T124
H8

To the Graduate Council:

I am submitting herewith a thesis written by Alan LaMont Fisher entitled "Two Alternative Methods for Computing the Matrix Cosine." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Steven M. Serbin

Steven M. Serbin, Major Professor

We have read this thesis
and recommend its acceptance:

Alan Karr

Carl G. Wagner

Accepted for the Council:

Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Alan LaMont Fisher

Date July 16, 1986

TWO ALTERNATIVE METHODS FOR COMPUTING THE MATRIX COSINE

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Alan LaMont Fisher

August 1986

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Steven M. Serbin, for his suggestions and accessibility. His guidance helped to make this work a very worthwhile experience. I would also like to thank Dr. C. Wagner and Dr. O.A. Karakashian for their helpful comments and suggestions. I would also like to thank everyone in the Department of Mathematics who made my stay here at The University of Tennessee, Knoxville, an enjoyable one.

Special thanks are due to Cindi Blair for the excellent and efficient typing job done.

ABSTRACT

We discuss and analyze two methods which may be used to approximate the cosine of a matrix. The two methods are similar in that they both form a rational approximation $R(kA)$ to the matrix cosine, $\cos(kA)$, and generate approximate schemes which are based on the exact representation $y(t + k) + y(t - k) = 2 \cos(kA)\, y(t)$. The methods also allow for our generated approximation to be corrected. The methods differ in that one forms the rational approximation in a partial-fraction form with only linear factors in the denominator, for computational efficiency, especially in a parallel environment. This method allows for the approximation of the local error at any time-step via simple calculations.

The double angle method is presented so that comparisons may be made between it and the time-stepping procedure.

Numerical experiments are performed in order to confirm that our theoretical predictions of convergence rates are valid.

TABLE OF CONTENTS

CHAPTER		PAGE
INTRODUCTION		1
I	THE TIME-STEPPING PROCEDURE WITH A POST-PROCESSOR	4
II	EXPRESSING THE RATIONAL OPERATOR IN A PARTIAL-FRACTION FORM	14
III	COMPUTING THE MATRIX COSINE EXACTLY AND USING THE DOUBLE ANGLE METHOD	23
IV	NUMERICAL RESULTS	25
REFERENCES		49
VITA		51

LIST OF TABLES

TABLE		PAGE
IV-1	Measuring $\|\hat{y}(1) - \tilde{y}(1)\|_E$ for the Regular Problem . .	.28
IV-2	Measuring $\|y(1) - \hat{y}(1)\|_E$ for the Regular Problem . .	.30
IV-3	Measuring $\|y(1) - \tilde{y}(1)\|_E$ for the Regular Problem . .	.31
IV-4	Measuring $\|y(1) - \tilde{y}(1)\|_E$ for the Large Problem . . .	.33
IV-5	Measuring $\|y(1) - \tilde{y}(1)\|$ for the Large Problem, But Ignoring Approximation of the Large Entries. . .	.35
IV-6	Comparing Double Angle Method to Time-Stepping Procedure for the Regular Problem.	.37
IV-7	Measuring $\|y(1) - \tilde{y}(1)\|$ When $R(kA_1)$ is Implemented in Partial-Fraction Form	.41
IV-8	Measuring $\|y(1) - \hat{y}(1)\|$ for the Regular Problem. . .	.42
IV-9	Error Resulting From Letting $\beta_2 = F(\beta_1) + .01$	.45
IV-10	Error Resulting From Letting $\beta_2 = F(\beta_1) + .05$	.45
IV-11	Measure of $\|e - \hat{e}\|$ for the Sequence Generated W/R to the 4^{th} Column of A_1.	.46
IV-12	Measure of $\|e - \hat{e}\|$ for the Sequence Generated W/R to the 4^{th} Column of A_2.	.46
IV-13	Measure of $\|e - \hat{e}\|$ for the Sequence Generated W/R to the First Column of A_1 and A_2	.48

INTRODUCTION

For many years, much attention has been focused on the problem of finding a highly accurate, yet cost efficient, method for solving the matrix differential equation

$$Y''(t) + A^2Y(t) = G(t) \tag{1}$$

on the interval $[a, b]$, subject to initial conditions

$$Y(a) = y_0 \quad \text{and} \quad Y'(a) = x_0 ,$$

where Y and G are $(n \times m)$ matrices and A is an $(n \times n)$ matrix. The solution to this matrix differential equation, as is the case of the similar scalar differential equation, consists of terms involving the cosine and sine functions.

Our concern here is in solving a special case of equation (1), mainly computing the solution to the homogeneous second-order matrix differential equation

$$Y''(t) + A^2Y(t) = 0 \tag{2}$$

on the interval $[0, 1]$, subject to initial conditions

$$Y(0) = I \quad \text{and} \quad Y'(0) = 0 ,$$

where A is an $(n \times n)$ real-valued, positive definite symmetric

matrix and I is the $(n \times n)$ identity matrix. It has been determined that the solution to this differential equation is

$$Y(t) = \cos(At) \equiv \sum_{j=0}^{\infty} (-1)^j \frac{(At)^{2j}}{(2j)!} \qquad \text{c.f. [3] .}$$

Hence, as a special case, setting $t = 1$, we are presented with the problem of calculating the matrix cosine of A, denoted $\cos(A)$.

Methods for approximating the cosine of a matrix have been developed, but many rely on knowing the eigenvalues and eigenvectors of the matrix (c.f. [1]) and this information, obviously, may not always be available or may not be economically obtained. We will discuss, and test the effectiveness of, two new methods which require no detailed information concerning the eigensystem of A. We shall assume, for simplicity, that A is a real-valued, positive definite symmetric matrix, but the exact eigenvalues need not be known.

The central idea of the two methods is the generation of schemes via the exact representation

$$y(t + k) + y(t - k) = 2\cos(kA)\, y(t) \; . \tag{3}$$

The first method to be described takes a well-known time-stepping procedure which approximates (3), and then uses a post-processing procedure which acts as a corrector. In this regard, it appears to be applicable only to homogeneous problems. Like the first, the second method constructs a rational approximation to the cosine of A, but implements this approximation in a partial fraction form.

We will see that this representation affords opportunities both for parallel computation within a time-step, as well as an efficient local error estimate. It is our intention to describe these methods, implement them, and to compare their performance in a practical computing environment with what theory predicts. As a part of the analysis, the post-processing method will be compared to a more known model, the Double Angle method.

CHAPTER I

THE TIME-STEPPING PROCEDURE WITH A POST-PROCESSOR

The generating scheme of equation (3) is a very useful one since our original problem, that is, finding $Y = \cos(A)$, is equivalent to the problem of solving the system of equations in (2) column by column, where each column corresponds to a different set of initial conditions. For example, for a given step size $k = 1/m$, where $m \geq 1$ (since our interval stops at 1), suppose we would like to calculate the first column of the solution $Y = \cos(A)$. Then, referring to equation (3), all we would need is the first column's solution initially $(y_0^{(1)})$ and its solution, or an approximation, at the first time step k $(y_1^{(1)})$. (See Figure 1).

0	k	2k	3k	. . .	(m-1)k	mk = 1
$y_0^{(1)}$	$y_1^{(1)}$	$y_2^{(1)}$	$y_3^{(1)}$		$y_{m-1}^{(1)}$	$y_m^{(1)}$

FIGURE 1

GRAPHICAL REPRESENTATION OF THE TIME-STEPPING PROCEDURE AS APPLIED TO THE FIRST COLUMN

We could then generate the first column of the solution $Y = \mathrm{Cos}(A)$ by simply applying the recursion formula of equation (3), but in the form

$$y(t + k) = 2\cos(kA)\, y(t) - y(t - k), \tag{4}$$

(m-1) times. Then, the entry which is listed as $y_m^{(1)}$ in Figure 1 would be the first column of the solution $Y = \cos(A)$. Applying this same procedure to all n columns would result in the calculation of the entire matrix cosine of A.

The problem with this procedure lies in the formation of the $\cos(kA)$ term of equation (3). After all, if we could form this, then we could also form $\cos(A)$ and our problem would be solved. So, instead of forming $\cos(A)$ exactly we approximate it by a rational operator $R(kA)$, where

$$R(kA) = Q^{-1}(kA)\, P(kA) ,$$

with P and Q being even polynomials in their argument. Then, in place of the exact scheme given in (3) are substituted approximate schemes of the form

$$w_{j+1} + w_{j-1} = 2R(kA)\, w_j , \tag{5}$$

or,

$$Q(kA)(w_{j+1} + w_{j-1}) = 2P(kA)\, w_j , \tag{6}$$

where $w_j \doteq y(t_j)$ and $t_j = jk$. As would be expected, the higher the order of the rational function $R(kA)$ as an approximation to $\cos(kA)$, the better the approximation schemes. But also, along with a higher order $R(kA)$ comes a more expensive approximating scheme.

Hence it seems that once again we should not expect accurate approximations to the cosine of A without paying for them. But, as we shall see, accurate approximations can be achieved without paying as much as it appears.

The key to the establishment of the post-processing procedure is a representation of exact and approximate solutions in terms of the data at $t = 0$ and $t = k$. We begin by recalling the Chebyshev polynomials of the second kind, defined by

$$S_{-2}(x) = -1 \ , \ S_{-1}(x) = 0 \ , \text{ and } \ S_j(x) = \frac{\sin(j+1)\theta}{\sin\theta} \ , \ j \geq -2$$

with $\theta = \cos^{-1}x$, which obey the recursion formula

$$S_{j+1}(x) = 2xS_j(x) - S_{j-1}(x) \ , \ j \geq -1 \ . \tag{7}$$

These polynomials can be used to establish the following theorem.

Theorem 1. Let $c = \cos(kA)$ and let $\rho = R(kA)$, where ρ represents any particular rational approximation. Then

$$y_j = S_{j-1}(c)\, y_1 - S_{j-2}(c)\, y_0 \tag{8}$$

and

$$w_j = S_{j-1}(\rho)\, w_1 - S_{j-2}(\rho)\, w_0 \ . \tag{9}$$

Proof. Induction on j will be used. Let $j = 0$; then

$$S_{-1}(c)\, y_1 - S_{-2}(c)\, y_0 = 0(y_1) - (-1)y_0 = y_0 .$$

When $j = 1$, we consider

$$S_0(c)\, y_1 - S_{-1}(c)\, y_0 . \tag{10}$$

But from (7) we know that

$$S_0(x) = 2xS_{-1}(x) - S_{-2}(x) \tag{11}$$

and substituting (11) into (10) yields

$$\begin{aligned} S_0(c)\, y_1 - S_{-1}(c)\, y_0 &= [2cS_{-1}(c) - S_{-2}(c)]y_1 - S_{-1}(c)\, y_0 \\ &= [0 - (-1)]\, y_1 - 0(y_0) = y_1 . \end{aligned}$$

Now we shall assume the relation holds for $j \leq n - 1$ and show it holds for $j = n$, $n \geq 2$. From (7) we have

$$\begin{aligned} & S_{n-1}(c)\, y_1 - S_{n-2}(c)\, y_0 \\ &\quad = [2cS_{n-2}(c) - S_{n-3}(c)]y_1 - [2cS_{n-3}(c) - S_{n-4}(c)]\, y_0 \\ &\quad = 2c\, [S_{n-2}(c)\, y_1 - S_{n-3}(c)\, y_0] - [S_{n-3}(c)\, y_1 - S_{n-4}(c)\, y_0] \\ &\quad = 2cy_{n-1} - y_{n-2} = y_n . \end{aligned}$$

The theorem is proved in a similar manner for w_n . This theorem will be needed a little later to establish an important result for the approximating schemes.

Recall also the Chebyshev polynomials of the first kind, defined by

$$T_j(x) = \cos(j\theta) ,$$

again with $\theta = \cos^{-1} x$. Chebyshev polynomials of the first and second kind are related by

$$T_j(x) = xS_{j-1}(x) - S_{j-2}(x) .$$

The terms $S_{n-1}(\rho)w_1$ and $S_{n-2}(\rho)w_0$ of equation (9) can be formed simultaneously in the basic representation given in equation (6); however, for the purpose of post-processing, it becomes necessary to separate out the different components, and hence form the two terms separately. But this does not represent a real problem since the terms can be generated in parallel. The idea is to generate the following sequences:

$$u_{j+1} = 2\rho u_j - u_{j-1} \tag{12}$$

with $u_0 = y_0 = w_0$, $u_1 = 0$, and

$$z_{j+1} = 2\rho z_j - z_{j-1} \tag{13}$$

with $z_0 = 0$, $z_1 = w_1$ where w_1 is the approximation to a column of $\cos(kA)$ at the first time step. Then,

$$w_j = u_j + z_j \ , \ j \geq 2 \ ,$$

will be the approximation to $y(t_j)$ where again $t_j = jk$, since

$$S_{j-1}(\rho)w_1 = S_{j-1}(\rho)z_1 - S_{j-2}(\rho)z_0 = z_j$$

and

$$S_{j-2}(\rho)w_0 = -[S_{j-1}(\rho)u_1 - S_{j-2}(\rho)u_0] = (-1)u_j \ .$$

Again, the accuracy of the sequences generated in (12) and (13) will depend on the order of ρ , and as was mentioned earlier, we would like to avoid using high-order operators since the cost of using such operators can become quite expensive. This is due to the amount of systems that have to be solved, at each time-step, when a higher-order rational function is used to generate the sequences in (6) . Also, when a lot of systems have to be solved, a lot of back solving has to be done and hence the cost of generating the sequence can be very expensive. So, we could just pick a high-order rational approximation, but rather than compute the sequences using the higher-order ρ , the idea of post-processing at a fixed time station is that we can use the Taylor series to approximate $S_j(\rho)$. Hence, we have

$$(14) \qquad S_j(\rho) = S_j(\gamma) + (\rho - \gamma)S_j'(\gamma) + \frac{(\rho - \gamma)^2}{2} S_j''(\gamma) + \ldots \ ,$$

where $\gamma \equiv G(kA)$ is some lower-order approximation to $\cos(kA)$ and $\rho \equiv R(kA)$ a higher-order approximation. The derivative terms though cannot be applied directly since there exist no corresponding term in the exact expansion. We can, however, derive expressions for the derivatives using identities of Chebyshev polynomials. One identity, for example, is

$$(1 - x^2)S_j'(x) = -jxS_j(x) + (j + 1)S_{j-1}(x)$$

from which the following can be derived:

$$(1 - x^2)^2 S_j''(x) = j[(j - 1)x^2 - 1]S_j(x) + (j + 1)(3 - 2j)xS_{j-1}(x) + j(j + 1)S_{j-2}(x) .$$

Terms involving higher derivatives can be derived by simply differentiating appropriate formulas and multiplying by the appropriate factor of $(1 - x^2)$. Then the Taylor approximation now becomes, for example:

$$\begin{aligned} S_j(\rho) = {} & S_j(\gamma) + (\rho - \gamma)(1 - \gamma^2)^{-1} [-j\gamma S_j(\gamma) + (j + 1)S_{j-1}(\gamma)] \\ & + \frac{(\rho - \gamma)^2}{2} (1 - \gamma^2)^{-2}\{j[(j - 1)\gamma^2 - 1]S_j(\gamma) \\ & + (j + 1)(3 - 2j)\gamma S_{j-1}(\gamma) + j(j + 1)S_{j-2}(\gamma)\} \\ & + \mathcal{O}(\|\rho - \gamma\|^3) \end{aligned}$$

where the constant in $\mathcal{O}(\|\rho - \gamma\|^3)$ depends on j .

So, having generated the sequences in (12) and (13) we can then apply the first step of the post-processing procedure. That is, form

$$S_j(\rho) \doteq S_j(\gamma) + (\rho - \gamma)(1 - \gamma^2)^{-1}[-j\gamma S_j(\gamma) + (j + 1)S_{j-1}(\gamma)]$$

which then yields

$$\begin{aligned} & S_{j-1}(\rho)w_1 - S_{j-2}(\rho)w_0 \\ & \doteq S_{j-1}(\gamma)w_1 - S_{j-2}(\gamma)w_0 + \frac{(\rho - \gamma)}{(1 - \gamma^2)}[-(j - 1)\gamma S_{j-1}(\gamma)w_1 \\ & \qquad + jS_{j-1}(\gamma)w_1 - (j - 2)\gamma S_{j-2}(\gamma)w_0 + (j - 1)S_{j-3}(\gamma)w_0] \\ & = \tilde{z}_j + \tilde{u}_j + \frac{(\rho - \gamma)}{(1 - \gamma^2)}[-(j - 1)\gamma\tilde{z}_j + j\tilde{z}_{j-1} \\ & \qquad - (j - 2)\gamma\tilde{u}_j + (j - 1)\tilde{u}_{j-1}] \end{aligned}$$

(where the $\sim$ indicates that the terms are computed using the lower-order scheme). Notice that for this method only the two previous values of $\tilde{z}$ and $\tilde{u}$ are necessary for generating the next $\tilde{z}$ and $\tilde{u}$ values in the sequence. Also, in order to form the first connection term only the last two values of $\tilde{z}$ and $\tilde{u}$ are needed, which implies that no large amount of storage is needed for this method. In general, if we wish to correct using m corrections, we will need to store the final $(m + 1)$ values of $\tilde{z}$ and $\tilde{u}$. But this should not cause us any major concern since usually only a few corrections are needed. Also, adding more and more corrections causes the amount of systems to be solved to grow quite quickly.

For more accuracy we can then add in the next term in the Taylor expansion:

$$\frac{1}{2}\left\{\frac{\rho - \gamma}{1 - \gamma^2}\right\}^2 \{(j - 1)[(j - 2)\gamma^2 - 1]\tilde{z}_j + j(5 - 2j)\gamma\tilde{z}_{j-1}$$
$$+ j(j - 1)\tilde{z}_{j-2} + (j - 2)[(j - 3)\gamma^2 - 1]\tilde{u}_j$$
$$+ (j - 1)(7 - 2j)\gamma\tilde{u}_{j-1} + (j - 1)(j - 2)\tilde{u}_{j-2}\}$$

and continue to correct as long as we wish, in principle, by including more terms in the Taylor series. In practice, though, truncation and roundoff error will also be multiplied by increasing powers of j, which limits the achieved accuracy.

The order of the post-processing procedure will now be analyzed. Suppose the error in the approximations of R and G are such that

$$|\cos(z) - G(z)| = \mathcal{O}(|z|^{\mu+2})$$

and

$$|\cos(z) - R(z)| = \mathcal{O}(|z|^{\nu+2})$$

with $\mu < \nu$ since $R(z)$ is of a higher-order than $G(z)$. It is known (c.f. [5]) that if $y_0 = \hat{y}_0$ and $\|y_1 - \hat{y}_1\| = \mathcal{O}(k^{\nu+1})$ (in any suitable norm) where y_i is the exact value of y and $\hat{y}_i$ is the sequence generated using the higher-order operator ρ, then $\|y_j - \hat{y}_j\| = \mathcal{O}(k^{\nu})$, that is, the order of the error in the generated

approximation is decreased by two from the local accuracy. Furthermore, $\|\rho - \gamma\| = \mathcal{O}(k^{\mu+2})$, $\|\rho - \gamma\|^2 = \mathcal{O}^{(2\mu+4)}$, $\|\rho - \gamma\|^3 = \mathcal{O}^{(3\mu+6)}$ etc. . Also, the bound on the term $|S_{j-1}^{(m)}(x)|$ can be shown to be $Cj^{(2m+1)}$ [2] . From this, it appears at first glance that the error committed when the first truncated term in the expansion is $S_{j-1}^{(m)}(\gamma)(\rho - \gamma)^m$, will be

$$\mathcal{O}(k^{m(\mu+2)} \cdot k^{-(2m+1)}) = \mathcal{O}(k^{m\mu-1}) .$$

Clearly this gives the wrong estimation for $m = 1$ (no correction). It turns out that a more delicate analysis confirms that the correct result is $\mathcal{O}(k^{m\mu})$. Therefore the accuracy of the processes value will be

$$\begin{aligned} \|y_j - \tilde{y}_j\| &= \|y_j - \hat{y}_j + \hat{y}_j - \tilde{y}_j\| \\ &\leq \|y_j - \hat{y}_j\| + \|\hat{y}_j - \tilde{y}_j\| \\ &= \mathcal{O}(k^{\min(\nu, m\mu)}), \end{aligned}$$

where $\tilde{y}_i$ is the sequence generated using the lower-order operator γ . The key thing to notice is that the order is improved multiplicatively.

CHAPTER II

EXPRESSING THE RATIONAL OPERATOR IN A PARTIAL-FRACTION FORM

Consider again the iterative scheme given by equation (5). For a given rational operator and step size it would be desirable to be able to approximate the local error present in our generated sequence. Representing the rational operator $R(kA)$ in a partial-fraction form allows us easily to approximate the local error at any of the $m-1$ steps. In addition, it would be of great computational benefit were we to be computing in a parallel environment to dole out equivalent portions of the action of $R(kA)$ to different processors. Further, since A^2 is often given implicitly as $M^{-1}K$, M, K sparse, a decomposition of $R(kA)$ using only terms linear in A^2 would be of interest for sparsity concerns. For these reasons, then, we will see that it behooves us to represent the rational operator in the form

$$(16) \qquad R(z) = C_1 + \frac{\alpha_1}{1 + \beta_1 z^2} + \frac{\alpha_2}{1 + \beta_2 z^2} + \ldots$$

with C_1, α_i, and β_i $(i = 1,2,\ldots)$ all being real constants. Also, to avoid having a representation of the form

$$(17) \qquad R(z) = C_1 + \frac{\alpha_1}{1 + \beta_1 z^2} + \frac{\alpha_2 + \alpha_3 z}{1 + \beta_1 z^2)^2} \quad ,$$

i.e. quadratic factors, we require that all the β_i in (16), and

thus the poles of $R(z)$, be distinct. So, now we must determine the coefficients (C_1, α_i, and β_i) for a rational operator of a given order.

We begin by analyzing the simplest case, i.e. $R(z)$ a rational operator of order two. We then wish to represent $R(z)$ by

$$R(z) = \frac{1 + p_1 z^2}{1 + q_1 z^2} \tag{18}$$

which means the error in this approximation can be

$$\cos(z) - R(z) = \cos(z) - \frac{P(z)}{Q(z)} = \mathcal{O}(z^4) ,$$

which gives

$$\begin{aligned} \cos(z)\, Q(z) - P(z) &= \mathcal{O}(z^4) \\ &= (1 - \frac{z^2}{2} + \frac{z^4}{24} - \ldots)(1 + q_1 z^2) - (1 + p_1 z^2) \\ &= (1 + (q_1 - \tfrac{1}{2})z^2 + \mathcal{O}(z^4) + \ldots) - (1 + p_1 z^2) \\ &= (q_1 - \tfrac{1}{2})z^2 - p_1 z^2 + \mathcal{O}(z^4) = (z^4) , \end{aligned}$$

and therefore $p_1 = (q_1 - \frac{1}{2})$ is a necessary and sufficient condition for the error to be fourth-order.

In addition to accuracy conditions, we also require stability. In our case, having prescribed that A have real eigenvalues, we shall ask for unconditional stability that

$$-1 \leq |R(z)| \leq 1 \qquad \text{for all real } z ,$$

or

$$-1 \leq \frac{P(z)}{Q(z)} \leq 1 \ .$$

The upper bound holds for all z since

$$\frac{P(z)}{Q(z)} = \frac{1 + (q_1 - \frac{1}{2})z^2}{1 + q_1 z^2} \leq 1$$

is equivalent to

$$1 + (q_1 - \tfrac{1}{2})z^2 \leq 1 + q_1 z^2 \ ,$$

which obviously is true for all real z . Now, for the lower bound to hold we must have

$$-1 \leq \frac{1 + (q_1 - \frac{1}{2})z^2}{1 + q_1 z^2}$$

or

$$-1 - q_1 z^2 \leq 1 + (q_1 - \tfrac{1}{2})z^2 \ ,$$

which is equivalent to

$$0 \leq 2 + (2q_1 - \tfrac{1}{2})z^2 \ .$$

In order to make this hold for all real z we require that the

coefficient of z^2 be non-negative, or

$$(19) \qquad 2q_1 - \frac{1}{2} \geq 0 \ ,$$

which can be written as

$$q_1 + (q_1 - \frac{1}{2}) = q_1 + p_1 \geq 0$$

and taking the boundary case (i.e. equality) yields

$$q_1 = -p_1 \ .$$

This result is of special interest since it can be easily shown that if $Q(z) = (q_1 z^2 + 1)$ were to be divided into $P(z) = (p_1 z^2 + 1)$ synthetically, the first term of the quotient would be

$$\frac{p_1}{q_1} \equiv -1 \ .$$

Now, when the partial-fraction form for $R(z)$ is constructed it will have the form

$$R(z) = C_1 + \frac{\alpha_1}{1 + \beta_1 z^2}$$

and we must also have

$$R(z) = \frac{1 + p_1 z^2}{1 + q_1 z^2}$$

Hence we have

$$R(z) = C_1 + \frac{\alpha_1}{1 + \beta_1 z^2} = \frac{C_1 + C_1\beta_1 z^2 + \alpha_1}{1 + \beta_1 z^2}$$

$$= \frac{(C_1 + \alpha_1) + C_1\beta_1 z^2}{1 + \beta_1 z^2}$$

$$= \frac{1 + p_1 z^2}{1 + q_1 z^2}$$

which implies that

$$C_1 + \alpha_1 = 1 \ , \ \beta_1 = q_1 \ , \text{ and } \ C_1\beta_1 = p_1 = (\beta_1 - \tfrac{1}{2}) \ .$$

Therefore, given β_1, C_1 is found using

$$C_1 = \frac{(\beta_1 - \frac{1}{2})}{\beta_1} = \frac{p_1}{q_1} = -1$$

and α_1 is found using

$$\alpha_1 = 1 - C_1 = 2 \ .$$

Notice also that in (19) we must have the condition

$$q_1 \geq \frac{1}{4}$$

and again taking the boundary case we get

$$q_1 = \frac{1}{4} .$$

But suppose that we would like to have approximations of the local error being committed in each term of the sequence. We illustrate the process first by assuming that the basic scheme is second-order and we use a higher-order scheme for estimation of the local error. We know that

$$y(t_{n+1}) - 2\text{Cos}(kA)\, y(t_n) + y(t_{n-1}) = 0$$

and therefore, assuming $y(t_n) = y_n$, $y(t_{n-1}) = y_{n-1}$ (as we do in the usual definition of local error),

$$y(t_{n+1}) = 2\cos(kA)y_n - y_{n-1}$$

and

$$y_{n+1} = 2R(kA)y_n - y_{n-1} .$$

Therefore

$$y(t_{n+1}) - y_{n+1} = 2[\text{Cos}(kA) - R(kA)]y_n$$

$$= 2\text{Cos}(kA)y_n - 2R(kA)y_n$$

is the true local error. But we could provide local approximations by doing the following: First, recall the scheme generated by $R(kA)$,

the rational operator of order two, given by

$$(20) \qquad y_{n+1} + y_{n-1} = 2R(kA)y_n = 2(C_1 + \alpha_1(I + \beta_1(kA)^2)^{-1})y_n$$

$$= 2C_1 y_n + \alpha_1 w$$

where $(I + \beta_1(kA)^2)w = y_n$. If at any of the $(m-1)$ steps we wish to approximate the error in the y_{n+1} term we could generate the term using a higher-order $R(kA)$. More specifically, the next higher-order $R(kA)$, say $R1(kA)$, which evolves from an expansion of the form

$$(21) \qquad R1(z) = \frac{1 + p_1 z^2 + p_2 z^2}{1 + q_1 z^2 + q_2 z^2} = C_2 + \frac{\alpha_1^*}{1 + \beta_1 z^2} + \frac{\alpha_2^*}{1 + \beta_2 z^2}$$

which may be selected to have an order of four.

Applying the same analysis to $R1$ that was done to R yields:

1. Given β_1, where $\beta_1 > \frac{1}{4}$ (notice the strict inequality now)
2. $\beta_2 \geq \frac{12\beta_1 - 1}{48\beta_1 - 12} \equiv F(\beta_1)$ is the condition for unconditional stability
3. $C_2 = \frac{24\beta_1\beta_2 - 12(\beta_1 + \beta_2) + 1}{24\ \beta_i\beta_2}$, $C_2 = -1$ when $\beta_2 = F(\beta_1)$
4. $\alpha_1^* = \frac{1 - 12\beta_2}{24\beta_1(\beta_1 - \beta_2)}$
5. $\alpha_2^* = \frac{12\beta_1 - 1}{24\beta_2(\beta_1 - \beta_2)}$

The value obtained using the higher-order method one time at this step satisfies

$$(22)\qquad \hat{y}_{n+1} + y_{n-1} = 2Rl(kA)y_n = 2(C_2 I + \alpha_1^*(I + \beta_1(kA)^2)^{-1} + \alpha_2^*(I + \beta_2(kA)^2)^{-1})y_n$$

$$= 2C_2 y_n + \alpha_1^* w + \alpha_2^* w_1 ,$$

where $(I + \beta_1(kA)^2)w = y_n$ and $(I + \beta_2(kA)^2)w_1 = y_n$. It is the following features that make this method so attractive. Notice that since in (20) w has already been calculated it will not be necessary to calculate it again in (21). The only terms that change when the order of the rational operator is increased are the coefficients. A new term w_1 will have to be calculated but since the system we will have to solve is linear, calculating w_1 will not prove to be time-consuming or costly. Notice also that if we did choose to use as our base scheme the rational operator in (21), the terms w and w_1 could be generated on a parallel processor and hence make for an even more efficient method. Now, the difference in (20) and (22) is

$$(23)\qquad \hat{y}_{n+1} - y_{n+1} = 2[(C_2 - C_1)y_n + (\alpha_1^* - \alpha_1)w + \alpha_2^* w_1] .$$

This implies that we can approximate the local error, again, by only solving the one system involving w_1 and taking linear combinations with the new coefficients.

Notice that since β_1 in $Rl(kA)$ is strictly greater than $1/4$, C_1 no longer is equal to minus one. This is because increasing the order of the rational operator has the effect of "pushing" the base scheme away from the boundary. The error in the generated sequence

is then approximated by (23) and if the term in (23) is large, then choosing a smaller step-size k or a higher-order base scheme should be considered.

For an analogous treatment using a fourth-order base scheme and a sixth-order estimator the conditions on $R(kA)$ of order six are

1. Given β_1 and β_2, where $(\beta_1 + \beta_2) > \frac{1}{4}$

2. $$\beta_3 \geq \frac{360\beta_1\beta_2 - 30(\beta_1 + \beta_2) + 1}{1440\beta_1\beta_2 - 360(\beta_1 + \beta_2) + 30} \equiv \tilde{F}(\beta_1\beta_2)$$

   (and for simplicity choose $\beta_3 = \tilde{F}(\beta_1\beta_2)$)

3. $C_3 = -1.0$

4. $$\hat{\alpha}_1 = \frac{48\beta_2\beta_3 - 12(\beta_2 + \beta_3) + 1}{24(\beta_2 - \beta_1)(\beta_3 - \beta_1)}$$

5. $$\hat{\alpha}_2 = \frac{48\beta_1\beta_3 - 12(\beta_1 + \beta_3) + 1}{24(\beta_1 - \beta_2)(\beta_3 - \beta_2)}$$

6. $$\hat{\alpha}_3 = \frac{48\beta_1\beta_2 - 12(\beta_1 + \beta_2) + 1}{24(\beta_2 - \beta_3)(\beta_1 - \beta_3)}$$

where

$$R(kA) = C_3 I + \hat{\alpha}_1(1 + \beta_1(kA)^2)^{-1} + \hat{\alpha}_2(1 + \beta_2(kA)^2)^{-1} + \hat{\alpha}_3(1 + \beta_3(kA)^2)^{-1} .$$

CHAPTER III

COMPUTING THE MATRIX COSINE EXACTLY AND USING THE DOUBLE ANGLE METHOD

The double angle method (c.f. [4]) and a way of computing the matrix cosine of A exactly will now be discussed. The double angle method is presented here since later there will be comparisons made between the double angle method and the time-stepping procedure. We will need to compute the matrix cosine exactly in order to calculate errors in our approximations.

In theory, the double angle method may be applied to a matrix A provided that A is a real, $(n \times n)$, normal matrix. The first step is to form a rational approximation to the cosine of A. But a necessary condition for stability of the rational approximation, as it will be used in a generating scheme, is that $\|R(A)\| \leq 1$. Hence, in order to insure that the rational approximation is stable we scale the matrix A by an appropriate power of two until $\|R(A/2^m)\| \leq 1$. The choice of the parameter m can be determined in terms of the spectral radius of A [4]. Having done this we then generate an approximation to the matrix cosine using a version of the double angle formula as applied to matrices, which is

$$\cos(2A) = 2\cos^2 A - I .$$

The scheme then is

$$U_{j+1} = 2U_j^2 - I , \quad j = 0, \ldots, m-1 ,$$

where $U_0 = R(A/2^m)$. Then, U_m approximates $\cos(A)$. Specific error bounds can be found in [4] . We observe that the double angle method is inherently a serial method. That is, the entire matrix must be formed at once. Also, the double angle method calls for the repeated squaring of each generated matrix U_j and this may be expensive for a large matrix.

In order to check the accuracy of the methods mentioned here it is necessary to be able to compute the cosine of a matrix exactly. To do this we use the fact that given a matrix A with

$$A = T^{-1} D T$$

where $T^{-1} = T^t$ (that is, T and T^{-1} are orthonormal matrices) and D a diagonal matrix, then

$$\cos(A) = T^{-1} \cos(D) T$$

where $\cos(D)$ is the matrix obtained by taking the cosine of all the diagonal entries. Similarly,

$$\cos(kA) = T^{-1} \cos(kD) T .$$

CHAPTER IV

NUMERICAL RESULTS

The matrices which were used to construct the test matrices were

$$T = \begin{bmatrix} 0 & -.5 & 0 & -.5 & 0 & .5 & 0 & 0 & .5 & 0 \\ 0 & 0 & 0 & 0 & 0 & .5 & .5 & -.5 & -.5 & 0 \\ -.5 & -.5 & 0 & 0 & 0 & -.5 & 0 & -.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & .5 & .5 & 0 & .5 & 0 & .5 & 0 \\ -.5 & 0 & 0 & 0 & -.5 & 0 & .5 & .5 & 0 & 0 \\ .5 & -.5 & 0 & .5 & -.5 & 0 & 0 & 0 & 0 & 0 \\ .4 & 0 & .6 & -.4 & 0 & -.4 & .4 & 0 & 0 & 0 \\ 0 & -.5 & 0 & 0 & .5 & 0 & 0 & .5 & -.5 & 0 \\ -.3 & 0 & .8 & .3 & 0 & .3 & -.3 & 0 & 0 & 0 \end{bmatrix}$$

$D_1 = \text{diag}[1,1,2,2,3,3,4,4,5,5]$, and $D_2 = \text{diag}[1,1,2,2,3,3,4,4,5,100]$ (recall that $T^{-1} = T^t$) . The test matrices

$$A_1 = T^{-1} D_1 T$$

and

$$A_2 = T^{-1} D_2 T ,$$

because of the magnitude of their largest eigenvalue, shall be referred

to as the regular problem and the large problem, respectively. As the values of $y_1^{(i)}$, $i = 1,2,\ldots, n$, we used the exact values at the first time-step, which corresponds to the ith column of $\cos(kA)$. In practice, this is usually not known, and has to be generated by a single-step method of appropriate order. The rational operator which served as ρ , the high-order approximation to $\cos(kA)$, in (14) was

$$(24) \qquad R(z) = \frac{1 + (2\beta_1 - 1/2)z^2 + (\beta_1^2 - \beta_1 + 1/24)z^4}{(1 + \beta_1 z^2)^2}$$

and the rational operator serving as the low-order function γ was

$$(25) \qquad R(z) = \frac{1 + (\beta - 1/2)z^2}{1 + \beta z^2} \quad .$$

The matrix which represents the actual cosine of A shall be denoted $Y(1)$, the matrix which represents the cosine approximation generated by the higher-order ρ shall be denoted $\hat{Y}(1)$, and the matrix which represents the cosine approximation generated by the lower-order γ , with corrections, shall be denoted $\tilde{Y}(1)$.

For the regular problem the values of β which produced accurate results were the unconditionally stable $\beta = 1/4$ and the conditionally stable $\beta = 1/12$. The value of β_1 which produced the most accurate results was $\beta_1 = (5 - \sqrt{15})/60$, which can be shown to be sixth-order conditionally stable approximation of the form (24). The parameter $\beta_1 = (5 + \sqrt{15})/60$ also produces accurate results and also can be

shown to be a sixth-order conditionally stable approximation of the form (24). Unless otherwise stated, all errors which are mentioned were measured in the Euclidean norm, denoted $\|\cdot\|_E$, where

$$\|A - B\|_E = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (a_{ij} - b_{ij})^2} \quad .$$

Table IV-1 lists the errors that result from using the low-order rational operator, with and without corrections, in the generating scheme

$$w_{j+1} + w_{j-1} = 2R(kA_1)w_j \ ,$$

as an approximation of the scheme generated with the higher-order operator. The results suggest that for the regular problem $\beta = 1/12$ serves as a better parameter than does $\beta = 1/4$. Also, $\beta = 1/12$ is so effective that once one or two corrections are performed, better approximations are produced with a step-size of .02 (50 intervals) than with much smaller step-sizes (probably due to roundoff error). With $\beta = 1/4$, decreasing the step-size always tends to reduce the error when one or no corrections are performed. When two corrections are performed and $\beta = 1/4$ a step-size of .005 (200 intervals) produces better results than do step-sizes much smaller than .005 (again this may be due to roundoff error). The number of intervals listed at the top of Table IV-1 indicate the number of time-steps.

Our predictions of the convergence of the time-stepping procedure are shown to be quite accurate. The rates listed were calculated

TABLE IV-1

MEASURING $\|\hat{y}(1) - \tilde{y}(1)\|_E$ FOR THE REGULAR PROBLEM

			Number of Intervals								
β	β_1	Corrections	10	Conv. rate	50	100	Conv. rate	200	500	Conv. rate	1000
$\frac{1}{4}$	-	0	.100(+0)	1.92	.452(-2)	.114(-2)	1.99	.287(-3)	.460(-4)	2.00	.115(-4)
$\frac{1}{12}$	-	0	.605(-3)	3.95	.104(-5)	.658(-7)	3.99	.413(-8)	.105(-9)		.170(-10)
$\frac{1}{4}$	$\frac{5-\sqrt{15}}{60}$	1	.247(-2)	3.77	.576(-5)	.370(-6)	3.98	.234(-7)	.603(-9)		.414(-9)
$\frac{1}{12}$	$\frac{5-\sqrt{15}}{60}$	1	.113(-6)	7.65	.510(-12)	.177(-11)		.788(-11)	.803(-10)		.481(-9)
$\frac{1}{4}$	$\frac{5-\sqrt{15}}{60}$	2	.139(-4)	5.81	.120(-8)	.195(-10)		.154(-11)	.147(-10)		.167(-10)
$\frac{1}{12}$	$\frac{5-\sqrt{15}}{60}$	2	.367(-11)		.241(-13)	.113(-12)		.683(-12)	.110(-10)		.485(-10)

using every other adjacent pair of columns. In the first and second row we have $\beta = 1/4$ and $\beta = 1/12$, and hence second-order and fourth-order schemes, respectively. Our expected rates then are $m_\mu = (1)(2) = 2$ and $m_\mu = (1)(4) = 4$, respectively, and our results agree with our predictions. In rows three and four we expect $2(2) = 4$ and $2(4) = 8$, respectively, and again we get what we expected. The rate in row five also satisfies our prediction. Some of the convergence rates are not listed because the roundoff errors which occur in our calculations cause them to be in error.

Since our ultimate goal is to approximate the actual matrix cosine it would be of interest to see which rational operator generates an approximation which is closest to the true matrix cosine. Table IV-2 lists the errors that result from comparing the actual matrix cosine of A_1 to the approximation generated using the rational operator $R(kA_1)$ in (24). Table IV-3 lists the errors that result from comparing the actual matrix cosine of A_1 to the approximation generated using the rational operator $R(kA_1)$ in (25), with and without corrections. It appears that we can use a lower-order scheme and yet still approach the accuracy which is achieved using a higher-order scheme. It is necessary for us to apply one correcting term, but considering that the system we are using has one term of correction which is linear, it would seem that using the lower-order scheme and correcting once is cheaper than generating the scheme using the higher-order scheme and having to solve a quadratic system at each time-step. Also note in Table IV-3 that the accuracy achieved with $\beta = 1/12$ and one correction is equivalent to

TABLE IV-2

MEASURING $\|y(1) - \hat{y}(1)\|_E$ FOR THE REGULAR PROBLEM

	Number of Intervals							
β_1	10	Conv. rate	50	100	Conv. rate	200	500	Conv. rate	1000
$\frac{5-\sqrt{15}}{60}$	.126(-5)	5.96	.854(-10)	.138(-11)		.100(-11)	.492(-11)		.222(-10)

TABLE IV-3

MEASURING $\|y(1) - \tilde{y}(1)\|_E$ FOR THE REGULAR PROBLEM

Number			Number of Intervals								
β	β_1	Corrections	10	Conv. rate	50	100	Conv. rate	200	500	Conv. rate	1000
$\frac{1}{4}$	-	0	.133(+0)	1.93	.597(-2)	.151(-2)	1.99	.379(-3)	.609(-4)	2.00	.152(-4)
$\frac{1}{12}$	-	0	.829(-3)	3.95	.143(-5)	.903(-7)	3.99	.567(-8)	.145(-9)		.144(-10)
$\frac{1}{4}$	$\frac{5-\sqrt{15}}{60}$	1	.831(-3)	3.75	.197(-5)	.126(-6)	3.98	.801(-7)	.212(-9)		.603(-10)
$\frac{1}{12}$	$\frac{5-\sqrt{15}}{60}$	1	.400(-6)	5.91	.296(-10)	.725(-12)		.242(-11)	.366(-10)		.161(-9)
$\frac{1}{4}$	$\frac{5-\sqrt{15}}{60}$	2	.436(-4)	5.81	.379(-8)	.612(-10)		.478(-11)	.451(-10)		.604(-10)
$\frac{1}{12}$	$\frac{5-\sqrt{15}}{60}$	2	.440(-6)	5.97	.297(-10)	.726(-12)		.242(-11)	.366(-10)		.161(-9)

to the accuracy achieved, using the same β and two corrections (agin roundoff error causes this lack of gain in accuracy).

The rates in Table IV-3 again agree with our predictions. In rows one and two we expect convergence rates of $\min(\nu = 6, m\mu = (1)(2)) = 2$ and $\min(\nu = 6, (1)(4)) = 4$, respectively. In rows three and four we expect rates of $\min(\nu = 6, (2)(2)) = 4$ and $\min(\nu = 6, (2)(4)) = 6$, respectively. The expected rates of convergence for the entries in rows five and six are $\min(6, 6) = 6$ and $\min(6, 3(4)) = 6$.

For the large problem in which we introduce a large eigenvalue, and thus for which stability becomes an issue, the values of β producing the most accurate results were $\beta = 1/4$, $\beta = 1/12$, and $\beta = (3 + \sqrt{2})/12$. The parameter $\beta = (3 + \sqrt{2})/12$ was selected since a goal of ours was to make the norm of the "corrector operator",

$$\frac{\rho - \gamma}{1 - \gamma^2},$$

small and therefore maybe keep the error amplification down. The best choice for the large problem was $\beta_1 = 1/4$. Table IV-4 lists the errors that result from approximating the exact matrix cosine solution with approximations generated using the lower-order operator with corrections. Notice that when a large step-size is chosen, the conditionally stable $\beta = 1/12$ is a bad choice for the parameter since the scheme does not converge. This is because the rational operator has a norm greater than one and therefore is not stable. But when the step-size is chosen to be small, $\beta = 1/12$ is the best of the three parameters. Neither $\beta = 1/4$ nor $\beta = (3 + \sqrt{2})/12$

TABLE IV-4

MEASURING $\|y(1) - \tilde{y}(1)\|_E$ FOR THE LARGE PROBLEM

			Number of Intervals								
β	β_1	Conditions	10	Conv. rate	50	100	Conv. rate	200	500	Conv. rate	1000
$\frac{3+\sqrt{2}}{12}$	-	0	.191(+6)		.891(+0)	.596(+0)		.156(+1)	.402(+0)	2.33	.801(-1)
$\frac{1}{12}$	-	0	*[a]		.782(+0)	.665(+0)	4.01	.413(-1)	.110(-2)	3.98	.699(-4)
$\frac{1}{4}$	-	0	.191(+6)		.119(+1)	.158(+0)		.164(+1)	.210(+0)	2.22	.449(-1)
$\frac{1}{4}$	$\frac{1}{4}$	1	.191(+6)		.573(+1)	.640(+1)	3.60	.524(+0)	.397(-1)	3.78	.228(-2)
$\frac{1}{4}$	$\frac{1}{4}$	2	.126(-4)		.120(-8)	.195(-10)		.150(-11)	.146(-10)		.173(-10)

[a]No convergence for this condition

yield accurate approximations when no correction is used, but $\beta = 1/4$ does yield reasonable results when two corrections are applied. Also, choosing $\beta = \beta_1$ cuts down a little of the work, in that a new factorization of $(I + \beta_1(kA)^2)$ need not be performed. The large errors in Table IV-4 are due to the large eigenvalues which are present in the matrix A_2. This is made obvious in Table IV-5 which lists the same error measurement in Table IV-4 except that the approximations to the terms corresponding to the large eigenvalues are ignored. Notice how much more accurate the approximations become. Examining the last row of Tables IV-4 and IV-5 we see that when two corrections are performed, the errors in the two tables are identical. Hence, if two corrections are performed this choice of β and β_1 may serve as adequate parameters and this is some evidence of the efficacy of the correction process. But if we wish to obtain reasonable results with no corrections we may choose to use $\beta = 1/12$ and carefully select a small step-size, or choose $\beta = 1/4$ and a large step-size and simply ignore those approximations corresponding to the large eigenvalues, which would be evidenced by large changes in the corresponding components in successive corrections.

The efficiency of the time-stepping method was then compared to that of the double angle method. The comparison involved choosing a desired amount of accuracy, for which there corresponds a value of m (minimum), and then calculating the number of multiplications and divisions required for that accuracy, relative to the exact solution. For the double angle method m here represents the exponent of two in the rational function $R(A/2^m)$ and for the time-stepping

TABLE IV-5

MEASURING $\|y(1) - \tilde{y}(1)\|$ FOR THE LARGE PROBLEM,
BUT IGNORING APPROXIMATIONS OF THE LARGE ENTRIES

			Number of Intervals								
β	β_1	Corrections	10	Conv. rate	50	100	Conv. rate	200	500	Conv. rate	1000
$\frac{3+\sqrt{2}}{12}$	-	0	.166(0)	1.91	.770(-2)	.195(-2)	1.99	.490(-3)	.786(-4)	2.00	.196(-4)
$\frac{1}{12}$	-	0	.362(-2)	3.89	.692(-5)	.438(-6)	3.99	.275(-7)	.706(-9)	3.79	.511(-10)
$\frac{1}{4}$	-	0	.978(-1)	1.91	.451(-2)	.114(-2)	1.99	.287(-3)	.460(-4)	2.00	.115(-4)
$\frac{1}{4}$	$\frac{1}{4}$	1	.173(-2)	3.74	.420(-5)	.270(-6)	3.98	.171(-7)	.443(-9)		.403(-9)
$\frac{1}{4}$	$\frac{1}{4}$	2	.126(-4)	5.75	.120(-8)	.195(-10)		.150(-11)	.146(-10)		.173(-10)

procedure m stands for the number of intervals (or the number of time-steps). We also recorded the required CPU time of each method. Table IV-6 contains the results of the comparison.

Since the double angle method must be performed in series, the entire matrix must be formed and the number of multiplications and divisions occurring when implementating this method on an $(n \times n)$ matrix is $m(n^3 + n^2)$. Since for the time-stepping method each column can be generated simultaneously, assuming access to a parallel processor, the multiplication and division count and the CPU time for the time-stepping procedure was based on the generation of only one column (it is reasonable to assume that these calculations would be the same for any column as long as we were dealing with a regular problem, as we are here). The multiplication and division count for one column of the time-stepping procedure with zero, one, and two corrections, is $3(m-1)n^2$, $2(3m-1)n^2 + 4n$, and $2n^3 + (6m+5)n^2 + 15n$, respectively. As a result of the way these calculations were done, the errors had to be computed on a relative scale. Hence the error for the double angle method, since it is based on all n^2 entries, was computed using

$$e = \frac{\sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (y_{ij} - \hat{y}_{ij})^2}}{n}$$

where y_{ij} is a component of the exact solution and $\hat{y}_{ij}$ is the corresponding generated approximation. This formula is adequate since if all the entries were in error by the same amount, say p, the result of summing up the square of the same error n^2 times

TABLE IV-6

COMPARING DOUBLE ANGLE METHOD TO TIME-STEPPING PROCEDURE
FOR THE REGULAR PROBLEM

	METHOD											
	Double Angle				Time-stepping with no Corrections			Time-stepping with one Correction			Time-stepping with two Corrections	
Desired accuracy	Scaling exponent	CPU time (secs)	Mult. Div. count	time steps	CPU time (secs)	Mult. Div. count	time steps	CPU time (secs)	Mult. Div. count	time steps	CPU time (secs)	Mult. Div. count
10^{-3}	6	.02	6600	40	.04	11,700	7	.01	4,040	5	.02	8,050
10^{-5}	9	.03	9900	398	.35	119,100	25	.04	14,840	12	.03	9,850
10^{-7}	12	.05	13200	3979	3.39	1,193,400	80	.16	47,840	25	.05	17,650
10^{-9}	*[a]	---	---	*[a]	--	---	253	.46	151,640	55	.10	35,650

[a]This accuracy was not achieved with our choice of rational operators

would result in

$$e = \frac{\sqrt{n^2 p^2}}{n} = P$$

which is what it should be. Similarly, the error for the time-stepping procedure, since it is based on only n entries, is

$$e = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} .$$

For the data listed in Table IV-6 reported for the matrix A_1, the choice of $R(z)$ for the double angle method was

$$R(z) = \frac{1 + (\beta - 1/2)z^2}{1 + \beta z^2}$$

with $\beta = 1/4$ and the choice of ρ and γ for the time-stepping procedure where

$$R(z) = \frac{1 + (2\beta_1 - 1/2)z^2 + (\beta_1^2 - \beta_1 + 1/24)z^4}{(1 + \beta_1 z^2)^2}$$

and

$$G(z) = \frac{1 + (\beta - 1/2)z^2}{1 + \beta z^2}$$

with $\beta_1 = (5 - \sqrt{15})/60$ and $\beta = 1/4$.

The double angle method is a much more efficient method than is the time-stepping method when no corrections are performed. But

neither can produce an accuracy of 10^{-9} or better, due to roundoff, although in theory they should be able to. Depending on the degree of accuracy desired, the double angle method and the time-stepping method with one correction are comparable. If the desired accuracy is 10^{-3}, then time-stepping and correcting once is faster, and hence cheaper, than the double angle method. However, if the desired accuracy is 10^{-5} or 10^{-7}, the double angle method is the more efficient of the two. An accuracy of 10^{-9}, which we could not achieve with the double angle method and our chosen $R(z)$, can be achieved with one correction of the time-stepping procedure but producing this accuracy is quite expensive.

The work that is required for accuracies of 10^{-3}, 10^{-5}, and 10^{-7} is practically the same for the double angle method and the time-stepping procedure with two corrections. The advantage that correcting twice has over the double angle method is that, for our choices of ρ and γ, time-stepping and correcting twice can produce an accuracy of 10^{-9} in a rather efficient manner, whereas the double angle method cannot. These results do not, though, predict any general conclusions which can be drawn concerning comparison of the methods.

Recall that another way the time-stepping procedure may be implemented is by representing the rational operator in a partial-fraction form. The operator we choose as our base scheme was that of the form

$$R(z) = \frac{1 + p_1 z^2 + p_2 z^4}{1 + q_1 z^2 + q_2 z^4}$$

which was shown to have the partial-fraction form

$$R(z) = C_2 + \frac{\alpha_1^*}{1 + \beta_1 z^2} + \frac{\alpha_2^*}{1 + \beta_2 z^2} .$$

Since there appears to be no optimal choice of real poles β_1 and β_2, conditions on which were given in Chapter III, we decided to choose values of β_1 in the interval $[.3, .7]$ and test the sensitivity of these values on our approximations. Table IV-7 lists the errors, as compared to the actual solution, of the solutions we generated for those chosen values, as applied to the regular problem. It would seem that the situation we would want to have occur is to have the values of β_1 and β_2 become relatively close since when they are equal the error constant is minimized. This is made evident in Table IV-8 in which we let $\beta_1 = \beta_2$ (and hence no longer implement $R(z)$ in the partial-fraction form). Notice that when we allow $\beta_1 = \beta_2$, except for two entries, the results we get are equivalent to the results achieved in row five of Table IV-7 (which contained the best results among those in which β_1 and β_2 are distinct). Hence, for certain choices of distinct β_1 and β_2, the rational approximation, which can be implemented in partial-fraction form, provides results of nearly equal accuracy to those which could be obtained with the approximation corresponding to equal roots. But when $R(z)$ is in partial-fraction form, the closer the values of the two parameters get to each other the larger the coefficients α_1^* and α_2^* become. It seems that we want to avoid having the coefficients grow too large as this would make our approximations not very reliable numerically. But the approximation in Table IV-7 which is the most accurate occurs when α_1^* and α_2^* achieve their largest value (in

TABLE IV-7

MEASURING $\|y(1) - \tilde{y}(1)\|$ WHEN $R(kA_1)$ IS IMPLEMENTED IN PARTIAL-FRACTION FORM

					Number of Intervals (m)					
β_1	β_2	C_2	α_1	α_2	10	50	100	200	500	1000
.30	1.08	-1.0	2.13	-.127	.314(-1)	.719(-4)	.458(-5)	.289(-6)	.742(-8)	.451(-9)
.34	.713	-1.0	2.48	-.482	.248(-1)	.535(-4)	.340(-5)	.214(-6)	.550(-8)	.387(-9)
.38	.571	-1.0	3.36	-1.36	.226(-1)	.477(-4)	.303(-5)	.190(-6)	.490(-8)	.349(-9)
.42	.495	-1.0	6.53	-4.53	.218(-1)	.457(-4)	.290(-5)	.182(-6)	.471(-8)	.371(-9)
.46	.448	-1.0	-34.2	36.2	.216(-1)	.453(-4)	.288(-5)	.181(-6)	.481(-8)	.292(-9)
.50	.417	-1.0	-3.99	5.99	.218(-1)	.458(-4)	.291(-5)	.183(-6)	.470(-8)	.299(-9)
.54	.394	-1.0	-1.96	3.96	.222(-1)	.468(-4)	.297(-5)	.187(-6)	.480(-8)	.303(-9)
.58	.376	-1.0	-1.24	3.24	.227(-1)	.480(-4)	.305(-5)	.192(-6)	.443(-8)	.298(-9)
.62	.363	-1.0	-.875	2.88	.233(-1)	.495(-4)	.315(-5)	.198(-6)	.508(-8)	.311(-9)
.66	.352	-1.0	-.659	2.66	.239(-1)	.511(-4)	.325(-5)	.204(-6)	.526(8)	.320(-9)
.70	.343	-1.0	-.518	2.52	.246(-1)	.529(-4)	.336(-5)	.211(-6)	.544(-8)	.336(-9)

TABLE IV-8

MEASURING $\|y(1) - \hat{y}(1)\|$ FOR THE REGULAR PROBLEM

	Number of Intervals								
	10	Conv. rate	50	100	Conv. rate	200	500	Conv. rate	1000
$\frac{6+\sqrt{24}}{24}$	.216(-1)	3.83	.453(-4)	.288(-5)	3.99	.181(-6)	.466(-8)	4.00	.291(-9)

terms of magnitude), but these values are not excessively large. Also, Table IV-7 demonstrates that the errors achieved are not exceptionally sensitive to changes in the parameters.

The approximations which were generated using this method are very accurate when the number of time-steps is properly chosen. But the features of this method which caused us to want to represent the rational operator in the partial-fraction form were the ease at which we can compute higher-order approximations to $R(z)$ and the ease at which we may approximate the local error of the generated sequence. Recall that once we have formed the term

$$(26) \quad 2R(kA)y_j = 2[C_1 I + \alpha_1^*(I + \beta_1(kA)^2)^{-1} + \alpha_2^*(I + \beta_2(kA)^2)^{-1}]y_j$$

$$= 2[C_2 y_j + \alpha_1^* w + \alpha_2^* w_1] ,$$

in the fourth-order approximation, where, as before, $(I + \beta_1(kA)^2)w = y_j$ and $(I + \beta_2(kA)^2 w_1 = y_j$, if we wish to increase the order of the rational operator by two (and hence have a sixth-order operator), and therefore form

$$(27) \quad 2R(kA)y_j = 2[C_3 I + \hat{\alpha}_1(I + \beta_1(kA)^2)^{-1} + \hat{\alpha}_2(I + \beta_1(kA)^2)^{-1}$$

$$+ \hat{\alpha}_3(I + \beta_3(kA)^2)^{-1}]y_j$$

$$= 2[C_3 y_j + \hat{\alpha}_1 w + \hat{\alpha}_2 w_1 + \hat{\alpha}_3 w_2 1 ,$$

we would not have to recalculate w and w_1 since they remain the same. We would have to calculate w_2 and the new coefficients, all

calculations which are simple considering the effect they will have on our accuracy.

Another important result that should be revealed is that the effect of "pushing" away from the boundary, that is, letting $\beta_2 > F(\beta_1)$, as opposed to letting $\beta_2 = F(\beta_1)$, does cause the error to change (become worse) slightly. The further β_2 is "pushed", the larger the error becomes (see Tables IV-9 and IV-10).

The other outstanding feature of the partial-fraction technique is its ability to approximate the local error (e) being committed at any time-step. Assuming we generate our sequence using the operator in (26) we can approximate the error of any term in the sequence by forming

$$(28) \qquad \hat{e} = \hat{y}_{n+1} - y_{n+1} = 2[(C_3 - C_2)y_j + (\hat{\alpha}_1 - \hat{\alpha}_1^*)w + (\hat{\alpha}_2 - \hat{\alpha}_2^*)w_1 + \hat{\alpha}_3 w_2]$$

where $\hat{y}_{n+1}$ is the approximation generated using (27) and y_{n+1} is the approximation generated using (26). Again, this calculation is simple since we already have w and w_1. We compared the approximate error (28) to the actual error that is committed at the second and last time-step of the procedure. The data listed in Table IV-11 are the differences between the exact error and the approximate errors (in absolute value) of the fourth column for the regular problem at the second and last time-step. The values of β_1, β_2, and β_3 used in Table IV-11 were 0.5, 0.7, and 0.89 respectively, and the number of time-steps used was 100. Table IV-12 lists the differences between

TABLE IV-9

ERROR RESULTING FROM LETTING $\beta_2 = F(\beta_1) + .01$

β_1	β_2	C_2	α_1	α_2	Number of Intervals 100	200
.30	1.09	-.997	2.12	-.125	.463(-5)	.291(-6)
.46	.458	-.980	-.256	.258	.296(-5)	.186(-6)
.54	.404	-.973	-2.18	4.15	.307(-5)	.193(-6)
70	.353	-.964	-.554	2.52	.349(-5)	.220(-6)

TABLE IV-10

ERROR RESULTING FROM LETTING $\beta_2 = F(\beta_1) + .05$

β_1	β_2	C_2	α_1	α_2	Number of Intervals 100	200
.30	1.13	-.985	2.10	-.115	.482(-5)	.303(-6)
.46	.458	-.908	11.7	-9.84	.329(-5)	.207(-6)
.54	.444	-.879	-3.46	5.34	.346(-5)	.218(-6)
.70	.392	-.836	-.719	2.55	.403(-5)	.253(-6)

TABLE IV-11

MEASURE OF $|e - \hat{e}|$ FOR THE SEQUENCE GENERATED W/R TO THE 4^{th} COLUMN OF A_1

Time Step	
Second	Last
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000077453	0.0000000000018254
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000

TABLE IV-12

MEASURE OF $|e - \hat{e}|$ FOR THE SEQUENCE GENERATED W/R TO THE 4^{th} COLUMN OF A_2

Time Step	
Second	Last
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0199592275406947	0.0316740417843800
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000

the exact errors and the approximate errors of the fourth column (which is where the entry corresponding to the largest eigenvalue is located) for the large problem at the second and last time-step. The parameters and step-size of Table IV-11 were also used for this table. Also, to show the effect the large eigenvalue has over some of the entries, Table IV-13 lists the differences between the exact errors and the approximate errors of the first column for the regular and large problem (it turns out to be the same) at the second and last time-step. For variety, the choice of parameters in this case was $\beta_1 = 0.5$, $\beta_2 = 2/3$, and $\beta_3 = 0.96$, and the number of time-steps used was 100. The results in the tables indicate that the approximations are accurate enough to determine whether our generated sequences are of an acceptable accuracy. We can see from Table IV-12 for example that for this choice of step-size our error is large and therefore we should consider decreasing our step-size or possibly increasing the order of our base scheme rational operator.

As a final bit of analysis, we should note that another advantage the time-stepping procedure has over the double angle method is that it has the capabilities to solve rectangular $(n \times m)$ systems where as the double angle method can only be applied to square systems. Also, although it was not done, the correcting procedure can be applied to the time-stepping procedure when $R(z)$ is formed in the partial-fraction form. We would simply have to generate two sequences as we did in Chapter II.

TABLE IV-13

MEASURE OF $|e - \hat{e}|$ FOR THE SEQUENCE GENERATED W/R TO THE FIRST COLUMN OF A_1 AND A_2

Time Step	
Second	Last
0.0000000000020095	0.0000000000004323
0.0000000000019422	0.0000000000004983
0.0000000000000333	0.0000000000000328
0.0000000000000000	0.0000000000000000
0.0000000000019743	0.0000000000004660
0.0000000000000000	0.0000000000000000
0.0000000000000013	0.0000000000000003
0.0000000000000256	0.0000000000000258
0.0000000000019756	0.0000000000004657
0.0000000000000192	0.0000000000000193

REFERENCES

REFERENCES

1. G.H. Golub, C.F. Van Loan, "Matrix Computations," The Johns Hopkins University Press, 1973.

2. Y.L. Luke, "The Special Functions and Their Approximations," Academic Press, 1969.

3. S.M. Serbin, "Rational Approximations of Trigonometric Matrices with Applications to Second-Order Systems of Differential Equations," Applied Mathematics and Computation, 5, 75-92 (1979).

4. S.M. Serbin, S.A. Blalock, "An Algorithm for Computing the Matrix Cosine," SIAM J. Sci. Stat. Comput., 1, 198-204 (1980).

5. S.M. Serbin, G.A. Baker, and V.A. Dougalis, "An Approximation Theorem for Second-Order Evolution Equations," Numer. Math., 35, 127-142 (1980).

VITA

Alan LaMont Fisher was born in Lexington, Kentucky on May 16, 1962. He graduated from Bryan Station Senior High School in June 1980. The following September he entered Centre College of Danville, Kentucky, and in June 1984 he received a Bachelor of Science degree, with a double major in mathematics and physics. In the Fall of 1984 he accepted a teaching assistantship at The University of Tennessee, Knoxville, and began study towards a Master's degree. This degree was awarded in August 1986.