property are called "SIRK"(singly-implicit Runge-Kutta) methods.

To avoid order reduction and to construct higher order methods with an error estimator, high stage order methods are proposed for solving stiff problems. In addition to the one-point spectrum property, the standard s-stage SIRK methods have stage order and order each equal to s. The coefficients of the method satisfy the stage order conditions $C(s)$,

$$\sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{1}{k} c_i^k, \quad i = 1, 2 \ldots, s, \quad k = 1, 2 \ldots, s, \tag{7}$$

are equivalent to

$$\sum_{j=1}^{s} a_{ij}\, p(c_j) = \int_0^{c_i} p(c) dc, \quad i = 1, 2 \ldots, s, \tag{8}$$

where $p(x)$ is any polynomial with degree $\leq s-1$. Let $c^k, k = 0, 1, \ldots$, denote the component-wise k-th power of c, and from

$$A^k e = A^{k-1}(Ae) = A^{k-1} c = \frac{1}{2} A^{k-2} c^2 = \cdots = \frac{1}{(k-1)!} A c^{k-1} = \frac{c^k}{k!},$$

by the singly-implicitness of A and the Cayley-Hamilton theorem, we have

$$0 = (A - \lambda I)^s e = \sum_{k=0}^{s} \binom{s}{k} (-1)^k \lambda^k A^{s-k} e = \sum_{k=0}^{s} \binom{s}{k} \frac{(-1)^k}{(s-k)!} \lambda^k c^{s-k}.$$

Multiplying both sides of the above equation by λ^{-s}, each component of c satisfies the equation

$$\sum_{k=0}^{s} \frac{(-1)^{s-k}}{(s-k)!} \binom{s}{k} \left(\frac{c_i}{\lambda}\right)^{s-k} = 0.$$

This is equivalent to $L_s(c_i/\lambda) = 0$, L_s is the s-degree Laguerre polynomial. Therefore, the abscissae $c_1, \ldots, c_s$ for an s-stage SIRK method, satisfy

$$c_i = \lambda \xi_i, \qquad i = 1, 2, \ldots, s,$$

where ξ_i are the zeros of L_s. If the eigenvalue λ satisfy $L_s(\frac{1}{\lambda}) = 0$ for L-stability, one of the abscissae must be chosen to be 1 [6]. For an s stage standard SIRK method, $b_j = a_{ij}$ when $c_i = 1$, $j = 1, 2, \ldots, s$. The coefficient matrix A is determined by the stage order conditions and $b^T = e_j^T A$, where e_j is the zero vector except

the j-th component is 1. Furthermore, since $L_k(0) = L_{k+1}(0) = 1$, $L'_{p+1}(z) = L'_p(z) - L_p(z)$ and (8), we have

$$\begin{aligned}\sum_{j=1}^{p} a_{ij} L_k(\xi_j) &= \sum_{j=1}^{p} a_{ij} L_k\left(\frac{c_j}{\lambda}\right) = \int_0^{c_i} L_k\left(\frac{c}{\lambda}\right) dc = \int_0^{\xi_i} \lambda L_k(\xi) d\xi \\ &= \lambda \int_0^{\xi_i} (L'_k(\xi) - L'_{k+1}(\xi)) d\xi \\ &= \lambda L_k(\xi_i) - \lambda L_{k+1}(\xi_i), \quad k = 0, 1, \ldots, p-1. \end{aligned} \tag{9}$$

Hence $T^{-1}AT = \bar{A} = \lambda(I - K)$, K is $p \times p$ matrix with all lower subdiagonal elements are 1. The first order L-stable SIRK method is the implicit Euler method, for order-2 method, the stability function

$$R(z) = \frac{P(z)}{Q(z)} = \frac{P(z)}{(1-\lambda z)^2} = \exp(z) + O(z^3),$$

and $1/\lambda$ is chosen to be a zero of L_2, the method obtained is order 2 but L-stable because the degree of $P(z)$ is less than 2. That is

$$P(z) = 1 + (1 - 2\lambda)z.$$

Using E-polynomial [15], we have

$$\begin{aligned} E(y) &= Q(iy)Q(-iy) - P(iy)P(-iy) = \left(1 + \lambda^2 y^2\right)^2 - \left(1 + (1-2\lambda)^2 y^2\right), \\ &= \left(-1 + 4\lambda - 2\lambda^2\right) y^2 + \lambda^4 y^4. \end{aligned}$$

This implies

$$E(y) \geq 0, \quad \forall y \Leftrightarrow 1 - \frac{\sqrt{2}}{2} \leq \lambda \leq 1 + \frac{\sqrt{2}}{2}.$$

Because the two zeros of $L_2\left(\frac{1}{z}\right)$ are $1 - \sqrt{2}/2$ and $1 + \sqrt{2}/2$, $\lambda = 1/\xi_2 = 1 - \sqrt{2}/2$ for smaller error constant, so $c_2 = 1$. Hence $c_1 = \lambda\xi_1 = \lambda(2 - \sqrt{2}) = 3 - 2\sqrt{2}$. By stage order conditions $C(2)$, we have the following equations

$$\begin{aligned} a_{11} + a_{12} &= c_1, & a_{21} + a_{22} &= c_2, \\ a_{11}c_1 + a_{12}c_2 &= \frac{c_1^2}{2}, & a_{21}c_1 + a_{22}c_2 &= \frac{c_2^2}{2}. \end{aligned}$$

The coefficient matrix A of the order-2 method is easy to determine uniquely. The weight b^T is the second row of A. We have the corresponding Butcher tableau,

$$\begin{array}{c|cc} 3 - 2\sqrt{2} & \frac{5-3\sqrt{2}}{4} & \frac{7-5\sqrt{2}}{4} \\ 1 & \frac{1+\sqrt{2}}{4} & \frac{3-\sqrt{2}}{4} \\ \hline & \frac{1+\sqrt{2}}{4} & \frac{3-\sqrt{2}}{4} \end{array}. \tag{10}$$

From (9), the transformation matrix T is given by

$$T = [v_1, v_2] = \begin{bmatrix} L_0(\xi_1) & L_1(\xi_1) \\ L_0(\xi_2) & L_1(\xi_2) \end{bmatrix} = \begin{bmatrix} 1 & -1 + \sqrt{2} \\ 1 & -1 - \sqrt{2} \end{bmatrix}.$$

The design of SIRK methods has high stage order (equal to overall order) and L-stability for very stiff problems. However, higher order SIRK methods have some abscissae outside the integration interval because of the stability requirement (some of abscissae are greater than 1 when $s \geq 3$). In fact, once the eigenvalue λ is chosen for stability, SIRK methods have no free parameters. Besides, they are inferior for small dimensional systems because of their relatively high transformation costs.

2.2 DESI methods

The main skeleton of DESI methods is to use the SIRK methods to create the input value for the following several diagonal stages within one step. Because of those extra internal stages, DESI methods can have more freedom in determining its coefficients, such as the abscissae. It turns out that their error constants can be chosen much smaller than those of the classical SIRK methods. This also means that the DESI methods will integrate further forward than their corresponding SIRK methods. For same integration interval, number of steps used by DESI methods are fewer. Therefore, the overall computation costs can be reduced considerately.

The s stage, order p DESI method with the corresponding tableau can be written as

$$\begin{array}{c|cccccccc}
c_1 & a_{11} & a_{12} & \dots & a_{1p} & 0 & 0 & \dots & 0 \\
c_2 & a_{21} & a_{22} & \dots & a_{2p} & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
c_p & a_{p1} & a_{p2} & \dots & a_{pp} & 0 & 0 & \dots & 0 \\
c_{p+1} & a_{p+1,1} & a_{p+1,2} & \dots & a_{p+1,p} & \lambda & 0 & \dots & 0 \\
c_{p+2} & a_{p+2,1} & a_{p+2,2} & \dots & a_{p+2,p} & a_{p+2,p+1} & \lambda & \dots & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
c_s & a_{s1} & a_{s2} & \dots & a_{sp} & a_{s,p+1} & a_{s,p+2} & \dots & \lambda \\
 & b_1 & b_2 & \dots & b_p & b_{p+1} & b_{p+2} & \dots & b_s
\end{array},$$

where the $p \times p$ matrix

$$A_p = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & & \vdots \\ a_{p1} & a_{p2} & \dots & a_{pp} \end{bmatrix},$$

has one-point spectrum $\{\lambda\}$ with λ as in the diagonal block. In order to retain the stage order to be p, $c_1, \dots, c_p$ are chosen as $\lambda\xi_1, \dots, \lambda\xi_p$ respectively where $\xi_1, \xi_2, \dots, \xi_p$ are the zeros of the Laguerre polynomial L_p. The matrix A_p thus can be found using stage order conditions. The method are proposed to be strongly stable at infinity, therefore an A-stable method can be L-stable, c_s is always chosen to be 1 and $b_i = a_{si}, \quad i = 1, \dots, s$. The other coefficients are determined to satisfy the stability requirement and the stage order conditions.

Consider methods with $p = 2, s = 4$. Use $(1-\lambda z)^4 \exp(z)$ to find the first two coefficients up to z^2 of $P(z)$, the numerator of the stability function $R(z) = P(z)/Q(z)$, it follows that

$$P(z) = 1 + (1-4\lambda)z + \left(\frac{1}{2} - 4\lambda + 6\lambda^2\right) z^2 + \alpha_3 z^3, \qquad Q(z) = (1-\lambda z)^4.$$

The E-polynomial can be derived as

$$\begin{aligned} E(y) = & \left(-\frac{1}{4} + 2\alpha_3 + (4-8\alpha_3)\lambda - 22\lambda^2 + 48\lambda^3 - 30\lambda^4\right) y^4 \\ & + \left(-\alpha_3^2 + 4\lambda^6\right) y^6 + \lambda^8 y^8. \end{aligned}$$

It can then be shown

$$\begin{aligned} E(y) \ \geq \ 0, \quad & \forall y \geq 0 \\ \Leftrightarrow \ & 0.12994576623707250434 4 \leq \lambda \leq 3.284267796136022544 \\ \wedge \ & -0.00573385343830948 \leq \alpha_3 \leq 0.00573385343830948. \end{aligned}$$

In order to minimize the error constant and place the absicssae inside the integration interval as possible as we can, we can choose $\lambda = 0.129945766237072504344$ and $\alpha_3 = 0.00573385343830948$. The abscissae c_1, c_2 are given by $c_i = \lambda\xi_i, i = 1, 2$. c_3 is a free parameter and $c_4 = 1$. We can choose $c_3 = \lambda/\lambda_t$ where $\lambda_t = 0.1804253064293985641$ is the eigenvalue in the case of $p = 2, s = 3$. The coefficients in the first three row in A can be obtained by stage order conditions $C(2)$. The coefficients in the last row of A can be derived using $C(2)$ and using the stability requirement

$$1 + ze_4^T A(I - zA)^{-1} e = \frac{P(z)}{(1-\lambda z)^4}.$$

The error constant C_3 is the coefficient of z^3 in $(1-\lambda z)^4 \exp(z) - \alpha_3$, Hence, the second-order L-stable DESI method with 4 stages has the following corresponding Butcher tableau

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & 0 & 0 \\ c_2 & a_{21} & a_{22} & 0 & 0 \\ c_3 & a_{31} & a_{32} & \lambda & 0 \\ 1 & a_{41} & a_{42} & a_{43} & \lambda \\ \hline & a_{41} & a_{42} & a_{43} & \lambda \end{array},$$

where

$$\begin{array}{llllll} c_1 = 0.0761204674887132, & c_2 = 0.4436625974595767, & c_3 = 0.7049034875501352, \\ a_{11} = 0.0840029999907146, & a_{12} = -0.0078825325020013, & a_{21} = 0.2677740649761463, \\ a_{22} = 0.1758885324834304, & a_{31} = 0.2672945180670744, & a_{32} = 0.3076632032459882, \\ a_{41} = 0.2738877005939397, & a_{42} = 0.2719103215907779, & a_{43} = 0.3242562115782104. \end{array}$$

The related coefficients $1/\lambda$ and $(1/\lambda^i)\alpha_i$ of DESI formulae for order $p = 1, 2 \ldots, 8$ and $s = p + n, n = 1, 2, \ldots, 6$ are given in [14].

DESI methods have more free parameters in determining coefficients for methods and in minimizing error constant with more extra diagonal stages. In general, DESI

methods with one more stage will have one more free parameter for coefficient of method and have one more free parameter for minimizing the error constant. For example, in the case of $s = p + 1$, DESI methods have no free parameter once the λ is chosen for stability. When $s = p + 2$, DESI methods have one free parameter to determine the absicssae and have one free parameter to minimize the error constant. We also note that for order 2 DESI methods the error constants for $s = 3$, $s = 4$ are respectively 0.012185 and 0.00642 approximately, while the error constant for order 2 SIRK method is 0.04044. It is clear to have the advantage from adding extra stages in the point of view of local error. In general, as the number of stages increases the error constant decrease. But there are some exceptions, for example, when $p = 3$, the error constant for $s = 6$ is smaller than $s = 7$. The error constants of SIRK and DESI of order up to 8 are given in Table 1.

3 ESIRK methods

For explicit Runge Kutta methods, the idea of effective order reduces the number of order conditions and makes it possible first break the Butcher barrier. Because of the stability requirement and the fact that SIRK methods are collocation methods, the choices of abscissae of the methods are restricted to be proportional to the roots of the Laguerre polynomial. It turns out that when $s \geq 3$, some of the components of c are greater than 1. Effective order SIRK methods (ESIRK) are generalizations of classical SIRK methods and they have the same stability properties as SIRKs, their stage order equal to the stage number. We assume ψ and ϕ represent the mappings associated with the starting method and the method itself. If the method satisfies

$$\psi(y(x_{n-1} + h)) - \phi(\psi(y(x_{n-1}))) = O(h^{s+1}),$$

then the method has effective order s. By using this generalized concept of order, we will be able to introduce more free parameters from the starting method ψ. In other words, the input value $y(x_{n-1})$ at the n-th step can now be replaced by

$$\psi(y(x_{n-1})) = y(x_{n-1}) + \alpha_1 h y'(x_{n-1}) + \cdots + \alpha_s h^s y^{(s)}(x_{n-1}),$$

Table 1 Error constants of SIRK and DESI

p	SIRK	$s = p + 1$	$s = p + 2$	$s = p + 3$	$s = p + 4$
2	4.4044×10^{-2}	1.2185×10^{-2}	6.4203×10^{-3}	3.6824×10^{-3}	2.5535×10^{-3}
3	2.5897×10^{-2}	3.7902×10^{-4}	3.5969×10^{-4}	4.6262×10^{-6}	2.6417×10^{-5}
4	1.1242×10^{-3}	8.8406×10^{-4}	1.6388×10^{-4}	8.0272×10^{-5}	4.1801×10^{-5}
5	5.3005×10^{-4}	4.5630×10^{-5}	1.7491×10^{-5}	2.4229×10^{-6}	1.3785×10^{-6}
6	1.5712×10^{-5}	2.7750×10^{-5}	2.7630×10^{-7}	1.6085×10^{-6}	8.5710×10^{-7}
7	5.7932×10^{-6}	1.1977×10^{-6}	3.5152×10^{-7}	8.3174×10^{-8}	3.5883×10^{-8}
8	1.3036×10^{-7}	1.9774×10^{-7}	4.8152×10^{-8}	1.6909×10^{-9}	2.7833×10^{-9}

and the output for this step becomes

$$\psi(y(x_{n-1}+h)) = y(x_{n-1}+h) + \alpha_1 h y'(x_{n-1}+h) + \cdots + \alpha_s h^s y^{(s)}(x_{n-1}+h).$$

This can be done by adding starting procedure and ensure the numerical solution obtained after this step is $\psi(y(x_{n-1}+h)) + O(h^{s+1})$. Hence, the stage values Y_i and output values y_n for order s ESIRK methods ϕ are

$$\begin{aligned} Y_i &= \psi(y(x_{n-1})) + h\sum_{j=1}^{s} a_{ij} f(Y_j), \\ y_n &= y(x_{n-1}) + \alpha_1 h y'(x_{n-1}) + \cdots + \alpha_s h^p y^{(s)}(x_{n-1}) + h\sum_{i=1}^{s} b_i f(Y_i), \\ &= \psi(y(x_{n-1}+h)), \end{aligned}$$

for $i = 1, 2 \ldots, s$. The stage order conditions (7) become

$$\sum_{j=1}^{s} a_{ij} \frac{c_j^{k-1}}{(k-1)!} + \alpha_k = \frac{c_i^k}{k!}, \quad i = 1, 2 \ldots, s, \quad k = 1, 2 \ldots, s. \tag{11}$$

and the order conditions for output values are

$$\sum_{i=1}^{s} b_i \frac{c_i^{k-1}}{(k-1)!} = \sum_{j=1}^{k} \frac{1}{(k-j+1)!}\alpha_{j-1}, \quad \alpha_0 = 1, \quad k = 1, 2, \ldots, s. \tag{12}$$

The single eigenvalue λ of the A matrix of an s-stage method is chosen as the one for the s-stage SIRK method. In addition to having the s^2 conditions in (11), the single-implicitness gives another s conditions. Consequently, once the abscissae c_j, $j = 1, 2, \ldots, s$ are choosen distinctly, the coefficient matrix A of the new method will be determined uniquely. It is easy to see that $\alpha_k = 0, \quad k = 1, 2, \ldots, s$ if the abscissae are chosen the same as SIRK methods. It is necessary to have a perturbed initial value of the differential equation in the first step and to undo the perturbation in the last step. For finishing methods, because the s-stage ESIRK methods have the stage order s, one can take any stage value Y_i to be the output value as long as for some $c_i = 1$. From (11), after the distinct abscissae c_i are given, the s^2 equations with $s^2 + s$ unknowns (s^2 a_{ij} and s α_i) plus s conditions from the characteristic polynomial of A, $p(z) = (z-\lambda)^s$, we can find the perturbations α_i and the coefficients a_{ij} for an order s ESIRK method. For example, the second order ESIRK methods $(A, b^T, c)_2$ with eigenvalue $\lambda = 1 - \sqrt{2}/2$, if choose any two distinct abscissae, say $c = [c_1, c_2]$, from (11), then A matrix are given by

$$a_{11} = \frac{c_1^2 - 2c_1c_2 - 2\alpha_2 + 2\alpha_1 c_2}{2(c_1 - c_2)}, \qquad a_{12} = \frac{c_1^2 - 2\alpha_1 c_1 + 2\alpha_2}{2(c_1 - c_2)},$$

$$a_{21} = \frac{-c_2^2 - 2\alpha_2 + 2\alpha_1 c_2}{2(c_1 - c_2)}, \qquad a_{22} = \frac{2c_1c_2 - c_2^2 - 2\alpha_1 c_1 + 2\alpha_2}{2(c_1 - c_2)},$$

with the characteristic polynomial of A is

$$p(z) = \frac{1}{2}c_1c_2 - \frac{1}{2}c_1\alpha_1 + \alpha_2 - \frac{1}{2}c_2\alpha_1 + \left(-\frac{1}{2}c_1 + \alpha_1 - \frac{1}{2}c_2\right)z + z^2,$$

and $p(z) = \det(zI_2 - A) = (z-\lambda)^2$. We have $\alpha_1 = (c_1+c_2)/2 - 2\lambda, \alpha_2 = \left(c_1^2+c_2^2\right)/4 - (c_1+c_2)\lambda + \lambda^2$. Also from (12), $b_1 = (c_1 - c_2 + 1 - 4\lambda)/(2c_1 - 2c_2), b_2 = (c_1 - c_2 - 1 + 4\lambda)(2c_1 - 2c_2)$. Let $c = [0, 1]$, the corresponding tableau is given by

$$\begin{array}{c|cc} 0 & \frac{9-6\sqrt{2}}{4} & \frac{-3+2\sqrt{2}}{4} \\ 1 & \frac{11-6\sqrt{2}}{4} & \frac{-1+2\sqrt{2}}{4} \\ & 2-\sqrt{2} & -1+\sqrt{2} \end{array}.$$

For starting methods, the c vector and the coefficient matrix A of the classical SIRK method will not be changed, b^T vector need to be modified in accordance with the required perturbation of ESIRK input values. Denote the modified vector by $\bar{b}^T = [\bar{b}_1, \bar{b}_2]$ and $e = [1, 1]^T$,

$$[\bar{b}_1, \bar{b}_2][e, \bar{c}] = \left[1 + \alpha_1, \tfrac{1}{2} + \alpha_1 + \alpha_2\right], \quad \bar{c} = [3 - 2\sqrt{2}, 1]^T. \tag{13}$$

The starting method is given by the tableau

$$\begin{array}{c|cc} 3-2\sqrt{2} & \frac{5-3\sqrt{2}}{4} & \frac{7-5\sqrt{2}}{4} \\ 1 & \frac{1+\sqrt{2}}{4} & \frac{3-\sqrt{2}}{4} \\ \hline 1 & \frac{3+\sqrt{2}}{8} & \frac{-7+7\sqrt{2}}{8} \end{array},$$

The coefficient matrix A of ESIRK methods have the similarity with the matrix named "doubly companion matices" C, and it has been shown that C have the similarity property with a matrix which has Jordan block form with a single diagonal element λ [9]. Consequently, the idea of the Butcher's transformation can be adapted to these methods.

4 DESIRE methods

It is allowed to choose any s distinct abscissae for ESIRK methods and the local truncation error can be reduced by applying a variable stepsize scheme to ESIRK methods [10], however the error constants for ESIRK (SIRK) methods are not small enough. The DESI formulae are good extensions of the SIRK methods because of the smaller error constants which allow larger stepsizes to be taken. L-stable DESI methods are available for every order. Besides, DESI also has advantages in estimating the error. However, the existing order p DESI formulae with stage number $s = p + 3$ suffers from the fact that their some abscissae are greater than one when $p > 4$. DESIRE methods are combination of DESI method and ESIRK method in

order to compensate for each other's weakness by using the strengths of these two types of methods [13]. Some features of DESIRE methods can be summarized as below.

(1) DESIRE methods have the same structures as DESI methods (including the singly-implicit part and the diagonally-implicit part).
(2) DESIRE methods have the singly-implicitness property.
(3) The singly-implicit part of DESIRE is an ESIRK method.
(4) The additional diagonal stages retain the same stage order as the singly-implicit part.
(5) DESIRE methods have the same stability property as their associated DESIs.

While free parameters of ESIRK methods come from their perturbed input approximations, DESI methods have extra parameters because of their additional diagonal stages. It can be expected that there are more freedom in choosing the coefficients for DESIRE methods. If we consider the order p DESIRE methods with number of stages $s = p + 1$, the additional diagonal stage can be determined immediately once c_{p+1} is assigned because the stage order is p. In this case, we have only $a_{p+1,1}, a_{p+1,2}, \ldots, a_{p+1,p}$ to specify (note that $a_{p+1,p+1} = \lambda$). From the weight vector b^T, there are $p + 1$ unknowns to be specified. Similar to ESIRK methods, we have to modify b^T so as to produce the perturbed initial value. From (12), with $s = p + 1$, there are p conditions to be satisfied. Also from the restriction of stability behaviour of an order p method, the stability function $R(z) = \frac{P(z)}{Q(z)}$ should satisfy $Q(z) = (1 - \lambda z)^s$ and the coefficient of z^{p+1} in $P(z)$ must be zero. It is noted that in this case, we don't have free parameter except the abscissae $c_1, \ldots, c_p$ should be chosen distinctly.

In the case of $s = p + 2$, we have extra $p + 1$ parameters, $a_{p+2,1}, a_{p+2,2}, \ldots, a_{p+2,p+1}$. Deducted by stage order conditions of stage value Y_{p+2}, we have one free parameter left to obtain better error estimation. In [13], it is proposed to use this free parameter to estimate the error for a variable order implementation. Consequently, we have another advantage over the classical DESI methods when effective order is applied to the order p DESI methods with $s = p+2$ stages; we can construct an error estimator for the purpose of changing order [13] whereas the classical DESI methods cannot achieve this conveniently unless $s \geq 3$.

Consider the second order DESIRE method with 4 stages. Let the abscissae $c = \left[0, \frac{1}{3}, \frac{2}{3}, 1\right]^T$, the stability function of the method satisfies

$$R(z) = \frac{1 + \gamma_1 z + \gamma_2 z^2 + \gamma_3 z^3 + \gamma_4 z^4}{(1 - \lambda z)^4}.$$

According to [7], the intervals of λ for L-stability are [0.129945766, 3.284267796], and $\gamma_4 = 0$. Choose $\lambda = 3/23$ (an approximation to 0.129945766). Because the

method is of order 2, we have $\gamma_1 = 11/23, \gamma_2 = 85/1058$. By the E-polynomial scheme, the method is A-stable if and only if

$$\frac{9(11-\sqrt{-49+36\sqrt{2}})}{12167\sqrt{2}} < \gamma_3 < \frac{9(11+\sqrt{-49+36\sqrt{2}})}{12167\sqrt{2}}.$$

We can choose γ_3 from this interval to obtain the smallest possible error constant. The error constant is given by

$$C_3 = (-1)^4 L_4'\left(\frac{1}{\lambda}\right)\lambda^3 - \gamma_3.$$

The lower bound of γ_3 makes the error constant smallest. For simplicity, we choose $\gamma_3 = 63/12167$, then $C_3 = 6.178 \times 10^{-3}$. From (11) and singly-implicitness,

$$\alpha_1 = -\frac{13}{138}, \qquad \alpha_2 = \frac{25}{19044}.$$

Choose $a_{43} = 1/3$ for estimating $h^4 y^{(4)}$ [13], the coefficients of A can be specified by the order conditions. b^T is found using (12) with $s = 2$, and the stability function

$$1 + zb^T(I_4 - zA)^{-1}e = \frac{1+\gamma_1 z+\gamma_2 z^2+\gamma_3 z^3}{(1-\lambda z)^4},$$

with $\gamma_3 = \frac{63}{12167}$. The resulting Butcher tableau is

$$\begin{array}{c|cccc} 0 & \frac{623}{6348} & -\frac{25}{6348} & 0 & 0 \\ \frac{1}{3} & \frac{1681}{6348} & \frac{1033}{6348} & 0 & 0 \\ \frac{2}{3} & \frac{1451}{6348} & \frac{2551}{6348} & \frac{3}{23} & 0 \\ 1 & \frac{407}{2116} & \frac{927}{2116} & \frac{1}{3} & \frac{3}{23} \\ \hline & \frac{126379}{559682} & \frac{236527}{559682} & \frac{145503}{559682} & \frac{51273}{559682} \end{array}.$$

For starting methods, c and A of the classical SIRK method will not be changed, the weight vector $\bar{b}^T = [9497(1+\sqrt{2})/38088, (25003-9497\sqrt{2})/38088]$ is found using (13). In a similar way, we can derive DESIRE methods with $s = p+3, p+4, \ldots$, but for simplicity of implementation, the case of $s = p+2$ is enough to estimate the error for variable order, we will not go further to discuss the case of $s > p+2$.

In attempting to examine which of the cases $s = p+1$ or $s = p+2$ is a better option, we compare $s = p+1$ and $s = p+2$ DESIRE methods for $p = 1, 2, 3, 4$. In Fig. 1, we show the work/precision diagrams for the different order p methods for solving the Kaps problem [16]. Although the case $s = p+2$ has one more stage than the case $s = p+1$, except $p = 3$ and $p = 1$ in low accuracy, the case $s = p+2$ seems to be more efficient than $s = p+1$. The different error constants of the methods seem able to offer an explanation. Referring to Table 1, the third order method with 4 stages has a smaller error constant than the method with 5 stages. This is also the case for the first order method. For variable order, methods with $s = p+2$ are good

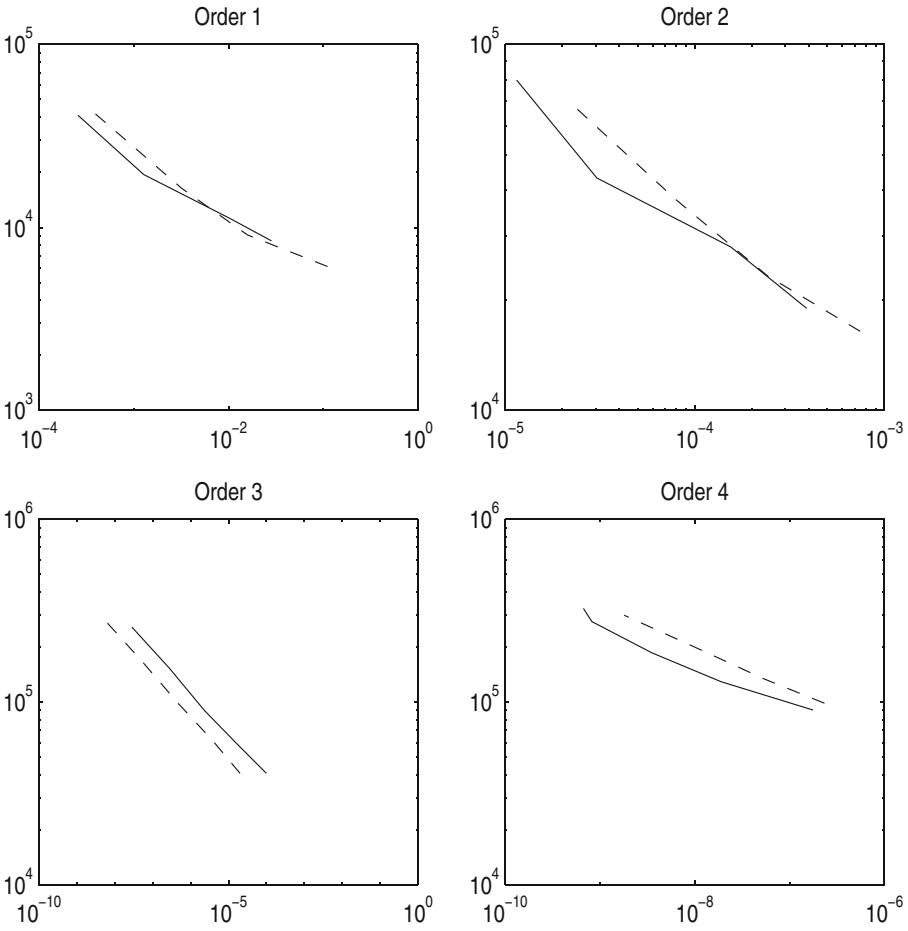

Fig. 1 Work/Precision diagram (flops/maximum error) for DESIRE methods with order $p = 1, 2, 3, 4$ solving the Kaps problem [16], $s = p + 1$: $--$, $s = p + 2$: $-$

candidates for implementation. Therefore, in the following context of this paper, the word "DESIRE" will indicate the case $s = p + 2$.

5 The efficiency of methods and numerical experiments

One may ask : how many internal diagonally stages should be added on? In general, the efficiency of a L-stable method for solving stiff problems is determined by global error (or accuracy) and number of operations. The error constants for SIRK and ESIRK methods are identical while DESI and DESIRE methods have the same error constants (much smaller than SIRK)(cf. Table 1). In count of the total operations, in addition to considering the cost for LU factorization and back substitutions, we also need to take the transformation costs into account. Approximately, for an s

stage, order p method solving a system of dimension N, the total operation number for each iteration can be summarized as follows.

Method	Operation count (p: order, s: stage no., N: system dimension)
SDIRK	$\frac{N^3}{3}$(factorization)+sN^2(back substitution)
SIRK, ESIRK	$\frac{N^3}{3} + pN^2$+$\underline{3Np^2}$(transformation)
DESI, DESIRE	$\frac{N^3}{3} + pN^2 + 3Np^2$+$\underline{(s-p)N^2}$(diagonal part)

The underlining in above table indicates the extra computation costs shown in parentheses. The cost of back substitution for SDIRK is the square of system dimension N^2 multiplied by the stage number s. In [1], R. Alexander shows that when $s \leq 3$, we can have L-stable SDIRK method with order $p = s$, but it is not possible to attain order 4 in 4 stages. In fact, we need at least 5 stages so as to obtain 4-th order L-stable SDIRK methods (see [15]). For DESI and DESIRE methods, the number of additional added diagonal stages is $s - p$. DESI (DESIRE) methods can compensate for their larger number of operations by the choice of a much smaller error constant.

If we suppose there is only one iteration required per step and use the length of the stepsize taken per flop at each step as the measurement of efficiency for order p methods, then the magnitude of efficiency will be proportional to

$$E = \frac{1}{N_{opn} C^{\frac{1}{p+1}}},$$

where N_{opn} is the number of operations and C is the error constant of the related method. The values $N = 2, 5, 50$ give the efficiency Table 2.

For $p = 1, 2, \ldots, 6$, except $p = 3, 6$, adding more diagonally stages improves efficiency for both small and large systems. When $p = 3$, one less added stage method, $s = p + 1$ is slightly more efficient than $s = p + 2$. This can be verified by the numerical results in Fig. 1. It also happens in the case of $s = p+2$ and $s = p+3$ when $p = 6$. Furthermore, the efficiency improved rapidly (more than 2 times) by adding one diagonally stage and efficiency improved 6 times more by adding three diagonally stages for both small and large systems.

In practice, it can not be always to have only one iteration for each step. As we have stated, because of the small error constant, the stepsize taken by DESI(DESIRE) methods are much larger than for usual methods. A good predictor is desirable and indeed necessary. There are many ways to find the starting value for the modified Newton iterations. For example, we can construct an interpolation polynomial based on the order conditions or, alternatively, use Lagrange's interpolation formulae to form the interpolation polynomial. The numerical experiments for DESI code have shown that from the information on $hf(Y_i)$ of the previous step, using the $C(p)$ conditions to derive a quadrature formula for predicting the starting

Table 2 Efficiency measurement of singly-implicit methods

	$p=2$	$p=3$	$p=4$	$p=5$	$p=6$
$N=2$					
SIRK(ESIRK)	8.1682×10^{-2}	3.6303×10^{-2}	3.3915×10^{-2}	2.0358×10^{-2}	2.0010×10^{-2}
$s=p+1$	1.1239×10^{-1}	9.8628×10^{-2}	3.4385×10^{-2}	2.9944×10^{-2}	1.8149×10^{-2}
$s=p+2$	1.2610×10^{-1}	9.4713×10^{-2}	4.6598×10^{-2}	3.4355×10^{-2}	3.4502×10^{-2}
$s=p+3$	1.3877×10^{-1}	2.6730×10^{-1}	5.2050×10^{-2}	4.6726×10^{-2}	2.6405×10^{-2}
$N=5$					
SIRK(ESIRK)	1.8670×10^{-2}	9.9052×10^{-3}	1.0189×10^{-2}	6.4896×10^{-3}	6.6365×10^{-3}
$s=p+1$	2.4598×10^{-2}	2.5905×10^{-2}	1.0034×10^{-2}	9.3353×10^{-3}	5.9164×10^{-3}
$s=p+2$	2.6680×10^{-2}	2.4071×10^{-2}	1.3242×10^{-2}	1.0490×10^{-2}	1.1064×10^{-2}
$s=p+3$	2.8569×10^{-2}	6.6007×10^{-2}	1.4437×10^{-2}	1.3992×10^{-2}	8.3360×10^{-3}
$N=50$					
SIRK(ESIRK)	5.9908×10^{-5}	4.9346×10^{-5}	7.1929×10^{-5}	6.0694×10^{-5}	7.8234×10^{-5}
$s=p+1$	8.7321×10^{-5}	1.3518×10^{-4}	7.2135×10^{-5}	8.7559×10^{-5}	6.9335×10^{-5}
$s=p+2$	1.0294×10^{-4}	1.3080×10^{-4}	9.6773×10^{-5}	9.8650×10^{-5}	1.2896×10^{-4}
$s=p+3$	1.1824×10^{-4}	3.7166×10^{-4}	1.0709×10^{-4}	1.0709×10^{-4}	9.6661×10^{-5}

values for singly-implicit stages, and making use of the recently evaluated singly-implicit stages to form the quadrature formulae for predicting the starting values for diagonally-implicit stages is a better predictor [14].

In an attempt to go further to analyze the outcomes due to the additional diagonally stages and the application of effective order, we compare the numerical results between the members of singly-implicit methods including SIRK, DESI, ESIRK and DESIRE methods. In our experiments on test problems, we choose

1. the Curtis problem,

$$\begin{cases} y' = f = Ay + du - Au, \quad y = [y_1, y_2]^T, & y_1(0) = 1 \\ u = [\cos(x), \sin(x)]^T, \quad du = [-\sin(x), \cos(x)]^T, & y_2(0) = 0 \\ A(1,1) = -1 - \lambda\cos(\theta x)^2, & \theta = \frac{1}{5} \\ A(2,2) = -1 - \lambda\sin(\theta x)^2, & \lambda = 1000 \\ A(1,2) = A(2,1) = \lambda\cos(\theta x)\sin(\theta x), & x \in [0, 10\pi] \end{cases} \tag{14}$$

2. the Robertson kinetic problem,

$$\begin{cases} y_1' = -0.04y_1 + 10^4 y_2 y_3, & y_1(0) = 1 \\ y_2' = 0.04y_1 - 10^4 y_2 y_3 - 3\times10^7 y_2^2, & y_2(0) = 0 \\ y_3' = 3\times10^7 y_2^2, & y_3(0) = 0 \\ x \in [0, 10^{10}], \end{cases} \tag{15}$$

3. the Van der Pol oscillator,

$$\begin{cases} y_1'(x) = y_2(x), & y_1(0) = 2 \\ y_2'(x) = 10^6(1 - y_1(x)^2)y_2(x) - y_1(x), & y_2(0) = 0 \\ x \in [0, 2]. \end{cases} \tag{16}$$

For the Robertson problem, we set $atol = 10^{-4}rtol$. For the rest of the problems, $atol = rtol$. For methods, we have chosen order 2, 3, 4 L-stable SIRK, DESI, ESIRK and DESIRE methods [12]. We use the stage number $s = p+3$ for implementing the order p DESI methods and $s = p + 2$ for the order p DESIRE methods. Therefore, for the same order DESI has the smallest error constant, followed by DESIRE, then SIRK and ESIRK methods. Besides, for effective order methods, the abscissae used are equally spaced in [0, 1] (cf. [12]).

For reasonable comparisons, in addition to having a good predictor and an error estimator for every method, some schemes introduced in [11] are adopted for the Newton iteration for all test methods. Besides, the numerical tests have been carried out using the same variable-step algorithm. We have also carefully chosen some crucial parameters for each method. These parameters, such as the safety factor for stepsize prediction, convergent rate for updating the Jacobian matrix, the relation parameter for stopping the iterations and the stepsize bound for constant stepsize, all

Table 3 Numerical results for Curtis (14) by testing order-2 singly-implicit methods

tol	nsteps	rej(err/newt)	nfcn	njac	lu	flops	maxerr	niter
SIRK								
10^{-3}	474	95/206	2832	474	775	505827	1.88×10^{-3}	2.49
10^{-4}	316	35/24	1862	316	375	316712	5.53×10^{-4}	2.65
10^{-5}	541	1/0	2828	541	542	481920	7.92×10^{-5}	2.61
10^{-6}	1138	0/0	4386	1138	1138	844924	2.05×10^{-5}	1.93
DESI								
10^{-3}	196	0/205	4668	196	401	844730	1.85×10^{-4}	4.76
10^{-4}	165	0/50	2875	165	215	581391	1.47×10^{-4}	3.48
10^{-5}	242	0/0	3157	242	242	649377	4.63×10^{-5}	2.61
10^{-6}	513	0/0	4650	513	513	1068513	9.33×10^{-6}	1.81
ESIRK								
10^{-3}	509	112/226	3110	508	846	599724	1.08×10^{-3}	2.50
10^{-4}	362	48/42	2166	361	451	393780	7.05×10^{-4}	2.64
10^{-5}	552	0/0	3040	551	551	535852	7.80×10^{-5}	2.75
10^{-6}	1179	0/0	4608	1178	1178	941861	1.88×10^{-5}	1.95
DESIRE								
10^{-3}	202	0/96	2097	201	297	471415	1.19×10^{-2}	2.59
10^{-4}	179	0/16	1743	178	194	366640	7.78×10^{-4}	2.43
10^{-5}	306	0/0	2212	305	305	515489	7.28×10^{-5}	1.81
10^{-6}	642	0/0	3662	641	641	949683	9.25×10^{-6}	1.43

effect the numerical performance directly. For example, for large tolerances, the stepsize taken is larger, the Newton iterations will converge slower, and we may lower the convergence rate for updating the Jacobian matrix so as to reduce the number of iterations. The parameters chosen usually are very dependent on the test problems, such as for a mildly stiff problem. The stepsize bound for constant stepsize is advisedly chosen to be smaller than the one for a strongly stiff problem. Since all numerical results for various testing differential equations using specific order methods are similar, we will show some numerical results for Curtis problem (14) by testing order-2 singly-implicit methods, and for Van der Pol equation (16) by testing order-3 method, and for Robertson equation (15) by testing order-4 methods. For each problem and for each fixed-order method, some numerical details are given in tables (cf. Tables 3, 4 and 5) in order that we can analyze the influence caused by the additional diagonally stages and by effective order. The details include: the accepted steps, the rejected steps (caused by both error and convergence), the function evaluations, the numbers of Jacobian evaluations, the LU factorizations, the total flops, the maximum global error and the average number of iterations per step.

Table 4 Numerical results for Van der Pol (16) by testing order-3 methods

tol	nsteps	rej(err/newt)	nfcn	njac	lu	flops	maxerr	niter
SIRK								
10^{-3}	310	18/17	3387	277	324	467644	5.89×10^{-4}	3.44
10^{-4}	486	16/0	3546	412	442	569033	4.61×10^{-4}	2.35
10^{-5}	837	17/0	5088	641	686	884954	6.61×10^{-5}	1.99
10^{-6}	1472	11/0	8163	1037	1091	1476563	1.36×10^{-5}	1.83
10^{-7}	2586	7/0	14838	1409	1503	2621753	2.62×10^{-6}	1.91
DESI								
10^{-3}	135	46/25	3711	111	199	636859	9.41×10^{-4}	3.42
10^{-4}	184	50/5	4008	148	226	725490	1.09×10^{-4}	2.85
10^{-5}	270	41/0	4361	213	282	849132	2.48×10^{-7}	2.34
10^{-6}	406	22/0	5313	317	372	1090620	5.92×10^{-7}	2.07
10^{-7}	638	11/0	7588	495	543	1596217	3.90×10^{-8}	1.95
ESIRK								
10^{-3}	317	25/0	1626	172	275	341005	2.03×10^{-3}	1.58
10^{-4}	518	20/0	2172	225	364	499879	2.44×10^{-4}	1.34
10^{-5}	853	18/0	3360	618	690	795337	6.37×10^{-5}	1.28
10^{-6}	1462	8/0	5481	490	688	1310768	1.40×10^{-5}	1.24
10^{-7}	2571	8/0	9108	757	998	2246127	2.12×10^{-6}	1.18
DESIRE								
10^{-3}	111	15/40	2554	87	158	442945	7.94×10^{-4}	4.05
10^{-4}	152	26/32	3006	15	197	552768	8.48×10^{-5}	3.37
10^{-5}	239	34/4	2878	177	244	602615	1.90×10^{-5}	2.11
10^{-6}	374	21/0	3566	269	326	790712	1.61×10^{-6}	1.80
10^{-7}	625	9/0	5012	438	494	1194759	3.02×10^{-7}	1.58

Table 5 Numerical results for Robertson (15) by testing order-4 methods

tol	nsteps	rej(err/newt)	nfcn	njac	lu	flops	maxerr	niter
SIRK								
10^{-3}	130	0/10	1544	126	140	398590	2.77×10^{-10}	2.97
10^{-4}	176	0/0	1984	172	176	517918	8.57×10^{-10}	2.82
10^{-5}	276	0/0	2672	267	273	748337	7.02×10^{-11}	2.42
10^{-6}	438	1/0	3728	428	437	1114730	2.90×10^{-11}	2.12
10^{-7}	694	1/0	5280	678	688	1674494	1.17×10^{-11}	1.90
DESI								
10^{-5}	114	2/0	2166	112	116	681725	3.31×10^{-10}	2.67
10^{-6}	163	0/0	2796	161	162	921812	8.68×10^{-11}	2.45
10^{-7}	247	1/0	3936	245	248	1357555	4.69×10^{-12}	2.27
10^{-8}	375	2/0	5697	371	376	2022189	1.27×10^{-13}	2.16
10^{-9}	576	2/0	8419	568	576	3055321	8.82×10^{-14}	2.08
ESIRK								
10^{-3}	156	1/0	752	152	155	343949	3.08×10^{-9}	1.19
10^{-4}	220	1/0	1012	159	194	470783	1.11×10^{-9}	1.14
10^{-5}	319	0/0	1428	192	239	667322	2.86×10^{-10}	1.11
10^{-6}	481	0/0	2116	253	315	991362	6.05×10^{-11}	1.10
10^{-7}	741	2/0	3216	341	414	1511945	1.20×10^{-11}	1.08
DESIRE								
10^{-5}	177	2/1	2158	159	174	796219	6.11×10^{-10}	2.01
10^{-6}	199	3/0	1864	193	201	845171	6.68×10^{-11}	1.53
10^{-7}	289	2/0	2350	277	288	1160511	7.69×10^{-12}	1.34
10^{-8}	439	2/0	3380	419	432	1724404	1.27×10^{-12}	1.28
10^{-9}	676	2/0	4975	625	648	2609413	1.82×10^{-13}	1.22

In the case of order 2 methods, DESI methods does not benefit from the three additonal stages. Even they take much less steps than those taken by their related SIRK methods, DESI methods still take more function evaluations and flops when we have same tolerence demanding. But it is not the case for effective order methods. DESIRE methods have smaller truncation error because of the smaller error constants. On the other hand, the ESIRK method does not benefit from effective order; its performance is very similar to the SIRK method but it has a slightly greater average number of iterations and is less efficient than the SIRK method. When using variable stepsize, the normalized error constant for the ESIRK method with abscissae in [0, 1] is larger than the classical SIRK method [10]. For DESI and DESIRE methods, the latter seems to have advantages over the former because of effective order.

For order 3 and order 4 methods, the positive influences of additional stages are apparent. When the methods attain the satisfied accuracy, DESI and DESIRE methods reduce the total flops from SIRK and ESIRK methods respectively. The

application of effective order also improves the efficiency. ESIRK methods are overall more efficient than SIRK methods. Some of the abscissae are greater than 1 for SIRK methods and the ESIRK methods have the advantage over the SIRK methods when using variable stepsize [10]. For DESI and DESIRE methods, it is not easy to converge when demanding less accuracy due to the large stepstize taken. But it still can be seen the positive effect from effective order.

Generally, it can be concluded that the second order SIRK method does not have abscissae greater than 1. Using "additional diagonal stages" or "effective order" does not improve the numerical performance. When order ≥ 3, the adoption of effective order successfully improve the numerical results by moving all the abscissae of SIRK methods inside the integration interval. For DESI or DESIRE type methods, because of the much smaller error constants, they are more efficient in high precision. Methods with order ≥ 3 are regarded as the "high order" methods. Because the stepsizes taken by DESI or DESIRE methods are much larger than the corresponding SIRK (ESIRK) methods, a good starting value and a severe stopping criterion for Newton iterations are necessary. Furthermore, some implementation strategies need more conservative parameters.

6 Conclusions

In this paper, we have shown that the family of singly-implicit methods is a good candidate for the solutions of stiff problems. It can also be summarized as below:

1. The second order SIRK method does not have abscissae greater than 1. Using "effective order" does not improve the numerical performance, it is not obvious that the additional diagonally stages can improve the numercial results.
2. The third order methods improve the numerical performance because of effective order and the additional diagonally stages. When adding one more stage, they improve the efficiency more than other order methods.
3. When order ≥ 3, the adoption of effective order seem have some advantages with variable stepsize.
4. In general, for mildly stiff problems, adding more additional diagonally stages improves the numerical efficiency more. For more stiff problems, efficiency improved by additional diagonally stages or by effective order can be summaried as below:

Order	Additional diagonally stages	Effective order
$p = 2$	not improved	not improved
$p = 3$	improved	improved
$p \geq 4$	improved	improved

5. Additional diagonally stages applied to ESIRK methods improves the efficiency for all order. In other words, the DESI type methods improve the efficiency because of the application of effective order.

Acknowledgments The author wants to thank Professor Butcher and two reviewers for their value comments and also thank the Mathematics Department of the University of Auckland for offering facilities while doing this research and to all members in the numerical group. This research is partly supported by the National Science Council, Taiwan.

References

1. Alexander, R.: Diagonally implicit Runge-Kutta methods for stiff ODEs. SIAM J. Numer. Anal. **14**, 1006–1021 (1977)
2. Burrage, K.: A special family of Runge-Kutta methods for solving stiff differential equations. BIT **18**, 22–41 (1978)
3. Burrage, K., Butcher, J.C., Chipman, F.H.: STRIDE: Stable Runge-Kutta integrator for differential equations. Computational Mathematics Report No. 20. University of Auckland (1979)
4. Butcher, J.C.: The effective order of Runge-Kutta methods, conference on the numerical solution of differential equations. Lect. Notes Math. **109**, 133–139 (1969)
5. Butcher, J.C.: On the implementation of implicit Runge-Kutta methods. BIT **16**, 237–240 (1976)
6. Butcher, J.C.: The numerical analysis of ordinary differential equations. Wiley (2008)
7. Butcher, J.C., Cash, J.: Towards efficient Runge-Kutta methods for stiff systems. SIAM J. Numer. Anal. **27**, 753–761 (1990)
8. Butcher, J.C., Chartier, P.: A generalization of Singly-Implicit Runge-Kutta methods. Appl. Numer. Math. **24**, 343–350 (1997)
9. Butcher, J.C., Chartier, P.: The Effective Order of Singly-Implicit Runge-Kutta methods. Numer. Algo. **20**, 269–284 (1999)
10. Butcher, J.C., Chen, D.J.L.: ESIRK methods and variable stepsize. Appl. Numer. Math. **28**, 193–207 (1998)
11. Butcher, J.C., Chen, D.J.L.: On the implementation of ESIRK methods for stiff IVPs. Numer. Algo. **26**, 201–218 (2001)
12. Chen, D.J.L.: The effective order of singly-implicit methods for stiff differential equations. PhD thesis, The University of Auckland, New Zealand (1998)
13. Butcher, J.C., Diamantakis, M.T.: DESIRE: diagonally extended singly implicit Runge-Kutta effective order methods. Numer. Algo. **17**, 121–145 (1998)
14. Diamantakis, M.T.: Diagonally extended singly implicit Runge-Kutta methods for stiff initial value problems. PhD thesis, Imperial College, University of London (1995)
15. Hairer, E., Wanner, G.: Solving ordinary differential equations II, stiff and differential-algebraic problems. Springer-Verlag, Berlin (1987)
16. Kaps, P.: The Rosenbrock-type methods. In: Dalhquist, G., Jeltsch, R. (eds.) Numerical methods for stiff initial value problems (Proceeding, Oberwolfach). Bericht Nr. 9, Institut fur Geometrie und Praktische Mathematik, RWTH Aachen, Germany (1981)