VIETNAM JOURNAL OF MATHEMATICS
Volume 23, Number 2, 1995

# AN IMPROVEMENT FOR EXPLICIT PARALLEL RUNGE-KUTTA METHODS[1]

NGUYEN HUU CONG and HOANG THI VI

**Abstract.** *In this paper, by means of replacing the last step-value predictors by two-step predictors and using informations from preceding step we construct the predictor-corrector (PC) methods which are of more accurate predictors. Moreover they reduce the number of sequential right-hand side evaluations and can be implemented on the same number of processors as in case of the parallel iterated Runge-Kutta (PIRK) methods of the same order. Having high-order predictors in addition with sufficiently large stability regions for nonstiff problems, the new PC methods show the improved efficiency when they are compared to the PIRK methods.*

## 1. INTRODUCTION

In this paper we study a class of (explicit) predictor-corrector (PC) methods obtained by predictor-corrector iteration (or fixed point iteration) of Runge-Kutta correctors for solving the initial-value problem (IVP) for nonstiff, ordinary differential equations (ODEs)

$$\frac{d\boldsymbol{y}(t)}{dt} = \boldsymbol{f}(\boldsymbol{y}(t)) \,, \quad \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \,, \quad t_0 \le t \le T \,. \tag{1.1}$$

The efficiency of this class of the PC methods which are based either on Runge-Kutta correctors (for first-order ODEs), or on Runge-Kutta-Nyström correctors (for special second-order ODEs) depends on the accuracy on the predictions. In [3] by using a large number of processors, together with the approximation to the step point value, a whole block of approximations to the exact solutions at the off-step point is computed. This block of approximations can be used in the next step for obtaining a high-order predictor formula. In this paper we use the

[1]This work is supported by the National Basis Research Program in Natural Sciences and Research Program B93-05-71.

stage values and the step point value computed at the previous step to construct more accurate prediction by extrapolation techniques. This approach was used in [5] for parallel diagonal-implicit iteration of RK methods. Here, we can not obtain high accurate prediction as in [3] because the stage order of the corrector methods is lower than the step point order and the block dimension is also smaller. However no additional processors are needed in the implementation. As an analogue of the improved PIRKN (IPIRKN) methods proposed in [7] to improve PIRKN methods, the PC methods considered in this paper will be termed *improved PIRK (IPIRK) methods.*

In order to facilitate a comparison of the IPIRK methods presented in this paper to sequential explicit RK and already available PIRK methods, we list below the main characteristic for the efficiency of these methods.

For an explicit RK method of order $p$, it is the number of stages (or right-hand side evaluations) per step that is the main characteristics for the efficiency of the method. In the case of PIRK methods, the method efficiency is mainly characterized by the number of stages $k$ per step in each processor. The minimal numbers of stages $k$ of various $p$-order sequential explicit RK and PIRK methods actually constructed are listed in Table 1.1.

*Table 1.1.* Numbers of sequential stages $k$ for various $p$-order methods

| Order $p$ | $\leq 4$ | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|
| Explicit RK methods | $k=p$ | $k=p+1$ | $k=p+1$ | $k=p+2$ | $k=p+3$ | - | $k=p+7$ |
| PIRK methods | $k=p$ | $k=p$ | $k=p$ | $k=p$ | $k=p$ | $k=p$ | $k=p$ |

For simplicity of notation, all formulas refer to IVPs for scalar equations and we shall extensively use "componentwise" notation, that is for any given vector $\boldsymbol{v} = (v_j)$, $f(\boldsymbol{v})$ denotes the vector with entries $f(v_j)$.

## 2. IMPROVED PIRK METHODS

The starting point is a fully implicit $s$-stage RK method. We shall consider only the IRK methods of Gauss – Legendre type (Gauss – Legendre IRK methods) because the step point value which is different

from stage values, gives a larger block dimension $s+1$ for generating higher order predictor formula (the parameters of Gauss–Legendre IRK methods of order up to 40 can be found in [1]). For a scalar equation, this method assumes the form.

$$\boldsymbol{Y}_n = y_n \boldsymbol{e} + hAf(\boldsymbol{Y}_n)\,, \tag{2.1a}$$

$$y_{n+1} = y_n + h\boldsymbol{b}^T f(\boldsymbol{Y}_n)\,, \tag{2.1b}$$

where $A$ is a $s \times s$ matrix, $\boldsymbol{b}, \boldsymbol{e}$ are $s$–dimensional vectors, $\boldsymbol{e} = (1, 1, \ldots, 1)^T$, and $\boldsymbol{Y}_n$ is the stage vector corresponding to the $n$–th step.

Consider the following fixed point iteration scheme:

$$\begin{aligned} \boldsymbol{Y}_n^{(1)} &= y_n \boldsymbol{e} + hAf(\boldsymbol{Y}_n^{(0)})\,, \\ \boldsymbol{Y}_n^{(j)} &= y_n \boldsymbol{e} + hAf(\boldsymbol{Y}_n^{(j-1)}),\ j = 2, \ldots, m\,, \end{aligned} \tag{2.2b}$$

$$y_{n+1} = y_n + h\boldsymbol{b}^T f(\boldsymbol{Y}_n^{(m)})\,. \tag{2.2c}$$

In order to start the iteration (2.2b), we need a predictor method to compute the initial approximation $\boldsymbol{Y}_n^{(0)}$. In [5] an one–step predictor based on last step–value was used. In this paper like for IPIKN and PDIRK methods (see e.g. [5], [7]), by using information from the preceding step, that is the values of $y_n$ and the stage vector $\boldsymbol{Y}_n^{(m)}$ computed in the last step, we constructed more accurate two–step predictors

$$\boldsymbol{Y}_n^{(0)} = V\boldsymbol{Y}_{n-1}^{(m)} + \boldsymbol{w}y_n\,, \tag{2.2a}$$

where $V$ is $s \times s$ matrix, $\boldsymbol{w}$ is $s$–dimensional vector, both being determined by the order conditions. Notice that the values of the stage vector $\boldsymbol{Y}_{n-1}^{(m)}$ and step value $y_n$ are already provided at the previous step. Moreover, the components of the vectors $\boldsymbol{Y}_n^{(0)}$ and $\boldsymbol{Y}_n^{(j)}$ can be computed in parallel, provided that only $s$ processors are available. Therefore, the computational time needed for one iteration of (2.2b) is equivalent to the time required to evaluate on right–hand side function on a sequential computer. The PC method (2.2) is of the same nature as the IPIRKN methods considered in [7]. Hence, the method (2.2) is called *improved PIRK (IPIRK) method.*

Assume that the function $f(y)$ is Lipschitz continuous and that (2.2a) defines a $q$–order predictor formula (i.e., $\boldsymbol{Y}_n^{(0)} - \boldsymbol{Y}_n = O(h^{q+1})$. Then we have the following order relations for the iteration errors associated with the stage vector and step point value

$$\begin{aligned} \boldsymbol{Y}_n - \boldsymbol{Y}_n^{(m)} &= O(h^{m+q+1}), \\ u_{n+1} - y_{n+1} &= h\boldsymbol{b}^T(f(\boldsymbol{Y}_n) - f(\boldsymbol{Y}_n^{(m)})) = O(h^{m+q+2}), \end{aligned} \tag{2.3}$$

where $u_{n+1}$ denotes the corrector solution at the step point $t_{n+1}$. The local truncation error of the IPIRK method (2.2) can be written as the sum of the truncation error of the $p$–order Gauss–Legendre IRK corrector method and the iteration error of the IPIRK method:

$$\begin{aligned} y(t_{n+1}) - y_{n+1} &= (y(t_{n+1}) - u_{n+1}) + (u_{n+1} - y_{n+1}) \\ &= O(h^{p+1}) + O(h^{m+q+2}). \end{aligned}$$

Thus we have following theorem analogous to Theorem 2.1 formulated in [7].

**Theorem 2.1.** *Let the generating corrector method (2.1) be of order $p$. Then on $s$–processor computers, the IPIRK method defined by (2.2) represents an explicit method of Runge–Kutta type of order $p^* = \min\{p,\ m+q+1\}$ requiring $m+1$ sequential right–hand side evaluations per step.*

*Remark 2.1.* From Theorem 2.1, by setting $m = p - q - 1$, we obtain a IPIRK method of maximum order $p^* = p$ (order of the corrector) with only $p - q$ sequential right–hand side evaluations per step.

### 2.1. Order conditions for the predictor methods

Since a $s$–stage Gauss–Legendre IRK corrector has the stage order $s$, and the block of approximations in the predictor method defined by (2.2a) has dimension $s+1$, we can construct predictor method of order $s$.

We now suppose that a fixed stepsize is used in the integration process. The conditions that the predictor is of order $s$ is derived by replacing $\boldsymbol{Y}_{n-1}^{(m)}$, $y_n$ and $\boldsymbol{Y}_n^{(0)}$ by the exact solution values $y(t_{n-1}\boldsymbol{e} + h\boldsymbol{c})$, $y(t_n)$ and $y(t_n\boldsymbol{e} + h\boldsymbol{c})$ (cf. [7], [8], [9]). Let us denote $\boldsymbol{a} = (\boldsymbol{e}^T, 1)^T$, $\boldsymbol{e}^* = (\boldsymbol{e}^T, 1)^T$. Then by requiring that the predictor method is of order $s$ in $h$, we are led to the conditions

$$\begin{aligned} &y(t_n\boldsymbol{e} + h\boldsymbol{c}) - V y(t_{n-1}\boldsymbol{e} + h\boldsymbol{c}) - y(t_n)\boldsymbol{w} = \\ &= y(t_n\boldsymbol{e} + h\boldsymbol{c}) - (V, \boldsymbol{w})y(t_n\boldsymbol{e}^* + h(\boldsymbol{a} - \boldsymbol{e}^*)) = O(h^{s+1}). \end{aligned} \tag{2.4a}$$

Using the relation $y(te+hx) = \exp\left(hx\frac{d}{dt}\right)y(t)$, we can expand left–hand side of (2.4a) in powers of $h$ and obtain

$$\left[\exp\left(h(c+e)\frac{d}{dt}\right) - (V, w)\exp\left(ha\frac{d}{dt}\right)\right]y(t_n) = O(h^{s+1})\,. \qquad (2.4b)$$

The relations (2.4) yield the conditions

$$(c+e)^j - (V, w)a^j = 0\,,\quad j = 0, \ldots, s\,. \qquad (2.5a)$$

Here, the powers of vectors mean componentwise powers. Now, let us define the matrices

$$P = (e,\ (c+e),\ (c+e)^2, \ldots,\ (c+e)^s)\,,\quad Q = (e^*, a, a^2, \ldots, a^s)\,,$$

where $P$ and $Q$ are $s\times(s+1)$ and $(s+1)\times(s+1)$ matrices, respectively. Then the condition (2.5a) can be written in the form

$$P - (V, w)\,Q = O\,. \qquad (2.5b)$$

Since the abscissas $a_j$ defined in this paper are distinct, we can derive

$$(V, w) = PQ^{-1}\,. \qquad (2.5c)$$

From Theorem 2.1 we deduce the following corollary:

**Corollary 2.1.** *Let the matrix $V$ and vector $w$ be defined according to (2.5), let $p$ be the order of the corrector method (2.1). Then (2.2) define an IPIRK method of order $p^* = \min\{p,\ m+s+1\}$.*

*Remark 2.2.* From Corollary 2.1, by setting $m = p - s - 1$, we obtain an IPIRK method of maximum order $p^* = p$ (order of the corrector) with only $p$–$s$ sequential right–hand side evaluations per step.

Since a $s$–stage Gauss–Legendre IRK method has the order of accuracy $p = 2s$, Remark 2.2 implies that for any given even $p$, a $p$–order IPIRK method requires only $k = p/2$ sequential right–hand side evaluations per step. Comparing the number $k$ of sequential stages of explicit RK and PIRK methods listed in Table 1.1 with those of IPIRK methods, we conclude that computational costs per step of IPIRK methods are much cheaper. On the other hand, the error constant of the extrapolation error associated with $Y_n^{(0)}$ is possibly large and therefore, in practical use, we may need a few more iterations which may compensate the large error constants.

### 2.2. Convergence boundaries

The convergence factors and convergence boundaries of the IPIRK methods are identical with those of the (B)PIRK methods studied in [3] and also in [8]. Hence we briefly mention the most important results.

The rate of convergence was determined by using the model test equation $y' = \lambda y$, where $\lambda$ runs through the eigenvalues of the Jacobian matrix $\partial \boldsymbol{f}/\partial \boldsymbol{y}$ (cf. e.g., [3, 6, 7, 8]). For this equation, we obtain the iteration error equation

$$\boldsymbol{Y}_n^{(j)} - \boldsymbol{Y}_n = zA\big[\boldsymbol{Y}_n^{(j-1)} - \boldsymbol{Y}_n\big]\,,\quad z := \lambda h\,,\quad j = 1, \ldots, m\,.$$

Hence, with respect to the model test equation, the rate of convergence is determined by the spectral radius $\rho(A)$ of the matrix $A$. We shall call $\rho(A)$ the *convergence factor* of IPIRK methods. Requiring that $\rho(zA) < 1$ leads us to the convergence condition

$$|z| < \frac{1}{\rho(A)} \quad \text{or} \quad h < \frac{1}{\rho(A)\,\rho(\partial \boldsymbol{f}/\partial \boldsymbol{y})}\,. \tag{2.6}$$

This convergence condition is of the same form as the stability condition associated with a RK method. In analogy with the notion of the *stability boundary*, we shall call $1/\rho(A)$ the *convergence boundary*. We refer to [3] and [8] for specifications of the convergence boundaries and convergence factors for the various (B)PIRK methods.

### 2.3. Stability boundaries

The linear stability of the IPIRK method (2.2) is investigated by using the model test equation $y' = \lambda y$, where $\lambda$ runs through the eigenvalues of $\partial \boldsymbol{f}/\partial \boldsymbol{y}$.

**Theorem 2.2.** *For the model test equation $y' = \lambda y$, the numerical solution obtained by the IPIRK method (2.2) satisfies the recursion*

$$\begin{pmatrix} \boldsymbol{Y}_n^{(m)} \\ y_{n+1} \end{pmatrix} = M_m(z) \begin{pmatrix} \boldsymbol{Y}_{n-1}^{(m)} \\ y_n \end{pmatrix}\,,\quad z = \lambda h\,, \tag{2.7a}$$

*where $M_m(z)$ is the amplification matrix*

$$M_m(z) = \begin{pmatrix} (zA)^m V & (zA)^m \boldsymbol{w} + (I - zA)^{-1}(I - (zA)^m)\boldsymbol{e} \\ z\boldsymbol{b}^T(zA)^m V & 1 + z\boldsymbol{b}^T\big((zA)^m \boldsymbol{w} + (I - zA)^{-1}(I - (zA)^m)\boldsymbol{e}\big) \end{pmatrix} \tag{2.7b}$$

*Proof.* Applying the IPIRK method (2.2) to the model equation (cf. [8]), we obtain

$$\begin{aligned} \boldsymbol{Y}_n^{(m)} &= \boldsymbol{e}y_n + zA\boldsymbol{Y}_n^{(m-1)} \\ &= (I + zA + (zA)^2 + \cdots + (zA)^{m-1})\boldsymbol{e}y_n + (zA)^m \boldsymbol{Y}_n^{(0)} \\ &= (zA)^m V \boldsymbol{Y}_{n-1}^{(m)} + ((I - zA)^{-1}(I - (zA)^m)\boldsymbol{e} \\ &\quad + (zA)^m \boldsymbol{w})y_n \,, \end{aligned} \tag{2.8a}$$

$$\begin{aligned} y_{n+1} &= y_n + z\boldsymbol{b}^T \boldsymbol{Y}_n^{(m)} \\ &= z\boldsymbol{b}^T (zA)^m V \boldsymbol{Y}_{n-1}^{(m)} + (1 + z\boldsymbol{b}^T((zA)^m \boldsymbol{w} \\ &\quad + (I - zA)^{-1}(I - (zA)^m)\boldsymbol{e}))y_n \,. \end{aligned} \tag{2.8b}$$

Combining the relations (2.7b), (2.8a), (2.8b), the one–step recursion (2.7a) of Theorem 2.2 is easily obtained.

The $(s+1) \times (s+1)$ matrix $M_m(z)$, which determines the stability of the IPIRK method, will be called the amplification matrix, and its spectral radius $\rho(M_m(z))$ the stability function. The stability region $D_m$, the real and imaginary stability intervals $(-\beta_{re}(m), 0)$, $(-\beta_{im}(m), \beta_{im}(m))$ of the IPIRK method corresponding to a given $m$ are respectively defined by

$$D_m := \{z \in C : \rho(M_m(z) \leq 1\} \cap \{z \in C : \mathrm{Re}(z) \leq 0\},$$

$$(-\beta_{re}(m), 0) := \{z \in C : \mathrm{Im}(z) = 0\} \cap D_m \,,$$

$$(-\beta_{im}(m), \beta_{im}(m)) := \{z \in C : \mathrm{Re}(z) = 0\} \cap D_m \,,$$

where $\beta_{re}(m)$, $\beta_{im}(m)$ are called the real and imaginary stability boundaries associated with $m$, respectively. From (2.8) we see that if $z$ satisfies (2.6), then $(zA)^m$ tends to zero matrix as $m \to \infty$. As a consequence, the spectral radius of $M_m(z)$ converges to the absolute value of the stability function of the corrector method as $m \to \infty$, i.e.,

$$\rho(M_m(z)) \to |R(z)| \text{ as } m \to \infty \,,$$

where $R(z) = 1 + z\boldsymbol{b}^T(I - zA)^{-1}\boldsymbol{e}$. Hence, the *asymptotic* stability region $D_\infty$ as $m \to \infty$ contains the intersection (on the left–half complex $z$–plane $C^-$) of the stability region of the generating corrector and the

region of convergence in the complex $z$–plane $C$ defined by (2.6). For IPIRK methods studied in this paper, where the corrector method is $A$–stable, its asymptotic stability region is larger than the convergence region in the left–half $z$–plane, and the real and imaginary asymptotic stability boundaries $\beta_{re}(\infty)$, $\beta_{im}(\infty)$ are not less than the convergence boundaries of the IPIRK methods, i. e.,

$$D_\infty \supset \{z \in C : |z| < 1/\rho(A)\} \cap C^- , \ \beta_{re}(\infty) = \beta_{im}(\infty) \geq 1/\rho(A) .$$

Table 2.1 list the stability pairs $(\beta_{re}(m), \beta_{im}(m))$ for various IPIRK method for a few values of $m$. The stability pairs corresponding to the minimal value of m needed to reach the order of the correctors are indicated in bold face. We observe that for small m, the stability of IPIRK methods is rather poor, but for $m \geq P/2$, their stability is large enough for nonstiff problems.

*Table 2.1.* Stability pairs $(\beta_{re}(m), \beta_{im}(m))$ for various IPIRK methods

| $p$–order methods | $m=1$ | $m=2$ | $m=3$ | $m=4$ | $m=5$ | ... | $m=\infty$ |
|---|---|---|---|---|---|---|---|
| $p=4$ | **(0.25, 0.17)** | (0.90, 0.27) | (1.60, 0.30) | (2.74, 0.30) | (2.10, 0.30) | ... | (≥3.46, ≥3.46) |
| $p=6$ | (0.05, 0.05) | **(0.45, 0.35)** | (0.97, 0.51) | (1.50, 0.65) | 1.97, 0.69) | ... | (≥4.65, ≥4.65) |
| $p=8$ | (0.01, 0.01) | (0.22, 0.23) | **(0.64, 0.63)** | (1.12, 0.83) | (1.62, 1.11) | ... | (≥6.06, ≥6.06) |
| $p=10$ | (0.00, 0.00) | (0.11, 0.11) | (0.42, 0.43) | **(0.83, 0.83)** | (1.29, 1.21) | ... | (≥7.30, ≥7.30) |

## 2.4. Iteration error for the test equation

The *iteration error* $u_{n+1} - y_{n+1} = O(h^{m+s+2})$ could be established by using (2.3) and $s$–order predictor constructed in Subsection 2.1 for any right–hand side function f(y(t)). In this subsection we will obtain an explicit expression for the local iteration error of IPIRK method (2.2) with respect to the model test equation $y' = \lambda y$. By defining the iteration error matrix

$$E_m(z) = \Big( -z\boldsymbol{b}^T(zA)^m V \quad z\boldsymbol{b}^T((I - zA)^{-1}(zA)^m \boldsymbol{e} - (zA)^m \boldsymbol{w}) \Big) \quad (2.9)$$

we have the following theorem:

**Theorem 2.3.** *For the model test equation $y' = \lambda y$, the local iteration error is explicitly defined by the equality*

$$u_{n+1} - y_{n+1} = E_m(z) \begin{pmatrix} \boldsymbol{Y}_{n-1}^{(m)} \\ y_n \end{pmatrix} = O(h^{m+s+2}) .$$

*Proof.* By means of (2.7), (2.9) and the stability function $R(z)$ of Gauss–Legendre IRK corrector method, we have the following representation

$$M_m(z) = \begin{pmatrix} (zA)^m V & (zA)^m \boldsymbol{w} + (I - zA)^{-1}(I - (zA)^m)\boldsymbol{e} \\ O_{1s} & R(z) \end{pmatrix} - \begin{pmatrix} O_{s,s+1} \\ E_m(z) \end{pmatrix}$$

where $O_{ij}$ is $i \times j$ matrix, with zero entries. In view of the order condition for the predictor methods (2.2a) and the recursion (2.7) in Theorem 2.2, Theorem 2.3 easily follows.

## 3. NUMERICAL EXPERIMENTS

In this section we report the numerical results obtained by the various PIRK and IPIRK methods. The absolute error obtained at the end of integration interval is presented in the form $10^{-d}$ (d may be interpreted as the number of correct decimal digits (NCD)). In order to see the efficiency of the various PIRK and IPIRK methods, we follows a dynamical strategy for determining the number of iterations in the successive steps (cf. [7], [8]). We have

$$\|\boldsymbol{Y}_n^{(m)} - \boldsymbol{Y}_n^{(m-1)}\|_\infty \leq Ch^p \text{ and } m \geq p/2 - 1 , \qquad (3.1)$$

where $p$ denotes the order of the corrector methods, $C$ is a problem– and method–dependent parameter. Furthermore, in the tables of results, $N_{\text{seq}}$ denotes the total number of sequential right hand side evaluations, and $N_{\text{steps}}$ denotes the total number of integrations steps.

### 3.1. Fehlberg problem

As a first numerical test, we apply the various PIRK and IPIRK methods to the often–used Fehlberg problem (cf. [2, p. 174]), for $0 \leq t \leq 5$,

$$\begin{aligned} \frac{dy_1(t)}{dt} &= 2ty_1(t)\log(\max\{y_2(t),\, 10^{-3}\}),\ y_1(0) = 1\,, \\ \frac{dy_2(t)}{dt} &= -2ty_2(t)\log(\max\{y_1(t),\, 10^{-3}\}),\ y_2(0) = e\,, \end{aligned} \tag{3.2}$$

with exact solution $y_1(t) = \exp(\sin(t^2))$, $y_2(t) = \exp(\cos(t^2))$. The results listed in Table 3.1 clearly show that the IPIRK methods are much more efficient than PIRK methods of the same order and of the same number of processors. To obtain the same accuracy (NCD), the number of sequential right–hand side evaluations ($N_{\text{seq}}$) required by IPIRK methods is in about 60% of that required by PIRK methods of the same order.

*Table 3.1.* Values of NCD / $N_{\text{seq}}$ for problem (3.2) obtained by the various $p$–order PIRK and IPIRK methods

| PC methods | $p$ | $N_{\text{steps}} = 100$ | $N_{\text{steps}} = 200$ | $N_{\text{steps}} = 400$ | $N_{\text{steps}} = 800$ | $N_{\text{steps}} = 1600$ | $C$ |
|---|---|---|---|---|---|---|---|
| PIRK | 4 | 2.7/392 | 4.0/842 | 5.2/1756 | 6.5/3650 | 7.7/7409 | $10^3$ |
| IPIRK | 4 | 2.6 /259 | 4.0/532 | 5.2/1125 | 6.5/2320 | 7.7/4794 | $10^3$ |
| PIRK | 6 | 5.2/601 | 7.0/1245 | 8.9/2542 | 10.7/5199 | 12.5/10488 | $10^3$ |
| IPIRK | 6 | 5.2/405 | 7.1/818 | 8.9/1634 | 10.7/3304 | 12.5/6694 | $10^3$ |
| PIRK | 8 | 7.8/774 | 10.2/1603 | 12.6/3297 | 15.1/6674 | 17.5/13468 | $10^3$ |
| IPIRK | 8 | 7.8/525 | 10.2/1070 | 12.6/2153 | 15.1/4276 | 17.5/8515 | $10^3$ |
| PIRK | 10 | 9.9/942 | 12.9/1947 | 15.9/3973 | 18.9/8134 | 22.0/16407 | $10^3$ |
| IPIRK | 10 | 9.9/636 | 12.9/1272 | 15.9/2537 | 18.9/5092 | 22.0/10176 | $10^3$ |

### 3.2. Euler's equation

Next, we solve Euler's equation of motion for the rigid body without external forces (problem JACB in [2, p. 236]) by various PIRK and IPIRK methods

$$\begin{aligned}
\frac{dy_1(t)}{dt} &= y_2(t)\, y_3(t), \qquad y_1(0) = 0, \\
\frac{dy_2(t)}{dt} &= -y_1(t)\, y_3(t), \qquad y_2(0) = 1, \quad 0 \le t \le 20, \qquad (3.3) \\
\frac{dy_3(t)}{dt} &= -k^2\, y_1(t)\, y_2(t), \quad y_3(0) = 1, \quad k^2 = 0.51.
\end{aligned}$$

The exact solution of the problem (3.3) is given by the Jacobian elliptic functions $y_1(t) = \mathrm{sn}(t\,;\,k)$, $y_2(t) = \mathrm{cn}(t\,;\,k)$, $y_3(t) = \mathrm{dn}(t\,;\,k)$. Following the same testing procedure as described above, we obtain the results as given in Table 3.2 which once again show the described advantage of IPIRK methods over PIRK methods.

*Table 3.2.* Values of NCD / $N_{\mathrm{seq}}$ for problem (3.3) obtained by the various $p$–order PIRK and IPIRK methods

| PC methods | $p$ | $N_{\mathrm{steps}} = 100$ | $N_{\mathrm{steps}} = 200$ | $N_{\mathrm{steps}} = 400$ | $N_{\mathrm{steps}} = 800$ | $N_{\mathrm{steps}} = 1600$ | $C$ |
|---|---|---|---|---|---|---|---|
| PIRK | 4 | 2.3/300 | 5.1/800 | 6.3/1600 | 7.5/3200 | 8.9/6571 | $10^1$ |
| IPIRK | 4 | 3.2/201 | 4.5/402 | 5.7/802 | 8.5/2002 | 9.5/4292 | $10^1$ |
| PIRK | 6 | 5.1/486 | 7.8/1126 | 11.2/2345 | 12.5/4775 | 14.3/9600 | $10^0$ |
| IPIRK | 6 | 5.9/307 | 8.1/639 | 10.0/1419 | 12.9/3202 | 14.8/6402 | $10^0$ |
| PIRK | 8 | 8.2/678 | 11.1/1470 | 14.0/3028 | 16.7/6195 | 19.1/12540 | $10^{-1}$ |
| IPIRK | 8 | 8.7/408 | 11.3/896 | 14.2/1840 | 17.2/3737 | 20.7/7600 | $10^{-1}$ |
| IPRK | 10 | 10.1/765 | 13.4/1655 | 16.8/3479 | 19.6/7095 | 23.2/14968 | $10^{-1}$ |
| IPIRK | 10 | 10.0/447 | 13.5/959 | 16.9/1956 | 20.3/4040 | 23.1/8393 | $10^{-1}$ |

## 4. CONCLUSIONS

This paper proposes a class of IPIRK methods which are obtained from the class of PIRK methods considered in [4] by means of replacing last step value predictors by high–order extrapolation predictors. The

higher precision in the predictor formulas has the effect to deduce considerably the number of sequential stages. The stability investigations have shown that the IPIRK methods have sufficiently large stability regions for nonstiff problems. The numerical experiments also have confirmed a better performance of the IPIRK methods in the comparison with the PIRK methods of the same order and the same number of processors.

These conclusions encourage us to pursue the investigations of IPIRK methods. In the forthcoming paper (cf. [10]) we will consider a stepsize strategy for IPIRK methods.

## REFERENCES


1. W. Glasmacher and D. Sommer, *Implizite Runge – Kutta Formeln*, Westdeutcher Verlag , Koln und Opladen, 1966.
2. E. Hairer, S. P. Norsett, and G. Wanner, *Solving ordinary differential equations, I. Nonstiff problems*, Springer – Verlag, Berlin, 1987.
3. P. J. van der Houwen and Nguyen Huu Cong, *Parallel block predictor–corrector methods of Runge – Kutta type*, Appl. Numer. Math., **13** (1993), 109 – 123.
4. P. J. van der Houwen and B. P. Sommeijer, *Parallel iteration of high–order Runge – Kutta methods with stepsize control*, J. Comput. Appl. Math., **29** (1990), 111 – 127.
5. P. J. van der Houwen and B. P. Sommeijer, *Analysis of parallel diagonal–implicit iteration of Runge – Kutta methods*, Proceedings of the International Conference on Parallel Methods for Ordinary Differential Equations, Grado (It), Sept. 10 – 13, 1991, Appl. Numer. Math., **11** (1993), 169 – 180.
6. Nguyen Huu Cong, *Note on the performance of direct and indirect Runge – Kutta – Nyström methods*, J. Comput. Appl. Math., **45** (1993), 347 – 355.
7. Nguyen Huu Cong, *An improvement for parallel–iterated Runge – Kutta – Nyström methods*, Acta math. Viet., **18** (2) (1993), 295 – 308.
8. Nguyen Huu Cong, *Parallel iteration of symmetric Runge – Kutta methods for nonstiff initial value problems*, J. Comput. Appl. Math., **51** (1994), 117 – 125.
9. Nguyen Huu Cong, and Taketomo Mitsui, *Collocation–based two–step Runge – Kutta methods*, Japan J. Indust. Appl. Math., **13** (1996), to appear.
10. Nguyen Huu Cong, *Variable stepsize strategies for parallel predictor corrector method of Runge – Kutta type*, in preparation.


*Received October 1, 1993*
*Revised February 19, 1994*

*Faculty of Mathematics,*
*Mechanics and Informatics,*
*University of Hanoi,*
*Hanoi, Vietnam*