IJMS 11 (1), 91-103 (2004)

DEVELOPING PARALLEL 3-POINT IMPLICIT BLOCK METHOD FOR SOLVING SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS DIRECTLY

ZURNI OMAR
Faculty of Quantitative Sciences
Universiti Utara Malaysia

ABSTRACT

Ordinary differential equations are commonly used for mathematical modeling in many diverse fields such as engineering, industrial mathematics, operation research, artificial intelligence, management, sociology and behavioural sciences. Numerous problems encountered in these fields require lengthy computation and immediate solution. In this paper, a new method called parallel 3-point implicit block method for solving second order ODEs is developed. This method takes full advantage of parallel computers because the numerical solution can be computed at three points simultaneously. As a result, the solution can be obtained faster if compared to the conventional methods where the numerical solution is computed at one point at a time. Computational advantages are presented comparing the results obtained by the new method with that of 1-point and 2-point implicit block methods. The numerical results show that parallel 3-point implicit block method reduces the total number of steps and execution time without sacrificing the accuracy.

ABSTRAK

Persaman pembezaan biasa sering digunakan untuk membentuk model matematik dalam pelbagai bidang seperti kejuruteraan, matematik industri, operasi penyelidikan, kepintaran buatan, pengurusan, sains sosial dan gelagat. Pelbagai masalah yang terdapat dalam bidang-bidang tersebut memerlukan pengiraan yang panjang dan penyelesaian segera. Dalam kertas ini, satu kaedah baru blok selari 3-titik tersirat dihasilkan bagi menyelesaikan persamaan pembezaan biasa peringkat dua. Kaedah ini memanfaatkan keupayaan komputer selari sepenuhnya kerana ianya menghitung penyelesaian berangka pada tiga titik serentak. Justeru, penyelesaian dapat diperoleh dengan lebih cepat jika dibandingkan dengan kaedah-kaedah konvensional di mana

penyelesaian berangka dihitung pada satu titik pada satu masa. Kelebihan kaedah baru ini dipersembahkan dengan membandingkannya dengan keputusan yang dicapai bila kaedah baru blok selari 1-titik dan 2-titik tersirat digunakan. Keputusan menunjukkan kaedah baru blok selari 3-titik tersirat dapat mengurangkan jumlah langkah dan tempoh pengiraan tanpa mengorbankan kejituan.

INTRODUCTION

Mathematical modeling is the art of translating problems from an application area into tractable mathematical formulations whose theoretical and numerical analysis provides insight, answer, and guidance useful for the originating application. For instance, Edward (2001) developed a differential equation model of North American cinematic box-office dynamics based on 442 new feature films released in 1999 on 37185 screens. He discovered that the gross for each film is really a sum of small contributions from a large number of units, which consist of either tens of thousands of screens or, tens of millions of individual filmgoers. His challenge and goal was to mathematically describe and understand the time-dependent behaviour of each unit by developing a mathematical model. His model has a great promise to describe and predict the box-offices grosses. We can also apply mathematical modeling in other areas as well, including management.

The building blocks of mathematical modeling are differential equations. In this paper, we focus on solving second order ordinary differential equation (ODE) of the following form:

$$y'' = f(x, y, y'),\ y(a) = y_0,\ y'(a) = y'_0,\ a \leq x \leq b \qquad (1)$$

Equation [1] can be solved by either using direct method as proposed by Gear (1966, 1971, 1978), Hall and Suleiman (1981) and Suleiman (1979, 1989) or reducing it to the equivalent system of first order equations and then solve it using first order ordinary differential equations (ODEs) methods. These methods, however, compute the numerical solution at one point at a time.

Birta and Abou-Rabia (1987), Chu and Hamilton (1987), Shampine and Watts (1969) and Tam (1989) introduced parallel block methods for numerical solutions of first order ODEs. 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. $y_{n+1}, y_{n+2},\ldots,y_{n+r}$ are obtained simultaneously at each iteration of the algorithm. 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.

The computational tasks at each point within a block are assigned to a single processor. Thus, the computations can be performed simultaneously.

DERIVATION OF THE 3-POINT IMPLICIT BLOCK METHOD

The method derived in this section is the extension of work done by Omar and Suleiman (1999a, 1999b). Let $x_{n+t} = x_n + th$, $t = 1, 2, 3$. Now, integrating [1] once gives

$$\int_{x_n}^{x_{n+t}} y''(x)dx = \int_{x_n}^{x_{n+t}} f(x, y, y')dx \tag{2}$$

Define $P_{k+1,\, n+t}(x)$ as the interpolation polynomial which interpolates f(x, y, y') at the set of points (x_{n+t-m}, f_{n+t-m}) for $m = 0, 1,\ldots,k$ as follows

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

where

$$s = \frac{x - x_{n+t}}{h}$$

Replacing $f(x, y, y')$ with $P_{k+1,\, n+t}(x)$ in [2] we now have

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

where

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

Subsituting $dx = hds$ and changing the limit of integration in [3] leads to

$$\begin{bmatrix} y'(x_{n+1}) \\ y'(x_{n+2}) \\ y'(x_{n+3}) \end{bmatrix} = \begin{bmatrix} y'(x_n) \\ y'(x_n) \\ y'(x_n) \end{bmatrix} + h \begin{bmatrix} \sum_{m=0}^{k} \gamma_m \nabla^m f_{n+1} \\ \sum_{m=0}^{k} \delta_m \nabla^m f_{n+2} \\ \sum_{m=0}^{k} \sigma_m \nabla^m f_{n+3} \end{bmatrix} \tag{4}$$

where

$$\gamma_m = \int_{-1}^{0} (-1)^m \binom{-s}{m}\, ds,$$

$$\delta_m = \int_{-2}^{0} (-1)^m \binom{-s}{m} ds, \text{ and}$$

$$\sigma_m = \int_{-3}^{0} (-1)^m \binom{-s}{m}\, ds.$$

In order to determine the values of γ_m, δ_m and σ_m, let $L(t)$, M(t)and N(t) be the generating functions defined as follows, respectively,

$$L(t) = \sum_{m=0}^{\infty} \gamma_m t^m = \sum_{m=0}^{\infty} (-t)^m \int_{-1}^{0} \binom{-s}{m}\, ds = \int_{-1}^{0} e^{-s\log(1-t)} ds \tag{5a}$$

$$M(t) = \sum_{m=0}^{\infty} \delta_m t^m = \sum_{m=0}^{\infty} (-t)^m \int_{-2}^{0} \binom{-s}{m}\, ds = \int_{-2}^{0} e^{-s\log(1-t)} ds \tag{5b}$$

$$N(t) = \sum_{m=0}^{\infty} \sigma_m t^m = \sum_{m=0}^{\infty} (-t)^m \int_{-3}^{0} \binom{-s}{m} ds = \int_{-2}^{0} e^{-s\log(1-t)} ds \tag{5c}$$

Using integration by parts on [5a]-[5c] leads to

$$L(t) = \sum_{m=0}^{\infty} \gamma_m t^m = \frac{t}{-log(1-t)} \tag{6a}$$

$$M(t) = \sum_{m=0}^{\infty} \delta_m t^m = \frac{t(t-2)}{log(1-t)} \tag{6b}$$

$$N(t) = \sum_{m=0}^{\infty} \sigma_m t^m = \frac{1-(1-t)^3}{-\,log(1-t)} = \frac{t(3-3t+t^2)}{-\,log(1-t)} \tag{6c}$$

Substituting $\dfrac{-\log(1-t)}{t} = (1 + \dfrac{1}{2} t + \dfrac{1}{3} t^2 + ...)$ in [6a]-[6c] and then expanding and rearranging terms gives the following solutions

$$\mu_0 = 1$$

$$\mu_m = -\sum_{r=0}^{m-1} \frac{\mu_r}{m+1-r} \quad \text{for } m = 1, 2, ...$$

$$\delta_0 = 2$$

$$\delta_1 = -\frac{\delta_0}{2} - 1 = -2$$

$$\delta_m = -\sum_{r=0}^{m-1} \frac{\delta_r}{m+1-r} \quad \text{for } m = 2, 3, ...$$

Fo

as

wh

In

wh

Th

int

where

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

Substituting $dx = hds$ and changing the limit of integration in [11] yields

$$\begin{bmatrix} y'(x_{n+1}) \\ y'(x_{n+2}) \\ y'(x_{n+3}) \end{bmatrix} = \begin{bmatrix} y(x_n) \\ y(x_n) \\ y(x_n) \end{bmatrix} + h \begin{bmatrix} y'(x_n) \\ 2y'(x_n) \\ 3y'(x_n) \end{bmatrix} + h^2 \begin{bmatrix} \sum_{m=0}^{k} \gamma^*_m \nabla^m f_{n+1} \\ \sum_{m=0}^{k} \delta^*_m \nabla^m f_{n+2} \\ \sum_{m=0}^{k} \sigma^*_m \nabla^m f_{n+3} \end{bmatrix} \tag{12}$$

where

$$\gamma^*_m = (-1)^m \int_{-1}^{0} (-s) \binom{-s}{m} ds \tag{13a}$$

$$\delta^*_m = (-1)^m \int_{-2}^{0} (-s) \binom{-s}{m} ds \tag{13b}$$

$$\sigma^*_m = (-1)^m \int_{-3}^{0} (-s) \binom{-s}{m} ds \tag{13c}$$

Let $L^*(t)$, $M^*(t)$ and $N^*(t)$ be defined as follows

$$L^*(t) = \sum_{m=0}^{\infty} \gamma_m t^m = \sum_{m=0}^{\infty} (-t)^m \int_{-1}^{0} \binom{-s}{m} ds = \int_{-1}^{0} (-s) e^{-s\log(1-t)} ds \tag{14a}$$

$$M^*(t) = \sum_{m=0}^{\infty} \gamma_m t^m = \sum_{m=0}^{\infty} (-t)^m \int_{-2}^{0} \binom{-s}{m} ds = \int_{-2}^{0} (-s) e^{-s\log(1-t)} ds \tag{14b}$$

$$N^*(t) = \sum_{m=0}^{\infty} \gamma_m t^m = \sum_{m=0}^{\infty} (-t)^m \int_{-3}^{0} \binom{-s}{m} ds = \int_{-3}^{0} (-s) e^{-s\log(1-t)} ds \tag{14c}$$

which leads to the relationships below

$$L^*(t) = \frac{(1-t) - L(t)}{\log(1-t)} \tag{15a}$$

$$M^*(t) = \frac{2(1-t)^2 - M(t)}{\log(1-t)} \tag{15b}$$

$$N^*(t) = \frac{3\,(1-t)^3 - N(t)}{\log(1-t)} \tag{15c}$$

whose solutions are

$$\gamma^*_0 = 1 + \gamma_1 = \frac{1}{2},$$

$$\gamma^*_{m+1} = \gamma_{m+2} - \sum_{r=0}^{m} \frac{\gamma^*_r}{m-r+2}$$

$$\delta^*_1 = 2^2 + \delta_1$$

$$\delta^*_1 = \delta_2 - 2 - \frac{\delta^*_0}{2}$$

$$\delta^*_m = \delta_{m+1} - \sum_{r=0}^{m} \frac{\delta^*_r}{m+1-r}$$

$$\nu^*_0 = 3^2 + \nu_1 ,$$

$$\nu^*_1 = 3^2 + \nu_2 - \frac{\nu^*_0}{2} ,$$

$$\nu^*_2 = 3 + \nu_3 - \sum_{r=0}^{1} \frac{\nu^*_r}{3-r} ,$$

$$\nu^*_{m+1} = \nu_{m+2} - \sum_{r=0}^{m} \frac{\nu^*_r}{m+2-r} \quad \text{for } m = 2, 3...$$

The constant step size formulation [12] can be written as

$$\begin{bmatrix} y(x_{n+1}) \\ y(x_{n+2}) \\ y(x_{n+3}) \end{bmatrix} = \begin{bmatrix} y(x_n) \\ y(x_n) \\ y(x_n) \end{bmatrix} + h \begin{bmatrix} y'(x_n) \\ 2y'(x_n) \\ 3y'(x_n) \end{bmatrix} + h^2 \begin{bmatrix} \sum_{m=0}^{k} \beta^*_{k,m} f_{n+1-m} \\ \sum_{m=0}^{k} \alpha^*_{k,m} f_{n+2-m} \\ \sum_{m=0}^{k} \tau^*_{k,m} f_{n+3-m} \end{bmatrix} \tag{16}$$

where

$$\beta^*_{k,m} = (-1)^m \sum_{r=m}^{k} \binom{r}{m} \gamma^*_r \tag{17a}$$

$$\alpha^*_{k,m} = (-1)^m \sum_{r=m}^{k} \binom{r}{m} \delta^*_r \tag{17b}$$

$$\tau^*_{k,m} = (-1)^m \sum_{r=m} \binom{r}{m} \sigma^*_r \tag{17c}$$

TEST PROBLEMS

The following problems were solved numerically using the 1-Point, 2-point and 3-point implicit 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: $y(x) = \frac{1}{2} x^2 e^{2x}$

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

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

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 implicit block method
P2PIB	Parallel implementation of the 2-point implicit block method
S3PIB	Sequential implementation of the 3-point implicit block method
P3PIB	Parallel implementation of the 3-point implicit block method

The maximum error, ratio step and ratio time are defined as follows:

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

$$\text{RATIO STEP} = \frac{\text{number of steps taken by 1 - Point method}}{\text{number of steps taken by parallel block method}}$$

$$\text{RATIO TIME} = \frac{\text{time taken by 1 - Point method}}{\text{time taken by parallel block method}}$$

The numerical results of the three problems of second order ODE are given in the following tables. Tables 1-3 show the performance comparison between the I1P, 2PIB and 3PIB methods in terms of the total number of steps taken, maximum error and the execution times (in microseconds). The results of the ratio steps and times are tabulated in Table 4.

Table 1

Comparison between the I1P, 2PIB and 3PIB Methods for Solving Problem 1 of Second Order ODE when k=5

h	MTD	STEPS	MAXE	TIME
	I1P	100	1.43154(-3)	133055
	S2PIB	53	1.43153(-3)	125804
10^{-2}	P2PIB	53	1.43153(-3)	267669
	S3PIB	36	1.43153(-3)	139970
	P3PIB	36	1.43153(-3)	297222
	I1P	1000	1.43166(-4)	1188010
	S2PIB	503	1.43166(-4)	1075821
10^{-3}	P2PIB	503	1.43166(-4)	1052437
	S3PIB	336	1.43166(-4)	1163381
	P3PIB	336	1.43166(-4)	888219
	I1P	10000	1.43167(-5)	11862787
	S2PIB	5003	1.43167(-5)	10699069
10^{-4}	P2PIB	5003	1.43167(-5)	9984578
	S3PIB	3336	1.43167(-5)	11526844
	P3PIB	3336	1.43167(-5)	8353568
	I1P	100000	1.43167(-6)	118502578
	S2PIB	50003	1.43167(-6)	106828469
10^{-5}	P2PIB	50003	1.43167(-6)	101130883
	S3PIB	33336	1.43167(-6)	115010473
	P3PIB	33336	1.43167(-6)	83393111

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

h	MTD	STEPS	MAXE	TIME
	I1P	100	1.25292(-2)	150265
	S2PIB	53	1.25811(-2)	145487
10^{-2}	P2PIB	53	1.25811(-2)	277465
	S3PIB	36	1.30459(-2)	133296
	P3PIB	36	1.30459(-2)	262630
	I1P	1000	1.25675(-3)	1355877
	S2PIB	503	1.25636(-3)	1260639
10^{-3}	P2PIB	503	1.25636(-3)	1114347
	S3PIB	336	1.25699(-3)	1113396
	P3PIB	336	1.25699(-3)	943335
	I1P	10000	1.25712(-4)	13542405
	S2PIB	5003	1.25707(-4)	12542291
10^{-4}	P2PIB	5003	1.25707(-4)	10785314
	S3PIB	3336	1.25709(-4)	11036813
	P3PIB	3336	1.25709(-4)	8779314
	I1P	100000	1.25716(-5)	135160215
	S2PIB	50003	1.25716(-5)	125133470
10^{-5}	P2PIB	50003	1.25716(-5)	104717695
	S3PIB	33336	1.25716(-5)	110132382
	P3PIB	33336	1.25716(-5)	87244821

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

h	MTD	STEPS	MAXE	TIME
	I1P	100	1.98934(-3)	107377
	S2PIB	53	1.98935(-3)	113613
10^{-2}	P2PIB	53	1.98935(-3)	224044
	S3PIB	36	1.98935(-3)	104714
	P3PIB	36	1.98935(-3)	248289
	I1P	1000	1.99846(-4)	926221
	S2PIB	503	1.99846(-4)	928675
10^{-3}	P2PIB	503	1.99846(-4)	788534
	S3PIB	336	1.99846(-4)	814743
	P3PIB	336	1.99846(-4)	687531

(continued)

10^{-4}	I1P	10000	1.99937(-5)	9244317
	S2PIB	5003	1.99937(-5)	9224021
	P2PIB	5003	1.99937(-5)	7530493
	S3PIB	3336	1.99937(-5)	8055093
	P3PIB	3336	1.99937(-5)	6256069
10^{-5}	I1P	100000	1.99947(-6)	92317214
	S2PIB	50003	1.99947(-6)	91981686
	P2PIB	50003	1.99947(-6)	73131842
	S3PIB	33336	1.99947(-6)	80320062
	P3PIB	33336	1.99947(-6)	62859845

Table 4
The Ratio Steps and Execution Times of the 2PIB and 3PIB Methods to the I1P Method for Solving Second Order ODEs when k=5

h	MTD	RATIO STEP	PROB 1	RATIO PROB 2	TIME PROB 3
10^{-2}	S2PIB	1.88679	1.05764	1.03284	0.94511
	P2PIB	1.88679	0.49709	0.54156	0.47927
	S3PIB	2.70270	0.95060	1.12730	1.02543
	P3PIB	2.70270	0.44766	0.57216	0.43247
10^{-3}	S2PIB	1.98807	1.10428	1.07555	0.99736
	P2PIB	1.98807	1.12882	1.21675	1.17461
	S3PIB	2.96736	1.02117	1.21779	1.13683
	P3PIB	2.96736	1.33752	1.43732	1.34717
10^{-4}	S2PIB	1.99880	1.10877	1.07974	1.00220
	P2PIB	1.99880	1.18811	1.25563	1.22759
	S3PIB	2.99670	1.02914	1.22702	1.14764
	P3PIB	2.99670	1.42009	1.54254	1.47765
10^{-5}	S2PIB	1.99988	1.10928	1.08013	1.00365
	P2PIB	1.99988	1.17177	1.29071	1.26234
	S3PIB	2.99967	1.03036	1.22725	1.14937
	P3PIB	2.99967	1.42101	1.54921	1.46862

COMMENTS ON THE RESULTS

The results indicate that the 2PIB method reduces the total number of steps to almost one half. In the case of the 3PIB method, the decrease in the total number of steps is more obvious, reducing the total number of steps by two-thirds. The accuracy of all the methods used are comparable and of the same order.

It can be observed that the execution times of the sequential implementation of 2PIB and 3PIB methods in all problems are better than the I1P despite the extra computations required in the former methods. The gains could have been contributed by the fact that the approximations at two and three points were calculated simultaneously in the 2PIB and 3PIB methods respectively and this made up for the time spent on the extra work.

The parallel implementation of both methods, as expected, required more time to perform the task at $h=10^{-2}$ due to parallel overheads. However, the timing gains in the parallel block methods began to show for $h<10^{-2}$ as the inherent parallelism within the block methods is fully exploited. The advantage of the parallel methods over the sequential methods became more obvious as the workload increased. It is also clear that the parallel implementation of 3PIB method is relatively the fastest among the other methods when the step size becomes smaller as shown in Table 4. This suggests that the strategy of using three processors to approximate numerical solutions at three different points simultaneously is the best choice especially for heavy workloads.

It can be concluded from the results that for a larger number of steps, it is recommended to employ parallel block methods as the given task can be completed faster. In addition, the reduction in the number of steps also provides great benefits for using the 2PIB and 3PIB methods instead of the I1P method.

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.

Edwards, D. A. (2001). A differential equation model of North American cinematic box-office dynamics. *IMA Journal of Management Mathematics, 12*, 41-74.

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.

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), 390-395 (ISBN 983-9700-71-5). 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.

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.