CWI Quarterly Volume 11 (1) 1998, pp. 101 – 111

Parallel MIRK Methods for ODEs

David Voss
Department of Mathematics
Western Illinois University
Macomb, IL 61455, U.S.A
e-mail: d-voss1@wiu.edu

Implicit Runge–Kutta (IRK) methods are cherished for their stability properties when solving Ordinary Differential Equations (ODEs). Unfortunately, computational complexities render them less competitive than implicit Linear Multistep methods overall. Several modifications have surfaced to make IRKs more viable. One such modification that arose almost two decades ago in an attempt to make IRKs similar in complexity to implicit Linear Multistep methods is denoted Mono-Implicit Runge-Kutta (MIRK) methods. In this paper, progress in this area is surveyed via parallel MIRK methods for initial value ODE systems.

1. Introduction

The numerical solution of systems of initial value ordinary differential equations, i.e. initial value problems (IVPs), of the form,

$$y'(t) = f(y(t)), \quad y(t_0) = y_0, \tag{1}$$

where $y \in \mathbb{R}^m$ and $f : \mathbb{R}^m \to \mathbb{R}^m$ has received considerable attention in the 20^{th} century. Classical accountings include Burrage [4], Butcher [7], Hairer and Wanner [15], and back to Henrici [16]. When the IVP is stiff, implicit Runge–Kutta (IRK) methods (see, for example [7] and references therein) are commonly used to provide a numerical solution. For the nth step, using a stepsize h, an s-stage IRK method has the form

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i f(Y_i), \tag{2}$$

with

$$Y_i = y_n + h \sum_{j=1}^{s} a_{i,j} f(Y_j), \quad i = 1, \ldots, s. \tag{3}$$

Note that each unknown, Y_i, is defined implicitly in terms of itself and the other unknowns. These schemes are normally given in terms of the compact Butcher tableau

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array},$$

where $c = Ae$, $c = (c_1, c_2, \ldots, c_s)^T$, $b = (b_1, b_2, \ldots, b_s)^T$ and A is the s by s matrix whose (i,j)th component is $a_{i,j}$, and e is the s-dimensional vector of 1's.

Newton's method is usually employed to solve the system of $s \times m$ nonlinear equations given in (3) in order to determine the intermediate values, Y_i. As pointed out by several authors (see, for example, REUSCH ET AL. [21]), this is one of the undesirable features of implicit Runge–Kutta methods rendering them less competitive than other methods, such as Backward Differentiation Formulas (BDFs). More specifically, Newton's method leads to an iteration matrix $(I_{ms} - hA \otimes J)$, where J is an approximation to the Jacobian $\frac{\partial f}{\partial y}$. Since the costs of the linear algebra associated with the solution of the resulting linear systems generally dominate the overall cost of the computation, many subclasses of IRK methods, such as diagonally implicit (DIRK) methods [1], singly implicit (SIRK) methods [5], mono–implicit (MIRK) methods [11], multi–implicit Runge–Kutta methods (for example, see [2] where they are also referred to as MIRK methods), and parallel diagonally–implicitly iterated Runge–Kutta (PDIRK) methods ([17], [12]) have been developed to attempt to reduce these costs, usually by decoupling this large system of $s \times m$ equations into s systems each of dimension m.

In this paper we are concerned with the design of MIRK methods that are inherently parallel in that the s systems of equations are apportioned to s concurrent processors, that is, parallelism across the method. The following sections consider the evolution of this design beginning with the original MIRKs discussed by CASH [8], and formally designated MIRKs by CASH and SINGHAL [11].

2. SEQUENTIAL MIRK SCHEMES

When applied to (1) with integration stepsize h, MIRK methods have the form

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i f(Y_i), \tag{4}$$

where

$$Y_i = (1 - v_i)y_n + v_i y_{n+1} + h\sum_{j=1}^{i-1} x_{i,j} f(Y_j), \quad i = 1, \ldots, s. \tag{5}$$

Thus the stages of a mono–implicit Runge–Kutta scheme are implicit only in y_{n+1}. MIRK methods are usually represented by the modified tableau

$$\begin{array}{c|c|c} c & v & X \\ \hline & & b^T \end{array},$$

where $v = (v_1, v_2, \ldots, v_s)^T$, $c = v + Xe$, and X, the s by s matrix whose (i,j)th component is $x_{i,j}$, is strictly lower triangular. The MIRK scheme (4)–(5) is equivalent to the IRK scheme (2)–(3) with $A = X + vb^T$.

The computational advantages associated with MIRK formulas, compared with fully implicit Runge–Kutta formulas, were first pointed out in Cash [8]. Cash proposed the general class of formulas to be of form (4)–(5), with

$$\begin{aligned} v_i &= 1, \quad 1 \le i \le r, \\ v_i &= 0, \quad x_{i,j} = 0, \quad r+1 \le i \le s, \quad 1 \le j \le r, \end{aligned} \tag{6}$$

where r is an integer satisfying $s - r \le [\frac{1}{2}s]$ which, consequently, yields the potential for A–stability. With this class of MIRKs, the system of m nonlinear equations implicitly defining y_{n+1} is given by

$$F(y_{n+1}) \equiv y_{n+1} - y_n - h\sum_{i=1}^{s} b_i f(Y_i), \tag{7}$$

and the Newton iteration scheme for the solution of this system is

$$J_F(y_{n+1}^{(l)})\Delta y_{n+1}^{(l)} = -F(y_{n+1}^{(l)}), \quad y_{n+1}^{(l+1)} = y_{n+1}^{(l)} + \Delta y_{n+1}^{(l)}, \quad l = 0, 1, \ldots. \tag{8}$$

Note that the expression for $J_F(y_{n+1}^{(l)})$ involves the evaluation of $\frac{\partial f}{\partial y}$ at several points. For example, with $s = 3$ and $r = 2$, this particular brand of MIRK formulas has form

$$\begin{aligned} y_{n+1} &= y_n + h\sum_{i=1}^{3} b_i f(Y_i) \\ Y_1 &= y_{n+1} \\ Y_2 &= y_{n+1} + hx_{2,1} f(Y_1). \\ Y_3 &= y_n. \end{aligned} \tag{9}$$

The corresponding system of equations to be solved has form (8) where

$$J_F(y_{n+1}^{(l)}) = I - hb_1 J_1(y_{n+1}^{(l)}) - hb_2 J_2(y_{n+1}^{(l)})[I + hx_{2,1} J_1(y_{n+1}^{(l)})] \tag{10}$$

and

$$J_1(y_{n+1}^{(l)}) = \frac{\partial f}{\partial y}(y_{n+1}^{(l)})$$
$$J_2(y_{n+1}^{(l)}) = \frac{\partial f}{\partial y}(y_{n+1}^{(l)} + hx_{2,1}f(y_{n+1}^{(l)})).$$

While the linear system has only dimension m, the Jacobian matrix $\frac{\partial f}{\partial y}$ must be evaluated twice and the product $J_2(y_{n+1}^{(l)})J_1(y_{n+1}^{(l)})$ computed. These computational aspects were considered in [9] where the coefficients of the MIRK formula were chosen so that, for example, the Newton iteration matrix (10) factorizes exactly as

$$J_F = (I - b_2hJ_2)(I - b_1hJ_1), \tag{11}$$

from which it necessarily follows that $b_2 = -x_{2,1}$. Even with this factorization, CASH and SINGHAL [11] note that the amount of work required to solve the system of equations is still twice that of linear multistep methods. The usual modification of Newton's method approximates the partial derivatives in J_F at the same point rendering it a polynomial in $J \equiv \frac{\partial f}{\partial y}$. With this modification, a particularly efficient second order MIRK arises as the Newton iteration matrix (10) factorizes as a perfect square

$$J_F = (I - \beta hJ)^2, \tag{12}$$

where $\beta = 1 - \frac{1}{\sqrt{2}}$, and the remaining coefficients are given by [11]:

$$b_3 = 1 - 2\beta, \quad b_2 = \frac{(b_3 - \frac{1}{2})^2}{\frac{1}{3} - b_3}, \quad b_1 = 1 - b_3 - b_2, \quad x_{2,1} = \frac{\frac{1}{3} - b_3}{b_3 - \frac{1}{2}}. \tag{13}$$

MIRKs of orders up to and including six have been proposed by CASH ([8], [9],[10]), and by VAN BOKHOVEN [3] where they were denoted implicit endpoint quadrature formulas. Higher order formulas do not readily admit perfect power factorizations of J_F and CASH and SINGHAL [11] followed the approach of SKEEL and KONG [22] wherein J_F in a sense nearly factorizes as a power of a single matrix to generate efficient higher order, L–stable MIRKs.

Applying MIRK schemes (4)–(5)–(6) to the scalar test equation $\dot{y} = \lambda y$, $Re\lambda < 0$, we obtain $y_{n+1} = R(q)y_n$, with rational stability function

$$R(q) = \frac{N(q)}{D(q)}, \quad q = \lambda h, \tag{14}$$

where $N(q)$ is a polynomial of degree $s - r$ and $D(q)$ is a polynomial of degree r. For A–stability, using the concept of an E–polynomial developed by Nørsett [19] and defining

$$E(y^2) = |D(iy)|^2 - |N(iy)|^2,$$

the corresponding methods will be A–stable if, and only if, $D(q)$ has no zeros in the left–hand plane $Re(q) < 0$, and $E(y^2) \geq 0$ for all positive real arguments. NØRSETT and WOLFBRANDT [20] considered rational approximations to e^q with real poles of the form

$$\frac{1+\alpha_1 q+\alpha_2 q^2+\alpha_n q^n}{(1-\beta_1 q)(1-\beta_2 q)\dots(1-\beta_m q)} \approx e^q, \tag{15}$$

and showed that, if $m \geq n$, the maximum obtainable order for A-stable methods is $n+1$ and that the approximation of order $n+1$ with least absolute value of the error constant occurs in the case of repeated poles, $\beta_i = \beta$, $1 \leq i \leq m$. For example, with $n = 1$ and $m = 2$ in(15),

$$\frac{1+\alpha q}{(1-\beta q)^2}, \quad \alpha = \sqrt{2}-1, \quad \beta = 1-\frac{\sqrt{2}}{2}, \tag{16}$$

is the second order approximation to e^q with smallest error constant

$$C_3^* = \frac{4-3\sqrt{2}}{6} \approx -0.0404, \tag{17}$$

and coincides with the stability function of the second order MIRK (13) possessing a perfect square iteration matrix (12).

3. Parallel MIRK schemes

MIRKs with perfect power Newton iteration matrices were built for sequential computers. The potential for parallelism across the method was investigated by Voss ([24], [25]) for a special brand of MIRKs, denoted PaMIRK(r), of form (4)–(5)–(6) with $s = 2r-1$, that is, with r implicit stages and $r-1$ explicit stages.

With r implicit stages, the Newton iteration has the form

$$J_F(y_{n+1}) = I - h\sum_{i=1}^{r} b_i J_i B_i \tag{18}$$

with

$$J_i \equiv J_i(y_{n+1}) = J_f(y_{n+1} + h\sum_{j=1}^{i-1} x_{i,j} f(Y_j))$$

and

$$B_i = I + h\sum_{j=1}^{i-1} x_{i,j} J_j B_j,$$

for $1 \leq i \leq r$.

As in [10], factorizing (18) into the r linear factors

$$J_F(y_{n+1}) = \prod_{i=0}^{r-1} (I - b_{r-i} h J_{r-i}) \tag{19}$$

requires that for $1 \leq j \leq r-1$,

$$x_{i,j} = -b_j, \quad i = j+1, \dots, r. \tag{20}$$

This natural factorization still leaves free $\frac{r}{2}(r+1)$ of the r^2 parameters occurring in the MIRKS. Natural parallelism surfaces if the usual modified Newton iteration scheme resulting from setting $J_i = J, 1 \leq i \leq r$ is used, and the inverse of the iteration matrix in (19) is decomposed into a partial fraction expansion of the form

$$J_F^{-1}(y_{n+1}) = \sum_{i=1}^{r} w_i (I - b_i h J)^{-1}, \tag{21}$$

from which it follows that

$$w_i = \frac{b_i^{r-1}}{\prod_{\substack{j=1 \\ j \neq i}}^{r} (b_i - b_j)}, \quad 1 \leq i \leq r, \tag{22}$$

provided $\{b_i\}_{i=1}^{r}$ are distinct.

The Butcher matrix for this brand of MIRKs has form

$$A = \begin{bmatrix} b_1 & b_2 & b_3 & \dots & b_r & b_{r+1} & b_{r+2} & \dots & \dots & b_s \\ 0 & b_2 & b_3 & \dots & b_r & b_{r+1} & b_{r+2} & \dots & \dots & b_s \\ 0 & 0 & b_3 & \dots & b_r & b_{r+1} & b_{r+2} & \dots & \dots & b_s \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & b_r & b_{r+1} & b_{r+2} & \dots & \dots & b_s \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & x_{r+2,r+1} & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & x_{r+2,r+1} & x_{r+2,r+2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 & x_{s,r+1} & x_{s,r+2} & \dots & x_{s,s-1} & 0 \end{bmatrix},$$

and it is immediately apparent that the natural factorization (19)–(20) results in the Butcher matrix having r real distinct eigenvalues, $b_i, i = 1 \dots r$, and a $(r-1)$–fold eigenvalue at 0. In general, the potential for parallelism in IRK methods arises when the Butcher matrix has real and distinct eigenvalues as a similarity transformation can be applied to decouple the stages so that each stage can be performed in parallel. In the case of PaMIRK(r) methods this transformation is unnecessary since with the expansion (21), using Newton's method to resolve the r implicit stages involves the solution of r independent real linear systems of the form

$$(I - b_i h J(y_{n+1}^{(l)}))\, \triangle_i\, y_{n+1}^{(l)} = -F(y_{n+1}^{(l)}), \quad 1 \leq i \leq r,$$

on r processors. The resulting increment is

$$\Delta y_{n+1}^{(l)} = \sum_{i=1}^{r} w_i\, \triangle_i\, y_{n+1}^{(l)}$$

and, as depicted by SWEET [23], a highly recursive multiplicative algorithm is converted into an additive parallel algorithm.

For this brand of MIRKs the stability function has form (14) where $N(q)$ is a polynomial of degree $r-1$ and

$$D(q) = \prod_{i=1}^{r}(1 - b_i q).$$

Consequently, the resulting methods will be strongly stable at infinity if $b_i \neq 0, 1 \leq i \leq r$. $D(q)$ has no zeros in the left–hand plane $Re(q) < 0$ if

$$b_i > 0, \quad 1 \leq i \leq r. \tag{23}$$

With the constraints (20) and (23), the remaining parameters were determined so that the corresponding parallel Runge–Kutta schemes listed in [25] were L–stable and possessed order r, $r \leq 4$. In particular, the second order PaMIRK(2) method

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{12}(3f(Y_1) + 4f(Y_2) + 5f(Y_3)) \\ Y_1 &= y_{n+1} \\ Y_2 &= y_{n+1} - \frac{h}{4}f(Y_1). \\ Y_3 &= y_n, \end{aligned} \tag{24}$$

was used to provide an efficient coarse grain time–stepping parallel algorithm in the solution of linear, multidimensional second order time dependent PDEs via the Method of Lines semidiscretization approach [26]. Its stability function

$$R(q) = \frac{1 + \frac{5}{12}q}{(1 - \frac{1}{3}q)(1 - \frac{1}{4}q)}$$

possesses error constant $C_3 = -\frac{1}{24} \approx C_3^*$, where C_3^* is the optimum error constant given in (17).

Unfortunately, in common with DIRKs, these MIRKs all possessed stage order one which is of some concern since the phenomenon of order reduction [14] can arise with the potential of causing these IRKs to behave as if their order were only their stage order. While convenient, the natural factorization (20) severely restricted the MIRK stage order due primarily to the presence of explicit stages. Recently Voss and Muir [27] investigated the full class of MIRKs, denoted MIRKspq, indicating a MIRK scheme having s stages, of order p, and having stage order q. Returning to the modified Newton iteration scheme, $J_i = J$, for this case, (10) has the form

$$J_F(y_{n+1}) = I - (b^T v)hJ - (b^T X v)(hJ)^2 - \ldots - (b^T X^{s-1} v)(hJ)^s, \tag{25}$$

and, again, parallelism across the method surfaces if (25) is expressible in the form

$$J_F(y_{n+1}^{(l)}) = \prod_{i=1}^{s}(I - \beta_i hJ(y_{n+1}^{(l)})), \tag{26}$$

where $\beta_i, i = 1 \ldots s$, are distinct. The Butcher matrix for the general MIRK class (4)–(5) is given by

$$A = \begin{bmatrix} v_1 b_1 & v_1 b_2 & \ldots & v_1 b_s \\ x_{2,1} + v_2 b_1 & v_2 b_2 & \ldots & v_2 b_s \\ \vdots & \vdots & \ddots & \vdots \\ x_{s,1} + v_s b_1 & x_{s,2} + v_s b_2 & \ldots & v_s b_s \end{bmatrix}.$$

For general IRKs (2)–(3) the linear stability function can be written in the form (see, for example, DEKKER and VERWER [13])

$$R(q) = \frac{det[I - qA + qeb^T]}{det[I - qA]}. \tag{27}$$

More recently, Muir and Enright [18] give $R(q)$ for MIRKs (4)–(5) in the form

$$R(q) = \frac{1 + qb^T(I - qX)^{-1}(e - v)}{1 - qb^T(I - qX)^{-1}v}. \tag{28}$$

Since X is strictly lower triangular,

$$(I - qX)^{-1} = I + qX + q^2X^2 + \ldots + q^{s-1}X^{s-1},$$

so that the denominator in (28) becomes

$$D(q) = 1 - (b^T v)q - (b^T X v)q^2 - \ldots - (b^T X^{s-1} v)q^s. \tag{29}$$

Clearly, $D(hJ) = J_F$ in (25) and the goal is to find distinct $\beta_i, i = 1 \ldots s$, that is, the eigenvalues of the Butcher matrix A, such that $\prod_{i=1}^{s}(1 - \beta_i q) = D(q)$. Expanding and equating the coefficients of like powers of q results in the system of equations for β_i:

$$\begin{aligned} &\beta_1 + \ldots + \beta_s = b^T v, \\ &\beta_1 \cdot \beta_2 + \beta_1 \cdot \beta_3 + \ldots + \beta_{s-1} \cdot \beta_s = -b^T X v, \\ &\quad \ldots, \\ &\beta_1 \cdot \ldots \cdot \beta_s = (-1)^{s-1} b^T X^{s-1} v. \end{aligned} \tag{30}$$

A MIRK scheme has order p if its local error is $O(h^{p+1})$; for Runge–Kutta schemes this is imposed by requiring the coefficients of the scheme to satisfy a set of equations called order conditions (see [7]). A MIRK scheme has stage order q if it has coefficients which satisfy the conditions,

$$Xc^{k-1} + \frac{v}{k} = \frac{c^k}{k}, \quad k = 1, \ldots, q. \tag{31}$$

Order barriers for this class of MIRKs were established by BURRAGE ET AL. [6] and are given in the next two theorems.

THEOREM 3.1. *The maximum order of an s–stage MIRK cannot exceed* $s+1$.

THEOREM 3.2. *The maximum stage order of an s–stage MIRK is min(s,3).*

In [27], parallel MIRK methods through order 4 were derived possessing stage order at most 3, the maximum possible. The derivation process employed consisted of selecting or determining families of MIRK schemes, in terms of the β_i parameters, with a given number of stages, a given order, and a given stage order. Subject to the restrictions that the β_i's be real, distinct, and positive, and that the MIRK scheme be A–stable, free parameters were chosen to yield optimal schemes according to the following criteria: (a) minimize $||T_{p+1}||$, the norm of the vector of truncation error coefficients of order $p+1$ associated with the MIRK scheme (see [7]), subject to the constraint that the ratio of $||T_{p+2}||$ to $||T_{p+1}||$ is not too large, (b) minimize $||w||$, the norm of the vector of w_i coefficients arising in (21), and (c) minimize $||\beta||$, the norm of the vector of β_i coefficients.

For example, an L–stable MIRK222 is given by the following tableau along with its stability function

$$\begin{array}{c|c|cc} 1 & 1 & 0 & 0 \\ \frac{4}{45} & \frac{344}{2025} & -\frac{164}{2025} & 0 \\ \hline & & \frac{37}{82} & \frac{45}{82} \end{array}, \quad R(q) = \frac{1+\frac{41}{90}q}{(1-\frac{q}{10})(1-\frac{4q}{9})}. \tag{32}$$

It has $||T_3|| \approx 0.086$, $||w|| \approx 1.3$, $||\beta|| \approx 0.46$, and $||T_4|| \approx 0.11$. The superior performance of this and other MIRK*spr* methods on problems (for example, see [12]) where DIRK schemes suffer order reduction appears in [27].

4. CONCLUSION

The design of parallel MIRKs was addressed, and on a machine with at least p processors, the computational complexity of a p^{th}–order parallel MIRK method is similar to that of an implicit Linear Multistep method as it effectively requires only one implicit stage per step whose solution involves a linear function of the Jacobian. Moreover, unlike DIRK methods which have stage order at most one, higher order MIRK methods with stage order up to and including three have been determined [27].

REFERENCES

1. R. ALEXANDER (1977). Diagonally implicit Runge–Kutta methods for stiff ODEs. *SIAM J. Numer. Anal.* **14**, 1006–1021.
2. C. BENDTSEN (1997). A parallel stiff ODE solver based on MIRKs. *Adv. Comput. Math.* **7**, 27–36.

3. W.M.G. van Bokhoven (1980). Implicit End–Point Quadrature formulae. *BIT* **3**, 87–89.
4. K. Burrage (1995). *Parallel and Sequential Methods for Ordinary Differential Equations*, Clarendon Press, Oxford.
5. K. Burrage (1978). A special family of Runge–Kutta methods for solving stiff differential equations. *BIT* **18**, 22–41.
6. K. Burrage, F.H. Chipman, P.H. Muir (1994). Order Results For Mono–Implicit Runge–Kutta Methods. *SIAM J. Numer. Anal.* **31**, 876–891.
7. J.C. Butcher (1987). *The Numerical Analysis of Ordinary Differential Equations*, Wiley, Chichester .
8. J.R. Cash (1975). A Class of Implicit Runge–Kutta Methods for the Numerical Integration of Stiff Differential Systems. *J. ACM* **22**, 504–511.
9. J.R. Cash (1977). On a class of implicit Runge–Kutta procedures. *J. Inst. Maths. Applics.* **19**, 455–470.
10. J.R. Cash (1977). A note on the computational aspects of a class of implicit Runge–Kutta procedures. *J. Inst. Maths. Applics.* **20**, 425–441.
11. J.R. Cash, A. Singhal (1982). Mono–Implicit Runge–Kutta Formulae for the Numerical Integration of Stiff Differential Systems. *IMA J. Numer. Anal.* **2**, 211–217.
12. N. Cong (1994). A parallel DIRK method for stiff initial–value problems. *J. Comp. Appl. Math.* **54**, 121–127.
13. K. Dekker, J.G. Verwer (1984). *Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations*. North–Holland, Amsterdam .
14. R. Frank, J. Schneid, C.W. Überhuber (1985). Order results for implicit Runge–Kutta methods applied to stiff systems. *SIAM J. Numer. Anal.* **23**, 515–534.
15. E. Hairer, G. Wanner (1991). *Solving Ordinary Differential Equations II, Stiff and Differential–Algebraic Equations*, Springer–Verlag, Berlin.
16. P. Henrici (1962). *Discrete Variable Methods in Ordinary Differential Equations*, Wiley, New York.
17. P.J. van der Houwen, B.P. Sommeijer, W. Couzy (1992). Embedded diagonally implicit Runge–Kutta algorithms on parallel computers. *Math. Comp.* **58**, 135–159.
18. P.H. Muir, W.H. Enricht (1987). Relationships among some classes of IRK methods and their stability functions. *BIT* **27**, 403–423.
19. S.P. Nørsett (1975). C-polynomials for rational approximation to the exponential function. *Numer. Math.* **25**, 39–56.
20. S.P.Nørsett, A. Wolfbrandt (1977). Attainable order of rational approximations to the exponential function with only real poles. *BIT* **17**, 200–208.
21. M.F. Reusch, L. Ratzan, N. Pomphrey, W. Park (1988). Diagonal Padé approximations for initial value problems. *SIAM J. Sci. Statist. Comput.* **9**, 829-838.
22. R.D. Skeel, A.K. Kong (1977). Blended linear multistep methods. *ACM*

Trans. Math. Software **3**, 326–345.
23. R.A. Sweet (1988). A parallel and vector variant of the cyclic reduction algorithm. *SIAM J. Sci. Statist. Comput.* **9**, 761–765.
24. D.A. Voss (1993). A second–order Mono–Implicit Runge–Kutta method for parallel computers. *Proceedings of Advances in Scientific Computing & Modeling*, 71-75.
25. D.A. Voss (1995). Parallel Mono–Implicit Runge–Kutta methods for stiff ODEs, Report, Dept. of Math., Western Illinois University.
26. D.A. Voss, D.A.Q.M. Khaliq (1995). Parallel LOD methods for second order time dependent PDEs. *Computers Math. Applic.* **30**, 25–35.
27. D.A. Voss, P.H. Muir (1997). Mono–Implicit Runge–Kutta schemes for the parallel solution of initial value ODEs, Report, Dept. of Math., Western Illinois University.