ANALISIS 6 (1&2), 61-74 (1999)

A NEW PARALLEL 3-POINT EXPLICIT BLOCK METHOD FOR SOLVING SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS DIRECTLY

ZURNI OMAR & MOHAMED SULEIMAN

ABSTRACT

A new parallel 3-point explicit block method for solving second order ordinary differential equations (ODEs) directly is developed. The method computes the numerical solution of the equations at three points simultaneously. Each problem was tested on the shared memory parallel computer Sequent S27 using both the sequential and parallel implementations of the new method and the conventional 1-point method. Numerical results are presented comparing the two methods in terms of the number of steps taken, accuracy and execution times. The results indicate that the parallel implementation of the new method is the best choice for solving second order ODEs directly, particularly when the step size becomes finer.

ABSTRAK

Satu kaedah baru blok selari 3-titik tak tersirat bagi menyelesaikan persamaan pembezaan biasa peringkat dua secara langsung dirumuskan. Kaedah tersebut menghitung penyelesaian berangka pada tiga titik serentak. Setiap masalah diuji pada komputer selari berkongsi ingatan Sequent S27 dengan menggunakan kedua-dua perlaksanaan bersiri dan selari kaedah baru tersebut dan kaedah 1-titik konvensional. Keputusan berangka dipersembahkan dengan membanding kedua-dua kaedah dari segi bilangan langkah, kejituan dan masa perlaksanaan. Keputusan tersebut menunjukkan implementasi selari bagi kaedah yang baru merupakan pilihan yang terbaik bagi menyelesaikan persamaan pembeza biasa peringkat dua terutamanya bila saiz langkah yang digunakan semakin kecil.

INTRODUCTION

In this paper, we consider solving the following second order ordinary differential equation (ODE),

$$y'' = f(x, y, y'), y(a) = y_0, y'(a) = y'_0, a \le x \le b. \qquad [1]$$

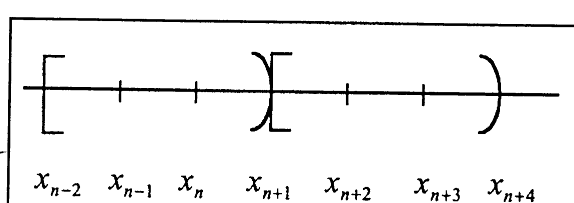
Currently, there are two techniques available for solving Equation [1]. The first technique is to reduce [1] to a system of first order equations and then solve it using first order ordinary differentials (ODEs) methods. These methods are very well established. The second technique is to solve [1] directly as suggested by Gear (1966, 1971, 1978), Hall and Suleiman (1981) and Suleiman (1979, 1989). However, these methods compute the numerical solution at one point at a time.

Parallel block methods for numerical solutions of first order ODEs have been proposed by several researchers such as Birta and Abou-Rabia (1987), Chu and Hamilton (1987), Shampine and Watts (1969) and Tam (1989). In a block method, a set of new values that are obtained by each application of the formula is referred to as "block". For instance, in a r - point block method, r new equally spaced solution values i.e., $y_{n+1}, y_{n+2}, ..., y_{n+r}$ are obtained simultaneously at each iteration of the algorithm. There are two types of block methods; *one-step block* and *multi-block*. In the one-step block method, the new block y_{n+i}, i = 1,2,..., r, is computed from the value y_n. On the other hand, in the multi-block method, the information from one or more previous blocks is used in the computation of the next block.

In the 3-point block method, the points whose solutions are to be computed in the first block of Figure 1 are x_{n-2}, x_{n-1} and x_n, while x_{n-2}, x_{n-1} and x_{n+3} are points of the next block.

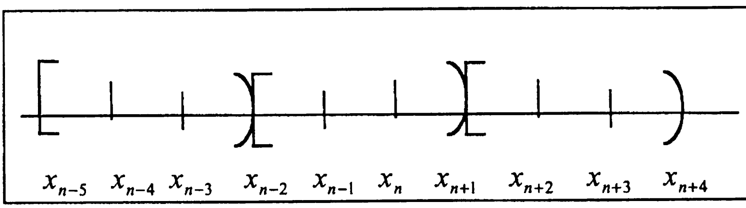
Figure 1

3-Point Method



The computation which proceeds in blocks is based on the computed values at the earlier blocks. If the computed values at the previous k blocks are used to compute the current block containing r points, then the method is called r-point k-block method. Similarly, if the computed values at the previous two blocks as shown in Figure 2 are used to compute the solutions at a current block containing three points, i.e., $x_{n+1}, x_{n+2}, x_{n+3}$, then the method is called the 3-point 2-block method.

Figure 2
3-Point 2-Block Method



The computational tasks at each point within a block are assigned to a single processor. Thus, the computations can be performed simultaneously. Though numerous parallel algorithms have been developed for solving first order ODEs, not enough effort has been made to derive parallel methods for solving ODEs of order greater than one directly.

DERIVATION OF THE 3-POINT BLOCK METHOD

The method derived in this section is the extension of work done by Omar and Suleiman (1999). Here, we attempt to find the numerical solution at three points simultaneously.

Let $x_{n+t} = x_n + th$, $t = 1,2,3$. Therefore,

$$\int_{x_n}^{x_{n+t}} y''(x)\,dx = \int_{x_n}^{x_{n+t}} f(x,y,y')\,dx$$

Define $P_{k,n}(x)$ as the interpolation polynomial which interpolates $f(x,y,y')$ at the k back values namely $\{x_{n-m} \mid m = 0,1,2,\ldots,k-1\}$ as follows

$$P_{k,n}(x) = \sum_{m=0}^{k-1} (-1)^m \binom{-s}{m} \nabla^m f_n$$

where

$$s = \frac{x - x_n}{h}$$

Approximating $f(x,y,y')$ with $P_{k,n}(x)$, we then have

$$\begin{bmatrix} y'(x_{n+1}) \\ y'(x_{n+2}) \\ y'(x_{n+3}) \end{bmatrix} = \begin{bmatrix} y'(x_n) \\ y'(x_n) \\ y'(x_n) \end{bmatrix} + \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} \qquad [2]$$

where

$$A_t = \int_{x_n}^{x_{n+t}} \sum_{m=0}^{k-1} (-1)^m \binom{-s}{m} \nabla^m f_n \, dx.$$

Replacing $dx = hds$ and changing the limit of integration in [2] leads to

$$\begin{bmatrix} y'(x_{n+1}) \\ y'(x_{n+2}) \\ y'(x_{n+3}) \end{bmatrix} = \begin{bmatrix} y'(x_n) \\ y'(x_n) \\ y'(x_n) \end{bmatrix} + h \begin{bmatrix} \sum_{m=0}^{k-1} \gamma_m \nabla^m f_n \\ \sum_{m=0}^{k-1} \delta_m \nabla^m f_n \\ \sum_{m=0}^{k-1} \sigma_m \nabla^m f_n \end{bmatrix} \qquad [3]$$

where

$$\gamma_m = \int_0^1 (-1)m \binom{-s}{m} ds$$ see Henrici (1962),

$$\delta_m = \int_0^2 (-1)^m \binom{-s}{m} ds,$$ and

$$\sigma_m = \int_0^3 (-1)^m \binom{-s}{m} ds$$

The values of γ_m and δ_m are given by (Omar and Suleiman, 1999)

$$\gamma_0 = 1, \gamma_{m+1} = 1 - \sum_{r=0}^{m} \frac{\gamma_r}{m+2-r}$$
$$\delta_0 = 2, \delta_{m+1} = (m+3) - \sum_{r=0}^{m} \frac{\delta_r}{m+2-r} \qquad [4]$$

The main concern now is to determine the values of σ_m. Let $J(t)$ be the generating function defined as follows,

$$\begin{aligned} J(t) &= \sum_{m=0}^{\infty} \sigma_m t^m \\ &= \sum_{m=0}^{\infty} (-t)^m \int_0^3 \binom{-s}{m} ds \\ &= \int_0^3 e^{-s \log(1-t)} ds . \end{aligned}$$

The result after performing the integration is given by,

$$J(t) = \frac{t(t^2 - 3t + 3)}{-(1-t)^3 \log(1-t)}$$

which can be written as,

$$\left(\sum_{m=0}^{\infty} \sigma_m t^m\right)\left(\frac{-\log(1-t)}{t}\right) = \frac{3 - 3t + t^2}{(1-t)^3} . \qquad [5]$$

Substituting $\frac{1}{(1-t)^3} = (1 + 3t + 6t^2 + 10t^3 + ...)$ and $\frac{-\log(1-t)}{t} = (1 + \frac{1}{2}t + \frac{1}{3}t^2 + ...)$ in [5], and on simplifying we get

$$\sigma_0 = 3,$$
$$\sigma_{m+1} = \frac{(m+4)(m+3)}{2} - \sum_{r=0}^{m} \frac{\sigma_r}{m+2-r} \qquad \text{for } m = 0,1,2,...$$

Note that formula [3] can be written in the form,

$$\begin{bmatrix} y'(x_{n+1}) \\ y'(x_{n+2}) \\ y'(x_{n+3}) \end{bmatrix} = \begin{bmatrix} y'(x_n) \\ y'(x_n) \\ y'(x_n) \end{bmatrix} + h \begin{bmatrix} \sum_{m=0}^{k-1} \beta_{k-1,m} f_{n-m} \\ \sum_{m=0}^{k-1} \alpha_{k-1,m} f_{n-m} \\ \sum_{m=0}^{k-1} \tau_{k-1,m} f_{n-m} \end{bmatrix} \quad [6]$$

where

$$\beta_{k-1,m} = (-1)^m \sum_{r=m}^{k-1} \binom{r}{m} \gamma_r \quad [7a]$$

$$\alpha_{k-1,m} = (-1)^m \sum_{r=m}^{k-1} \binom{r}{m} \delta_r \quad [7b]$$

$$\tau_{k-1,m} = (-1)^m \sum_{r=m}^{k-1} \binom{r}{m} \sigma_r . \quad [7c]$$

To estimate $y(x_{n+t})$, we integrate $f(x, y, y')$ twice, i.e.,

$$\int_{x_n}^{x_{n+t}} \int_{x_n}^{x} y''(x)\, dxdx = \int_{x_n}^{x_{n+t}} \int_{x_n}^{x} f(x, y, y')\, dxdx. \quad [8]$$

which can be written as,

$$y(x_{n+t}) - y(x_n) - thy'(x_n) = \int_{x_n}^{x_{n+t}} (x_{n+t} - x) f(x, y, y')\, dx. \quad [9]$$

Substituting $f(x,y,y')$ in Equation [9] with the interpolation polynomial $P_{k,n}(x)$ lead to the following result,

$$\begin{bmatrix} y(x_{n+1}) \\ y(x_{n+2}) \\ y(x_{n+3}) \end{bmatrix} = \begin{bmatrix} y(x_n) \\ y(x_n) \\ y(x_n) \end{bmatrix} + h \begin{bmatrix} y'(x_n) \\ 2y'(x_n) \\ 3y'(x_n) \end{bmatrix} + h^2 \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} \quad [10]$$

where

$$B_t = \int_{x_n}^{x_{n+t}} (x_{n+t} - x) \sum_{m=0}^{k-1} (-1)^m \binom{-s}{m} \nabla^m f_n \, dx \, , \; t = 1,2,3.$$

Replacing $dx = hds$ and changing the limit of integration in [10] gives,

$$\begin{bmatrix} y(x_{n+1}) \\ y(x_{n+2}) \\ y(x_{n+3}) \end{bmatrix} = \begin{bmatrix} y(x_n) \\ y(x_n) \\ y(x_n) \end{bmatrix} + h \begin{bmatrix} y'(x_n) \\ 2y'(x_n) \\ 3y'(x_n) \end{bmatrix} + h^2 \begin{bmatrix} \sum_{m=0}^{k-1} \gamma_m^* \nabla^m f_n \\ \sum_{m=0}^{k-1} \delta_m^* \nabla^m f_n \\ \sum_{m=0}^{k-1} \sigma_m^* \nabla^m f_n \end{bmatrix} \qquad [11]$$

where

$$\gamma_m^* = (-1)^m \int_0^1 (1-s) \binom{-s}{m} ds \qquad [12a]$$

$$\delta_m^* = (-1)^m \int_0^2 (2-s) \binom{-s}{m} ds \qquad [12b]$$

$$\sigma_m^* = (-1)^m \int_0^3 (3-s) \binom{-s}{m} ds \qquad [12c]$$

The values of γ^*_m and σ^*_m are given by,

$$\gamma_0^* = \gamma_1 = \frac{1}{2}, \gamma_{m+1}^* = \gamma_{m+2} - \sum_{r=0}^{m} \frac{\gamma_r^*}{m-r+2} \qquad [13]$$

$$\delta_0^* = \delta_1, \delta_{m+1}^* = \delta_{m+2} - \sum_{r=0}^{m} \frac{\delta_r^*}{m+2-r}$$

To determine the value of σ_m, we first define $J^*(t)$ the generating function as follows,

$$J^*(t) = \sum_{m=0}^{\infty} \sigma_m^* t^m \, . \qquad [14]$$

Substituting [12c] in [14] produces the following resu'

$$\sum_{m=0}^{\infty}\sigma_m^{*}t^m = \sum_{m=0}^{\infty}(-t)^m\int_0^3(3-s)\binom{-s}{m}ds \qquad [15]$$

$$= \int_0^3 (3-s)e^{-s\log(1-t)}\,ds.$$

Solving the term on the right hand side of Equation [15] by applying integrati on by parts gives the following relationship,

$$\sum_{m=0}^{\infty}\sigma_m^{*}t^m = \frac{t(3-3t+t^2)}{(1-t)^3[(\log(1-t)]^2} + \frac{3}{\log(1-t)}. \qquad [16]$$

Substituting $\frac{1}{(1-t)^3} = 1+3t+6t^2+10t^3+15t^4+...,$

$\frac{t}{[\log(1-t)]^2} = \frac{1}{2(h_1\frac{t}{2}+h_2\frac{t^2}{3}+h_3\frac{t^3}{4}+...)}$ and $\frac{1}{\log(1-t)} := -\frac{1}{(t+\frac{1}{2}t^2+\frac{1}{3}t^3+...)}$ on the right hand side of Equation [16] yields

$$\sum_{m=0}^{\infty}\sigma_m^{*}t^m = [(3-3t+t^2)(1+3t+6t^2+10t^3+...)(t+\frac{1}{2}t^2+\frac{1}{3}t^3+\frac{1}{4}t^4+...)$$

$$-6(h_1\frac{t}{2}+h_2\frac{t^2}{3}+h_3\frac{t^3}{4}+...)]/[2(h_1\frac{t}{2}+h_2\frac{t^2}{3}+h_3\frac{t^3}{4}+...)(t+\frac{1}{2}t^2+\frac{1}{3}t^3\cdots)]$$

Rewrite the above equation in the following form

$$2\sum_{r=2}^{m}\sigma_{r-2}^{*}I_{m+2-r} = \frac{3(3+1)...(3+m-r)}{m!} + \sum_{r=2}^{m}\frac{3(3+1)(3+2)...(3+m-r)}{r(m-r+1)!} \qquad [17]$$

$$-\frac{6h_m}{m+1} \qquad \text{for } m = 2,3,...$$

where

$$I_m = \frac{h_1}{2(m-1)} + \frac{h_2}{3(m-2)} + ... + \frac{h_{m-2}}{(m-1)2} + \frac{h_{m-1}}{m}$$

The solution to [17] is,

$$\sigma_0^* = \sigma_1 = \frac{9}{2}$$

$$\sigma_{m+1}^* = \sigma_{m+2} - \sum_{r=0}^{m} \frac{\sigma_r^*}{m+2-r} \qquad \text{for } m = 0,1,2,\ldots$$

It follows from [11] the constant step size formulation can be written as,

$$\begin{bmatrix} y(x_{n+1}) \\ y(x_{n+2}) \\ y(x_{n+3}) \end{bmatrix} = \begin{bmatrix} y(x_n) \\ y(x_n) \\ y(x_n) \end{bmatrix} + h \begin{bmatrix} y'(x_n) \\ 2y'(x_n) \\ 3y'(x_n) \end{bmatrix} + h^2 \begin{bmatrix} \sum_{m=0}^{k-1} \beta_{k-1,m}^* f_{n-m} \\ \sum_{m=0}^{k-1} \alpha_{k-1,m}^* f_{n-m} \\ \sum_{m=0}^{k-1} \tau_{k-1,m}^* f_{n-m} \end{bmatrix} \qquad [18]$$

To determine $\beta^*_{k-1,m}$, the parameters $\beta_{k-1,m}$ and γ_r are substituted with $\beta^*_{k-1,m}$ and γ^*_m respectively using the relationship in [7a]. The values $\alpha^*_{k-1,m}$ and $\tau^*_{k-1,m}$ can be obtained in the same manner by using the relationship in [7b] and [7c] respectively.

TEST PROBLEMS

The following problems were solved numerically using the 3-point explicit block methods:

Problem 1: $y'' = -y + 2\cos x$, $y(0) = 1, y'(0) = 0, 0 \le x \le 1$

Solution: $y(x) = \cos x + x \sin x$

Problem 2: $y'' = 4y' - 4y + e^{2x}$, $y(0) = 0, y'(0) = 0, 0 \le x \le 1$

Solution: $y(x) = \frac{x^2 e^{2x}}{2}$

Problem 3: $y'' = y$, $y(0) = 1, y'(0) = 1, \le x \le 1$

Solution: $y(x) = e^x$

NUMERICAL RESULTS

The tables below show the numerial results for the problems described in the previous section when solved using the explicit 1-point (E1P) method, both sequential and parallel implementations of the 3-point explicit block (3PEB) methods with k=5. For the E1P and sequential implementaions of 3PEB methods only one processor was used. Three processors were employed for the parallel scheme of the 3PEB method.

The following notations are used in the tables:

h	Step size used
STEPS	Total number of steps taken to obtain the solution
MTD	Method employed
MAXE	Magnitude of the maximum error of the computed solution
TIME	The execution time in microseconds needed to complete the integration in a given range using the parallel computer Sequent S27.
S3PEB	Sequential implementation of the 3-point explicit block method
P3PEB	Parallel implementation of the 3-point explicit block method

The comparison of the 3PEB methods with the E1P method for solving the test problems in terms of the total number of steps, maximum error and execution times are tabulated in Tables 1-3. The maximum error is defined as follows,

$$\text{MAXE} = \max_{1 \le i \le \text{STEPS}} \left(\left| y_i - y(x_i) \right| \right).$$

Table 4 shows the ratio of steps and times of the 2PEB and 3PEB methods to E1P Methods for Solving Problem 2 of Second Order ODE When k=5. The ratio of the two parameters are obtained by dividing the parameters of the latter method with the corresponding parameters of the former methods. Hence, the ratios which are greater than one for both parameters indicate the efficiency of the 3PEB method. The ratio of time is also known as speedup.

Table 1
Comparison Between the E1P and 3PEB Methods for Solving Problem 1 of Second Order ODE When k=5

H	MTD	STEPS	MAXE	TIME
	E1P	100	4.21186(-3)	91936
10^{-2}	S3PEB	37	4.25416(-3)	78786
	P3PEB	37	4.25416(-3)	197282
	E1P	1000	4.20781(-4)	838927
10^{-3}	S3PEB	337	4.20825(-4)	670298
	P3PEB	337	4.20825(-4)	549022
	E1P	10000	4.20740(-5)	8387775
10^{-4}	S3PEB	3337	4.20740(-6)	6664967
	P3PEB	3337	4.20740(-6)	5045721
	E1P	100000	4.20736(-6)	83757255
10^{-5}	S3PEB	33337	4.20736(-6)	66524713
	P3PEB	33337	4.20736(-6)	49926496

Table 2
Comparison Between the E1P and 3PEB Methods for Solving Problem 2 of Second Order ODE When k=5

h	MTD	STEPS	MAXE	TIME
	E1P	100	3.70066(-2)	96312
10^{-2}	S3PEB	37	2.06948(-1)	82151
	P3PEB	37	2.06948(-1)	219374
	E1P	1000	3.69514(-3)	886320
10^{-3}	S3PEB	337	6.82362(-3)	712245
	P3PEB	337	6.82362(-3)	616433
	E1P	10000	3.69459(-4)	8858609
10^{-4}	S3PEB	3337	4.02459(-4)	7083500
	P3PEB	3337	4.02459(-4)	5627391
	E1P	100000	3.69454(-5)	88444224
10^{-5}	S3PEB	33337	3.72771(-5)	70711893
	P3PEB	33337	3.72771(-5)	55391405

Table 3
Comparison Between the E1P and 3PEB Methods for Solving Problem 3 of Second Order ODE When k=5

h	MTD	STEPS	MAXE	TIME
10^{-2}	E1P	100	5.87907(-3)	71262
	S3PEB	37	5.82299(-3)	60400
	P3PEB	37	5.82299(-3)	194967
10^{-3}	E1P	1000	5.87631(-4)	636554
	S3PEB	337	5.87570(-4)	492891
	P3PEB	337	5.87570(-4)	453872
10^{-4}	E1P	10000	5.87604(-5)	6362439
	S3PEB	3337	5.87603(-5)	4890910
	P3PEB	3337	5.87603(-5)	4102285
10^{-5}	E1P	100000	5.87601(-6)	63517003
	S3PEB	33337	5.87601(-6)	48780333
	P3PEB	33337	5.87601(-6)	40328641

Table 4
The Ratio Steps and Execution Times of the 3PEB Method to the E1P Method for Solving Second Order ODEs When k=5

TOL	MTD	RATIO STEP	RATIO TIME		
			PROB.1	PROB.2	PROB.3
10^{-2}	S3PEB	2.70270	1.16691	1.17238	1.17983
	P3PEB	2.70270	0.46601	0.43903	0.36551
10^{-3}	S3PEB	2.96736	1.25157	1.24440	1.29147
	P3PEB	2.96736	1.52804	1.43782	1.40250
10^{-4}	S3PEB	2.99670	1.25849	1.25060	1.30087
	P3PEB	2.99670	1.66235	1.57420	1.55095
10^{-5}	S3PEB	2.99967	1.25904	1.25077	1.30210
	P3PEB	2.99967	1.67761	1.59671	1.57499

CONCLUSION

It can be seen that in terms of the number of steps, the 3PEB method has been shown to be superior to the E1P method. As the step size becomes finer, the 3PEB method reduces the number of steps to almost one half and one third, respectively. In general, both methods are found to have the same order of accuracy.

The results demonstrate that the execution times for the sequential implementation of 3PEB method are better than those of the E1P in all test problems. As expected, the execution times taken by the parallel implementation of the 3PEB method are more than those taken by the sequential counterparts at $h = 10^{-2}$. This is because the number of steps taken is small and most of the execution times are dominated by the parallel overheads. However, the timings of the parallel versions of the 3PEB method are better when $h < 10^{-2}$. The reason for these gains is that as the step size gets smaller, more steps are taken to complete the computation. By using three processors instead of one, the computation can be performed faster. In conclusion, the parallel 3PEB method is recommended for solving second order ODEs directly when the step size becomes finer.

REFERENCES

Birta, L.G. & Abou-Rabia, O. (1987). Parallel block predictor-corrector methods for ODEs. *IEEE Transactions on Computers*, C-36(1), 299-311.

Chu, M.T. & Hamilton H. (1987). Parallel solution of ODEs by multi-block methods. *Siam J. Sci. Stat. Comput.*, 8(1), 342-353.

Gear, C.W. (1966). The numerical integration of ordinary differential equations. *Math. Comp.*, 21, 146-156.

Gear, C.W. (1971). *Numerical Initial Value Problems in Ordinary Differential Equations.* New Jersey: Prentice Hall.

Gear, C.W. (1978). The stability of numerical methods for second-order ordinary differential equations. *SIAM J. Numer. Anal.*, 15(1), 118-197.

Hall, G. & Suleiman, M.B. (1981). Stability of Adams-type formulae for second-order ordinary differential equations. *IMA J. Numer. Anal.*, 1, 427-428.

Henrici, P. (1962). *Discrete Variable Methods in Ordinary Differential Equations.* New York: John Wiley & Sons, Inc..

Omar, Z.B & Suleiman, M.B. (1999). *Solving second order ODEs directly using parallel 2-point explicit block method.* Presented at Kolokium Kebangsaan, Perayaan Ulang Tahun Ke-25 P.P.S. Matematik, Universiti Sains Malaysia.

Shampine L.F. & Watts H.A. (1969). Block implicit one-step methods. *Math. Comp.*, 23, 731-740.

Suleiman, M.B. (1979). *Generalised multistep Adams and backward differentiation methods for the solution of stiff and non-stiff ordinary differential equations.* Unpublished doctoral dissertation, University of Manchester.

Suleiman, M.B. (1989). Solving higher order ODEs directly by the direct integration method. *Applied Mathematics and Computation*, 33, 197-219.

Tam H.W. (1989). *Parallel methods for the numerical solution of ordinary differential equations* (Report No. UIUCDCS-R-89-1516). Urbana-Champaign: Department of Computer Science, University of Illinois.