Journal of Computational Mathematics, Vol.18, No.4, 2000, 419–430.

TWO ITERATION METHODS FOR SOLVING LINEAR ALGEBRAIC SYSTEMS WITH LOW ORDER MATRIX A AND HIGH ORDER MATRIX B: $Y = (A \otimes B)Y + \Phi$*1)

Shuang-suo Zhao
(*Depart. of Math. & Computing Engieering, Ningxia University, Yinchuan, 750021, China*)
Zhang-hua Luo
(*Depart. of Math. & Computer Science, Guangxi Normal University, Guilin, 541004, China*)
Guo-feng Zhang
(*Depart. of Math., Lanzhou University, Lanzhou, 730000, China*)

Abstract

This paper presents optimum an one-parameter iteration (OOPI) method and a multi-parameter iteration direct (MPID) method for efficiently solving linear algebraic systems with low order matrix A and high order matrix B: $Y = (A \otimes B)Y + \Phi$. On parallel computers (also on serial computer) the former will be efficient, even very efficient under certain conditions, the latter will be universally very efficient.

Key words: System of algebraic equations, Iteration method, Iteration direct method, Solution method for stiff ODEs

1. Introduction

It is well known that for IVP of stiff ODEs

$$y' = f(y), \quad t_0 < t \le T \quad y(t_0) = y_0 \in R^m, \quad f : \Omega \in R^m \in R^m, \quad m \gg 0 \tag{1.1}$$

implicit method with good stability have to be used, e.g., IRK methods[7], implicit block methods[4,12−17,18], etc. At each integral step, each of all these methods brings about solving block nonlinear equation systems

$$Y = h(A \otimes I_m)F(Y) + \Phi_1, \quad A \in R^{s*s}, \quad Y, F(Y), \Phi \in R^{ms}, \quad ms \gg 0, \tag{1.2}$$

where h is the stepsize, $\otimes$ kronecker product, $I_m \in R^m$ identity matrix, $Y = (y_1^T, y_2^T, \cdots, y_s^T)^T$, $F(Y) = (f(y_1)^T, \cdots, f(y_s)^T)^T$. Now, efficiently solving (1.2) become a key of efficiently solving (1.1).

* Received May 26, 1997.
1) Supported by the national natural science foundation of China (19671039).

Let

$$f(y) = \overline{B}y + g, \tag{1.3}$$

where $\overline{B} \in R^{m*m}$ is a constant matrix, $g \in R^m$ is a constant vector. For the definite problems of linear evolution equations systems

$$\frac{\partial u}{\partial t} = Lu + l(t, x_1, x_2, \cdots, x_q),$$

where L is a linear partial differential operator with respect to the space variables $x_1, x_2, \cdots, x_q$,
$l(t, x_1, x_2, \cdots, x_q)$ is a known continuous function of the time variable t and space variable $x_1, x_2, \cdots, x_q$. Using the semi-discrete method, we can obtain (1.1)(1.3). Under the condition (1.3), (1.2) can be written as a linear equation system

$$Y = (A \otimes B)Y + \Phi, \tag{1.4}$$

here $B = h\overline{B}$, $\Phi = h((Ae) \otimes g) + \Phi_1, e = (1, 1, \cdots, 1)^T \in R^s$.

The research of solution method for (1.2) have had a number of results[2,3,5,6,17]. We attempt to set up an universal efficient solution method for (1.1), (1.2) by the way of the construction of efficient solution method for (1.4). This aim have been achieved. As the space is limited, the paper only discusses solution methods for (1.4). As to solution methods for (1.2), we shall discuss then in another paper.

In order to set up an universal efficient solution method for (1.1)(1.3)(1.4) which can be generalized to establish an universal efficient solution method for (1.1)(1.2), we do some analyses for (1.4) produced from (1.2)(1.3).

Unlike general linear systems, (1.4) produced from (1.1)(1.3) have following features:

i) A in (1.4) is only determined by the method used by solution of (1.1)(1.3), its orders is lower. Usually, $s \in [2, 6]$, about at most doesn't exceed 10;

ii) To ensure the accuracy of numerical solution, the discrete stepsize h_i adopted in the directions of the space variable $x_i, i = 1(1)q$ are sufficient small, therefore m is a large number. When $m \gg 0$, to solve (1.4) need to use parallel computers(or vector computer) usually.

iii) For the accurate solution $Y = Y^*$ of (1.4), there is an initial approximation Y_0 with good accuracy.

Establishing an efficient solution method for (1.1)(1.2) (or ((1.1)(1.3)(1.4)) we must consider all of the three points.

For a matrix equation

$$A_1X + XB_1 = C_1, \quad A_1 \in R^{s*s}, \quad B_1 \in R^{m*m}, \quad C_1, \quad X \in R^{s*m}, \tag{1.5}$$

which is equivalent to

$$\overline{X} = -(A_1^{-1} \otimes B_1^T)\overline{X} + (A_1^{-1} \otimes I_m)\overline{C}_1$$

as A_1 is a regular matrix, where A_1, B_1 and C_1 are known, X is unknown,

$$\overline{X} = (x_{11}, x_{12}, \cdots, x_{1m}; x_{21}, \cdots, x_{s1}, x_{s2}, \cdots, x_{sm})^T,$$

$$\overline{C}_1 = (c_{11}, c_{12}, \cdots, c_{1m}; c_{21}, \cdots, c_{s1}, c_{s2}, \cdots, c_{sm})^T,$$

when the dimensions of matrices A_1 and B_1 are moderate size (e.g $\leq$ 500) at the most and dense, there are very efficient direct solution methods: the Bartels-Stewart algorithm(B-SA)[1] and the Hessenberg-Schur algorithm(H-SA)[9]. The former(later) requires[9,10]

$$10m^3 + 10s^3 + \frac{5}{2}(m^2 s + ms^2) \quad (\frac{5}{3}m^3 + 10s^3 + 5m^2 s + 2.5ms^2) \tag{1.6}$$

multiplication operations and

$$2m^2 + 2s^2 + ms \quad (3m^2 + 2s^2 + ms) \tag{1.7}$$

storages. But the two algorithms can not be generlized for solving (1.2) efficiently because the Schur decomposition or Hessenberg decomposition for matrix B is contained in the computation formulate. Otherwise, they can not use the good initial approximate to reduce operation count. [11] presented an iterative method for (1.5), but using this method, at each iterative step, one has to compute the inverses of both one $m * m$ matrix and one $s * s$ matrix as well as the products of some matrixes. Therefore, the operation count is very large.

We notice that eigenvalues of A in (1.4) may easily be calculated because A is a low order matrix. Even, in some cases, e.g., the diagonal and simple implicit R-K methods[7] and some block metheds[12,13,16,17] for solving ODEs, they are known. Based on this fact, this paper presents two iterative methods for solving (1.4). After a discussion of the convergence of one-parameter iteration method, by means of all different eigenvalues of A (no need any eigenvalue of B!), the paper presents an optimum one-parameter iteration (OOPI) method (see theorem 3.1) and a multi-parameter iteration direction (MPID) method which are suitable for use on serial computer as well as parallel computers.

It is not difficult to conclude that (1.4) have a unique solution if and only if

$$\nu_k \lambda_j \neq 1, \quad k = 1(1)s, \quad j = 1(1)m, \tag{1.8}$$

where $\nu_k, \quad \lambda_j$ denote eigenvalues of A, B, respectively. To ensure the existence and uniqueness of the solution of (1.4), we assume (1.8) holds throughout.

Section 2 in the paper discusses the convergence of the one-parameter iteration method. Section 3 set up the OOPI method, and gives calculation method of the optimum parameter μ_{opt}. Section 4 set up the MPID method and gives its implementation strategy. Finally, section 5 gives a number of numerical tests. The results show that

the OOPI method is efficient even is very efficient under the certain condition, MPID method is universally very efficient.

2. Convergence

The one-parameter iteration method for (1.4) is

$$\begin{cases} W(\mu)(Y_i - Y_{i-1}) = D(Y_{i-1}), \quad i = 1, 2, \cdots, \quad \mu \in R, \\ W(\mu) := I_s \otimes (I_m - \mu B), \quad D(Y) := \Phi - (I_{ms} - A \otimes B)Y, \end{cases} \tag{2.1}$$

where $I_n \in R^{n*n}$ is the identical matrix.

Theorem 2.1. *let* $\nu_k = u_k + iv_k, \quad k = 1(1)s, \quad \lambda_j = x_j + iy_j, \quad j = 1(1)m, \quad i^2 = -1, \quad u_k, v_k, x_j, y_j \in R, \quad \|\cdot\| := \|\cdot\|_2, \quad E_{k,j} := \|\lambda_j\|^2 u_k - x_j, \quad \Omega_0 := \{(k,j)|E_{k,j} = 0\}, \quad \Omega_+ := \{(k,j)|E_{k,j} > 0\}, \quad \Omega_- := \{(k,j)|E_{k,j} < 0\}.$ *Then* (2.1) *is convergent iff*

i) *eigenvalue condition:* $\|\nu_k\| \cdot \|\lambda_j\| < 1, \forall (k,j) \in \Omega_0$ (*this item would be canceled if* $\Omega_0 = \phi$), *and*

ii) *iteration parameter conditions:* μ *satisfies*

ii)$_1$. $1 - 2\mu x_j + \mu^2 \|\lambda_j\|^2 \neq 0, \quad j = 1(1)m,$

ii)$_2$. $$\max_{(k,j)\in\Omega_+} \frac{(\|\nu_k\| \cdot \|\lambda_j\|)^2 - 1}{2E_{kj}} < \mu < \min_{(k,j)\in\Omega_-} \frac{(\|\nu_k\| \cdot \|\lambda_j\|)^2 - 1}{2E_{kj}}$$

(*the left* (*right*) *side inequation would be canceled if* $\Omega_+ = \phi \quad (\Omega_- = \phi)$).

Proof. Error $E_i = Y - Y_i$ satisfies $E_i = G(\mu)E_{i-1}, \quad G(\mu) = (A - \mu I_s) \otimes ((I_m - \mu B)^{-1}B)$. Eigenvalues of $G(\mu)$ are

$$\Phi_{k,j} = \frac{(u_k - \mu)x_j - v_k y_j - i(v_k x_j + y_j(u_k - \mu))}{(1 - \mu x_j) - i\mu y_j}. \tag{2.2}$$

$|\Phi_{kj}| < 1$ iff

$$(\|\nu_k\| \cdot \|\lambda_j\|)^2 - 1 < 2\mu E_{k,j}, \quad \forall (k,j) \in \Omega_0 \cup \Omega_+ \cup \Omega_-.$$

From definition of $\Omega_0, \Omega_\pm$, the sufficiency conclusion follows. The proof of necessity is evident. □

When $\Omega_0 = \Omega_+ = \phi$, or $\Omega_0 = \Omega_- = \phi$, or $\Omega_+ = \Omega_- = \phi$ but

$$\|\nu_k\| \cdot \|\lambda_j\| < 1, \quad k = 1(1)s, j = 1(1)m \tag{2.3}$$

hold, there always exists such μ that (2.1) is convergent. But, this theorem can't be used as the basic of the selection of μ because it depend on the eigenvalues of B.

We firstly consider a very important case in practice applications for which the eigenvalues of A satisfy the condition

$$u_k \geq 0, \quad k = 1(1)s \quad \text{and} \quad u_k = 0 \Rightarrow v_k = 0. \tag{2.4}$$

If (1.4) is produced from solving stiff ODEs, (2.4) is always satisfied. Write

$$E(\mu):\ 2x_j\mu \le 1\ (\text{if}\ \prod_{l=1}^{s} u_l \ne 0),\quad 2x_j\mu < 1\ \ (\text{if}\ \prod_{l=1}^{s} u_l = 0),\quad j = 1(1)m.$$

Theorem 2.2. *Suppose* i) $\nu_k (1 \le k \le s)$ *satisfy* (2.4); ii) μ *satisfies* $E(\mu)$ *and*

$$\mu > \max_{k\in\Delta} \|\nu_k\|^2/(2u_k),\quad \Delta := \{k|u_k > 0, k = 1(1)s\}. \tag{2.5}$$

Then (2.1) *is convergent.*

Proof. Using (2.4), $E(\mu)$, from (2.2) it follows that

$$|\Phi_{k,j}|^2 = \frac{((\mu-u_k)^2+v_k^2)\|\lambda_j\|^2}{1-2\mu x_j+\mu^2\|\lambda_j\|^2} \begin{cases} \le ((\mu-u_k)^2+v_k^2)/\mu^2 & \text{if}\ \ \prod_{l=1}^{s} u_l \ne 0, \lambda_j \ne 0 \\ < ((\mu-u_k)^2+v_k^2)/\mu^2 & \text{if}\ \ \prod_{l=1}^{s} u_l = 0, \lambda_j \ne 0 \\ = 0 & \text{if}\ \ \lambda_j = 0. \end{cases} \tag{2.6}$$

From (2.6), the desired conclusion follows.

From (2.5), when (2.4) and $E(\mu)$ hold, this theorem can be used as the basic of the selection of μ (see section 3).

From the expression for $|\Phi_{k,j}|^2, E(\mu)$ and $B = h\overline{B}$, we can see that the rate of the convergence of (2.1) increase as h decrease. This is right what a good solution method for (1.1)(1.3)(1.4) must possess.

If (1.4) is produced from solving linear stiff ODEs (1.1)(1.3), then $E(\mu)$ means that the step length h has a restriction of independent of stiff property $\|R\|$. This is also what a good solution method for (1.1), (1.3), (1.4) must possess.

3. Optimum One-Parameter Iteration Method

In this section, we always assume that both (2.4) and $E(\mu)$ hold. All nonzero ν_k are divided into p set $\Omega_k = \{\nu_k, \cdots\}, k = 1(1)p$, the real parts of all elements in the every set are equal, the imaginary parts are equal in the absolute value, and satisfy

$$\frac{\|\nu_k\|^2}{u_k} < \frac{\|\nu_{k+1}\|^2}{u_{k+1}},$$

or

$$\frac{\|\nu_k\|^2}{u_k} = \frac{\|\nu_{k+1}\|^2}{u_{k+1}},\quad \frac{v_k^2}{\|\nu_k\|^2} < \frac{v_{k+1}^2}{\|\nu_{k+1}\|^2},$$

or

$$\frac{\|\nu_k\|^2}{u_k} = \frac{\|\nu_{k+1}\|^2}{u_{k+1}},\quad \frac{v_k^2}{\|\nu_k\|^2} = \frac{v_{k+1}^2}{\|\nu_{k+1}\|^2}.\ \ u_k < u_{k+1},\quad k = 1(1)p-1.$$

When all elements in $\{\nu_k\}$ satisfy (2.4), the paper always assume the number of ν_k satisfies this rule. Write

$$G_k(\mu) := ((\mu-u_k)^2+v_k^2)/\mu^2,\quad \Omega = \{\mu|\mu > \max_{1\le k\le p, u_k>0}(\|\nu_k\|^2/(2u_k))\}. \tag{3.1}$$

The solution of problem:

$$\min_{\mu\in\Omega}\max_{1\le k\le p} G_k(\mu) \tag{3.2}$$

is called the optimum iteration parameter for (2.1), which is denoted by μ_{opt}. The iteration method (2.1) corresponding to μ_{opt} is called the optimum one-parameter iteration (OOPI) method. It is easily verified that $G_k(\mu)$ $(1 \le k \le p)$ has following characters:

i) there is a unique stable (minimum) point. The stable point and minimum value are $\mu_k^{(1)} = \|\nu_k\|^2/u_k > 0$, $G_k(\mu_k^{(1)}) = v_k^2/\|\nu_k\|^2 < 1$, respectively.

ii) $G_k(\mu)$ is the monotonic increasing function, monotonic decreasing function and monotonic increasing function in $(-\infty, 0)$, $(0, \mu_k^{(1)})$ and $(\mu_k^{(1)}, \infty)$ respectively, and $\lim_{\mu\to 0\pm} G_k(\mu) = \infty$, $\lim_{\mu\to\pm\infty} G_k(\mu) = 1$.

iii) Denote $\mu_k^{(0)} = \mu_k^{(1)}/2$, then $G_k(\mu) < 1$ $\forall\mu \in (\mu_k^{(0)}, \infty)$; $G_k(\mu_k^{(0)}) = 1, G_k(\mu) > 1$, $\forall\mu \in (-\infty, \mu_k^{(0)})$; $G_k(\mu) < 1$, $\forall\mu \in (\mu_p^{(0)}, \infty)$.

iv) In $(-\infty, \infty), G_i(\mu)$ and $G_j(\mu)$ $(1 \le i \le p, i \ne j)$ have and only have one intersecting point. The abscissa of the point is

$$\mu_{i,j} = (\|\nu_i\|^2 - \|\nu_j\|^2)/(2(u_i - u_j)) \quad (\text{definition } \mu_{ij} = \infty \text{ when } u_i = u_j);$$

and, when $j < i$, if $u_j \le u_i$ then $\mu_{ij} \ge \mu_i^{(0)}$ else $\mu_{ij} < \mu_i^{(0)}$.

By the characters mentioned above, μ_{opt} can be calculated by following steps:

step 1 r:=p, $\alpha := \mu_p^{(0)}, \beta := 2\alpha$

step 2 $\mu_{r,j_1} := \min\{\beta, \mu_{r,j} | \mu_{r,j} \in [\alpha, \beta], j = 1(1)r-1\}$

step 3 if $\mu_{r,j_1} = \beta$ then $\mu_{opt} := \beta$, end

step 4 if $\mu_{r,j_1} \ge 2\mu_{j_1}^{(0)}$ then $\mu_{opt} := \mu_{r,j_1}$, end

step 5 $\alpha := \mu_{r,j_1}, \beta := \mu_{j_1}^{(1)}, r := j_1$ go to step 2.

It is easily known that if $v_j \equiv 0$ $(j = 1(1)s)$ then

$$\mu_{opt} = (\min_{1\le i\le p} u_i + \max_{1\le i\le p} u_i)/2.$$

Now we have confirmed:

Theorem 3.1. *Suppose* i) *eigenvalues of* A, $\nu_k = u_k + iv_k, k = 1(1)s$, *satisfy* (2.4); ii) $E(\mu_{opt})$ *holds. Then the OOPI method*

$$W(\mu_{opt})(Y_i - Y_{i-1}) = D(Y_{i-1}), \quad i = 1, 2, \cdots \tag{3.3}$$

converges with the most fast rate; and, after each time iteration, the error component corresponding to each eigenvector in a linear independent eigenvector system of $G(\mu_{opt})$ *decrease at least by factor*

$$\rho := \begin{cases} \max_{1\le k\le p}\left(\frac{\sqrt{(\mu_{opt}-u_k)^2+v_k^2}}{\mu_{opt}}\right), & \nu_k \ne 0 \\ \max_{1\le j\le m}\left(\frac{\mu_{opt}\|\lambda_j\|}{\sqrt{1-2\mu_{opt}x_j+\mu_{opt}^2\|\lambda_j\|^2}}\right). & \nu_k = 0; \end{cases} \tag{3.4}$$

□

Now, we compare the OOPI with the B-SA and H-SA. The OOPI require about $2m^2+s^2+2ms$ storages, which are less than (1.7). When A have only one eigenvalue $u \neq 0$ (e.g., A in single implicit R-K method[7] and block θ method[12]), obviously, $\mu_{opt} = u$. Therefore $\rho = 0$. This means that if the maximum order of Jordan blocks is $r(r \leq s)$, then that r times iteration at the most are carried accordance with (3.3) can obtain the accurate solution of (1.4) from any Y_0. Therefore, at this case, (3.3) is a iterative direct method, and needs not more than about $\frac{1}{3}m^3 + O(m^2) + r(2sm^2 + s^2m)$ $(r \leq s)$ multiplication and division operations, where $O(m^2)$ is independent of r. When A is a nonsingular matrix and has multiple difference eigenvalues, then $\rho \neq 0$. If we determine q so that

$$\rho^q \leq \epsilon \Longleftrightarrow \quad q \geq \frac{log\epsilon}{log\rho},$$

then roughly speaking, that about q times iteration are carried accordance with (3.3) can obtain the approximate solution of (1.4) satisfying

$$\|Y^* - Y_q\| \leq \epsilon$$

from any Y_0. This process needs

$$\frac{1}{3}m^3 + O(m^2) + q(2sm^2 + s^2m)$$

multiplication and division operations, where $O(m^2)$ is independent of q. It can be seen from above that for sufficient large m, if $0 < \rho << 1$ then the OOPI is superior to the B-SA and H-SA.

When A is a nonsingular matrix and $v_k \equiv 0, k = 1(1)p$, it can be seen easily that $\rho = \frac{u_p - u_1}{u_p + u_1} < 1$.

4. Multi-Parameter Iteration Direct Method

In this section, we shall set up an iteration solution method of (1.4) for arbitrary A under (1.8).

Lemma 4.1. *let ν be a complex number so that $W(\mu)$ is a nonsingular matrix; both Y_0 and Φ be arbitrary real vectors. Then the solution Y_2 defined by*

$$\begin{cases} W(\nu)(Y_1 - Y_0) = D(Y_0) \\ W(\overline{\nu})(Y_2 - Y_1) = D(Y_1) \end{cases}$$

is a real vector, and may be obtained by

$$W_1(\nu)(Y_2 - Y_0) = (I_{ms} + (A - (\nu + \overline{\nu})I_s) \otimes B)D(Y_0) \tag{4.1}$$

by means of real arithmetic, where

$$W_1(\nu) = I_s \otimes (I_m - (\nu + \overline{\nu})B + \nu\overline{\nu}B^2) = I_s \otimes ((I_m - \nu B)(I_m - \overline{\nu}B)).$$

Lemma 4.2. *Let ν_i be arbitrary complex numbers so that $W(\nu_i)$ are nonsingular matrix $(i = 1, 2)$; both Y_0 and Φ be arbitrary real vector. Then $Y_2 = Z_2$, where Y_2 and Z_2 are defined by*

$$\begin{cases} W(\nu_1)(Y_1 - Y_0) = D(Y_0) \\ W(\nu_2)(Y_2 - Y_1) = D(Y_1) \end{cases}$$

and

$$\begin{cases} W(\nu_2)(Z_1 - Y_0) = D(Y_0) \\ W(\nu_1)(Z_2 - Z_1) = D(Z_1), \end{cases}$$

respectively.

Lemma 4.3. *let A have α $(\alpha \geq 0)$ different real eigenvalues $\nu_{j_1}, \nu_{j_2}, \cdots, \nu_{j_\alpha}$ (may contain zero and negative number), β $(\beta \geq 0)$ different conjugate complex eigenvalue pairs $\nu_{j_{\alpha+1}}, \nu_{j_{\alpha+2}} = \overline{\nu}_{j_{\alpha+1},}, \cdots, \nu_{j_{\alpha+2\beta-1}}, \nu_{j_{\alpha+2\beta}} = \overline{\nu}_{j_{\alpha+2\beta-1}}$, the maximum of orders of all Jordan blocks of A corresponding to ν_{j_l} be r_l; $\delta_l = \sum_{j=1}^{l} r_j$. Then when $\nu_k \lambda_j \neq 1$ $(k = 1(1)s, j = 1(1)m)$, for any $Y_0 \in R^{ms}, Y_{\delta\alpha+2\beta}$ obtained by*

$$\begin{cases} W(\nu_{j_l})(Y_i - Y_{i-1}) = D(Y_{i-1}), \\ i = \delta_{l-1} + 1(1)\delta_l, \quad l = 1(1)\alpha + 2\beta, \quad \delta_0 = 0 \end{cases} \tag{4.2}$$

is the accurate solution of (1.4).

Proof. $Y_{\delta_{\alpha+2\beta}} - Y^* = E_{\delta_{\alpha+2\beta}}$ satisfies

$$E_{\delta_{\alpha+2\beta}} = (\prod_{l=1}^{\alpha+2\beta} ((A - \nu_{j_l} I_s) \otimes ((I_m - \nu_{j_l} B)^{-1} B))^{r_l}) E_0 =$$

$$(C \otimes I_m)[(\prod_{l=1}^{\alpha+2\beta} (J - \nu_{j_l} I_s)^{r_l}) \otimes (\prod_{l=1}^{\alpha+2\beta} ((I_m - \nu_{j_l} B)^{-1} B)^{r_l})](C^{-1} \otimes I_m) E_0,$$

where $C^{-1}AC = J$ is the Jordan standard form of A. Note $(I_m - \nu_{j_l} B)$ are nonsingular matrices. Because the matrix in the square brackets [ ] is zero, the desired conclusion follows. □

Theorem 4.4. *let A have α $(\alpha \geq 0)$ different real eigenvalues $\nu_{j_1}, \nu_{j_2}, \cdots, \nu_{j_\alpha}$ (may contain zero and negative number), β $(\beta \geq 0)$ different conjugate complex eigenvalue pairs $\nu_{j_{\alpha+l}}, \overline{\nu}_{j_{\alpha+l}}, l = 1(1)\beta$; the maximum of orders of all Jordan blocks of A corresponding to ν_{j_l} be r_l; $\eta_i = \sum_{j=1}^{i} r_j, i = 1(1)\alpha + \beta$. Then when $\nu_k \lambda_j \neq 1$ $(k = 1(1)s, j = 1(1)m)$, for any $Y_0 \in R^{ms}, Y_{\eta_{\alpha+\beta}}$ obtained by the MPID method*

$$W(\nu_{j_l})(Y_i - Y_{i-1}) = D(Y_{i-1}), \quad i = \eta_{l-1} + 1(1)\eta_l, l = 1(1)\alpha, \quad \eta_0 = 0, \tag{4.3}$$

$$\begin{cases} W_1(\nu_{j_l})(Y_i - Y_{i-1}) = (I_{ms} + (A - (\nu_{j_l} + \overline{\nu}_{j_l}) I_s) \otimes B) D(Y_{i-1}) \\ i = \eta_{l-1} + 1(1)\eta_l, \quad l = (\alpha + 1)(1)(\alpha + \beta) \end{cases} \tag{4.4}$$

is the accurate solution of (1.4).

Proof. We can prove this theorem from lemma 4.1, 4.2, 4.3. □

When we don't know some of $\eta_i, i = 1(1)(\alpha+\beta)$, noticing $\max_{1\le i\le\alpha+\beta} r_i \le r := s-(\alpha+2\beta)+1$, we can rewrite theorem 4.4 as

Theorem 4.5. *let A have α $(\alpha \ge 0)$ different real eigenvalues $\nu_{j_1}, \nu_{j_2}, \cdots, \nu_{j_\alpha}$, β $(\beta \ge 0)$ different conjugate complex eigenvalue pairs, Then when $\nu_k\lambda_j \ne 1$ $(k = 1(1)s, j = 1(1)m)$, for any $Y_0 \in R^{ms}$, $Y_{r,\alpha+\beta}$ obtained by the MPID method*

$$\begin{cases} W(\nu_{j_l})(Y_{k,l} - Y_{k,l-1}) = D(Y_{k,l-1}), \quad l = 1(1)\alpha \\ W_1(\nu_{j_l})(Y_{k,l} - Y_{k,l-1}) = (I_{ms} + (A - (\nu_{j_l} + \overline{\nu}_{j_l})I_s) \otimes B)D(Y_{k,l-1}) \\ l = \alpha + 1(1)\alpha + \beta, \\ k = 1(1)r, \quad Y_{1,0} = Y_0, \quad Y_{q+1} = Y_{q,\alpha+\beta} \end{cases} \tag{4.5}$$

is the accurate solution of (1.3). □

The estimate for r can't be improved further, e.g., when $r_i \equiv 1, r = 1, s-(\alpha+2\beta)+1 = 1$.

Since A in theorem 4.4,4.5 is arbitrary under (1.8), the MPID method have wide application field, provided that eigenvalues of A are known or easily be computed.

Theorem 4.6. *Suppose that all eigenvalues of A are nonnegative real numbers and*

$$u_i \ge u_p/2, \quad i = 1(1)p-1, \quad u_1 < u_2 < \cdots < u_p^{[16,17]}; \tag{4.6}$$

and $E(\mu_k)$ $(k = 1(1)p)$ hold. Then the MPID method (4.3) *make any error component decrease at each iteration step.*

Proof. The use of (4.3),(2.6) and the characters of $G_k(\mu)(k = 1(1)p)$ can prove the theorem. □

When $\alpha = 1$ and $\beta = 0$, the computation formulae in theorem 4.4,4.5 and 3.1 identify in form. But, $\nu_{j_1} = u_{j_1}$ in theorem 4.4 and 4.5 is arbitrary, but $w_{opt} = u_{j_1} \ge 0$ in theorem 3.1.

Suppose the solution of (1.4) are computed on the serial computer. If $\alpha = 2$ and $\beta = 0^{[16]}$, then the LU-decomposition is made only for two matrix $(h\nu_{j_l})^{-1}I_m - B$ $(l = 1, 2)$. Otherwise the LHL^{-1} decomposition is made for $(h\nu_{j_1})^{-1}I_m - B$, and

$$\begin{cases} W(\nu_{j_l}) = I_s \otimes (h\nu_{j_l}L(((h\nu_{j_l})^{-1} - (h\nu_{j_1})^{-1})I_m + H)L^{-1}) \\ W_1(\nu_{j_l}) = I_s \otimes [h^2|\nu_{j_l}|^2 L(((h\nu_{j_l})^{-1} - (h\nu_{j_1})^{-1})I_m + H)(((h\overline{\nu}_{j_l})^{-1} - (h\nu_{j_1})^{-1})I_m + H)L] \\ l = (\alpha+1)(1)(\alpha+\beta), \end{cases} \tag{4.7}$$

where H is the upper Hessenberg matrix, L is the upper triangular matrix. Therefore multiplication and divison operation count needed by the MPID method is at most

$$Q = \begin{cases} Q_1 + Q_2 + Q_3, & \text{if } (\alpha \ge 3 \text{ and } \beta = 0) \text{ or } \beta \ne 0 \text{ by(4.3)(4.4)}, \\ Q_1 + Q_4 + Q_5, & \text{if } (\alpha \ge 3 \text{ and } \beta = 0) \text{ or } \beta \ne 0 \text{ by(4.5)}, \\ \frac{2}{3}m^3 + O(m^2) + \eta_2(2sm^2 + s^2m), & \text{if } \alpha = 2 \text{ and } \beta = 0, \end{cases} \tag{4.8}$$

where $Q_2 = \eta_{\alpha+\beta}(O(m^2))$, $Q_3 = (2\eta_{\alpha+\beta}-\eta_\alpha)(sm^2+s^2m)$, $Q_4 = (\alpha+\beta)(O(m^2))$, $Q_5 = (2\beta+\alpha)(sm^2+s^2m)$; $Q_1 = \frac{5}{6}m^3 + O(m^2)$ is multiplication and divison operations required by the LHL^{-1} -decomposition for an $m*m$-matrix, sm^2+s^2m is that required by the computaion of $(I_{ms} + C \otimes D)F(Z)$, $O(m^2)$ in Q_2 and Q_4 is that required by the use of (4.7) to compute Y_i from Y_{i-1} in accordance with (4.3)(4.5) (or (4.5)).

It is easily known that MPID method requires at most about

$$3m^2 + s^2 + 3ms \tag{4.9}$$

storages.

Comparing (4.8)(4.9) with (1.6)(1.7) we conclude that MPID method requires slight more storage but far less operation count for sufficient large m.

From (4.3), (4.4) or (4.5) it can be seen that the MPID method is also a parallel algorithm. But the B-SA and H-SA are not. On parallel computers with s available processors, if we compute Y^* according to (4.3)(4.4), each processor only need to carry out at most

$$\overline{Q}_1(\alpha,\beta) = \begin{cases} \frac{1}{3}m^3 + \lceil\frac{m}{s}\rceil m^2 + O(m^2) + (3\eta_{\alpha+\beta} - \eta_\alpha)m^2 + (2\eta_{\alpha+\beta} - \eta_\alpha)sm, & \beta \neq 0, \\ \frac{1}{3}m^3 + O(m^2) + \eta_\alpha(2m^2 + sm), & \beta = 0, \end{cases}$$

multiplication operations. Similarly, if we compute Y^* according to (4.5), each processor only need to carry out at most

$$\overline{Q}_2(\alpha,\beta) = \begin{cases} \frac{1}{3}m^3 + \lceil\frac{m}{s}\rceil m^2 + O(m^2) + r((3\beta - \alpha)m^2 + (2\beta - \alpha)sm), & \beta \neq 0, \\ \frac{1}{3}m^3 + O(m^2) + r\alpha(m^2 + sm), & \beta = 0, \quad \alpha \leq s \end{cases}$$

multiplication operations. In $\overline{Q}_1(\alpha,\beta)$ and $\overline{Q}_2(\alpha,\beta)$, $\lceil\cdot\rceil$ denotes the upper integral number.

5. Numerical Experiment

Using the semi-discrete method for the difinite problem of the linear evolution equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad u(0,t) = u(1,t) = 0, \quad u(x,0) = sin(\pi x), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq T, \tag{5.1}$$

we obtain IVP of stiff ODEs

$$\begin{cases} y'(t) = By, \quad y = (y_1, y_2, \cdots, y_m)^T, \quad (m+1)h_1 = 1, \\ B = \frac{1}{h_1^2}\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2 \end{bmatrix}, \quad y(0) = \begin{pmatrix} sin(\pi h_1) \\ sin(2\pi h_1) \\ \vdots \\ sin(m\pi h_1) \end{pmatrix}. \end{cases} \tag{5.2}$$

The accurate solution of (5.2) is $y(t) = exp(-\frac{4t}{h_1^2} sin^2(\frac{\pi h_1}{2}))y(0)$. We set m=10 and use respectively following methods in order that the numericl integration for (5.2) advance one step (or one block).

1) the 3-point order 3 block θ method[12] ($\theta = 0.5$), 2) the 2-stage order 4 R-K method[7], 3) the 4-point order 5 method in [17], 4) the 3-stage order 6 R-K method[7], and use respectively the OOPI method in the paper for solving linear equation systems produced by them. Results obtained are listed in table 1. We also use respectively the MPID method in the paper for solving the equation sysytem produced by 2), 3) and 4). Results obtained are listed in table 2.

In table 1, 2, $\triangle := -\ln \|Y^* - Y_i\|_\infty / \ln 10$, where Y_i is the numerical solution obtained; $\epsilon = 10^{-9}$ is the tolerance of iterative error; in (a,b), a is the number of different real eigenvalues of A, b is the number of the different complex conjugatel eigenvalues pair of A; in a/b, a is the datum corresponding to $h = 0.01$, b is the datum corresponding to $h = 0.001$; T_0 is the distance that numerical integration advance one step or one block.

Table 1 Results obtained by the OOPI method(3.3)

method	order	(a,b)	iterative number	$\triangle$	T_0
1	3	(1,0)	3/3	3.8/7.7	0.03/0.003
2	4	(0,1)	6/4	7.9/12.9	0.01/0.001
3	5	(4,0)	6/4	5.4/11.4	0.04/0.004
4	6	(1,1)	9/5	12.0/15.0	0.01/0.001

Table 2 Results obtained by the MPID(4.3)(4.4)

method	order	(a,b)	iterative number	$\triangle$	T_0
2	4	(0,1)	1/1	7.9/12.9	0.01/0.001
3	5	(4,0)	4/4	5.4/11.6	0.04/0.004
4	6	(1,1)	2/2	12.0/15.0	0.01/0.001

The date in table 1 and 2 are consistent completly with the theory. We compute also some others examples in which matrix B is full. The conclusion is the same, too.

References

[1] R.H.Bartels and G.W. Stewart, Algorithm 432, Solution of the matrix equation $AX + XB = C$, Comm ACM, 15(1972), 820-826

[2] T.A. Bickart, An efficient solution process for implicit Runge-Kutta metheds, SIAM J Numer Anal, 14(1977)6, 1022-1027

[3] J.C. Butcher, On the implementation of implicit Runge-Kutta metheds, BIT, 16(1976)2, 237-240

[4] P.Chartier, L-stable parallel one-block methods for ordinary differential equation, SIAM J. Numer. Anal., 31(1994)2, 552-571

[5] G.J. Cooper and J.C. Butcher, An iteration scheme for implicit Runge-Kutta methods, IMA J. Numer. Anal., 3(1983), 127-140

[6] G.J. Cooper and R. Vignesvaran, Some schemes for the implementation of implicit Runge-Kutta metheds, J Comp Appl Math, 45(1993), 213-225

[7] K. Dekker and J.G. Verwer, Stability of Runge-Kutta methods for stiff nonlinear differential equations, North-Holland, Amsterdam · New York · Oxford, 1984, 62-93

[8] W.H. Enright, Improving the efficiency of matrix operations in the numerical solution of stiff ordinary differential equations, ACM TMS 4(1978)2, 127-136

[9] G.H. Golub, S. Nash and C.Van Loan, A Hessenberg-shur method for the problem $AX + XB = C$, IEEE Troms. Automat. Contr, AC-24(1979)6,809-913

[10] G.H. Golub and C.F. Van Loan, Matrix Computations, Johns Hopkins University press, 1983

[11] W.D. Hoskong, D.S. Meek and D.J. Watten, The numerical solution of the matrix equation $XA + AY = F$, BIT, 17(1977), 184-190

[12] Lu Lianhua, The stability of the block θ-methods, IMA Numer. Anal., 13(1993), 101-114

[13] B.P. Sommeijer, W. Couzy and P.J Van der Houwen, A-stable parallel block methods for ordinary and integro-differential equations, Appl. Numer. Math., 9(1992)2, 267-271

[14] H.A. Watts and L.F. Shampine, A-stable block implicit one-step methods. BIT, 12(1972), 252-266

[15] Zhao Shuangsuo and Zhang Guofeng, A class of one-block methods and their implementation scheme, Abstract of the paper in Chinese academic periodical (newsflash on science and technology), 2(1996)5, 109

[16] Zhao Shuangsuo and Zhang Guofeng, A class of one-block implicit methods with non-derogatory coefficient matrix having only two different positive eigenvalues, Math. Numer., Sinica, 19(1997)1, 47-57

[17] Zhao Shuangsuo, Zhang Guofeng and Wang Changyin, The convergence of two Newton-like methods for solving block nonlinear equations and a class of r-point $(r+1)$st-order A-stable one-block methods, Appl. Numer. Math., 25(1997)1,117-133

[18] Zhou Bing, A-stable and L-stable implicit one block methods, JCM, 3(1985)4, 328-341