

Parallel block methods for solving nonstiff equations with stepsize control

Kazufumi OZAWA† and Susumu YAMADA‡

小澤一文†, 山田進‡

† Education center for information processing, Tohoku University

‡ Graduate school of information science, Tohoku University

E-mail: ozawa@dais.is.tohoku.ac.jp, yamada@ecip.tohoku.ac.jp

Abstract

Parallel block methods with a stepsize strategy are developed for solving nonstiff initial value problems on parallel computers. The block methods use the predictor-corrector iteration, which makes it easy to exploit the parallelism across the method, for the solutions of nonlinear equations. The stepsize strategy used in the methods is based on Milne's device which estimates the local error of the overall method using the predicted and corrected values. Numerical experiments are carried out on a KSR-1 parallel computer in order to show the performance of the block methods. The computational costs, errors and speed-up ratios of the block methods are compared with those of the PIRK (parallel iterated Runge-Kutta) by Houwen et al.

1 Introduction

Consider the initial value problems (IVPs)

$$y' = f(x, y), \quad y(x_0) = \eta. \quad (1.1)$$

Here the equation (1.1) is assumed to be nonstiff. We propose predictor-corrector (P-C) type parallel block methods with a stepsize strategy for solving (1.1).

When solving nonstiff equations one often use explicit Runge-Kutta (ERK) methods. For the parallelization of ERKs, Iserles and Nørsett [9] has shown that the orders of parallel ERKs never exceed the number of the effective stages in parallel execution. This is a serious drawback since the higher the degree of parallelism (the lower the effective stages), the lower the accuracy. Consider, for example, a five-stage ERK whose Butcher array has the sparsity pattern

| | | | | | |
|-------|-------|-------|-------|-------|-------|
| c_1 | 0 | | | | |
| c_2 | x | 0 | | | |
| c_3 | x | x | 0 | | |
| c_4 | x | x | 0 | 0 | |
| c_5 | x | x | 0 | 0 | 0 |
| | b_1 | b_2 | b_3 | b_4 | b_5 |

Fig1. Butcher array of some parallel ERK

where the symbol x denotes a nonzero element. In this method, the third, fourth and fifth stages can be computed in parallel after the computations of the first and second stages, if at least three processors are available. Therefore the number of effective stages of the method is not 5 but 3, so that the speed-up ratio becomes $5/3$. In this case, however, the attainable order with the method is only 3, since the number of effective stages is 3. This result suggests that the parallel ERKs have nothing or little to recommend it, particularly when the method is of order ≤ 4 ; in such a case, as is well known, there exists a sequential ERK of the order equal to the number of (effective) stages.

Another possibility of constructing parallel methods for nonstiff ODEs is to use PIRK (parallel iterated Runge-Kutta) developed by Houwen and Sommeijer[8]. Consider the following s -stage (implicit or explicit) Runge-Kutta formula

$$\begin{cases} y_{n+1} = y_n + h \sum_{i=1}^s b_i f(x_n + c_i h, z_i), \\ z_i = y_n + h \sum_{j=1}^s a_{ij} f(x_n + c_j h, z_j), \quad i = 1, \dots, s. \end{cases} \quad (1.2)$$

If the method (1.2) is explicit we must compute each stage sequentially, except for the particular case that the Butcher array has the sparsity pattern as shown in Fig. 1. On the other hand, if the method is implicit then (1.2) is a system of nonlinear algebraic equations, which leads to the evaluation and LU decomposition of a Jacobi matrix. Thus, in either case, it is difficult to parallelize the whole part of the scheme. However, the fixed-point iteration

$$\begin{cases} z_i^{[\nu]} = y_n + h \sum_{j=1}^s a_{ij} f(x_n + c_j h, z_j^{[\nu-1]}), \quad \nu = 1, 2, \dots, \\ z_i^{[0]} = y_n, \quad i = 1, \dots, s, \\ y_{n+1}^{[\nu]} = y_n + h \sum_{i=1}^s b_i f(x_n + c_i h, z_i^{[\nu]}) \end{cases} \quad (1.3)$$

associated with the Runge-Kutta formula (1.2) is easy to carry out in parallel. Houwen and Sommeijer [8] called the method (1.3) PIRK (parallel iterated Runge-Kutta). As was pointed out by [8], although the PIRK do not have an excellent stability property such as A - or L -stability, even if the underlying Runge-Kutta does, the PIRK can have the same order as that of the underlying Runge-Kutta. It is shown by [7] and [8] that $y_{n+1}^{[\nu]}$ by (1.3) is being an approximation to $y(x_n + h)$ of order

$$p = \min\{\nu + 1, p^*\}, \quad (1.4)$$

where p^* denotes the order of the underlying Runge-Kutta. This means that the PIRK is a kind of embedded method, and therefore it is easy to incorporate the stepsize control mechanism into the method. It is also shown in [8] that the PIRK is equivalent to the $(\nu + 1)s$ -stage Runge-Kutta formula whose Butcher array is given by

$$\begin{array}{c|cccccc} 0 & 0 & & & & & \\ c & A & 0 & & & & \\ c & 0 & A & 0 & & & \\ \vdots & \vdots & & \ddots & \ddots & & \\ c & 0 & \dots & 0 & A & 0 & \\ \hline & 0 & \dots & 0 & 0 & b^T & \end{array}$$

Fig. 2 Butcher array of the PIRK

where $A = (a_{ij})_{i,j=1}^s$, $b = (b_1, \dots, b_s)^T$, $c = (c_1, \dots, c_s)^T$, that is, the PIRK is a parallel ERK with effective stage $\nu + 1$, if at least s processors are available.

Since the PIRK is a parallel ERK, it has the same drawback as stated before. To overcome this drawback, or to reduce the effective stages, it is necessary to start the

iteration with the starting values $z_i^{[0]}$ of higher order. The parallel block methods presented here are a predictor-corrector type, which consists of Runge-Kutta type correctors and Adams type predictors of the same orders as that of the correctors.

2 Parallel P-C block method

Consider the set of $r(> 1)$ abscissas

$$X_m = \{x_{m,1} < x_{m,2} < \cdots < x_{m,r}\}, \quad m = 0, 1, \dots,$$

which satisfy the following conditions:

$$\begin{cases} x_{m,\nu} = x_{m-1,r} + \sigma_\nu h_m, & \nu = 1, 2, \dots, r, \\ 0 \leq \sigma_1 < \sigma_2 < \cdots < \sigma_{r-1} < \sigma_r = 1, \\ x_{0,1} = x_0, \\ x_{m-1,r} \leq x_{m,1}. \end{cases} \quad (2.1)$$

We call X_m the m -th block, and denote the length of the block by h_m , i.e.,

$$h_m = x_{m,r} - x_{m-1,r}.$$

Hereafter we will refer to h_m as stepsize, for brevity.

Consider the implicit one block method of the type

$$Y_{m+1} = y_{m,r} \mathbf{e} + h_{m+1} B^c F(Y_{m+1}) \quad (2.2)$$

for solving Eq.(1.1), where \mathbf{e} is the r -dimensional vector with all entries equal to unity, and Y_m and $F(Y_m)$ are the vectors defined by

$$\begin{aligned} Y_m &= (y_{m,1}, y_{m,2}, \dots, y_{m,r})^T, \\ F(Y_m) &= (f(y_{m,1}), f(y_{m,2}), \dots, f(y_{m,r}))^T, \end{aligned} \quad (2.3)$$

which approximate the solutions and its derivatives in the m th block X_m , respectively.

The method (2.2) is considered to be a set of r one step methods, and therefore each of these methods is more stable than linear multistep methods. It is, however, almost impossible to compute r entries of Y_{m+1} in parallel, if B^c is a full matrix, since we must solve a system of the nonlinear algebraic equations in method (2.2) at each step of the integration. On the other hand, if we take a diagonal or lower triangular matrix as B^c , which makes it easy to carry out all the methods in parallel, then each of the methods cannot have a high order; if, for example, B^c is diagonal, the method has order at most 1. A reasonable compromise is to take B^c as a full matrix, and to use a fixed-point iteration for the solution of the nonlinear algebraic equations.

Let the entries of $B^c = (b_{ij}^c)_{i,j=1}^r$ in (2.2) be

$$b_{ij}^c = \int_0^{\sigma_i} \prod_{\substack{k=1 \\ k \neq j}}^r \frac{t - \sigma_k}{\sigma_j - \sigma_k} dt, \quad i, j = 1, 2, \dots, r, \quad (2.4)$$

then each of the methods is of order at least r , since if the exact solution $y(x)$ is a polynomial of degree r , then, by assumption (2.1), there exists a unique polynomial

$\varphi_{m+1}(x)$ of degree $r - 1$ which interpolates $y'(x) = f(x, y)$ at $x = x_{m+1,\nu}$ ($\nu = 1, 2, \dots, r$), so that

$$\begin{aligned} y(x_{m+1,i}) - y(x_{m,r}) &= \int_{x_{m,r}}^{x_{m+1,i}} \varphi_{m+1}(x) dx \\ &= h_{m+1} \int_0^{\sigma_i} \varphi_{m+1}(x_{m,r} + h_{m+1}t) dt \\ &= h_{m+1} \sum_{j=1}^r b_{ij}^c f(x_{m+1,j}, y(x_{m+1,j})). \end{aligned} \quad (2.5)$$

Although the particular choice of σ_ν 's leads to a higher order corrector, it is not necessary desirable that one of the r methods is of higher order than that of the others, since if this is the case, it is difficult to supply the set of starting values of the same order as that of the highest corrector; the values $y_{m,\nu}$, $f_{m,\nu}$ ($\nu = 1, \dots, r$) are used in the predictor for the calculation of the starting values (predicted values) at the next block.

Therefore we take the following sets of equally spaced data points:

- Type 1 $\sigma_\nu = \nu/r$, $\nu = 1, 2, \dots, r$
- Type 2 $\sigma_\nu = (\nu - 1)/(r - 1)$, $\nu = 1, 2, \dots, r$

Note that Type 2 is equivalent to the r -point Newton-Cotes formulae of numerical quadrature, and therefore the order of the methods are r and $r + 1$ when r is even and odd, respectively. Note also that the number of processors required for the parallel execution of Type 2 is not r but $r - 1$, since $x_{m-1,r} = x_{m,1}$.

Next we consider the predictor for our P-C scheme. The coefficient matrix of the predictor must be determined as a function of the stepsize ratio, so as to supply the sufficient starting values for any stepsize ratio. We consider the one block predictor of the type

$$\begin{aligned} Y_{m+1} &= y_{m,r}e + \theta_{m+1}h_m B^p(\theta_{m+1})F(Y_m), \\ \theta_{m+1} &= h_{m+1}/h_m. \end{aligned} \quad (2.6)$$

which gives a set of starting values at block X_{m+1} , using the previously calculated values at X_m . In (2.6) the entries of the coefficient matrix $B^p(\theta) = (b_{ij}^p(\theta))$ are given by

$$b_{ij}^p(\theta) = \int_0^{\sigma_i} \prod_{\substack{k=1 \\ k \neq j}}^r \frac{1 + \theta t - \sigma_k}{\sigma_j - \sigma_k} dt, \quad i, j = 1, 2, \dots, r. \quad (2.7)$$

It is easily shown, as is the case for the corrector, that each of the methods in predictor (2.6) is of order r for any $\theta > 0$ and any types of the abscissas.

3 Local truncation error and stepsize strategy

In what follows we assume that the solution of (1.1) has as many higher derivatives as required. Then the analysis of the orders in the previous section leads to the following the asymptotic expansions of the local errors:

$$\begin{cases} y(x_{m,\nu}) - y_{m,\nu}^p = c_\nu^p(\theta_m) h_{m-1}^{r+1} y^{(r+1)}(x_{m,\nu}) + O(h_{m-1}^{r+2}), \\ y(x_{m,\nu}) - y_{m,\nu}^c = c_\nu^c(\theta_m) h_{m-1}^{r+1} y^{(r+1)}(x_{m,\nu}) + O(h_{m-1}^{r+2}), \end{cases} \quad \nu = 1, 2, \dots, r, \quad (3.1)$$

where $y_{m,\nu}^p$ and $y_{m,\nu}^c$ are the values calculated by the predictor and corrector, respectively, and $c_\nu^p(\theta_m)$ and $c_\nu^c(\theta_m)$ are the constants independent of h . We call these constants the error constants of the methods. The next theorem gives the basis on which the stepsize strategy used in this paper is valid.

Theorem 1

(i) The error constants $c_\nu^c(\theta)$ ($\nu = 1, 2, \dots, r$) of the corrector formula are of the form

$$c_\nu^c(\theta) = M_\nu \theta^{r+1}, \quad \nu = 1, 2, \dots, r,$$

where M_ν are constants independent of θ .

(ii) For any choice of σ_ν ($\nu = 1, \dots, r$) satisfying the condition (2.1), we have $c_r^p(\theta) \neq c_r^c(\theta)$, if $\theta > 0$.

Proof

(i) Using the error formula of interpolating polynomials, we can express the error constants $c_\nu^c(\theta)$ in the form

$$\begin{aligned} c_\nu^c(\theta) &= \frac{1}{r!} \int_0^{\theta\sigma_\nu} (x - \theta\sigma_1) \cdots (x - \theta\sigma_r) dx \\ &= \frac{\theta^{r+1}}{r!} \int_0^{\sigma_\nu} (x - \sigma_1) \cdots (x - \sigma_r) dx, \end{aligned} \quad (3.2)$$

which proves the assertion.

(ii) For given σ 's and any a 's, we define the constants w_i and f_i by

$$\begin{cases} w_i = \int_0^1 \prod_{j=1}^i (t - a_j) \prod_{j=i+1}^r (t - \sigma_j) dt, & i = 0, \dots, r, \\ f_{i+1}(x) = \int_0^x \prod_{j=1}^i (t - a_j) \prod_{j=i+2}^r (t - \sigma_j) dt, & i = 0, \dots, r-1, \end{cases} \quad (3.3)$$

where we use the convention that

$$\prod_{j=1}^0 (t - a_j) = \prod_{j=r+1}^r (t - \sigma_j) = 1.$$

Integration by parts leads to the following relations for w_i 's

$$\begin{aligned} w_i &= (1 - \sigma_{i+1})f_{i+1}(1) - \int_0^1 f_{i+1}(x) dx, \\ w_{i+1} &= (1 - a_{i+1})f_{i+1}(1) - \int_0^1 f_{i+1}(x) dx. \end{aligned} \quad (3.4)$$

If we take the constants a_{i+1} ($i = 0, \dots, r-1$) in such a way that

$$a_{i+1} = \begin{cases} 0, & f_{i+1}(1) \geq 0, \\ 1, & f_{i+1}(1) < 0, \end{cases} \quad (3.5)$$

then we have

$$\begin{aligned} w_0 \leq w_1 \leq \cdots \leq w_r &= \int_0^1 t^{r-k}(t-1)^k dt \\ &= (-1)^k \frac{(r-k)!k!}{(r+1)!} \leq \frac{1}{r+1}, \end{aligned} \quad (3.6)$$

where k denotes the number of a_i 's which satisfy $a_i = 1$. It follows from (3.2) and (3.6) we have

$$c_r^c(\theta) = \frac{\theta^{r+1}}{r!} w_0 \leq \frac{\theta^{r+1}}{(r+1)!}. \quad (3.7)$$

On the other hand, the error constant $c_r^p(\theta)$ of the predictor is given by

$$\begin{aligned} c_r^p(\theta) &= \frac{1}{r!} \int_0^\theta (t + (1 - \sigma_1)) \cdots (t + (1 - \sigma_r)) dt \\ &= \frac{1}{r!} \int_0^\theta (t^r + s_1 t^{r-1} + \cdots + s_{r-1} t) dt \\ &= \frac{1}{r!} \left(\frac{1}{r+1} \theta^{r+1} + \frac{s_1}{r} \theta^r + \cdots + \frac{s_{r-1}}{2} \theta^2 \right), \end{aligned} \quad (3.8)$$

where s_i is the i th fundamental symmetric expression consists of

$$0 = 1 - \sigma_r < 1 - \sigma_{r-1} < \cdots < 1 - \sigma_1 \leq 1.$$

We have thus for $\theta > 0$

$$c_r^p(\theta) - c_r^c(\theta) \geq \frac{1}{r!} \left(\frac{s_1}{r} \theta^r + \cdots + \frac{s_{r-1}}{2} \theta^2 \right) > 0, \quad (3.9)$$

since $s_i > 0$ ($i = 1, \dots, r-1$), which proves the assertion. QED

Using (3.1) the maximum local truncation error at X_m of the corrector can be approximated by

$$\begin{aligned} \max_{1 \leq \nu \leq r} |y(x_{m,\nu}) - y_{m,\nu}^c| &\approx \max_{1 \leq \nu \leq r} |c_\nu^c(\theta_m) y^{(r+1)}(x_{m,\nu})| h_{m-1}^{r+1} \\ &\approx \max_{1 \leq \nu \leq r} |c_\nu^c(\theta_m)| |y^{(r+1)}(x_{m,r})| h_{m-1}^{r+1}, \end{aligned} \quad (3.10)$$

and the $(r+1)$ st derivative of the solution at $x = x_{m,r}$ is estimated by

$$h_{m-1}^{r+1} y^{(r+1)}(x_{m,r}) \approx \frac{y_{m,r}^p - y_{m,r}^c}{c_r^c(\theta_m) - c_r^p(\theta_m)}, \quad (3.11)$$

which is valid because of the result of Theorem 1. We have thus

$$\max_{1 \leq \nu \leq r} |y(x_{m,\nu}) - y_{m,\nu}^c| \approx \max_{1 \leq \nu \leq r} |c_\nu^c(\theta_m)| \left| \frac{y_{m,r}^p - y_{m,r}^c}{c_r^c(\theta_m) - c_r^p(\theta_m)} \right|. \quad (3.12)$$

Therefore, using (3.12), we can estimate the maximum local truncation error of the corrector by sampling the predicted and corrected values only at $x = x_{m,r}$.

Here we consider the implementation of our P-C scheme. Our P-C scheme based on (2.6) and (2.2) is

$$\begin{cases} Y_{m+1}^{[0]} = y_{m,r} \mathbf{e} + \theta_{m+1} h_m B_p(\theta_{m+1}) F_m, \\ Y_{m+1}^{[\kappa]} = y_{m,r} \mathbf{e} + \theta_{m+1} h_m B_c F(Y_{m+1}^{[\kappa-1]}), & \kappa = 1, 2, \dots, \mu_m, \\ m = 0, 1, \dots, \end{cases} \quad (3.13)$$

where $Y_m^{[\kappa]}$, $F(Y_m^{[\kappa]})$, Y_m and F_m are given by

$$\begin{aligned} Y_m^{[\kappa]} &= (y_{m,1}^{[\kappa]}, y_{m,2}^{[\kappa]}, \dots, y_{m,r}^{[\kappa]})^T, & F(Y_m^{[\kappa]}) &= (f_{m,1}^{[\kappa]}, f_{m,2}^{[\kappa]}, \dots, f_{m,r}^{[\kappa]})^T, \\ f_{m,\nu}^{[\kappa]} &= f(x_{m,\nu}, y_{m,\nu}^{[\kappa]}), & \nu &= 1, 2, \dots, r, \\ Y_m &= Y_m^{[\mu_m]}, & F_m &= F(Y_m^{[\mu_m]}), \end{aligned}$$

and μ_m is the maximum number of iterations, which will be stated later.

Let Err be the estimate of the maximum local truncation error at X_m given by (3.12), i.e.,

$$Err := \max_{1 \leq \nu \leq r} |c_\nu^c(\theta_m)| \left| \frac{y_{m,r}^p - y_{m,r}^c}{c_r^c(\theta_m) - c_r^p(\theta_m)} \right|, \quad (3.14)$$

then the stepsize ratio which guarantees a local truncation error at X_{m+1} less than a prescribed tolerance TOL is

$$\theta_{m+1} = \alpha \left(\frac{TOL}{Err} \right)^{\frac{1}{r+1}}, \quad (3.15)$$

where α is a safety factor such that $0 < \alpha < 1$.

There remains the problem how to stop the iteration for our predictor-corrector scheme. Here we simplify the corrector iteration by

$$Y^{[\kappa]} = h B_c F(Y^{[\kappa-1]}) + \text{const}, \quad \kappa = 1, 2, \dots, \quad (3.16)$$

and denote its exact solution by Y^* . In principle, we must use the exact solution Y^* , which is also the exact value satisfying the corrector equation (2.2), in order to get Err by (3.14), but it is almost impossible to get Y^* in a finite iterations. To avoid this difficulty it is desirable to use the $Y^{[\kappa]}$ satisfying

$$\|Y^{[\kappa]} - Y^*\| < TOL, \quad (3.17)$$

where TOL is the tolerance for the local error. As is easily shown from (3.16), the error norm $\|Y^{[\kappa]} - Y^*\|$ is bounded by

$$\|Y^{[\kappa]} - Y^*\| \leq (hL \|B_c\|)^\kappa \|Y^{[0]} - Y^*\|, \quad (3.18)$$

where L is the Lipschitz constant of $f(x, y)$ with respect to y . If the right-hand side in (3.18) is bounded by TOL , then we have

$$\begin{aligned} \|Y^{[\kappa]} - Y^{[\kappa-1]}\| &\leq \|Y^{[\kappa]} - Y^*\| + \|Y^* - Y^{[\kappa-1]}\| \\ &\leq (hL \|B_c\|)^{\kappa-1} (hL \|B_c\| + 1) \|Y^{[0]} - Y^*\| \\ &\leq TOL (hL \|B_c\| + 1) / (hL \|B_c\|). \end{aligned} \quad (3.19)$$

Therefore we adopt the stopping criterion such that

$$\|Y^{[\kappa]} - Y^{[\kappa-1]}\| \leq TOL (1 + hL\|B_c\|)/(hL\|B_c\|), \quad (3.20)$$

and use the PE(CE) $^{\mu_m}$ mode, where μ_m is determined adaptively.

The parallel block method with stepsize strategy is executed as follows:

1. Chose an initial stepsize h_0 by the procedure developed by [6] so that all the entries in Y_0 , which are calculated any of one-step methods with constant stepsize h_0 , have local truncation errors less than TOL .

For $m = 1, 2, \dots$ repeat the following steps:

2. Calculate the predicted value $Y_m^{[0]}$ by the predictor.
3. For $\kappa = 1, 2, \dots$, calculate the corrected values $Y_m^{[\kappa]}$ by the corrector repeatedly until the condition (3.20) is satisfied. We set here the maximum iteration number 3; if not satisfied with the condition even for $\kappa = 3$ then stop the corrector iteration.
4. Compute Err by (3.14), where $y_{m,r}^p$ and $y_{m,r}^c$ in (3.14) are the r th components of $Y_m^{[0]}$ and $Y_m^{[\mu_m]}$, respectively, and
 - If $Err \leq TOL$ then change the stepsize ratio to the one with which the maximum truncation error in the next block is of modulus less than TOL .
 - If $Err > TOL$ then reject this block.
5. Let $Y_m := Y_m^{[\mu_m]}$, $F_m := F(Y_m^{[\mu_m]})$.

In the above procedure the function evaluations $F(Y_m^{[\kappa]})$ are carried out in parallel.

4 Numerical experiments

Let consider the differential equation

$$\begin{cases} y_1' = -y_1 + y_1^2 y_2 + \cos x - \cos^2 x \sin x - \sin x, \\ y_2' = -y_2 + y_1 y_2^2 + \sin x - \cos x \sin^2 x + \cos x, \end{cases} \quad (4.1)$$

$$y_1(0) = 1, \quad y_2(0) = 0,$$

which has the exact solution

$$y_1(x) = \cos x, \quad y_2(x) = \sin x.$$

Here we integrate the equation from $x = 0$ to $15\pi/4$ by the block methods, PIRKs and RKF45, and for each of the methods compute 'global error' at $x = 15\pi/4$, i.e.,

$$\text{global error} = \max \{|y_1 - y_1(15\pi/4)|, |y_2 - y_2(15\pi/4)|\},$$

y_1 and y_2 are the numerical solution at $x = 15\pi/4$.

The stepsize control mechanisms are incorporated into all the codes. Here we compare the number of total steps, function evaluations, and etc of the block methods with those of PIRKs and RKF45. The results are shown in Table 1.

It follows from the results that the number of function evaluations of the block method of Type 2 with $r = 5$ is the smallest for all TOL .

Table 1. Numbers of total steps and function calls by the block methods, PIRKs and RKF45 for problem (4.1).

(a) Type 1 ($r = 4$)

| TOL | steps | | function calls | | GE |
|------------|-------|--------|----------------|----------|--------|
| | total | reject | serial | parallel | |
| 10^{-5} | 28 | 0 | 424 | 106 | -5.35 |
| 10^{-6} | 43 | 0 | 668 | 167 | -6.32 |
| 10^{-7} | 67 | 0 | 936 | 234 | -7.03 |
| 10^{-8} | 105 | 0 | 1352 | 338 | -7.74 |
| 10^{-9} | 166 | 0 | 2060 | 515 | -8.57 |
| 10^{-10} | 262 | 0 | 3192 | 798 | -9.32 |
| 10^{-11} | 414 | 0 | 5004 | 1251 | -10.10 |
| 10^{-12} | 655 | 1 | 7780 | 1945 | -10.89 |

(b) Type 2 ($r = 4$)

| TOL | steps | | function calls | | GE |
|------------|-------|--------|----------------|----------|--------|
| | total | reject | serial | parallel | |
| 10^{-5} | 25 | 0 | 282 | 94 | -4.50 |
| 10^{-6} | 37 | 0 | 429 | 143 | -5.90 |
| 10^{-7} | 58 | 0 | 666 | 222 | -7.01 |
| 10^{-8} | 91 | 0 | 975 | 325 | -7.76 |
| 10^{-9} | 142 | 0 | 1398 | 466 | -8.66 |
| 10^{-10} | 224 | 0 | 2070 | 690 | -9.38 |
| 10^{-11} | 354 | 0 | 3228 | 1076 | -10.15 |
| 10^{-12} | 560 | 1 | 5058 | 1686 | -10.94 |

(c) Type 1 ($r = 5$)

| TOL | steps | | function calls | | GE |
|------------|-------|--------|----------------|----------|--------|
| | total | reject | serial | parallel | |
| 10^{-5} | 20 | 2 | 375 | 75 | -4.52 |
| 10^{-6} | 28 | 3 | 520 | 104 | -5.58 |
| 10^{-7} | 37 | 2 | 700 | 140 | -6.71 |
| 10^{-8} | 51 | 2 | 980 | 196 | -7.75 |
| 10^{-9} | 73 | 2 | 1375 | 275 | -8.61 |
| 10^{-10} | 105 | 2 | 1835 | 367 | -9.49 |
| 10^{-11} | 152 | 2 | 2495 | 499 | -10.34 |
| 10^{-12} | 221 | 2 | 3295 | 659 | -11.23 |

(d) Type 2 ($r = 5$)

| TOL | steps | | function calls | | GE |
|------------|-------|--------|----------------|----------|--------|
| | total | reject | serial | parallel | |
| 10^{-5} | 18 | 3 | 280 | 70 | -4.39 |
| 10^{-6} | 23 | 2 | 344 | 86 | -5.33 |
| 10^{-7} | 32 | 3 | 472 | 118 | -6.19 |
| 10^{-8} | 43 | 3 | 656 | 164 | -7.77 |
| 10^{-9} | 60 | 3 | 920 | 230 | -8.92 |
| 10^{-10} | 85 | 3 | 1312 | 328 | -9.99 |
| 10^{-11} | 121 | 3 | 1836 | 459 | -10.97 |
| 10^{-12} | 175 | 3 | 2496 | 624 | -11.82 |

(e) PIRK (2-stage Gauss)

| TOL | steps | | function calls | | GE |
|------------|-------|--------|----------------|----------|--------|
| | total | reject | serial | parallel | |
| 10^{-5} | 128 | 20 | 1024 | 512 | -6.31 |
| 10^{-6} | 204 | 20 | 1632 | 816 | -7.40 |
| 10^{-7} | 348 | 26 | 2784 | 1392 | -8.37 |
| 10^{-8} | 590 | 25 | 4720 | 2360 | -9.19 |
| 10^{-9} | 1027 | 29 | 8216 | 4108 | -10.07 |
| 10^{-10} | 1788 | 22 | 14304 | 7152 | -10.86 |
| 10^{-11} | 3146 | 14 | 25168 | 12584 | -11.66 |
| 10^{-12} | 5571 | 10 | 44568 | 22284 | -12.53 |

(f) PIRK (3-stage Radau IA)

| TOL | steps | | function calls | | GE |
|------------|-------|--------|----------------|----------|--------|
| | total | reject | serial | parallel | |
| 10^{-5} | 75 | 17 | 1125 | 375 | -5.71 |
| 10^{-6} | 107 | 21 | 1605 | 535 | -6.43 |
| 10^{-7} | 154 | 23 | 2310 | 770 | -7.38 |
| 10^{-8} | 226 | 23 | 3390 | 1130 | -8.23 |
| 10^{-9} | 341 | 24 | 5115 | 1705 | -9.20 |
| 10^{-10} | 520 | 24 | 7800 | 2600 | -10.00 |
| 10^{-11} | 802 | 22 | 12030 | 4010 | -10.87 |
| 10^{-12} | 1251 | 21 | 18765 | 6255 | -11.72 |

(g) RKF 45

| TOL | steps | | function calls | | GE |
|------------|-------|--------|----------------|----------|--------|
| | total | reject | serial | parallel | |
| 10^{-5} | 59 | 14 | 354 | – | -5.31 |
| 10^{-6} | 88 | 20 | 528 | – | -6.19 |
| 10^{-7} | 118 | 16 | 708 | – | -7.06 |
| 10^{-8} | 179 | 19 | 1074 | – | -8.10 |
| 10^{-9} | 268 | 19 | 1608 | – | -9.09 |
| 10^{-10} | 409 | 18 | 2454 | – | -10.07 |
| 10^{-11} | 630 | 15 | 3780 | – | -11.07 |
| 10^{-12} | 983 | 12 | 5898 | – | -12.06 |

TOL : tolerance for local errors

total: number of total steps

reject: number of rejected steps

GE : \log_{10} global error

serial: number of function calls in serial computations

parallel: number of function calls in parallel computations

Next we consider the system of $2N$ differential equations which describes the behaviour of the electric currents in a linear lattice circuit with N closed loops (see Fig. 2). The numerical results by the P-C block methods and the PIRKs on a KSR-1 parallel computer are shown in Table 2. The equation to be solved is

$$\begin{cases} y'_j = y_{j+1}, & j = 1, 3, \dots, 2N-1, \\ y'_j = -\frac{y_1 - y_3}{LC} - \frac{y_2 R}{L} - \frac{2\pi f \sin(2\pi f x)}{L}, & j = 2, \\ y'_j = \frac{y_{j-3} - 2y_{j-1} + y_{j+1}}{LC} - \frac{y_j R}{L}, & j = 4, 6, \dots, 2N-2, \\ y'_j = \frac{y_{2N-3} - 2y_{2N-1} - y_{2N} R}{LC}, & j = 2N, \end{cases} \quad (4.2)$$

$$y_j(0) = 0 \quad (j = 1, 2, \dots, 2N)$$

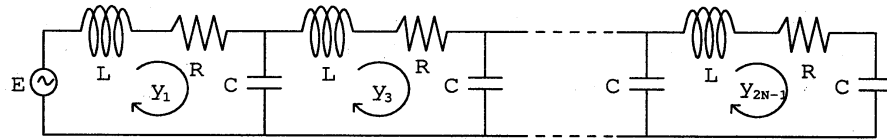


Fig. 3 LRC circuit of N closed loops

Table 2. CPU-times (sec), speed-up ratios, and efficiencies for problem (4.2).

| Type 1 ($r = 4$, order=4, $P = 4$) | | | | Type 2 ($r = 4$, order=4, $P = 3$) | | | |
|---------------------------------------|----------|----------|-----------|---------------------------------------|----------|----------|-----------|
| | $N = 10$ | $N = 50$ | $N = 100$ | | $N = 10$ | $N = 50$ | $N = 100$ |
| T_s (sec) | 55.43 | 323.3 | 647.39 | T_s (sec) | 40.93 | 214.45 | 427.85 |
| T_p (sec) | 41.54 | 180.22 | 343.77 | T_p (sec) | 48.31 | 144.00 | 264.03 |
| $S (= T_s/T_p)$ | 1.33 | 1.79 | 1.88 | $S (= T_s/T_p)$ | 0.85 | 1.49 | 1.62 |
| $E (= S/P)$ | 0.33 | 0.45 | 0.47 | $E (= S/P)$ | 0.28 | 0.50 | 0.54 |

| Type 1 ($r = 5$, order=5, $P = 5$) | | | | Type 2 ($r = 5$, order=6, $P = 4$) | | | |
|---------------------------------------|----------|----------|-----------|---------------------------------------|----------|----------|-----------|
| | $N = 10$ | $N = 50$ | $N = 100$ | | $N = 10$ | $N = 50$ | $N = 100$ |
| T_s (sec) | 32.34 | 163.22 | 327.60 | T_s (sec) | 20.06 | 100.24 | 200.49 |
| T_p (sec) | 22.39 | 82.80 | 158.77 | T_p (sec) | 16.03 | 57.02 | 108.88 |
| $S (= T_s/T_p)$ | 1.44 | 1.97 | 2.06 | $S (= T_s/T_p)$ | 1.25 | 1.76 | 1.84 |
| $E (= S/P)$ | 0.29 | 0.39 | 0.41 | $E (= S/P)$ | 0.31 | 0.43 | 0.46 |

| PIRK (2-stage Gauss, order=4, $P = 2$) | | | | PIRK (3-stage Radau IA, order=5, $P = 3$) | | | |
|---|----------|----------|-----------|--|----------|----------|-----------|
| | $N = 10$ | $N = 50$ | $N = 100$ | | $N = 10$ | $N = 50$ | $N = 100$ |
| T_s (sec) | 223.51 | 1116.12 | 2213.13 | T_s (sec) | 147.71 | 729.27 | 1460.94 |
| T_p (sec) | 326.05 | 941.37 | 1738.35 | T_p (sec) | 159.27 | 473.30 | 897.08 |
| $S (= T_s/T_p)$ | 0.69 | 1.19 | 1.27 | $S (= T_s/T_p)$ | 0.93 | 1.54 | 1.63 |
| $E (= S/P)$ | 0.34 | 0.59 | 0.64 | $E (= S/P)$ | 0.31 | 0.51 | 0.54 |

T_s : CPU-time in serial computation,
 T_p : CPU-time in parallel computation,
 S : speed-up ratio,

P : number of processors,
 E : efficiency.

It follows from the experiments that our block methods are much faster than the PIRKs, and that the block method of Type 2 with $r = 5$ is the fastest one although this method is the most inefficient. It is, in general, true that the smaller the number of processors, the higher the efficiency, and that the greater the dimension of the equation, the higher the efficiency. This may be caused by the unavoidable overheads, like communication and sequential parts in the algorithm.

5 Conclusion

The detailed discussions on the stability and stepsize control mechanism of the block methods are shown in [10]. Further extension for reducing the overheads in parallel computations will be expected.

Acknowledgment. The authors wish to thank the courtesy of the members of Canon Supercomputing S.I. Inc. that helped to use a KSR-1 parallel computer. They would like to thank Hisashi Nara, Professor Emeritus of Tohoku University, for his encouragement.

References

- [1] Birta, L. G. and Abou-Rabia, O., Parallel block predictor-corrector methods for ode's, IEEE Trans. on Computers C-36(1987), 229-311.
- [2] Burrage, K., Parallel and Sequential Methods for Ordinary Differential Equations, Oxford University Press, Oxford, 1995.
- [3] Butcher, J. C., The Numerical Analysis of Ordinary Differential Equations, John Wiley & Sons, New York, 1987.
- [4] Chu, M. T. and Hamilton, H., Parallel solution of ODE's by multi-block methods, SIAM J. Sci. Stat. Comput. 3(1987), 342-353.
- [5] Franklin, M. A., Parallel Solution of Ordinary Differential Equations, IEEE Trans. on Computers C-27(1978), 413-420.
- [6] Gladwell, I., Shampine, L. F., and Brankin, R. W., Automatic selection of the initial step size for an ODE solver, J. Comp. Appl. Math. 18(1987), 175-192.
- [7] Hairer, E., Nørsett S. P., and Wanner, G., Solving Ordinary Differential Equations I (Nonstiff Problems, 2nd Ed.), Springer-Verlag, Berlin, 1993.
- [8] Houwen, P. J. van der and Sommeijer, B.P., Parallel Iteration of High-order Runge-Kutta Methods with Stepsize Control, J. Comp. Appl. Math. 29(1990), 111-127.
- [9] Iserles, A. and Nørsett, S.P., On the theory of parallel Runge-Kutta methods, IMA J. Numer. Anal. 10(1990), 463-488.
- [10] Yamada, S. and Ozawa, K., Parallel Block Methods of Predictor-Corrector Type and Its Stepsize Strategy (in Japanese), in preparation.