Journal of ICT, 3 (1), pp: 53-66

PARALLEL R-POINT IMPLICIT BLOCK METHOD FOR SOLVING HIGHER ORDER ORDINARY DIFFERENTIAL EQUATIONS DIRECTLY

*Z. Omar and **M. Sulaiman

** Faculty of Quantitative Sciences, Universiti Utara Malaysia, 06010 UUM Sintok, Kedah*

*** Department of Mathematics, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor*

Tel: 04-928 4900, Fax: 04-928 5779, Email: zurni@uum.edu.my

ABSTRACT

Most of the existing methods for solving ordinary differential equations (ODEs) of higher order are sequential in nature. These methods approximate a numerical solution at one point at a time and therefore do not fully exploit the capability of parallel computers. Hence, the development of parallel algorithms to suit these machines becomes essential. In this paper, a new method called parallel R-point implicit block method for solving higher order ODEs directly using constant step size is developed. This method calculates the numerical solution at more than one point simultaneously and is parallel in nature, thus suitable for parallel computation. Computational advantages are presented comparing the results obtained by the new method with that of conventional 1-point method. The numerical results show that the new method reduces the total number of steps and execution time. The accuracy of the parallel block and the conventional 1-point methods are comparable particularly when finer step sizes are used.

Key words: parallel R-point, implicit block method, higher order ODEs, directly

1.0 INTRODUCTION

In the subsequent discussion we consider solving the following n th order ordinary differential equation (ODE)

$$y^n = f(x, y, y', y'', ..., y^{n-1}), \;\; y^{(i)} = \eta_i, \;\; a \leq x \leq b \tag{1}$$

There are two techniques available for solving Equation [1]. The first technique is to reduce [1] to the equivalent first order system and then solve it using first order ordinary differential (ODEs) methods. Another approach to solve [1] is using direct method as suggested by Gear (1966, 1971, 1978), Hall and Suleiman (1981) and Suleiman (1979, 1989). These methods, however, compute the numerical solution at one point at a time.

Several researchers have attempted to estimate the numerical solution of first order ODEs at more than one point simultaneously. One of these methods is called parallel block method. In a block method, a set of new values that are obtained by each application of the formula is referred to as a "block". For instance, in an r-point block method, r new equally spaced solution values, i.e. $y_{t+1}, y_{t+2}, ..., y_{t+r}$ are obtained simultaneously at each iteration of the algorithm. Since the numerical solutions of ODEs can be found simultaneously, the block method is suitable for parallel computation, i.e one processor is assigned to perform a task at each point of the numerical solution. 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 Watt, (1969) and Tam (1989). However, parallel block method for solving higher order ODEs has not yet been discussed in previous literature.

2.0 DERIVATION OF R-POINT IMPLICIT BLOCK METHOD FOR HIGHER ORDER ODEs

The method derived in this section is the extension of work done by Omar and Suleiman (1999a, 1999b, 2000a, 2000b, 2001) and Omar *et al.* (2002). Let $x_{t+d} = x_t + dh$, where $d = 1,2,..,R$. The constant order formulation at the point x_{t+d} of R-point implicit method using k back values when [1] is solved by direct integration is given by:

$$y_{t+d}^{(n-p)} - y_t^{n-p} - (dh)y_t^{(n-p+1)} - \frac{(dh)^2}{2!}y_t^{(n-p+2)} - \dots$$
$$- \frac{(dh)^{p-2}}{(p-2)!}y_t^{n-2} - \frac{(dh)^{p-1}}{(p-1)!}y_t^{(n-1)} = h^p \sum_{m=0}^{k} \delta_{k,m}^{(d,p)} f_{t+d-m} \qquad (2)$$

where

$$h^p \sum_{m=0}^{k} \delta_{k,m}^{(d,p)} f_{t+d-m} = \underbrace{\int_{x_n}^{x_{n+d}} \int_{x_n}^{x} \int_{x_n}^{x} \dots \int_{x_n}^{x}}_{\leftarrow \text{p times} \rightarrow} P_{k,t+d}(x)\, dx \dots dx$$

and $P_{k,t+d}(x)$ is defined as follows

$$P_{k,t+d}(x) = \sum_{m=0}^{k} (-1)^m \binom{-s}{m} \nabla^m f_{t+d}$$

The coefficients $\left\{\delta_{k,m}^{(d,p)} \mid m = 0,1,\dots,k\right\}$ in [2] are given by

$$\delta_{k,m}^{(d,p)} = (-1)^m \sum_{r=m}^{k} \binom{r}{m} \alpha_r^{(d,p)} \qquad (3)$$

where $\alpha_r^{(d,p)}$ are the coefficients of the backward difference formulation of [2] which can be represented as

$$y_{t+d}^{(n-p)} - y_t^{n-p} - (dh)y_t^{(n-p+1)} - \frac{(dh)^2}{2!}y_t^{(n-p+2)} - \dots$$
$$- \frac{(dh)^{p-2}}{(p-2)!}y_t^{n-2} - \frac{(dh)^{p-1}}{(p-1)!}y_t^{(n-1)} = h^p \sum_{m=0}^{k} \alpha_m^{(d,p)} \nabla^m f_{t+d} \qquad \textit{for } p = 1,2,..,n.$$

In order to determine the general formulation for $\alpha_m^{(d,p)}$, we employ the generating functions strategy.

The generating functions when [1] is integrated once and twice, respectively, are given by:

$$S_d^{(1)}(t) = \int_{-d}^{0} e^{-s\log(1\text{-t})}\, ds \qquad \textit{for } d = 1,2,\ldots,\mathrm{R} \qquad [4]$$

$$S_d^{(2)}(t) = \int_{-d}^{0} (-s)e^{-s\log(1\text{-t})}\, ds \qquad [5]$$

which leads to the following relationship

$$S_d^{(2)}(t) = \frac{d(1-t)^d - S_d^{(1)}(t)}{\log(1-t)}. \qquad [6]$$

Integrating [1] three times, we have

$$y_{t+3}^{(n-3)} - y_t^{(n-3)} - (rh)y_t^{(n-2)} - \frac{(rh)^2}{2!}y_t^{(n-2)} = h^3 \sum_{m=0}^{k} \alpha_m^{(d,3)} \nabla^m f_t$$

where

$$\alpha_m^{(d,3)} = (-1)^m \int_{-d}^{0} \frac{(d-s)^2}{2!} \binom{-s}{m} ds\ . \qquad [7]$$

Substituting [7] in the generating function $S_d^{(3)}(t) = \sum_{m=0}^{\infty} \alpha_m^{(d,3)} t^m$ gives

$$S_d^{(3)}(t) = \int_{-d}^{0} \frac{(-s)^2}{2!} e^{-s\log(1\text{-t})}\, ds\ . \qquad [8]$$

Solving the integration on the right hand side of [8] produces the following relationship

$$S_d^{(3)}(t) = \frac{d^2(1-t)^d - 2!\,S_d^{(2)}(t)}{2!\log(1\text{-t})}$$

Using mathematical induction, it can be easily proven that the relationship of the generating functions is

$$S_d^{(p)}(t) = \frac{d^{(p-1)}(1-t)^d - (p-1)!\,S_d^{(p-1)}(t)}{(p-1)!\log(1-t)}\ ,\quad p = 2,3,\ldots,n \qquad [9]$$

where $S_d^{(p)}(t)$ and $S_d^{(p-1)}(t)$ are the generating functions when we integrate [1] p and p-1 times respectively at d th point of the implicit R-point explicit block method.

It follows from [9] that

$$(p-1)!\left[-\left(t+\frac{1}{2}t^2+\frac{1}{3}t^3+\ldots+\frac{1}{m}t^m+\ldots\right)\right]\left(\sum_{m=0}^{\infty}\alpha_m^{(d,p)}t^m\right)$$
$$=d^{(p-1)}(1-t)^d-(p-1)!\left(\sum_{m=0}^{\infty}\alpha_m^{(d,p-1)}t^m\right) \quad [10]$$

with

$$\alpha_m^{(d,p)}=(-1)^m\int_{-d}^{0}\frac{(-s)^{p-1}}{(p-1)!}\binom{-s}{m}ds$$

and

$$\alpha_m^{(d,p-1)}=(-1)^m\int_{-d}^{0}\frac{(-s)^{p-2}}{(p-2)!}\binom{-s}{m}ds.$$

Equating coefficients of t^m in [10], the following solution is obtained

$$\alpha_0^{(d,p)}=\alpha_1^{(d,p-1)}-\frac{d^{p-1}\binom{d}{1}(-1)}{(p-1)!}=\alpha_1^{(d,p-1)}+\frac{d^p}{(p-1)!} \quad [11a]$$

$$\alpha_m^{(d,p)}=\alpha_{m+1}^{(d,p-1)}-\frac{d^{p-1}\binom{d}{m+1}(-1)^{m+1}}{(p-1)!}-\sum_{r=0}^{m-1}\frac{\alpha_r^{(d,p)}}{(m-1)+2-r},\ m=1,2,\ldots,d-1 \quad [11b]$$

$$\alpha_m^{(d,p)}=\alpha_{m+1}^{(d,p-1)}-\sum_{r=0}^{m-1}\frac{\alpha_r^{(d,p)}}{(m-1)+2-r},\quad m=d,d+1,\ldots,k+n-p \quad [11c]$$

where $p=2,3,\ldots,n$ and $d=1,2,\ldots,R$.

Integrating [1] once, i.e. p=1 produces the following results

$$\alpha_0^{(d,1)} = d \qquad [12a]$$

$$\alpha_m^{(d,1)} = -\binom{d}{m+1}(-1)^{m+1} - \sum_{r=0}^{m-1} \frac{\alpha_r^{(d,1)}}{(m-1)+2-r}, \quad m = 1,2,\ldots,d-1$$

[12b]

$$\alpha_m^{(d,1)} = -\sum_{r=0}^{m-1} \frac{\alpha_r^{(d,1)}}{(m-1)+2-r}, \quad m = d, d+1, \ldots, k+n \qquad [12c]$$

where $d = 1,2,\ldots,R$.

For the purpose of illustration, let's apply the formulae above to derive the constant formulation at x_{t+1} when a second order ODE is integrated once using 2-point implicit block method with five back values. Replacing $n = 2$, $k = 5$, $d = 1$ and $p = 1$ in equations [12a]-[12c] gives the results as shown in Table 1 below:

Table 1: Integration Coefficients of the First Point of 2-Point Implicit Block Method When a Second Order ODE is Integrated Once

m	0	1	2	3	4	5
$\alpha_m^{(1,1)}$	1	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{3}{160}$

Substituting the integration coefficients obtained in Table 1 into equation [3] gives

$$\begin{aligned}
\delta_{5,0}^{(1,1)} &= (-1)^0 \sum_{r=0}^{5} \binom{r}{0} \alpha_m^{(1,1)} &&= \frac{475}{1440} \\
\delta_{5,1}^{(1,1)} &= (-1)^1 \sum_{r=1}^{5} \binom{r}{1} \alpha_m^{(1,1)} &&= \frac{1427}{1440} \\
\delta_{5,2}^{(1,1)} &= (-1)^2 \sum_{r=2}^{5} \binom{r}{2} \alpha_m^{(1,1)} &&= -\frac{798}{1440} \\
\delta_{5,3}^{(1,1)} &= (-1)^3 \sum_{r=3}^{5} \binom{r}{3} \alpha_m^{(1,1)} &&= \frac{482}{1440} \\
\delta_{5,4}^{(1,1)} &= (-1)^4 \sum_{r=4}^{5} \binom{r}{4} \alpha_m^{(1,1)} &&= -\frac{173}{1440} \\
\delta_{5,5}^{(1,1)} &= (-1)^5 \sum_{r=0}^{5} \binom{r}{5} \alpha_m^{(1,1)} &&= \frac{27}{1440}
\end{aligned} \qquad [13]$$

Now, we replace the coefficients obtained in [13] into [2] to get the constant formulation at x_{t+1} when a second order ODE is integrated once using 2 point implicit block method with five back values.

$$y'_{t+1} = y'_t + \frac{h}{1440}\left(475 f_{t+1} + 1427 f_t - 798 f_{t-1} + 482 f_{t-2} - 173 f_{t-3} + 27 f_{t-4}\right)$$

The constant formulation at other points using *R*-point implicit block method for solving higher order ordinary differential equations directly can be derived in the same manner.

3.0 TEST PROBLEMS

The following problems were solved numerically using the 2-point and 3-point implicit block methods:

Problem 1: $y''' = y + 3e^x, \quad y(0) = 0,\ y'(0) = 1,\ y''(0) = 2,$
$0 \le x \le 1$

Solution: $y(x) = xe^{x}$

Source: Krogh (1968).

Problem 2: $y''' = 8y' - 3y - 4e^{x}, \quad y(0) = 2, \; y'(0) = -2, \; y''(0) = 10,$
$0 \le x \le 1$

Solution: $y(x) = e^{x} + e^{-3x}$

Source: Suleiman (1989).

Problem 3: $y^{(iv)} = (x^4 + 14x^3 + 49x^2 + 32x - 12)e^{x},$
$y(0) = y'(0) = 0, y''(0) = 2, y'''(0) = -6, \; 0 \le x \le 1.$

Solution: $y(x) = x^2(1-x)^2 e^{x}.$

Source: Russel and Shampine (1972).

4.0 NUMERICAL RESULTS

The following notations are used in the tables:

h	Step size used
k	The number of back values 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.
I1P	Implicit 1 point method
S2PIB	Sequential implementation of the 2-point explicit block method
P2PIB	Parallel implementation of the 2-point explicit block method
S3PIB	Sequential implementation of the 3-point explicit block method
P3PIB	Parallel implementation of the 3-point explicit block method

The maximum error is defined as follows

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

Below are the results for the test problems when solved using the I1P, 2PIB and 3PIB methods. The performance comparisons of the methods are presented in Tables 2 to 4. In order to determine which algorithm gives better accuracy and timing, the ratio steps and times are tabulated in Table 5.

Table 2: Comparison between the I1P, 2PIB and 3PIB Methods for Solving Problem 1 of Higher Order ODE When k=5

h	MTD	STEPS	MAXE	TIME
	I1P	100	2.57702(-3)	204738
	S2PIB	53	2.56931(-3)	180067
10^{-2}	P2PIB	53	2.56931(-3)	305420
	S3PIB	36	2.54259(-3)	203792
	P3PIB	36	2.54259(-3)	305888
	I1P	1000	2.59298(-4)	1867372
	S2PIB	503	2.59290(-4)	1565157
10^{-3}	P2PIB	503	2.59290(-4)	1495489
	S3PIB	336	2.59262(-4)	1711414
	P3PIB	336	2.59262(-4)	1238822
	I1P	10000	2.59457(-5)	18660438
	S2PIB	5003	2.59457(-5)	15573376
10^{-4}	P2PIB	5003	2.59457(-5)	14914246
	S3PIB	3336	2.59456(-5)	16954815
	P3PIB	3336	2.59456(-5)	11744420
	I1P	100000	2.59474(-6)	186062037
	S2PIB	50003	2.59472(-6)	155331642
10^{-5}	P2PIB	50003	2.59472(-6)	145015244
	S3PIB	33336	2.59473(-6)	168929484
	P3PIB	33336	2.59473(-6)	117587775

5.0 COMMENTS ON THE RESULTS

In terms of the number of steps taken to find the numerical solution, the results confirm the superiority of the 2PIB and 3PIB methods to the I1P method by reducing the total number of steps taken to one half and one third respectively.

Table 3: Comparison between the I1P, 2PIB and 3PIB Methods for Solving Problem 2 of Higher Order ODE When k=5

h	MTD	STEPS	MAXE	TIME
10^{-2}	I1P	100	4.04194(-2)	204963
	S2PIB	53	4.04375(-2)	199023
	P2PIB	53	4.04375(-2)	310574
	S3PIB	36	4.04970(-2)	201467
	P3PIB	36	4.04970(-2)	314840
10^{-3}	I1P	1000	3.98784(-3)	1876443
	S2PIB	503	3.98786(-3)	1751291
	P2PIB	503	3.98786(-3)	1556215
	S3PIB	336	3.98792(-3)	1715376
	P3PIB	336	3.98792(-3)	1422511
10^{-4}	I1P	10000	3.98245(-4)	18750622
	S2PIB	5003	3.98245(-4)	17444141
	P2PIB	5003	3.98245(-4)	15189914
	S3PIB	3336	3.98245(-4)	17024108
	P3PIB	3336	3.98245(-4)	13850862
10^{-5}	I1P	100000	3.98191(-5)	187156497
	S2PIB	50003	3.98191(-5)	174032396
	P2PIB	50003	3.98191(-5)	151188843
	S3PIB	33336	3.98191(-5)	169750332
	P3PIB	33336	3.98191(-5)	135635017

In general, the accuracy of all methods for solving the test problems are comparable and of the same order.

The results obtained in the test problems show that the sequential 2PIB and 3PIB methods require less execution time than the I1P method at $h=10^{-2}$. However, the execution times for the parallel implementation of 2PIB and 3PIB are higher than the other methods at this step size due to the parallel overheads, but as the step size gets smaller, the parallel versions make up for the loss. It is displayed by the execution times obtained in all test problems that the parallel 2PIB and 3PIB methods outperform other methods when $h \leq 10^{-3}$.

Table 4: Comparison between the I1P, 2PIB and 3PIB Methods for Solving Problem 3 of Higher Order ODE When k=5

H	MTD	STEPS	MAXE	TIME
10^{-2}	I1P	100	3.42897(-3)	366335
	S2PIB	53	3.55338(-3)	359506
	P2PIB	53	3.55338(-3)	426027
	S3PIB	36	3.99807(-3)	363767
	P3PIB	36	3.99807(-3)	548111
10^{-3}	I1P	1000	3.40542(-4)	3468913
	S2PIB	503	3.40664(-4)	3299383
	P2PIB	503	3.40664(-4)	2643350
	S3PIB	336	3.41092(-4)	3250331
	P3PIB	336	3.41092(-4)	2525767
10^{-4}	I1P	10000	3.40304(-5)	34682919
	S2PIB	5003	3.40305(-5)	32886800
	P2PIB	5003	3.40305(-5)	26039852
	S3PIB	3336	3.40310(-5)	32304811
	P3PIB	3336	3.40310(-5)	24985844
10^{-5}	I1P	100000	3.40280(-6)	345745159
	S2PIB	50003	3.40280(-6)	327716782
	P2PIB	50003	3.40280(-6)	259397309
	S3PIB	33336	3.40280(-6)	322034763
	P3PIB	33336	3.40280(-6)	246441955

Based on the numerical results, it can be concluded that the parallel implementation of the 2PIB and 3PIB methods should be employed for solving higher ODEs directly when either the number of steps taken is large or the step size is small. For large step size, the numbers of steps taken by both 2PIB and 3PIB methods are considerably small. Hence, it is not advisable to use the parallel implementation of block methods since most of the execution times are dominated by the parallel overheads. In fact, the execution times of the sequential implementation of 2PIB and 3PIB methods when compared to the I1P method are also not impressive. This is due to more work involved in computing extra integration coefficients in the block methods. However, the coefficients are calculated only once at the beginning of the algorithms. As the number of steps becomes larger, the parallel 2PIB and 3PIB methods begin to

Table 5: The Ratio Steps and Execution Times of the 2PIB and 3PIB Methods to the I1P Method for Solving Higher Order ODEs When k=5

h	MTD	RATIO	RATIO TIME		
		STEP	PROB.1	PROB.2	PROB.3
10^{-2}	S2PIB	1.88679	1.13701	0.97102	1.01900
	P2PIB	1.88679	0.67035	0.65995	0.85989
	S3PIB	2.70270	1.00464	1.01735	1.00706
	P3PIB	2.70270	0.66932	0.65101	0.66836
10^{-3}	S2PIB	1.98807	1.19309	1.07146	1.05138
	P2PIB	1.98807	1.24867	1.20577	1.31232
	S3PIB	2.96736	1.09113	1.09390	1.06725
	P3PIB	2.96736	1.50738	1.31911	1.37341
10^{-4}	S2PIB	1.99880	1.19823	1.07490	1.05462
	P2PIB	1.99880	1.25118	1.23441	1.33192
	S3PIB	2.99670	1.10060	1.10142	1.07362
	P3PIB	2.99670	1.58888	1.35375	1.38810
10^{-5}	S2PIB	1.99988	1.19784	1.07541	1.05501
	P2PIB	1.99988	1.28305	1.23790	1.33288
	S3PIB	2.99967	1.10142	1.10254	1.07363
	P3PIB	2.99967	1.58233	1.37985	1.40295

reveal significant gains in the execution times by accelerating the integration process. This is achieved by distributing the job to more than one processor and therefore reducing the computation time. The numerical results also suggest that the accuracy of the block methods is comparable with the I1P method.

Table 4 shows the ratio of steps and times of the 2PIB and 3PIB methods to I1P method. The ratios 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 3PIB method. The ratio of time is also known as speedup.

Finally, if one is to choose between the parallel 2PIB and parallel 3PIB for solving higher order ODEs directly, the latter method seems to be a better choice. This is because the latter method is superior than the former method in terms of the number of steps and execution times and the accuracy of both methods are comparable.

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.

Krogh, F. T. (1968). A variable step, variable multistep method for the numerical solution of ordinary differential equation. *Proceedings of the IFIP Congress in Information Processing*, *68*,194.199.

Omar, Z. B., & Suleiman, M.B. (1999a). Solving second order ODEs directly using parallel 2-Point explicit block method. *Prosiding Kolokium Kebangsaan Pengintegrasian Teknologi Dalam Sains Matematik (27-28 mei,1999)*, pp. 390-395, Universiti Sains Malaysia.

Omar, Z. B., & Suleiman, M. B. (1999b). A new parallel 3-point explicit block method for solving second order ordinary differential equations directly. *ANALISIS*, *6* (1&2), pp. 63-76, Universiti Utara Malaysia.

Omar, Z. B., & Suleiman, M. B. (2000a). *Parallel 3-point explicit block method for solving higher order ordinary differential equations directly.* Paper presented at Asian Technology Conference in Mathematics, Chiagmai, Thailand.

Omar, Z. B., & Suleiman, M. B. (2000b). *Derivation of parallel 3-point implicit block method for solving higher order ordinary differential equations directly.* Paper presented at International Conference on Mathematics and its Applications in the New Millennium, UPM, 18-19 July 2000.

Omar, Z. B., & Suleiman, M. B. (2001). *Solving higher order ordinary differential equations using parallel 2-point explicit block method.* Paper presented at International Conference on Numerical Algorithm, Marrakech, Morocco, 2001.

Omar, Z. B., & Suleiman, M. B., Saman, M. Y, Evans, D.J. (2002). Parallel R-point explicit block method for solving second order ordinary differential equations directly. *International Journal of Computer Mathematics*, *79*, Issue March 2002, pp 289-298, Taylor & Francis, United Kingdom.

Russel, R. D., & Shampine, L. F. (1972). A collocation method for boundary value problems. *Num. Math*, *19*, 1-28.

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

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

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. Department of Computer Science, University of Illinois at Urbana-Champaign, United States of America.