J. Korean Math. Soc. **31** (1994), No. 4, pp. 633-643

ON THE NUMERICAL COMPUTATION OF THE MATRIX EXPONENTIAL

DONG WON YU

1. Introduction

Let us consider the initial-value problem of dimension m:

$$\frac{d}{d\tau}\mathbf{y}(\tau) = \mathbf{f}(\tau, \mathbf{y}(\tau)), \qquad \mathbf{y}(0) = \mathbf{y}_0, \qquad \tau \geq 0, \tag{1.1}$$

where $\mathbf{f} = (f_1, f_2, \cdots, f_m)$ and $\mathbf{y} = (y_1, y_2, \cdots, y_m)$. We assume that

$$\lim_{\tau\to\infty} \frac{\frac{d}{d\tau}y_i(\tau)}{y_i(\tau)} = \lim_{\tau\to\infty} \frac{f_i(\tau, \mathbf{y}(\tau))}{y_i(\tau)} = \alpha_i, \qquad i = 1, 2, \cdots, m. \tag{1.2}$$

Then $\alpha = \max\{\alpha_1, \alpha_2, \cdots, \alpha_m\}$ is called the **exponentially dominant order** (cf. [10]) of the system (1.1).

Using the function $\mathbf{z}(\tau) = e^{-\alpha\tau}\mathbf{y}(\tau)$, the problem (1.1) with the dominant order α is transformed to the initial-value problem for $\mathbf{z}(\tau)$ (cf. [5]) :

$$\frac{d}{d\tau}\mathbf{z}(\tau) = e^{-\alpha\tau}\mathbf{g}(\tau, e^{\alpha\tau}\mathbf{z}(\tau)), \qquad \mathbf{z}(0) = \mathbf{y}_0, \tag{1.3}$$

where $\mathbf{g}(\tau, \mathbf{y}) = \mathbf{f}(\tau, \mathbf{y}) - \alpha\mathbf{y}$.

Apply the Runge-Kutta methods defined by the Butcher array (1.4) to the transformed problem (1.3).

$$\begin{array}{c|c} \mathbf{c} & D \\ \hline & \mathbf{b}^T \end{array} = \begin{array}{c|ccc} c_1 & d_{11} & \cdots & d_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & d_{s1} & \cdots & d_{ss} \\ \hline & b_1 & \cdots & b_s \end{array} \tag{1.4}$$

Received July 24, 1993.
This paper was supported by NON DIRECTED RESEARCH FUND, Korea Research Foundation, 1992.

And if we define that $\mathbf{y}_n$ and $\mathbf{z}_n$ are approximations of $\mathbf{y}(\tau_n)$ and $\mathbf{z}(\tau_n) = e^{-\alpha\tau_n}\mathbf{y}(\tau_n)$ respectively, then the **generalized Runge-Kutta (GRK) method** is derived as follows (cf. [10]):

$$\mathbf{y}_{n+1} = e^{\alpha h}\mathbf{y}_n + h\sum_{i=1}^{s} b_i e^{(1-c_i)\alpha h}\mathbf{k}_i, \tag{1.5}$$

where

$$\mathbf{k}_i = \mathbf{g}(\tau_i + c_i h,\ \mathbf{y}_{n,i}),$$

$$\mathbf{y}_{n,i} = e^{c_i\alpha h}\mathbf{y}_n + h\sum_{j=1}^{s} d_{ij}e^{(c_i-c_j)\alpha h}\mathbf{k}_j$$

Here h is the constant step size and $\tau_n = nh$.

All methods for the initial-value problem (1.1) can be generalized as above. For instance the explicit and the implicit Euler methods are generalized as follows:

$$\mathbf{y}_{n+1} = e^{\alpha h}\left[\mathbf{y}_n + h\mathbf{g}(\tau_n, \mathbf{y}_n)\right], \tag{1.6}$$

and

$$\mathbf{y}_{n+1} = e^{\alpha h}\mathbf{y}_n + h\mathbf{g}(\tau_{n+1}, \mathbf{y}_{n+1}).$$

In Section 2, we derive a new algorithm (2.3) for the computation of the matrix exponential. In Section 3, we prove that the local truncation error bound of the generalized explicit Euler method (1.6) is less than the local truncation error bound of the explicit Euler method. In the last section, it is shown through numerical examples that, for the problem (1.1) with $\alpha \neq 0$, the algorithm (2.3) is more effective than the original algorithm (2.2) and Ward's algorithm (2.4) (cf. [7]).

2. Algorithms for the matrix exponential

Let $\mathbb{R}^{m\times m}$ denote the set of real $m \times m$ matrices and $\mathbb{R}^m$ be the m dimensional column vector space. The exponential of a matrix $A = [a_{ij}] \in \mathbb{R}^{m\times m}$ is defined by

$$e^{\tau A} = \sum_{k=0}^{\infty}\frac{(\tau A)^k}{k!}, \qquad \text{where } \tau \geq 0.$$

The importance of this matrix function in applied mathematics is derived from the fact that the matrix exponential is the unique solution $X(\tau) = [\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^m] = [x_{ij}] \in \mathbb{R}^{m\times m}$ to the initial-value problem:

$$\frac{d}{d\tau}X(\tau) = AX(\tau), \qquad X(0) = I, \qquad \tau \geq 0. \tag{2.1}$$

where $I = [\mathbf{i}^1, \mathbf{i}^2, \cdots, \mathbf{i}^m] = [\delta_{ij}]$ is the identity matrix.

In this paper we will assume that the matrix A has no complex eigenvalues. And suppose that A has a full set of m linearly independent eigenvectors. Letting $\mathbf{p}^1, \mathbf{p}^2, \cdots, \mathbf{p}^m (\in \mathbb{R}^{m\times 1})$ denote these eigenvectors and $\lambda_1, \lambda_2, \cdots, \lambda_m (\delta = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m = \alpha)$ the corresponding eigenvalues, form the matrix P such that $P = [\mathbf{p}^1\ \mathbf{p}^2\ \cdots\ \mathbf{p}^m]$. Then the solution of (2.1) is given by

$$X(\tau) = C_1 e^{\lambda_1 \tau} + C_2 e^{\lambda_2 \tau} + \cdots + C_m e^{\lambda_m \tau},$$

where $C_i = \mathbf{p}^i(\mathbf{q}^i)^T \in \mathbb{R}^{m\times m}$ and $\mathbf{q}^i$ is the ith row vector of P^{-1}. Thus $X(\tau)$ tends to $C_m e^{\alpha\tau}$ as τ tends to ∞. Hence we have the following proposition.

PROPOSITION 2.1. *The exponentially dominant order of the problem (2.1) is the largest eigenvalue α of the matrix A.*

Application of the Runge-Kutta method (1.4) to the equation (2.1) yields the stability function

$$\mathcal{R}(hA) = \frac{\text{Det}\ [I - (D - \mathbf{u}\mathbf{b}^T) \otimes hA]}{\text{Det}\ [I - D \otimes hA]}.$$

In this case, $\otimes$ denotes Kronecker product, Det represents the cell determinant (cf. [9]), $\mathbf{b} = [b_1, b_2, \cdots, b_s]^T \in \mathbb{R}^s$ and $D = [d_{ij}] \in \mathbb{R}^{s\times s}$ are the Butcher arraies (1.4), and $\mathbf{u} = [1, 1, \cdots, 1]^T \in \mathbb{R}^s$. Then the matrix exponential $e^{\tau_n A} = X(\tau_n)$ can be approximately computed by this rational function as follows:

$$e^{\tau_n A} = X(\tau_n) \approx X_n = \mathcal{R}(hA)X_{n-1} = (\mathcal{R}(hA))^n. \tag{2.2}$$

REMARK 2.1. In order to improve the accuracy for matrices with large norm, Lawson [5] proposed a new idea using diagonal Padé approximation and the identity

$$e^A = (e^{2^{-n}A})^{2^n}.$$

It is said to be the scaling and squaring algorithm.

While applying the GRK method (1.5) to the problem (2.1) with the exponentially dominant order α, its stability function $\mathcal{K}(hA)$ is derived as follows (cf. [9]):

$$\mathcal{K}(hA) = e^{\alpha h}\mathcal{R}(hB), \qquad B = A - \alpha I.$$

So we arrive at an algorithm for the numerical computation of the matrix exponential:

$$e^{\tau_n A} = X(\tau_n) \approx X_n = \mathcal{K}(hB)X_{n-1} = (\mathcal{K}(hA))^n = e^{\alpha\tau_n}(\mathcal{R}(hB))^n. \tag{2.3}$$

REMARK 2.2. Using the trace $tr(A)$ of the matrix 4, Ward [7] splits the matrix A as follows:

$$A = B + \beta I, \qquad \text{where } \beta = tr(A)/n,$$

and utilizes the **splitting, scaling, and squaring algorithm:**

$$e^{\tau_n A} \approx e^{\beta\tau_n}(\mathcal{R}(hB))^n. \tag{2.4}$$

REMARK 2.3. The (p, q)-Padé approximations (cf. [6]),

$$e^A \approx R_{pq}(A) = [D_{pq}(A)]^{-1}[N_{pq}(A)],$$

can be transformed to the generalized (p, q)-Padé approximation as follows:

$$e^{\tau_n A} \approx e^{\alpha\tau_n}(R_{pq}(hB))^n,$$

where $B = A - \alpha I$.

3. Comparison of error bounds

In order to make a comparative study of the algorithms (2.2) and (2.3), we apply the explicit Euler method and the generalized explicit Euler method (1.6) to the problem (2.1) with the exponential order $\alpha(\alpha \neq 0)$. Because of convenience of unitary invariance, we work exclusively with the 2-norm in the vector spaces $\mathbb{R}^m$ and $\mathbb{R}^{m\times m}$, and denote the constants $\eta, \bar{\eta}, k_n, \bar{k}_n, k$ and $\bar{k}$ as follows:

$$\begin{aligned} \eta &= \|A\|, & \bar{\eta} &= \|B\|, \\ k_n &= |X''(\tau_n + \zeta)|, & \bar{k}_n &= |\alpha^2 X(\tau_n + \xi) - 2\alpha X'(\tau_n + \xi) \\ & (0 < \zeta < h) & & \quad + X''(\tau_n + \xi)|, \\ & & & (0 < \xi < h) \\ k &= \max\{k_1, k_2, \cdots, k_n\}, & \bar{k} &= \max\{\bar{k}_1, \bar{k}_2, \cdots, \bar{k}_n\} \end{aligned}$$

From the equation (2.1), we have $\alpha^2 X(\tau_n+\xi) - 2\alpha X'(\tau_n+\xi)+X''(\tau_n+\xi) = (\alpha I - A)^2 X(\tau_n+\xi) = B^2 X(\tau_n+\xi)$ and $X''(\tau_n+\zeta) = A^2 X(\tau_n+\zeta)$. Hence we have

$$k_n = |A^2 X(\tau_n + \zeta)|, \qquad \bar{k}_n = |B^2 X(\tau_n + \xi)|, \qquad 0 < \xi < h. \quad (3.1)$$

Since the global truncation error of the generalized explicit Euler method (1.6) is given by

$$|\mathbf{y}(\tau_{n+1}) - \mathbf{y}_{n+1}| \leq e^{\alpha h}(1 + h\bar{\eta})|\mathbf{y}(\tau_n) - \mathbf{y}_n| + \frac{e^{\alpha(h-\xi)}\bar{k}_n h^2}{2},$$

the local truncation error bounds ℓ_n and $\bar{\ell}_n$ of the explicit Euler method and the generalized explicit Euler method (1.6) are given by

$$\ell_n = \frac{k_n h^2}{2}, \qquad \bar{\ell}_n = \frac{e^{\alpha(h-\xi)}\bar{k}_n h^2}{2}. \quad (3.2)$$

The global truncation error bound $\mathcal{G}_n$ and the total error bound $\mathcal{J}_n$ of the explicit Euler method are transformed as follows:

$$\mathcal{G}_n = \frac{(e^{\eta\tau_n} - 1)kh}{2\eta}, \qquad \mathcal{J}_n = e^{\eta\tau_n}\left\{\gamma_0 + \frac{1}{\eta}\left(\frac{kh}{2} + \frac{\gamma}{h}\right)\right\},$$

and the global truncation error bound $\mathcal{G}_n$ and the total error bound $\bar{\mathcal{J}}_n$ of the generalized explicit Euler method (1.6) may be classified as follows:

α	$\mathcal{G}_n$	$\bar{\mathcal{J}}_n$
$\alpha \geq 0$	$\frac{\left\{e^{(\alpha+\bar{\eta})\tau_n}-1\right\}\bar{k}h}{2\bar{\eta}}$	$e^{(\alpha+\bar{\eta})\tau_n}\left\{\gamma_0+\frac{1}{\bar{\eta}}\left(\frac{\bar{k}h}{2}+\frac{\gamma}{h}\right)\right\}$
$0>\alpha>-\bar{\eta}$	$\frac{\tau_n e^{(\alpha+\bar{\eta})\tau_n}\bar{k}h}{2}$	$e^{(\alpha+\bar{\eta})\tau_n}\left\{\gamma_0+\tau_n\left(\frac{\bar{k}h}{2}+\frac{\gamma}{h}\right)\right\}$

where

1. γ_0 is the initial round-off error bound committed evaluating $X(0)$,
2. γ is the maximum possible round-off error bound of the computer.

PROPOSITION 3.1. *Let the matrix A be normal and have no complex eigenvalues, and let $B = A - \alpha I$. If $0 < \alpha/2 \leq \delta \leq \lambda_i \leq \alpha$ or $\delta \leq \lambda_i \leq \alpha < 0$, then*

(i) $|B\mathbf{x}| \leq |A\mathbf{x}|$,
(ii) $\bar{\eta} \leq \eta$.

Proof. (i) Let $\mathbf{u}^1, \mathbf{u}^2, \cdots, \mathbf{u}^m$ be a set of orthonormal eigenvectors of A with associated eigenvalues $\delta = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m = \alpha$. If $\mathbf{x} = \sum x_i \mathbf{u}_i$, then $A\mathbf{x} = \sum x_i \lambda_i \mathbf{u}_i$, and

$$
\begin{aligned}
&|B\mathbf{x}|^2 \\
&= \langle (A-\alpha I)\mathbf{x},\ (A-\alpha I)\mathbf{x}\rangle = \langle A\mathbf{x}, A\mathbf{x}\rangle - 2\alpha\langle A\mathbf{x}, \mathbf{x}\rangle + \alpha^2\langle \mathbf{x}, \mathbf{x}\rangle \\
&= \sum x_i^2\lambda_i^2 + \alpha\left\{\alpha\sum x_i^2 - 2\sum x_i^2\lambda_i\right\} = |A\mathbf{x}|^2 + \alpha\sum x_i^2(\alpha - 2\lambda_i).
\end{aligned}
$$

If $0 < \alpha/2 \leq \delta \leq \lambda_i \leq \alpha$ or $\delta \leq \lambda_i \leq \alpha < 0$, then $\alpha\sum x_i^2(\alpha - 2\lambda_i) \leq 0$. So we have

$$|B\mathbf{x}| \leq |A\mathbf{x}|.$$

(ii) $\bar{\eta} = \|B\| = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{|B\mathbf{x}|}{|\mathbf{x}|} \leq \sup_{\mathbf{x}\neq\mathbf{0}} \frac{|A\mathbf{x}|}{|\mathbf{x}|} = \|A\| = \eta.$ □

PROPOSITION 3.2. *Let the matrix A be normal and have no complex eigenvalues. Then, for any $\epsilon > 0$, we have*

$$(i) \quad k_n \leq \epsilon k_n,$$
$$(ii) \quad \ell_n \leq \epsilon \ell_n$$

when n is sufficiently large.

Proof. Let $P = [\mathbf{u}^1\ \mathbf{u}^2\ \cdots\ \mathbf{u}^m] \in \mathbb{R}^{m\times m}$, then $D = P^TAP$ and $A = PDP^T$. Since $X(\tau) = e^{\tau A} = Pe^{\tau D}P^T$, we have

$$B^2X(\tau) = (A^2 - 2\alpha A + \alpha^2)e^{\tau A} = P\left\{(D^2 - 2\alpha D + \alpha^2)e^{\tau D}\right\}P^T$$
$$= P\begin{bmatrix} (\lambda_1-\alpha)^2e^{\tau\lambda_1} & & 0 \\ & \ddots & \\ 0 & & (\lambda_m-\alpha)^2e^{\tau\lambda_m} \end{bmatrix}P^T$$

Then, because P is an orthogonal matrix

$$|B^2X(\tau)|^2 = \sum_{i=1}^{m}(\lambda_i-\alpha)^4e^{2\tau\lambda_i}.$$

Similary,

$$|A^2X(\tau)|^2 = \sum_{i=1}^{m}\lambda_i^4e^{2\tau\lambda_i}.$$

(i) From (3.1) we have

$$0 \leq \frac{\bar{k}_n^2}{k_n^2} = \frac{|B^2X(\tau_n+\xi)|^2}{|A^2X(\tau_n+\zeta)|^2} = \frac{\sum_{i=1}^m(\lambda_i-\alpha)^4e^{2(\tau_n+\xi)\lambda_i}}{\sum_{i=1}^m\lambda_i^4e^{2(\tau_n+\zeta)\lambda_i}}$$
$$\leq \rho_1\frac{\sum_{i=1}^m(\lambda_i-\alpha)^4e^{2\tau_n\lambda_i}}{\sum_{i=1}^m\lambda_i^4e^{2\tau_n\lambda_i}},$$

where $\rho_1 = \rho_1(\lambda(A), \xi, \zeta)$. But

$$\lim_{n\to\infty}\frac{\sum_{i=1}^m(\lambda_i-\alpha)^4e^{2\tau_n\lambda_i}}{\sum_{i=1}^m\lambda_i^4e^{2\tau_n\lambda_i}} = \lim_{n\to\infty}\frac{\sum_{i=1}^m(\lambda_i-\alpha)^4e^{2\tau_n(\lambda_i-\alpha)}}{\sum_{i=1}^m\lambda_i^4e^{2\tau_n(\lambda_i-\alpha)}} = 0.$$

So, we have $\lim_{n\to\infty} \frac{k_n}{k_n} = 0$.

(ii) Similary, from (3.2), we have

$$\begin{aligned} 0 \le \frac{\ell_n^2}{\ell_n^2} &= \frac{e^{2\alpha(h-\xi)} k_n^2 h^4}{k_n^2 h^4} \\ &\le \frac{e^{2\alpha(h-\xi)} \sum_{i=1}^m (\lambda_i - \alpha)^4 e^{2(\tau_n+\xi)\lambda_i}}{\sum_{i=1}^m \lambda_i^4 e^{2(\tau_n+\zeta)\lambda_i}} \\ &\le \rho_2 \frac{\sum_{i=1}^m (\lambda_i - \alpha)^4 e^{2\tau_n\lambda_i}}{\sum_{i=1}^m \lambda_i^4 e^{2\tau_n\lambda_i}}, \end{aligned}$$

where $\rho_2 = \rho_2(\lambda(A), h, \xi, \zeta)$. Thus we have $\lim_{n\to\infty} \frac{\bar{\ell}_n}{\ell_n} = 0$. □

REMARK 3.1. Fair and Luke [3] have shown that the error in approximating the matrix exponential by the (q,q)-Padé approximation increases as the norm of the matrix increases. Ward [7] attempts to minimize the matrix norm by the translation, $e^A = e^\beta e^B$, where $\beta = tr(A)/m$ and $B = A - \beta I$. In general, $\|A - \beta I\|$ is smaller than $\|A - \alpha I\|$. But $|e^{\tau A}|$ does not tend to $e^{\tau\beta}$ but $e^{\tau\alpha}$ as τ increases. Hence, in this case, we can not verify that $\bar{\ell}_n \le \ell_n$. In the last section we exemplify that the relative total error bounds of the original and Ward's algorithms increase as τ increases.

REMARK 3.2. It is also shown through numerical examples that Proposition 3.2 holds for the arbitrary matrix A which is not normal.

4. Numerical tests

In Section 2, the algorithms (2.2) for the computation of the matrix exponential $e^{\tau A}$ is transformed to the algorithm (2.3) by the exponentially dominant order α. Using the mean of eigenvalues of A, Ward [7] organizes the algorithm (2.4). The various Runge-Kutta methods with their transformed algorithms (2.3) and (2.4) are investigated in this section through the following example matrices. We implement our algorithms on a personal computer in double precision.

Example 1.([4])

$$A = \begin{bmatrix} -1+\delta & 4 \\ 0 & -1-\delta \end{bmatrix},$$

where $\delta = 10^{-6}$

Example 2.

$$A = \begin{bmatrix} -2 & 4 \\ 0 & -2-\delta \end{bmatrix},$$

Example 3. ([4],[7],[8])

$$A = \begin{bmatrix} 4 & 2 & 0 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{bmatrix},$$

Example 4.

$$A = \begin{bmatrix} -3 & 1 & 0 \\ 0 & -3 & 1 \\ 4 & -8 & 2 \end{bmatrix},$$

Example 5. ([7],[8])

$$A = \begin{bmatrix} -131 & 19 & 18 \\ -390 & 56 & 54 \\ -387 & 57 & 52 \end{bmatrix},$$

Example 6.

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & -1 \end{bmatrix},$$

Example 7.

$$A = \begin{bmatrix} -2 & 2 & 2 \\ 2 & -5 & 1 \\ 2 & 1 & -5 \end{bmatrix},$$

Example 8.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

The example matrices are specified as the following table:

Ex	$\lambda(A)$	$\|A\|$	$\alpha(A)$	$\|B\|$	$\mathrm{tr}(A)/m$	$\|C\|$	$\mu(A)$	Note
1	$-1+\delta, -1-\delta$	4.2361	$-1+\delta$	4	-1	4	1	nearly defective
2	$-2, -2-\delta$	4.8284	-2	4	$-2-\delta/2$	4	-5×10^{-7}	nearly confluent
3	3, 3, 6	6.0779	6	3.6886	4	2.3073	6.0566	defective
4	-1, -1, -2	9.8398	-1	9.7846	-1.3333	9.8246	4.1471	defective
5	-1, -2, -20	575.95	-1	575.91	-7.6667	575.77	277.21	widely spread eigenvalues
6	1, -1, -1	1.9319	1	2.7651	-1/3	1.8095	1.0663	defective, different sign
7	0, -6, -6	6	0	6	-4	4	0	nondefective, symmetric
8	1, -1, i, -i	1	1	2	0	1	1	complex eigenvalues

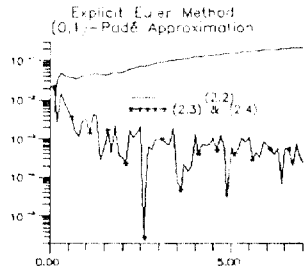
where $B = A - \alpha I$, $C = A - (tr(A)/m)I$, and $\mu(A) = \max\{\lambda_i : \lambda_i \in \lambda[(A + A^T)/2]\}$.

The relative total error of the approximation $X_n = V = [v_{ij}]$ to $X(\tau_n) = e^{\tau_n A} = W = [w_{ij}]$, computed by our algorithm (2.3),

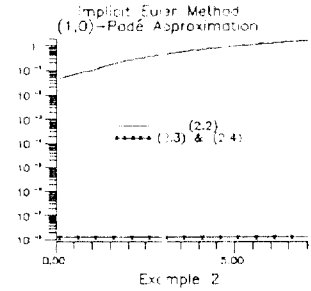
$$\sum_{i=1}^{m}\sum_{j=1}^{m}\left|\frac{w_{ij} - v_{ij}}{w_{ij}}\right|$$

is compared with those obtained by the original algorithm (2.2) and Ward's algorithm (2.4) as follows:

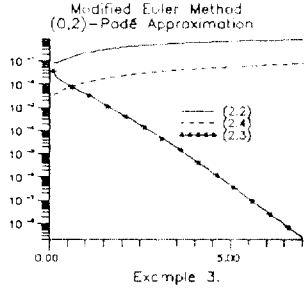
Comparison of the Behaviours of Relative Total Errors
of the Algorithms (2.2), (2.3) and (2.4) for the Matrix Exponential



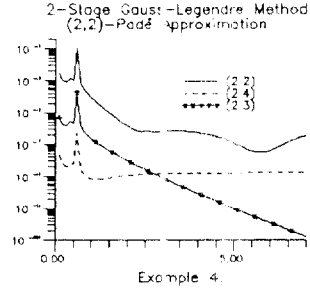
Example 1.



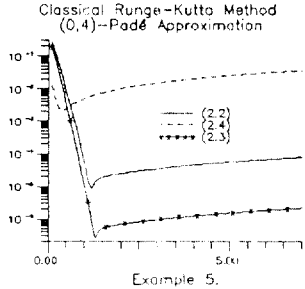
Example 2.



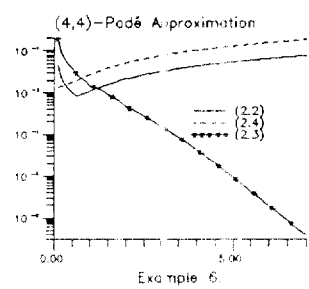
Example 3.



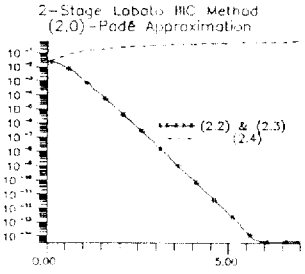
Example 4.



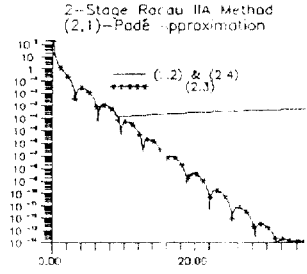
Example 5.



Example 6.



Example 7.



Example 8.

References

1. G. G. Dahlquist, *A special stability problem for linear multistep methods*, BIT **3** (1963), 27-43.
2. K. Dekker and J. D. Verwer, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, North-Holland, Amsterdam, 1984.
3. W. Fair and Y. L. Luke, *Padé approximations to the operator exponential*, Numer. Math. **14** (1970), 379-382.
4. B. Kagstrom, *Bounds and perturbation bounds for the matrix exponential*, BIT **17** (1977), 39-57.
5. J. D. Lawson, *Generalized Runge-Kutta processes for stable systems with large Lipschitz constants*, SIAM J. Num. Anal **4** (1967), 372-380.
6. C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Review **20** (1978), 801-836.
7. R. C. Ward, *Numerical computation of the matrix exponential with accuracy estimate*, SIAM J. Numer. Anal. **14** (1977), 600-610.
8. D. Westreich, *A practical method for computing the exponential of a matrix and its integral*, Communications in applied numerical methods **6** (1990), 375-380.
9. D. W. Yu, *Stability functions of partly-linearized Runge-Kutta method*, Bull. Korean Math. Soc. **25** (1988), 67-76.
10. D. W. Yu, *A study on numerical methods for ordinary differential equations*, Bull. Korean Math. Soc. **28** (1991), 207-217.

Department of Mathematics
Chung-Ang University
Seoul 156-756, Korea