NAWCWPNS TP 8356

# Using Diagonally Implicit Multistage Integration Methods for Solving Ordinary Differential Equations. Part 2: Implicit Methods

by
Jack VanWieren
*Research & Technology Group*

**August 1997**

**NAVAL AIR WARFARE CENTER WEAPONS DIVISION**
**CHINA LAKE, CA 93555-6100**


# Naval Air Warfare Center Weapons Division

## FOREWORD

Diagonally implicit multistage integration methods (DIMSIMs) hold great promise for providing more efficient, more accurate and more robust software for solving systems of ordinary differential equations numerically. The author was supported by Navy In-House Independent Research Funds.

This report is a working document subject to change and was reviewed for technical accuracy by Professor Zdzislaw Jackiewicz, Arizona State University, Tempe, Arizona.

Approved by
R. L. Derr, *Head*
*Research and Technology Group*
30 August 1997

Under authority of
J. V. CHENEVEY
RADM, U.S. Navy
*Commander*

Released for publication by
S. HAALAND
*Director for Research and Engineering*

NAWCWPNS Technical Publication 8356

Published by . . . . . Technical Information Division
Collation . . . . . Cover, 59 leaves
First printing . . . . . 30 copies

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | August 1997 | Interim report—October 1994-September 1996 |

**4. TITLE AND SUBTITLE**
Using Diagonally Implicit Multistage Integration Methods for Solving Ordinary Differential Equations. Part 2: Implicit Methods

**5. FUNDING NUMBERS**
N0001495WX30085
N0001495WX20167

**6. AUTHOR(S)**
Jack M. Van Wieren

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Naval Air Warfare Center Weapons Division
China Lake, CA 93555-6100

**8. PERFORMING ORGANIZATION REPORT NUMBER**
NAWCWPNS TP 8356

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
In-House Laboratory Independent Research
Dr. Ronald Derr
Code 4B0000D
Naval Air Warfare Center Weapons Division
China Lake, CA 93555-6100

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12A. DISTRIBUTION/AVAILABILITY STATEMENT**
A Statement; public release; distribution unlimited.

**12B. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

(U) New theoretical results for implicit methods of the recently invented and highly promising class of diagonally implicit multistage integration methods (DIMSIMs) are describeded. Some new A-stable and L-stable methods are derived, and conditions for a method to have an A-stable or L-stable FASAL (First Approximately Same As Last) implementation are derived. New alternative error estimates are proposed and successfully tested. The stability regions for alternative predictor-corrector DIMSIM implementations are derived. Implementation parameters for stiff solvers based on second and fifth order DIMSIMs are derived. Development of prototype stiff solvers of both second and fifth order stiff DIMSIM solvers is described. Results of successful tests on the Prothero-Robinson problem are reported and demonstrate that DIMSIMs may be used to develop efficient stiff solvers.

**14. SUBJECT TERMS**
Differential equations, DIMSIMs, stiff systems, numerical analysis, mathematical software

**15. NUMBER OF PAGES**
118

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL |

## CONTENTS



---

Figures:

---

Tables:

# INTRODUCTION

## BACKGROUND

This report is an extension of work previously reported (Reference 1) to initial value problems involving stiff systems of ordinary differential equations (ODEs). As pointed out by Hairer and Wanner (Reference 2), the term "stiff" is universally applied by numerical analysts to certain systems but does not as yet have a generally accepted precise mathematical definition. An early paper by Curtiss and Hirschfelder (Reference 3) supplies a practical definition, that stiff systems are systems of ODEs for which certain types of implicit numerical methods work far better than explicit methods. For stiff problems, numerical stability tends to become a bigger constraint on step-size selection than accuracy. As a result, methods with limited stability regions, including all explicit methods, tend to require a very small step size to maintain stability, which greatly increases the computational work and solution time, and may even cause serious roundoff errors to render solution infeasible. Hairer and Wanner (Reference 2) give examples of stiff systems that show they may consist of a single differential equation, often (but not necessarily) yield solutions consisting of a transient phase followed by a smooth steady-state solution, are often associated with physical processes that involve components of both very fast and very slow rates, and can result from method-of-lines discretization of partial differential equations. Shampine (Reference 4) points out that explicit methods may work well within the transient region of an otherwise stiff problem because step size is limited by accuracy, not stability. He concludes by noting that there is much about the theory and practice of solving stiff equations that is not understood.

Since Curtiss and Hirschfelder first introduced backwards differentiation formula (BDF) methods (Reference 3), there has been a considerable amount of work done to develop software to use these methods for stiff problems. Gear (see for example Reference 5) first developed sophisticated solvers based on BDF methods, and these are now typically called Gear methods. EPISODE (Reference 6), LSODE (Reference 7), and VODE (Reference 8) are widely used codes based on Gear's approach and include adaptive step-size selection and also adaptive order selection. Variable order is important because the BDF methods are not A-stable above second order. The entire negative real axis is included through sixth order, beyond which the methods are not even zero-stable, but the stability region becomes increasingly restricted in the area of the left half of the complex plane near the imaginary axis. Thus higher order methods may be used in regions where the Jacobian of the the derivative function has eigenvalues that are real or have small imaginary parts, which makes these methods effective with method-of-lines solution of parabolic partial differential equations. Reduction to second order is necessary for stiff oscillatory problems where the imaginary parts of the eigenvalues of the Jacobian dominate. Thus these methods can become very ineffective in solving hyperbolic partial differential equations, for example. And sixth order is considered to have a sufficiently restrictive stability region that it is not used in BDF codes such as LSODE (Reference 7).

Explicit methods of high order (8-12) have been routinely used for nonstiff problems to produce efficient solutions of high accuracy. Thus restriction to order two for stiff oscillatory problems and maximum order of five under optimal circumstances has not been accepted as representing true limitations, and efforts have been made to find more efficient stiff methods. Implicit Runge-Kutta methods were among the first to be developed. High-order methods with high stability have been derived, but the principal drawback has been the amount of work required to solve the systems of nonlinear equations for computation of the stages. In general this is proportional to $(Ms)^3$, where M is the number of equations in the system and s is the number of stages, which depends on both the order and class of the method. For example, with Gauss methods the order is 2s, for Radau methods 2s-1, and for Lobatto 2s-2. Singly diagonally implicit Runge-Kutta (SDIRK) methods provide a significant reduction in work by enabling separate solution for each stage, a savings of a factor of $s^2$. But analysis showed that with the stiff problems for which they were intended, the low stage order of the methods reduced the observed order of the convergence (see for example Reference 9). Thus singly implicit Runge-Kutta (SIRK) methods were developed by Burrage (Reference 10) and Butcher (Reference 11, see also Reference 9), which require repeated linear transformations to convert back and forth to a lower triangular system with a constant diagonal to be solved. The transformations are proportion to $M^2$ and so require less work for large systems. The code STRIDE (Reference 12) utilizes this approach, but its efficiency is seriously reduced by the amount of transformation work that is necessary. Butcher and Cash (Reference 13) have recently described an approach that reduces this transformation work somewhat.

The class of general linear methods was formulated by Burrage and Butcher (Reference 14, see also Reference 9) to include linear multistep methods (such as BDF methods), Runge-Kutta methods, and a number of other methods that had been invented over the years, within one theoretical formulation that would also include methods yet to be developed. A number of subfamilies of methods of this more general family have been studied recently. Blended multistep methods were developed by Skeel and Kong (Reference 15). SECDER (Reference 16) implements second derivative multistep methods studied by Enright (Reference 17). Cash first found extended BDF methods (Reference 18) with A-stability through fourth order and A($\alpha$)-stability through ninth order, and then (Reference 19) produced a much more efficient modification. Cash and Considine (Reference 20) have now developed codes based on these modified extended backwards differentiation formula (MEBDF) methods. Their code MEBDF uses variable order and variable step size with methods of orders 2-8. Hairer and Wanner (Reference 2) describe multistep Runge-Kutta methods including multistep collocation methods and multistep methods of Radau type in particular. Jackiewicz and Tracogna (Reference 21) and Tracogna (Reference 22) have described two-step Runge-Kutta methods. Butcher (Reference 23) proposed diagonally implicit multistage integration methods (DIMSIMs). A class of diagonally implicit single eigenvalue methods (DIMSEMs), influenced by Butcher's DIMSIMs, was investigated by Enenkel (Reference 24) and Enenkel and Jackson (Reference 25). At this time no stiff solvers based on methods that are A-stable beyond second order have achieved widespread acceptance as standard recommended codes. Because of their high stage order and certain other desirable design features, it is hoped that such a standard stiff solver may be developed based on implicit DIMSIMs.

Butcher's initial paper on DIMSIMs in 1992 (Reference 23) laid out the essential elements of a new family of general linear methods. His stated purpose was to overcome the glaring weaknesses of existing methods, that is, lack of A-stability for high-order linear

multistep methods and low stage order and high implementation costs for A-stable implicit Runge-Kutta methods. These methods are diagonally implicit and hence have computational complexity properties similar to diagonally implicit Runge-Kutta (DIRK) methods, but utilize additional parameters generated through the general linear design to overcome the stage order (and hence stiff order) limitations of DIRK methods. Explicit DIMSIMs are not technically "diagonally implicit," since the diagonal is actually 0, but have an advantage over Runge-Kutta methods in that the order barriers for explicit Runge-Kutta methods do not apply, and p-stage methods of order p do indeed exist for all positive integers p. The original concept called for stage order q to equal the number of internal stages s, the number of external stages r, and the order p. In a subsequent paper with Jackiewicz (Reference 26) the family of DIMSIMs was extended to include adjacent methods for which s + 1 = r = q, p = q or q + 1, s = r + 1 = q, p = q or q + 1, and s = r = q, p = q + 1. A significant step toward practical utilization was taken with another paper by Butcher and Jackiewicz (Reference 27) that lays out techniques for error estimation, interpolation, and step-size changing. Jackiewicz, Vermiglio, and Zennaro (Reference 28) devised an alternative step-size changing strategy and showed how incorporation of an additional external stage could provide a satisfactory continuous method. In a separate paper (Reference 29) they also showed that there exist explicit DIMSIMs with regularity properties not possessed by explicit Runge-Kutta methods. Butcher, Chartier, and Jackiewicz, in an unpublished manuscript, "Nordsieck representation of DIMSIMs," recently proposed an alternative representation of DIMSIMs with promise of simplifying analysis and implementation. Both explicit and implicit DIMSIMs up to the order 8 have now been found with appropriate stability properties and were announced by Butcher, Jackiewicz, and Mittelmann (Reference 30), extending techniques described in earlier papers by Butcher and Jackiewicz (References 31 and 32). A wide range of implementation issues was resolved and second and fifth order explicit DIMSIM computer codes were developed and successfully tested in my previous report (Reference 1).

## A QUICK OVERVIEW OF DIMSIMS

To make this report as self-contained as is practical, a survey of some DIMSIM results that are important for orientation is provided. A more extensive discussion is provided in (Reference 1). We consider the initial value problem of dimension M:

$$\frac{dy}{dt} = f(t,y),\ \ y(t_0) = y_0,\ \ t \in [t_0, T]. \tag{1}$$

We define matrices A, U, B and V such that A is s x s, V is r x r, U is s x r, and B is r x s. Let Y be composed of the s internal stages, F be composed of the s stage derivatives ($F_j = f(t_n + c_j h, Y_j)$) and $y^{[n]}$ be composed of the r external stage values. Then if h is the step size, the solution advances one step through the relationships:

$$Y_i = h\sum_{j=1}^{s} a_{ij}F_j + \sum_{j=1}^{r} u_{ij}y_j^{[n-1]}$$
$$y_i^{[n]} = h\sum_{j=1}^{s} b_{ij}F_j + \sum_{j=1}^{r} v_{ij}y_j^{[n-1]}. \tag{2}$$

The external stages are defined through Taylor expansion so that if

$$y_i^{[n-1]} = \sum_{j=0}^{p} \alpha_{ij} h^j y^{(j)}(t_{n-1}) + O\left(h^{p+1}\right), \tag{3}$$

then we must have, for some constants $\alpha_{ij}$,

$$y_i^{[n]} = \sum_{j=0}^{p} \alpha_{ij} h^j y^{(j)}(t_n) + O\left(h^{p+1}\right), \tag{4}$$

for a method of order p. The s values $c_i$ are chosen initially and other method parameters are then determined in a way to produce high stage order, that is, so that:

$$Y_i = y(t_{n-1} + hc_i) + O\left(h^{q+1}\right), \quad i = 1,2,...,s. \tag{5}$$

where q is defined to be the stage order. We will restrict ourselves in this report to methods for which p = q = r =s.

The conditions of Equation 2—Equation 5 may be re-expressed in a convenient form as follows. Let W be the r x (p+1) matrix of $\alpha_{ij}$ values, and we denote the vector consisting of the kth column of W as $\alpha_{k-1}$. Let Z be a vector with element $Z_j = z^{j-1}$. Then define w(z) = WZ. Furthermore we may define $e^{cz}$ as the vector

$$e^{cz} = \begin{bmatrix} e^{c_1 z} \\ e^{c_2 z} \\ \vdots \\ e^{c_s z} \end{bmatrix}. \tag{6}$$

Then the following theorem of Butcher (Reference 23) may be used in determining the coefficients of the method:

**Theorem 1**: A DIMSIM (Equation 2) has order p and stage order p if and only if

$$\begin{aligned} e^{cz} &= zAe^{cz} + Uw(z) + O\left(z^{p+1}\right), \\ e^{z}w(z) &= zBe^{cz} + Vw(z) + O\left(z^{p+1}\right). \end{aligned} \tag{7}$$

The coefficient methods are expressed in a tableau consisting of the matrices A, U, B, and V, arranged as follows:

$$\begin{bmatrix} A & U \\ B & V \end{bmatrix}. \tag{8}$$

DIMSIMs were classed by Butcher (Reference 23) according to the form of A. A Type 3 method has A = 0. Type 2 and Type 4 methods have a single diagonal value, with Type 4 methods having a diagonal A and Type 2 methods a lower triangular A. Type 1 methods have a strictly lower triangular A. And V is chosen to be of rank 1, $V = ev^T$, and v is chosen such that $v^Te = 1$.

It turns out that a restriction on the matrix elements of B provides assurance that order and stage order conditions are met, according to the following theorem derived by Butcher (Reference 23).

**Theorem 2**. Let r = s = p, Ve = e. Then the DIMSIM

$$\begin{bmatrix} A & I \\ B & V \end{bmatrix}$$

is of order p and stage order q = p if and only if $B = B_0 - AB_1 - VB_2 + VA$, where

$$B_{0,i,j} = \frac{\int_0^{1+c_i} \phi_j(x)dx}{\phi_j(c_j)},$$

$$B_{1,i,j} = \frac{\phi_j(1+c_i)}{\phi_j(c_j)}, \tag{9}$$

$$B_{2,i,j} = \frac{\int_0^{c_i} \phi_j(x)dx}{\phi_j(c_j)},$$

and where

$$\phi_j(x) = \prod_{k \neq j}(x - c_k). \tag{10}$$

Note that using this theorem eliminates the elements of B as free parameters when deriving methods. It also has the following immediate corollary, since for any specified vector c and matrices A and V a construction may be completed, leaving stability as the principal issue in deriving new methods.

**Corollary**: For each integer p > 1, DIMSIMs of order p and stage order q = p exist for s = r = p, U = I, where s is the number of internal stages and r is the length of the external stage vector.

The stability matrix for a DIMSIM is (Reference 23)

$$M(z) = V + zB(I - zA)^{-1}. \tag{11}$$

This is easily seen from the method (Equation 2) using the standard test problem $y' = \lambda y$, $y(t_0) = y_0$. We solve the first method equation for Y and, noting that $F(Y) = \lambda Y$, we obtain, setting $z = h\lambda$,

$$Y = (I - zA)^{-1} y^{[n-1]}.$$

Then we may substitute this expression in the second equation to obtain

$$y^{[n]} = \left(zB(I - zA)^{-1} + V\right)y^{[n-1]}.$$

Thus the region of absolute stability of a DIMSIM is the region

$$A = \{z : w \in \sigma(M(z)) \Rightarrow w < 1\}.$$

If A includes the entire open left half plane, the method is called A-stable. We also define the associated stability polynomial

$$p(w,z) = \det(wI - M(z)), \tag{12}$$

A method may typically be verified to be A-stable either by using the Schur criterion (see, for example, Lambert (Reference 33)), or by reducing the stability polynomial to a familiar form associated with a Runge-Kutta method known to be A-stable.

Nordsieck techniques are used extensively in the development and implementation of DIMSIMs. These use a vector of derivatives scaled with the step size and usually also with a factor of 1/(j-1)! where j is the component number of the vector. Here we omit the exra factor, which can be readily provided through multiplication by a constant diagonal matrix

$$S = \text{diag}(1/0!,1/1!,1/2!,...,1/p!),$$

and use the term "Nordsieck vector" to refer to a closely related vector that frequently appears here. We define the Nordsieck vector of length p + 1 as

$$\tilde{y}(t_n) = \begin{bmatrix} y(t_n) \\ hy'(t_n) \\ \vdots \\ h^p y^{(p)}(t_n) \end{bmatrix}. \tag{13}$$

We also use the term to refer to the computed approximation to the exact Nordsieck vector.

Two matrices are defined which are used to relate the Nordsieck vector to the internal and external stages of the method. Let $F\left(Y^{[n]}\right)$ be a vector with kth component $f\left(Y_k^{[n]}\right)$. Then we can find matrices $\tilde{B}$ and $\tilde{V}$ such that

$$\tilde{y}(t_n) = h\tilde{B}F\left(Y^{[n]}\right) + \tilde{V}y^{[n-1]} + O\left(h^{p+1}\right). \tag{14}$$

These matrices can be calculated using the following theorem, announced in Butcher (Reference 23) and proven rigorously in Butcher and Jackiewicz (Reference 27). We first define $\tilde{z}$ as a vector of length p + 1,

$$\tilde{z} = \begin{bmatrix} 1 \\ z \\ \vdots \\ z^p \end{bmatrix}. \tag{15}$$

**Theorem 3** (Butcher and Jackiewicz, Reference 27)**:** Assume that the method (Equation 2) has order p and stage order q = p or q = p - 1. Then the approximations (Equation 12) are correct to $O\left(h^{p+1}\right)$ if and only if

$$e^z \tilde{z} = z\tilde{B}e^{cz} + \tilde{V}w(z) + O\left(z^{p+1}\right). \tag{16}$$

The Nordsick vector provides an output value and a convenient interpolant, as well as a predictor for implicit methods. To change step size we also define the diagonal matrix

$$D = diag\left(1, \delta, \delta^2, ..., \delta^p\right), \tag{17}$$

where $\delta = \delta_n = \frac{h_n}{h_{n-1}}$. Then after using the expression for the Nordsieck vector at $t_{n-1}$, that we can rescale to accommodate a change of step size by using the formula

$$\hat{y}^{[n-1]} = h_{n-1} WD\tilde{B}F\left(Y^{[n-1]}\right) + WD\tilde{V}\hat{y}^{[n-2]}. \tag{18}$$

We now have a modified numerical process, as follows:

$$\begin{aligned} Y^{[n]} &= h_n AF\left(Y^{[n]}\right) + \hat{y}^{[n-1]}, \\ y^{[n]} &= h_n BF\left(Y^{[n]}\right) + V\hat{y}^{[n-1]}. \end{aligned} \tag{19}$$

## CALCULATING INTERNAL STAGES WITH IMPLICIT DIMSIMS

The description "diagonally implicit" comes from the form of the matrix As. If we restrict A to be lower triangular with a constant along the diagonal, the complexity of the solution of the system of nonlinear equations determining Y in Equation 2 is greatly reduced. If A is dense and a standard Gaussian elimination approach is used in a modified Newton method, the arithmetic complexity is $O((Ns)^3)$, or $O(N^3s^3)$, where s is the number of stages of the method and N is the dimension of the initial value problem. Simply requiring A to be lower triangular separates the system of ns simultaneous nonlinear equations into s systems of N simultaneous equations to be solved in sequence, each with arithmetic complexity $O(N^3)$ for total complexity $O(sN^3)$, a reduction by a factor of $O(s^2)$.

The advantage of having a single value along the diagonal may be seen from a closer examination of the solution process. The Newton iteration to solve the nonlinear system $g(x) = 0$ takes the form

$$x_{n+1} = x_n - J(x_n)^{-1} g(x_n), \tag{20}$$

where J is the Jacobian of g. Of course the inverse of the Jacobian is not actually calculated and instead a technique such as Gaussian elimination is utilized, and the process followed is to calculate a correction:

$$\begin{aligned} J(x_n)\delta_{n+1} &= -g(x_n), \\ x_{n+1} &= x_n + \delta_{n+1}. \end{aligned} \tag{21}$$

An LU or PLU factorization of $J(x_n)$ is called for here at each iteration, which is $O(N^3)$, where N is the size of the system, and this is the most time-consuming step. In practice a new Jacobian is evaluated and G is factored only when convergence seems too slow. In the case of DIMSIMs, the equation for Y takes the form

$$Y_j = h\sum_{k=1}^{j-1} a_{jk} f(t_{n-1} + c_k h, Y_k) + h a_{jj} f(t_{n-1} + c_j h, Y_j) + y_j^{[n-1]}, \tag{22}$$

with U = I as is usually the case. Then we are solving an equation of the form $g_j(Y_j) = 0$, where $g_j$ takes the form:

$$g_j(Y_j) = Y_j - h a_{jj} f(t_{n-1} + c_j h, Y_j) - h\sum_{k=1}^{j-1} a_{jk} f(t_{n-1} + c_k h, Y_k) - y_j^{[n-1]}. \tag{23}$$

All the dependency on $Y_j$ is contained in the first two terms and so only these terms affect the calculation of the jacobian. Now if all diagonal elements $a_{jj}$ are the same the jacobians $J_j$ will vary from stage to stage only with the solution, and new jacobians and their LU decompositions will only have to be computed when convergence seems too slow, which should be relatively rare for small step size h. This can result in substantial savings.

A similar reduction follows from utilization of a nonsingular A (perhaps dense) with a single eigenvalue that may be transformed into a diagonally implicit method, and this has been applied in the implicit Runge-Kutta code STRIDE developed by Burrage, Butcher and Chipman (Reference 12). Because of the work required for linear transformations, this is to be avoided if possible due to the number of matrix multiplications that become necessary. Of course the low stage order of SDIRK methods, which drastically reduces the observed order of the method for stiff equations to second order, made this alternative approach attractive for use in STRIDE.

## INTERPOLATION AND ERROR ESTIMATION

An efficient adaptive solver requires error estimation to enable step-size control and interpolation to free the choice of step size from the selection of output points. A brief summary of results developed more extensively and tested in (Reference 1) is provided here as an aid to the reader.

A family of DIMSIM interpolants was first described in Butcher and Jackiewicz (Reference 27), and, for those DIMSIMs for which they exist, they provide continuous interpolation of maximal order with no derivative function evaluation or system solution cost. These interpolants are of the form

$$\eta(t_{n-1}+\theta h_n) = h_n\beta_0(\theta)F\left(Y^{[n]}\right)+\gamma_0(\theta)\hat{y}^{[n-1]}. \tag{24}$$

Here we define $\beta_0(\theta)=[\beta_{01}(\theta),\beta_{02}(\theta),...,\beta_{0s}(\theta)]$ and $\gamma_0(\theta)=[\gamma_{01}(\theta),\gamma_{02}(\theta),...,\gamma_{0r}(\theta)]$, where the components are polynomials of degree p (or lower if certain coefficients become set to zero). In order for compatibility with the equation for the first component of the Nordsieck vector, $\beta_0(1)$ and $\gamma_0(1)$ must be equal to the first rows of $\tilde{B}$ and $\tilde{V}$, respectively. The following theorem derived by Butcher and Jackiewicz (Reference 27) is used to derive these interpolants.

**Theorem 4** (Butcher and Jackiewicz): If a DIMSIM has order p and stage order q = p or q = p - 1, then $\eta$ approximates y with uniform order p if and only if

$$z\beta_0(\theta)e^{cz}+\gamma_0(\theta)w(z) = e^{\theta z}+O\left(z^{p+1}\right),\ \theta\in(0,1], \tag{25}$$

and w(z) is as defined above. Moreover, the interpolant $\eta$ is continuous on the whole interval of integration if and only if

$$\begin{aligned}
&\beta_0(0) = 0, \\
&\gamma_0(0)WD\tilde{B} = \beta_0(1), \\
&\gamma_0(0)WD\tilde{V} = \gamma_0(1).
\end{aligned} \tag{26}$$

Very few Type 2 methods that have otherwise been found provide these interpolants, and it was shown in the previous report (Reference 1) that the Nordsieck vector may always be utilized for this purpose, even with methods for which the Butcher-Jackiewicz interpolant does not exist.

Butcher and Jackiewicz (Reference 27) also derived error estimation results. We note that the following general definition for local discretization error of the external stages applies to all DIMSIMs. The idea is to identify what is to be called the local discretization error of the external stages with the term of order $h^{p+1}$ in the difference between the exact external stages at $t_n$ and the calculated external stages, assuming that an exact Nordsieck vector is used initially at $t_{n-1}$ and that stage and order conditions are met.

**Definition**: The local discretization error $le_i(t_n)$ of the ith external stage $y_i^{[n]}$ of the method of Equation 2 at the point $t_n$ is given by

$$le_i(t_n) = \sum_{k=0}^{p} \alpha_{ik} y^{(k)}(t_n) h^k - h \sum_{k=1}^{s} b_{ik} f\left(t_{n-1} + c_k h, Y_k^{[n]}\right) - \sum_{j=1}^{r} \sum_{k=0}^{p} v_{ij} \alpha_{jk} y^{(k)}(t_{n-1}) h^k, \tag{27}$$

where

$$Y_k^{[n]} = h \sum_{j=1}^{s} a_{ij} f\left(t_{n-1} + c_j h, Y_j^{[n]}\right) + \sum_{j=1}^{r} \sum_{i=0}^{p} u_{kj} \alpha_{ji} y^{(i)}(t_{n-1}) h^i, \quad k = 1,2,...,s. \tag{28}$$

For the case p = q = r = s, U = I, and assuming that the local condition for stage order holds, we make these substitutions in Equation 27 to obtain the simplified expression

$$\begin{aligned}
le_i(t_n) = &\sum_{k=0}^{p} \alpha_{ik} y^{(k)}(t_n) h^k - h \sum_{j=1}^{p} b_{ij} f\left(t_{n-1} + c_j h, y\left(t_{n-1} + c_j h\right) + \xi_j y^{(p+1)} h^{p+1}\right) \\
&- \sum_{j=1}^{p} \sum_{k=0}^{p} v_{ij} \alpha_{jk} y^{(k)}(t_{n-1}) h^k +, \quad i = 1,...,p\,.
\end{aligned} \tag{29}$$

In any case, the vector of values $le_i(t_n)$ we designate as the local discretization error of the external stages.

The following result derived by Butcher and Jackiewicz (Reference 27) shows that the local truncation error is proportional to the (p + 1)st derivative of the solution vector.

**Theorem 5** The local discretization error $le(t_n)$ of the external stages of Equation 2 at the point $t_n$ is given by

$$le(x_n) = \varphi_p y^{(p+1)}(x_{n-1})h^{p+1} + O(h^{p+2}), \tag{30}$$

where

$$\varphi_p = \sum_{k=1}^{p+1} \frac{\alpha_{p+1-k}}{k!} - \frac{Bc^p}{p!}. \tag{31}$$

To obtain a variable step estimator, we define $h_n$ as $t_n$-$t_{n-1}$ and $\delta = h_n/h_{n-1}$, and we seek vectors $\beta = \beta(\delta)$ and $\gamma = \gamma(\delta)$ such that

$$v^T \varphi_p h_n^{p+1} y^{(p+1)}(t_{n-1}) = h_n \beta^T(\delta) F\left(Y^{[n]}\right) + \gamma^T(\delta)\tilde{y}^{[n-1]} + O(h_n^{p+2}). \tag{32}$$

It should be noted in the following modification (Reference 1) of a theorem by Butcher and Jackiewicz (Reference 27) that the error formula includes the effect of rescaling and that error estimation is not carried out by simply using a fixed step formula with a rescaled external stage vector as is done in interpolation. Also note that variable step size does not apply before the second step. The validity of the local stage order condition is assumed. Define a matrix

$$G = \left[0, e, c, \ldots, \frac{c^p}{p!}\right], \tag{33}$$

and a matrix

$$\tilde{T} = \left[\tilde{t}_0, \tilde{t}_1, \ldots, \tilde{t}_{p+1}\right] = \begin{bmatrix} 1 & -1 & \frac{1}{2} & \cdots & \frac{(-1)^{p+1}}{(p+1)!} \\ 0 & 1 & -1 & \cdots & \frac{(-1)^{p}}{p!} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{34}$$

**Theorem 6:** Assume that the method in Equation 2 with p = q = r = s, U = I, is implemented in variable step size mode using the Nordsieck technique and that V is rank one, Ve = e. Then

$$v^T le(t_n) = h_n \beta^T(\delta) F\left(Y^{[n]}\right) + \gamma^T(\delta)\tilde{y}^{[n-1]} + O\left(h_n^{p+2}\right), \tag{35}$$

if $\beta = \beta(\delta)$ and $\gamma = \gamma(\delta)$ satisfy the system of equations

$$\begin{aligned} &1)\ \gamma^T WD\tilde{V} = 0 \\ &2)\gamma^T e = 0 \\ &3)\left(\beta^T + \tfrac{1}{\delta}\gamma^T WD\tilde{B}\right)e = 0 \\ &4)\beta^T \frac{c^{i-1}}{(i-1)!} + \frac{1}{\delta^i}\gamma^T WD\tilde{B}G\tilde{t}_i = 0, \quad i = 2,3,\ldots,p \\ &5)\beta^T \frac{c^p}{p!} + \frac{1}{\delta^{p+1}}\gamma^T WD\tilde{B}G\tilde{t}_{p+1} = v^T\varphi_p. \end{aligned} \tag{36}$$

For the frequently occurring case where the first row of $\tilde{V}$ is $v^T$ and the other rows are 0, the first condition simplifies to $\gamma^T e = 0$, eliminating it as a separate condition.

The following theorem was derived in Reference 1 for the intial step not covered by the Butcher-Jackiewicz theory and also may be used to produce a suitable starting step size.

**Theorem 7 (Initial Step Error Estimate):** If the solution y(x) to the problem in Equation 1 is sufficiently smooth and the starting vector is calculated by a method correct up to $O(h^{p+2})$, then the error in $y_1$, the approximation to $y(x_1)$ calculated using the method in Equation 2 is

$$lte = y(t_1) - y_1 = \left( \frac{1}{(p+1)!} - \sum_{j=1}^{p} \tilde{B}_{1j} \frac{c_j^p}{p!} \right) \left( h\beta^T F\left(Y^{[1]}\right) + \gamma^T y^{[0]} \right) + O\left(h^{p+2}\right), \tag{37}$$

provided vectors $\beta$ and $\gamma$ meet the following conditions:

$$\begin{aligned}
&1) \gamma^T e = 0 \\
&2) \beta^T e + \gamma^T \alpha_1 = 0 \\
&3) \sum_{j=1}^{p} \beta_j \frac{c_j^{k-1}}{(k-1)!} + \gamma^T \alpha_k = 0, k = 2,\ldots,p \\
&4) \sum_{j=1}^{p} \frac{\beta_j c_j^p}{p!} = 1
\end{aligned} \tag{38}$$

## SOME IMPLICIT DIMSIMS

### A TYPE 2 METHOD WITH BUTCHER-JACKIEWICZ-TYPE INTERPOLANT

Butcher's second order Type 2 method described in (Reference 23), although A-stable, was shown (Reference 1) not to have a Butcher-Jackiewicz-type interpolant. A method with a Butcher-Jackiewicz-type interpolant was found, however, and this process is now described.

A MATHEMATICA™ program was created which incorporated the interpolant conditions of Theorem 4 in the search and also tested for A stability using the Schur criterion (Reference 33). Examination of results led to discovery of an A-stable Type 2 method with c = [1/2,1] and with Butcher tableau

$$\begin{bmatrix}
\frac{5}{8} & 0 & 1 & 0 \\
\frac{11}{16} & \frac{5}{8} & 0 & 1 \\
\frac{71}{64} & \frac{1}{32} & \frac{1}{4} & \frac{3}{4} \\
\frac{107}{64} & -\frac{23}{32} & \frac{1}{4} & \frac{3}{4}
\end{bmatrix}.$$

For this method,

$$W = \begin{bmatrix} 1 & -\frac{1}{8} & -\frac{3}{16} \\ 1 & -\frac{5}{16} & -\frac{15}{32} \end{bmatrix},$$

and

$$\tilde{B} = \begin{bmatrix} \frac{47}{64} & \frac{17}{32} \\ 0 & 1 \\ -2 & 2 \end{bmatrix}, \quad \tilde{V} = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

For a Butcher-Jackiewicz-type interpolant of the form of Equation 24, we obtain

$$\beta_0 = \begin{bmatrix} \frac{3\theta}{2} - \frac{49\theta^2}{64} \\ \frac{17\theta^2}{32} \end{bmatrix}$$

and

$$\gamma_0 = \begin{bmatrix} \frac{5}{3} - \frac{8\theta}{3} + \frac{5\theta^2}{4} \\ -\frac{2}{3} + \frac{8\theta}{3} - \frac{5\theta^2}{4} \end{bmatrix}.$$

Using the conditions of Theorem 6 a Butcher-Jackiewicz-type error estimate was found of the form

$$est = \varsigma(\delta)\left( h_n \beta^T F\left( Y^{[n]} \right) + \gamma^T \tilde{y}^{[n-1]} \right),$$

where $\varsigma(\delta) = \frac{223\delta}{3+7\delta}$, $\beta = \begin{bmatrix} \frac{1}{96} \\ -\frac{1}{48} \end{bmatrix}$, and $\gamma = \begin{bmatrix} \frac{1}{18} \\ -\frac{1}{18} \end{bmatrix}$. For the first step, the conditions of Theorem 7 yield

$$est = h\beta_i^T F\left(Y^{[1]}\right) + \gamma_i^T y^{[0]},$$

where $\beta_i = \frac{293}{1344}\begin{bmatrix} 1 \\ -2 \end{bmatrix}$ and $\gamma_i = \frac{293}{252}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$. The error constant for this method is -223/768, which is approximately 30% larger than the error constant for the second order BDF method.

## L-STABLE TYPE 2 SECOND ORDER METHODS

The condition of L-stability (see for example Reference 2) requires that the method be A-stable, typically determined using the Schur criterion, and also that the eigenvalues of the stability matrix approach 0 as z approaches infinity. For the convenient c values of [0,1] this condition was evaluated. Note that if we specify the method matrix A as

$$A = \begin{bmatrix} \lambda & 0 \\ a & \lambda \end{bmatrix},$$

and $V = ev^T$ where $v^T = [v_1, v_2] = [v, 1-v]$, the free parameters are v, $\lambda$, and a. The stability polynomial is of the form

$$p_2(z)w^2 + p_1(z)w + p_0(z) = 0.$$

In general polynomials $p_2$, $p_1$, and $p_0$ are all of degree 2. We note that the quadratic formula yields eigenvalues of the form

$$w = \frac{-p_1(z) \pm \sqrt{p_1^2(z) - 4p_2(z)p_0(z)}}{2p_2(z)}.$$

Then the L-stability condition will be met if the degree of $p_2$ is greater than the degree of $p_1$ and $p_0$. This can be assured by calculating these coefficients (using MATHEMATICA™) and then finding conditions which make the coefficients of $z^2$ in $p_1$ and $p_0$ equal to 0. These were found to yield the equations

$$-\lambda + 2\lambda^2 + 2v - av - 2\lambda v = 0$$

and

$$3\lambda - 4\lambda^2 - av + 2\lambda v = 0.$$

Solving for v and a in terms of $\lambda$ we obtain

$$v = \frac{\lambda(3\lambda - 2)}{2\lambda - 1}, \quad a = \frac{-2\lambda^2 + 6\lambda - 3}{3\lambda - 2}.$$

We use these values with MATHEMATICA™ to obtain a stability polynomial and apply the Schur criterion along the imaginary axis. We obtain a number of expressions which evaluate to 0 or positive numbers, and three conditions on the parameter $\lambda$ for a method to be A-stable. These are

$$-1 + 8\lambda - 12\lambda^2 + 16\lambda^3 - 4\lambda^4 > 0,$$

which is satisfied for $\lambda \in (.1533, 3.2609)$,

$$-1 + 4\lambda - 8\lambda^3 + 6\lambda^4 > 0,$$

which is satisfied for $\lambda > .2872$, and

$$-11 + 64\lambda - 100\lambda^2 + 64\lambda^3 - 12\lambda^4 > 0,$$

which is satisfied for $\lambda \in (.261, 3.2449)$. The intersection of these sets is the interval (.2872,3.2449). The error coefficient is given by the formula

$$v^T \varphi_2 = \frac{5 - 24\lambda + 18\lambda^2}{12}.$$

This has a zero near 1.08 so we choose $\lambda$ = 13/12 to obtain an L-stable method with an error constant of 1/96. We do not attempt to use the exact zero so that the stage order remains equal to the order and also so that an error estimate exists. The full method, along

$$\begin{bmatrix} A & U \\ B & V \end{bmatrix} = \begin{bmatrix} \frac{13}{12} & 0 & 1 & 0 \\ \frac{83}{90} & \frac{13}{12} & 0 & 1 \\ \frac{811}{480} & -\frac{65}{96} & \frac{65}{56} & -\frac{9}{56} \\ \frac{1091}{480} & -\frac{1703}{1440} & \frac{65}{56} & -\frac{9}{56} \end{bmatrix}.$$

The matrix W for computing the starting vector is given by

$$W = \begin{bmatrix} 1 & -\frac{13}{12} & 0 \\ 1 & -\frac{181}{180} & -\frac{7}{12} \end{bmatrix}.$$

For computation of the Nordsieck vector and rescaling we have

$$\tilde{B} = \begin{bmatrix} \frac{811}{480} & \frac{13}{32} \\ 0 & 1 \\ -1 & 1 \end{bmatrix}, \quad \tilde{V} = \begin{bmatrix} \frac{65}{56} & -\frac{9}{56} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

A Butcher-Jackiewicz-type interpolant does not exist so a Nordsieck or continuous Nordsieck interpolant should be used. Finally, the error estimation vectors are given by

$$\beta = \frac{1}{48(1+\delta)}\left(\frac{811-6734\delta-3645\delta^2+3815\delta^3}{75\delta^2}, \delta\right)^T,$$

and

$$\gamma = \frac{1}{280\delta^2(1+\delta)}\left(811+811\delta+280\delta^2+270\delta^3, -811-811\delta+10\delta^3\right)^T.$$

For the first step error estimate we find that

$$\beta_i = \left[\frac{119}{1440}, -\frac{7}{96}\right]^T, \quad \gamma_i = \left[\frac{1}{8}, -\frac{1}{8}\right]^T.$$

Interestingly, the first step error constant = -7/96.

A search of other second order A-stable DIMSIMs with small error constants revealed the method a = 6/5, $\lambda$ = 3/10, $v$ = 4/5, $c_1$ = 0, $c_2$ = 1 with error constant -13/300. Further exploration turned up the DIMSIM a = 4/3, $v$ = 3/4, $\lambda$ = 1/4, $c_1$ = 0, $c_2$ = 1, with error constant of -1/48, 4 times better than that of the trapezoidal rule. The small error constants should make these competitive even with two internal stages to evaluate.

## SECOND ORDER METHODS WITH A-STABLE FASAL IMPLEMENTATIONS

First approximately same as last (FASAL) DIMSIM implementations were first studied in the previous report (Reference 1). For methods in which $c_1$ = 0 and $c_p$ = 1, because of the high stage order of DIMSIMs, there is no sacrifice in the order of accuracy through using the last stage derivative of the previous step in place of the first stage derivative of the current step, a reduction of one full stage evaluation. Because of the reduction in work with FASAL implementations, a study was made to determine what second order DIMSIMs have FASAL implementations that are L-stable or at least A-stable. These methods must have $c_1$ = 0, $c_2$ = 1, leaving only three free parameters, a, $\lambda$, and v, where A is given by

$$A = \begin{bmatrix} \lambda & 0 \\ a & \lambda \end{bmatrix}$$

and $V = ev^T$, where v = [v, 1-v]. The form of the matrix B is given by Theorem 2, and for a second order method with p = q = r = s = 2 this requires that $B = B_0 - AB_1 - VB_2 + VA$, where

$$B_0 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 2 \end{bmatrix}, \; B_1 = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}, \; B_2 = \begin{bmatrix} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Then we determine the stability matrix for the FASAL implementation using the techniques previously developed (Reference 1). We thus first produce

$$\hat{A} = \begin{bmatrix} 0 & 0 \\ a & \lambda \end{bmatrix}, \; \hat{B} = \begin{bmatrix} a & \lambda \\ & B \end{bmatrix}, \; \hat{U} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ and } \hat{V} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & & \\ 0 & & V \end{bmatrix}.$$

The 3x3 stability matrix is then given by

$$\hat{M} = z\hat{B}\left(I - z\hat{A}\right)^{-1}\hat{U} + \hat{V}.$$

We now form the stability polynomial

$$p(w,z) = \det\left(\hat{M} - wI\right).$$

This gives the expression

$$p(w,z) = \frac{\left(\begin{array}{l} 2w^2 - 2w^3 + vz - 2\lambda vz - wz + 2\lambda wz - 2vwz + \\ 4\lambda vwz + 3w^2z - 4\lambda w^2z + vw^2z - 2\lambda vw^2z + 2\lambda w^3z \end{array}\right)}{2(-1+\lambda z)}$$

In order for a method to be L-stable, the coefficient of the highest power of w in the numerator must be a higher degree polynomial than any of the coefficients of other powers of w. Thus we must have

$w^0$: $v(1-2\lambda) = 0,$

$w^1$: $-1 + 2\lambda - 2v + 4\lambda v = 0,$

$w^2$: $3 - 4\lambda + v - 2\lambda v = 0.$

From the 0th degree condition, we learn that either v = 0 or $\lambda$ = 1/2. But if v = 0, the second equation yields $\lambda$ = 1/2, while the third yields $\lambda$ = 3/4. So we must have $\lambda$ = 1/2. The second equation will then be automatically satisfied, but the third yields the contradiction 1 = 0. Thus there are no L-stable p = q = s = r = 2 (0,1) methods and we limit our search to A-stable methods.

In order for a FASAL implementation to be A-stable, all the roots w of p must have $|w| \le 1$ for $\mathrm{Re}(z) \le 0$, and the roots with |w| = 1 must be simple. As noted by Burrage (Reference 34), the work of Chartier concerning parallel one-block methods (Reference 35) may be extended to include all general linear methods. If A has a spectrum with positive real part (the diagonal is typically chosen as positive), then $\rho(M(z))$ will be analytic in the left half plane. This enables application of the maximum modulus principle to conclude that in the left half plane, $\rho(M(z))$ will take its largest value on the imaginary axis. Thus we may restrict ourselves to the imaginary axis and let z = iy, where y is real. If all the roots w

have $|w| \leq 1$ along the imaginary axis and at least one point in the left half plane also satisfies this condition, then the region of absolute stability will include the entire left half plane and the method is A-stable.

To evaluate this condition along the imaginary axis, the Schur criterion is applied, since a polynomial is a Schur polynomial if all the roots have modulus less than 1 (see Reference 33). Because of the lengthy expressions that result, the work was done using MATHEMATICA™. We note that we may write the stability polynomial in the form

$$p(w) = g_3 w^3 + g_2 w^2 + g_1 w + g_0.$$

We then form

$$\hat{p}(w) = g_0^* w^3 + g_1^* w^2 + g_2^* w + g_3^*.$$

We have as condition 1 that $|\hat{p}(0)| > |p(0)|$. We then form

$$p_1(w) = \frac{1}{w}\left[\hat{p}(0)p(w) - p(0)\hat{p}(w)\right].$$

This is a second degree polynomial in w. If $p_1$ is a also, a Schur polynomial and condition 1 is met, then p is a Schur polynomial. We now may write $p_1$ in the form

$$p_1(w) = h_2 w^2 + h_1 w + h_0.$$

We then form

$$\hat{p}_1(w) = h_0^* w^2 + h_1^* w + h_2^*.$$

We have as condition 2 that $|\hat{p}_1(0)| > |p_1(0)|$. We then form

$$p_2(w) = \frac{1}{w}\left[\hat{p}_1(0)p_1(w) - p_1(0)\hat{p}_1(w)\right].$$

If $p_2$ is a Schur polynomial and conditions 1 and 2 are met, then p is a Schur polynomial. This is a linear equation that may be solved for w and leaves only condition 3, that $|\hat{p}_2(0)| > |p_2(0)|$.

Each of the three conditions is a rational function in y. We first examine the denominators. For condition 1 this is $4(1+\lambda^2y^2)$. This is clearly always positive. For condition 2 this is $16(1+\lambda^2y^2)^2$ also always positive. Finally, for condition 3 we obtain a denominator of $32(1+\lambda^2y^2)^4$. Thus all the denominators are positive and the numerators must be investigated. And these all have only even powers of y, so we simply determine requirements to ensure that these coefficients are all positive. We list them by Schur condition number and by degree in y:

Condition 1, 2nd Degree:

$$(-2\lambda + v - 2\lambda v)(-2\lambda - v + 2\lambda v) \geq 0$$

Condition 1, 0th Degree:

$$4 \geq 0$$

Condition 2, 4th Degree:

$$(-2\lambda - 3v + 6\lambda v)(2\lambda - 8\lambda^2 + 3v - 6\lambda v + 2v^2 - 8\lambda v^2 + 8\lambda^2 v^2) \geq 0$$

Condition 2, 2nd Degree:

$$4(-1 + 4\lambda + 4\lambda^2 - 2v + 8\lambda v - 8\lambda^2 v - 3v^2 + 12\lambda v^2 - 12\lambda^2 v^2) \geq 0$$

Condition 2, 0th Degree:

$$16 \geq 0$$

Condition 3, 8th Degree:

$$(-1 + 2\lambda)(1 + v)(2\lambda + v - 2lbv)(2\lambda - v + 2\lambda v)(-2\lambda - 3v + 6\lambda v)^2 \geq 0$$

Condtion 3, 6th Degree:

$$16(-1+2\lambda)\left(2\lambda^2+3\lambda v-8\lambda^2 v+2v^2-5\lambda v^2+2\lambda^2 v^2+3v^3-12\lambda v^3+12\lambda^2 v^3\right)\geq 0$$

Condition 3, 4th Degree:

$$16(-1+2\lambda)(1-3v)\geq 0$$

Condition 3, 2nd Degree:

$$0\geq 0$$

Condition 3, 0th Degree:

$$0\geq 0$$

We immediately note that the quantity a does not appear and hence stability is determined only by v and $\lambda$. There are a total of 6 nontrivial inequalities to satisfy. We begin with Condition 3, 4th Degree. Here either $\lambda\geq\frac{1}{2}$ and $v\leq\frac{1}{3}$ or $\lambda\leq\frac{1}{2}$ and $v\geq\frac{1}{3}$ must be true. We then examine Condition 3, 8th Degree. The square term must be always positive. The product $(2\lambda+v-2\lambda v)(2\lambda-v+2\lambda v)$ is the same as Condition 1, 2nd Degree. Then assuming this condition to be met, we must have $(-1+2\lambda)(1+v)\geq 0$. Then either $\lambda\geq\frac{1}{2}$ and $v\geq -1$ or $\lambda\leq\frac{1}{2}$ and $v\leq -1$. But since for $\lambda\leq\frac{1}{2}$ we cannot have simultaneously $v\leq -1$ and $v\geq\frac{1}{3}$, we find that we have the necessary conditions for A-stability of $\lambda\geq\frac{1}{2}$ and $v\in\left[-1,\frac{1}{3}\right]$. Careful analysis of the remaining inequalities shows that this is also a sufficient condition. Condition 1, 2nd Degree may be rewritten as

$$\left(-1-(-1+2\lambda)(1+v)\right)\left(-1+(-1+2\lambda)(-1+v)\right)\geq 0.$$

We now assume the requirements on $\lambda$ and v to note that both factors will be negative and thus this condition will be met. This also ensures that Condition 3, 8th Degree will be met. W rewrite Condition 2, 2nd Degree as

$$4\left[-(-1+2\lambda)\left[(-1+2\lambda)(1+\nu)(-1+3\nu)-4\right]+2\right]\geq 0$$

Again using our requirements on $\lambda$ and $\nu$, we find that -1 + 3v is nonpositive while 1 + v and -1 + 2$\lambda$ are nonnegative. Thus $(-1+2\lambda)(1+\nu)(-1+3\nu)\leq 0$, subtracting 4 makes this negative, and multiplying by the nonpositive quantity -(-1 + 2$\lambda$) makes the inequality true.

Condition 3, 6th Degree will be satisfied if

$$2\lambda^2+3\lambda\nu-8\lambda^2\nu+2\nu^2-5\lambda\nu^2+2\lambda^2\nu^2+3\nu^3-12\lambda\nu^3+12\lambda^2\nu^3\geq 0.$$

This may be rewritten in a more convenient form as

$$\tfrac{1}{3}\left[(-1+3\nu)\left(\tfrac{3}{2}(-1+2\lambda)\left(2(1+\nu)\left(\tfrac{1}{2}(-1+2\lambda)(-1+2\nu)-\tfrac{1}{2}\right)-1\right)-\tfrac{1}{2}\right)+1\right]\geq 0.$$

We examine this from the inside outward. Since -1 + 2v is nonpositive, the innermost expression must be nonpositive. Since it is multiplied by a nonnegative expression, the next nested expression must also be nonpositive. This too is multiplied by a nonnegative expression and so the next nested expression is also nonpositive. But -1 + 3v is nonpositve, and so the whole inequality becomes true.

The last inequality that must be satisfied is Condition 2, 4th Degree. The left-hand side consists of a product of two factors. The first, -2$\lambda$ - 3v + 6$\lambda$v, may be rewritten as

$$(-1+2\lambda)(-1+3\nu)-1<0$$

since -1 + 3v is nonpositive and -1 + 2$\lambda$ is nonnegative. The requirement for the other factor may be rewritten as

$$(-1+2\lambda)(1+\nu)(2(-1+2\lambda)(-1+\nu)-3)-1\leq 0$$

Again using the restrictions on v and $\lambda$ we find that the innermost expression is nonpositive and then is multiplied by nonnegative quantities to produce a negative, from which 1 is subtracted. We thus have the following theorem.

**Theorem 8**: For any DIMSIM with p = q = r = s = 2, the FASAL implementation of the method will be A-stable if and only if the diagonal element $\lambda$ of the A matrix is not less than 1/2 and the element v in the first column of the V matrix is in the set [-1,1/3].

Proof: See above. ■

The following result shows that for a FASAL implementation the error constant can be as good as that of the trapezoidal rule (-1/12) but not better.

**Theorem 9**: For any DIMSIM with p = q = r = s = 2 and with an A-stable FASAL implementation, the error constant $C \le -\frac{1}{12}$, with the optimal value reached for the diagonal element $\lambda$ of the A matrix equal to 1/2.

Proof:

The error constant for all second order DIMSIMs with $c_1 = 0$, $c_2 = 1$ is given by

$$C = v^T \varphi_2 = \frac{5 - 12\lambda - 6v + 12\lambda v}{12}.$$

We rewrite this as

$$C = \tfrac{1}{2}(-1 + 2\lambda)(-1 + v) - \tfrac{1}{12}.$$

Since -1 + v < 0, this quantity is always negative. It reaches its smallest magnitude, -1/12, for $\lambda$ =1/2. ■

For these optimal methods, it is not possible to calculate an error estimate of the form found in Theorem 6.

**Theorem 10**: DIMSIMs with $c_1 = 0$, $c_2 = 1$, and $\lambda = 1/2$ do not have error estimates of the form

$$err = h_n \beta^T(\delta) F\left(Y^{[n]}\right) + \gamma^T(\delta) y^{[n-1]}.$$

Proof: The error estimate must satisfy the conditions of Theorem 9. We define $\beta = \begin{bmatrix} \beta_1(\delta) \\ \beta_2(\delta) \end{bmatrix}$ and $\gamma = \begin{bmatrix} \gamma_1(\delta) \\ \gamma_2(\delta) \end{bmatrix}$. Imposing the condition that $\gamma^T e = 0$, we find that $\gamma_2 = -\gamma_1$. We then apply order conditions and find that there are two conditions on b2:

1) $\frac{-a+b_2d^2-v+av}{\delta^2}=0$, which gives $b_2=\frac{v+a-av}{\delta^2}$

2) $\frac{6a+\delta^3+6b_2\delta^3+6v-6av}{12\delta^3}=0$, which gives $b2=\frac{6av-6v-6a-\delta^3}{6\delta^3}$.

These cannot both be satisfied for all values of δ. ■

However, it is possible to chose methods that do have a FASAL implementation and are A-stable and that have an error estimate to have an error constant arbitrarily close to that of the trapizoidal rule. These methods can be of Type 4 as well.

## THIRD ORDER DIMSIMS WITH A-STABLE FASAL IMPLEMENTATIONS

Searching has been conducted for third order Type 2 DIMSIMs with c = [0,$c_2$,1] for which a FASAL implementation is A-stable. To date none have been found and it might be conjectured that there are no A-stable FASAL implementations of DIMSIMs of order higher than 2. If this conjecture is true, it constitutes an unfortunate order barrier.

## A FIFTH ORDER TYPE 2 DIMSIM

Butcher and Jackiewicz (Reference 32) found an L-stable Type 2 DIMSIM with p = q = r = s = 5. Implementation parameters are here derived to enable use of this method as a stiff solver. The stage points are given by c = [0,1/4,1/2,3/4,1]. The A matrix is

$$\begin{bmatrix} .2780538411364521 & 0 & 0 & 0 & 0 \\ .2204522761825798 & .2780538411364521 & 0 & 0 & 0 \\ 2.2948198957363657 & -.6023667080712847 & .2780538411364521 & 0 & 0 \\ 5.0546209011538535 & -1.5298762183097632 & .0971191414988231 & .2780538411364521 & 0 \\ 9.3451677801081329 & -1.4121335130997734 & -1.8834019985178697 & .7825339554468704 & .2780538411364521 \end{bmatrix}$$

and we set $V=ev^T$ and $\tilde{V}=\hat{e}_1v^T$, where

$$v=\begin{bmatrix} -.0793854651324349 \\ .5543175729105773 \\ -1.5695895491441551 \\ 2.3320745924436815 \\ -.2374171510776688 \end{bmatrix}.$$

U = I. We use Theorem 2 to calculate B. This gives

$$\begin{bmatrix} 6.044855283302179 & -2.020000467205476 & .03293453364122479 & .5935789859233151 & -.2266648512058528 \\ 5.853954219943505 & -1.072092372634326 & -1.839270544389963 & 2.410922952843391 & -.899263047489796 \\ 6.004175007913425 & -2.014097375842605 & .6108454298803939 & -.963490004887004 & -.4051827602739023 \\ 6.002703177071046 & -2.556003283230891 & 3.151551366098853 & -5.493514217893924 & .4481026180673921 \\ 4.481882795290198 & 2.672564354868939 & -1.413660973235832 & -8.05815479374699 & .909905877341711 \end{bmatrix}$$

We use the approach developed in the first report (Reference 1) with a form for $\tilde{V}$ with the first row of $v^T$ and the rest of the elements equal to 0 and then apply Theorem 2 to calculate $\tilde{B}$. This gives

$$\begin{bmatrix} 6.044855283302179 & -2.020000467205476 & .03293453364122479 & .5935789859233151 & .05138898993059922 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & -\frac{16}{3} & 12 & -16 & \frac{25}{3} \\ \frac{44}{3} & -\frac{224}{3} & 152 & -\frac{416}{3} & \frac{140}{3} \\ 96 & -448 & 768 & -576 & 160 \\ 256 & -1024 & 1536 & -1024 & 256 \end{bmatrix},$$

and W is given by Theorem 1 as

$$\begin{bmatrix} 1 & -.2780538411364521 & 0 & 0 & 0 & 0 \\ 1 & -.2485061173190319 & -.03826346028411303 & -.006085015868847461 & -.0005613381279595107 & -.00003711813820580275 \\ 1 & -1.470507028801533 & .1365647564495951 & .004900562818504469 & -.001619958388073782 & -.0003656404215677 \\ 1 & -3.149917665479366 & .4066191029756902 & .02777859631520006 & -.004406329750950594 & -.00169212095991602 \\ 1 & -6.110220065073812 & .929780069812273 & .0871064932281099 & -.01678258627228048 & -.00843432100142294 \end{bmatrix}.$$

Attempts to impose the conditions of Theorem 4 to derive a Butcher-Jackiewicz-type interpolant unfortunately led to a contradiction and thus an interpolant of this type does not exist for this method. The Nordsieck and continuous Nordsieck interpolants are available, however. The conditions of Theorem 6 were used to derive a Butcher-Jackiewicz-type error estimate. Free parameters were used to impose an additional condition $\beta^T e=0$ (see Reference 27) and to ensure that there were no poles on the positive real axis (see Reference 1). The estimate is of the form

$$est = \varsigma(\delta)\left(h_n \beta^T F\left(Y^{[n]}\right) + \gamma^T \tilde{y}^{[n-1]}\right),$$

where

$$\varsigma(\delta) = \frac{\delta^4}{\sum_{j=0}^{4} d_j \delta^j},$$

with

$d_0$= 6.64091327569422,

$d_1$= 18.61705794940406,

$d_2$= 17.91032428174742,

$d_3$= 10.11460326636336,

and

$d_4$= 4.180423658325783.

Also, we calculate

$$\beta = \begin{bmatrix} 13.32250820717907 \\ -6.014932565502038 \\ -10.93733645614135 \\ 4.651916710919581 \\ -1.022155896455255 \end{bmatrix},$$

and

$$\gamma = \begin{bmatrix} 134.9644699887695 \\ -134.284141177105 \\ 0 \\ 0 \\ -.6803288116644014 \end{bmatrix}.$$

An initial error estimate was obtained using the conditions of Theorem 7. It is of the form

$$est = h\beta_0^T F\left(Y^{[1]}\right) + \gamma_0^T y^{[0]},$$

where

$$\beta_0 = \begin{bmatrix} -.935098307887532 \\ .1838804364783428 \\ .209786992326761 \\ -.04441560222950508 \\ -.02083333013682731 \end{bmatrix}$$

and

$$\gamma_0 = \begin{bmatrix} .1040230658993777 \\ 0 \\ 0 \\ 0 \\ -.1040230658993777 \end{bmatrix}.$$

The error coefficient for steps after the first is $v^T\varphi_5 = -.000530049899772722$ and for the first step it is calculated to be -0.000205316. All calculations were performed using MATHEMATICA™.

## ALTERNATIVE ERROR ESTIMATES

Error estimates seem to be a bit more difficult for stiff problems than for nonstiff problems. This is because higher order terms may dominate the error in the range of moderate accuracy and higher hL, where L is the relevant Lipshitz constant for the problem. The development of some alternative approaches for error estimation is described.

## BUTCHER-JACKIEWICZ-TYPE ERROR ESTIMATES

These estimates, of the form

$$est = v^T le = h_n \beta^T(\delta) F\left(Y^{[n]}\right) + \gamma^T(\delta)\tilde{y}^{[n-1]},$$

are used because the external stage error propagates in the direction of v and because this has successfully approximated the observed local error. An important theorem (theorem 5) states that the local error is given by

$$le(t_n) = \varphi_p h^{p+1} y^{(p+1)}(t_{n-1}) + O\left(h^{p+2}\right),$$

where $v^T\varphi_p$ is then the error constant characteristic of the method and $\varphi_p$ is given by

$$\varphi_p = \sum_{k=1}^{p+1} \frac{\alpha_{p+1-k}}{k!} - \frac{Bc^p}{p!}.$$

The Butcher-Jackiewicz error estimation conditions are then derived by expanding the difference between the calculated value for the external stage vector and the theoretical by equating Taylor expansion terms to 0 up to $O(h^p)$ and equating the $O(h^{p+1})$ term to the local external stage error. For variable step size the rescaling step must be included in the formulation. Conditions are then obtained which provide values for the vectors $\beta$ and $\gamma$. This is the approach that was used exclusively in the first report (Reference 1), along with some alternative equivalent formulas. Some examples of Butcher-Jackiewicz-type error estimates are derived in the previous section.

## USING HIGH STAGE ORDER

The equality of the stage order q to the order p in Butcher's original DIMSIM design (Reference 23) carries with it some interesting implications that have already been described in the first report (Reference 1) for FASAL implementations. The ocal stage order condition requires that

$$Y_j^{[n]} = y(t_{n-1} + c_j h_n; t_{n-1}, y_{n-1}) + O\left(h_n^{p+1}\right),$$

where here y denotes the local solution of the ODE through the indicated initial point. The internal stage values are actually used only after application of the derivative function and multiplied by the step size. We note that if the local stage order condition is met,

$$hf\left(t_{n-1}, Y_j^{[n]} + O\left(h^{p+1}\right)\right) = hf\left(t_{n-1}, Y_j^{[n]}\right) + O\left(h^{p+2}\right),$$

and so it is possible where this is advantageous to substitute stage values computed using other methods, other step sizes, or for other steps if they are computed for the same stage point without creating inaccuracies. As noted in the discussion of FASAL implementations in the first report (Reference 1), if the results are carried forward there may be stability implications. However, error estimation typically does not entail carrying a result forward, and the high stage order may thus be used for this application without generating stability problems. It should be noted that higher order terms ($O(h^{p+2})$) will be changed. In some cases they will be fortuitously decreased, but in others they will be increased. But error estimates and other calculated quantities tend to depend only on terms that are $O(h^{p+1})$. This is the fundamental idea that is used below to obtain new classes of no-cost DIMSIM error estimators.

## COMPANION METHOD ERROR ESTIMATES

A popular approach long used with Runge-Kutta methods for estimating error for a method of order p is to find companion methods of adjacent orders which differ only in the b vector and in the addition of one extra stage. The higher order answer is taken to closely approximate the solution and the error is then the difference between the solution given by the higher order method and the solution given by the method of interest. This difference should be $O(h^{p+1})$, with the higher order method producing hopefully a smaller error with principal part $O(h^{p+2})$. Sometimes no additional function evaluations are required, but then the lower order method will typically not then use the minimum number of function evaluations). This is the error estimation approach used with the popular Runge-Kutta-Fehlberg solver RKF45 (see for example Reference 36).

For a DIMSIM with p = q = r = s, the high stage order may be used as described above, along with the DIMSIM feature that finding a method of the appropriate order is just a matter of calculating the B matrix. Thus it is possible to find a companion method which may be used with no extra function evaluations for which the internal stage values are approximately the same (differing in $O(h^{p+1})$) and for which the error constant is 0 or at least different. It would seem that ideally the W matrix should be the same so that the external stage vector has the same meaning and may be used for either method. This is simply a matter of using the same c values for internal stage points and the same A matrix, which then guarantees the same W matrix. The vector $v$ is then chosen to make the error constant 0, if possible, and otherwise at least significantly different from the error constant of the original method. This is done without regard to stability since it is only used for one step and zero stability is guaranteed by the form of V. V and $\tilde{V}$ are then determined in the usual way as $ev^T$ and $\hat{e}_1 v^T$, respectively. The B matrix and the $\tilde{B}$ matrix must also be calculated for this new method, as determined in the customary way from the foregoing choices. Then if the step size is fixed, and if the original method is indicated by 1 and the

second method by 2, beginning with the external stage vector of the previous step we calculate

$$Y^{[n]} = hAF\left(Y^{[n]}\right) + y^{[n-1]},$$

$$y_1^n = h\tilde{B}_1^1 F\left(Y^{[n]}\right) + v_1^T y^{[n-1]} = y(t_n) - C_1 h^{p+1} y^{(p+1)}(t_{n-1}) + O\left(h^{p+2}\right),$$

and

$$y_2^n = h\tilde{B}_1^2 F\left(Y^{[n]}\right) + v_2^T y^{[n-1]} = y(t_n) - C_2 h^{p+1} y^{(p+1)}(t_{n-1}) + O\left(h^{p+2}\right).$$

Then the error can be estimated using the difference in the Nordsieck vector first components:

$$y_2^n - y_1^n = h\left(\tilde{B}_2(1) - \tilde{B}_1(1)\right)F\left(Y^{[n]}\right) + \left(v_2^T - v_1^T\right)y^{[n-1]}$$

$$= (C_1 - C_2)h^{p+1} y^{(p+1)}(t_{n-1}) + O\left(h^{p+2}\right).$$

Thus we obtain the estimate

$$est = \frac{C_1}{C_1 - C_2}\left(h\left(\tilde{B}_1^2 - \tilde{B}_1^1\right)F\left(Y^{[n]}\right) + \left(v_2^T - v_1^T\right)y^{[n-1]}\right). \tag{39}$$

Of course the leading coefficient ratio is exactly 1 for $C_2 = 0$.

A calculation for the Type 1, second order DIMSIM first developed by Butcher (Reference 23) and used extensively in the previous report (Reference 1) indicates that the same result is obtained for fixed step size as for the companion method approach outlined above. The method, using c = [0,1], is

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ \frac{5}{4} & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} \\ \frac{3}{4} & -\frac{1}{4} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

The error constant $C_1$ is 1/6. We choose another c = [0,1] method of the general form

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ b_{11} & b_{12} & v & 1-v \\ b_{21} & b_{22} & v & 1-v \end{bmatrix}.$$

Using Butcher's theorem we obtain a B matrix of the form

$$\begin{bmatrix} 2-\frac{3v}{2} & \frac{v}{2} \\ -\frac{3}{2}(-1+v) & \frac{1}{2}(-1+v) \end{bmatrix}.$$

The error constant $C_2$ for this method is

$$v^T\varphi_2 = \tfrac{1}{12}(5-6v).$$

We note that for explicit methods with $c_1 = 0$ and $\tilde{V}$ chosen in the customary way, the first row of $\tilde{B}$ is the same as the first row of B. Then we may calculate the same vectors $\beta$ and $\gamma$ for fixed step sizes that were defined for the Butcher-Jackiewicz-type error estimate, but now based on the idea of subtracting the first solution from the second,

$$\beta = \frac{C_1}{C_1 - C_2}\left(B_1^2 - B_1^1\right) = \left[-\tfrac{1}{2}, \tfrac{1}{6}\right]^T.$$

Similarly,

$$\gamma = \frac{C_1}{C_1 - C_2}\left(v_2^T - v_1^T\right) = \left[\tfrac{1}{3}, -\tfrac{1}{3}\right]^T.$$

Note that these results are independent of v and are identical to those obtained using the Butcher-Jackiewicz approach.

Noting that a value of v = 5/6 produces $C_2 = 0$, we now analyze error when simply subtracting the first result from the second. This now yields

$$\beta = B_1^2 - B_1^1 = \left[-\tfrac{1}{2}, \tfrac{1}{6}\right]$$

and

$$\gamma = \left[\tfrac{1}{3}, -\tfrac{1}{3}\right].$$

Thus the two approaches agree for this simple method with fixed step size, and similar results have been obtained for other methods.

If step size is not fixed, it must be noted that the step change enters in as a difference between two methods in the rescaling process, and this must be done separately for the two methods or the difference will reduce to the fixed step-size formula. Thus we have

$$y_1^n = h_n \tilde{B}_1^1 F\left(Y^{[n]}\right) + v_1^T \hat{y}_1^{[n-1]},$$

where

$$\hat{y}_1^{[n-1]} = h_{n-1} WD\tilde{B}^1 F\left(Y^{[n-1]}\right) + e v_1^T \hat{y}^{[n-2]},$$

and similarly for $y_2^n$. Then

$$y_2^n - y_1^n = h_n\left(\tilde{B}_1^2 - \tilde{B}_1^1\right)F\left(Y^{[n]}\right) + v_2^T\hat{y}_2^{[n-1]} - v_1^T\hat{y}_1^{[n-1]},$$

$$= h_n\left(\tilde{B}_1^2 - \tilde{B}_1^2\right)F\left(Y^{[n]}\right) + v_2^T\left(h_{n-1}WD\tilde{B}^2F\left(Y^{[n-1]}\right) + ev_2^T\hat{y}^{[n-2]}\right)$$

$$- v_1^T\left(h_{n-1}WD\tilde{B}^1F\left(Y^{[n-1]}\right) + ev_1^T\hat{y}^{[n-2]}\right).$$

Since $v^Te = 1$, this may finally be written as

$$y_2^n - y_1^n = h_n\left(\tilde{B}_1^2 - \tilde{B}_1^1\right)F\left(Y^{[n]}\right) + h_{n-1}\left(v_2^TWD\tilde{B}^2 - v_1^TWD\tilde{B}^1\right)F\left(Y^{[n-1]}\right)$$

$$+\left(v_2^T - v_1^T\right)\hat{y}^{[n-2]}.$$

We thus propose the following alternative heuristic error estimate.

**Companion Error Estimate**: Select two DIMSIMs with the same c and A but with different choices of v, with error constants $C_1$ and $C_2$, respectively. Then the local truncation error of the solution at the nth step may be estimated as

$$est = \left(\frac{C_1}{C_1 - C_2}\right)\left(y_2^n - y_1^n\right)$$

$$= \left(\frac{C_1}{C_1 - C_2}\right)\left(h_n\left(\tilde{B}_1^2 - \tilde{B}_1^1\right)F\left(Y^{[n]}\right) + h_{n-1}\left(v_2^TWD\tilde{B}^2 - v_1^TWD\tilde{B}^1\right)F\left(Y^{[n-1]}\right) + \left(v_2^T - v_1^T\right)\hat{y}^{[n-2]}\right),$$

where the indices 1 and 2 are used to refer to the separate methods.

Note that this measures the error in the computed solution at the end of a step using the same output internal stages from the previous step and the same external stage vector from two steps previous. This approach differs from the Butcher-Jackiewicz approach in the limit of constant step size, since for the variable step approach in the case of a step ratio of 1, a separate rescaling would be performed since $WD\tilde{B}^1 = B^1 \neq B^2 = WD\tilde{B}^2$ and $v_1 \neq v_2$.

## COMPARISON TESTING FOR SECOND ORDER

A Type 2 DIMSIM with a more moderately small error constant of -5/128 was selected for testing. This c = [0,1] method is described by the Butcher DIMSIM tableau

$$\begin{bmatrix} \frac{25}{24} & 0 & 1 & 0 \\ \frac{311}{324} & \frac{25}{24} & 0 & 1 \\ \frac{16477}{10368} & -\frac{75}{128} & \frac{225}{208} & -\frac{17}{208} \\ \frac{22093}{10368} & -\frac{11275}{10368} & \frac{225}{208} & -\frac{17}{208} \end{bmatrix}.$$

We also calculate

$$W = \begin{bmatrix} 1 & -\frac{25}{24} & 0 \\ 1 & -\frac{649}{648} & -\frac{13}{24} \end{bmatrix},$$

we choose the standard form for the rescaling matrices:

$$\tilde{V} = \begin{bmatrix} \frac{225}{208} & -\frac{17}{208} \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

and thus compute

$$\tilde{B} = \begin{bmatrix} \frac{16477}{10368} & \frac{175}{384} \\ 0 & 1 \\ -1 & 1 \end{bmatrix}.$$

For the Butcher-Jackiewicz-type error estimate we obtain

$$\beta = \frac{5\delta}{64(1+\delta)}\left[\frac{29}{27}, -1\right]^T$$

and

$$\gamma = \frac{15\delta}{104(1+\delta)}[1,-1]^T.$$

For the first step we have

$$\beta_0 = \left[\frac{1363}{10368}, -\frac{47}{384}\right]^T$$

and

$$\gamma_0 = \frac{47}{208}[1,-1]^T.$$

For a suitable companion method with error constant 0, we find the method with Butcher tableau

$$\begin{bmatrix} \frac{25}{24} & 0 & 1 & 0 \\ \frac{311}{324} & \frac{25}{24} & 0 & 1 \\ \frac{1057}{648} & -\frac{5}{8} & \frac{15}{13} & -\frac{2}{13} \\ \frac{176}{81} & -\frac{365}{324} & \frac{15}{13} & -\frac{2}{13} \end{bmatrix}.$$

The W matrix is unchanged and we select

$$\tilde{V} = \begin{bmatrix} \frac{15}{13} & -\frac{2}{13} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

in the customary way, and compute

$$\tilde{B} = \begin{bmatrix} \frac{1057}{648} & \frac{5}{12} \\ 0 & 1 \\ -1 & 1 \end{bmatrix}.$$

The method turns out to be A-stable.

Finally, for a companion method with a nonzero error constant, we select the method

$$\begin{bmatrix} \frac{25}{24} & 0 & 1 & 0 \\ \frac{311}{324} & \frac{25}{24} & 0 & 1 \\ \frac{26429}{20736} & -\frac{75}{256} & \frac{225}{416} & \frac{191}{416} \\ \frac{37661}{20736} & -\frac{16475}{20736} & \frac{225}{416} & \frac{191}{416} \end{bmatrix}.$$

It again has the same W matrix, and we select

$$\tilde{V} = \begin{bmatrix} \frac{225}{416} & \frac{191}{416} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

and compute

$$\tilde{B} = \begin{bmatrix} \frac{26429}{20736} & \frac{575}{768} \\ 0 & 1 \\ -1 & 1 \end{bmatrix}.$$

The error constant is -85/256 and the method is also A-stable.

We wish to use the Prothero-Robinson problem (Reference 37), as in the first report (Reference 1), to test convergence of error estimate as fixed step size is reduced, behavior of error estimate with varying step size, and behavior with increasing stiffness. The Prothero-Robinson problem displays stiffness in a single equation for an appropriate choice of parameters. The general form of the Prothero-Robinson equation is:

$$y' = \lambda(y - p(t)) + p'(t), \; y(t_0) = y_0. \tag{40}$$

The exact solution is

$$y(t) = (y_0 - p(t_0))\exp(\lambda(t - t_0)) + p(t). \tag{41}$$

It is interesting to note that for $y_0 = p(t_0)$ the solution is simply p(t). The sine function was used for p(t) and the interval of interest is $t \in [0,20]$. The value for $\lambda$ was -2 unless otherwise indicated.

A quantity r is determined for each step, where r is defined as

$$r \equiv \left| \left| \frac{err}{est} \right| - 1 \right|.$$

Here err represents the local error as described in Equation 1, that is, the solution of the initial value problem of Equation 1 but beginning at a DIMSIM solution point $(t_{n-1},y_{n-1})$ with exact local solution at $t_n$ given by $y(t_n;t_{n-1},y_{n-1})$, and with calculated solution at $t_n$ of $y_n$. Thus we test

$$err_n = y_n - y(t_n;t_{n-1},y_{n-1}).$$

It should be noted that it is not appropriate to restart the solution in seeking to obtain the expression for *err*. A DIMSIM changes character after the first step, and this must be preserved in the testing process. In particular the external stage vector is not recalculated using a starting procedure, otherwise the separate error estimate for an intial step would be required. Testing will only then truly reveal the behavior of the error estimator in its use within a solver.

The results of a test may be indicated both graphically and using a number of statistics. The quantity *est* is used to denote the result of the use of the error estimate provided for the method. A graph showing *err* and *est* together is sometimes very revealing, but it is also vital to look at the largest error and the solution range. And a table showing the percentages of error estimates yielding r less than 1, 5, 10, 25, 50, and 100% has proven to be of great value. Of course it is expected that test results will vary greatly with the tolerance used and also with the problem solved and even with the problem parameters. Thus any testing of implementation parameters must be considered preliminary to the more extensive testing required of a full solver.

Test 1 was first described by Butcher and Jackiewicz (Reference 26). A scheme was devised for testing the effects of rapid step changes on error estimation accuracy. An initial step size $h_0$ is chosen, along with a parameter $\rho$ that is varied from 1.25 to 2. The new step size is calculated using a ratio given by

$$r = \rho^{(-1)^k \sin\left(\frac{4\pi t}{b-a}\right)}. \qquad (42)$$

Thus a cyclic pattern of step sizes is used, alternately lengthening and shrinking in ratios from $1/\rho$ to $\rho$ until the end of the interval of interest is reached. This should be considered to be a very stringent test, especially for higher values of $\rho$, and serves as an excellent preliminary check. A quick calculation similar to the one shown above for the Type 1 DIMSIM shows that all three approaches lead to the same error estimate for fixed step size without rescaling, so fixed step size tests without separate rescaling is not repeated for each method.

The results shown in Figures 1 through 4 and Tables 1 through 4 were obtained with $\lambda = -2$. The dotted line is the estimate and the solid line is the true local error.

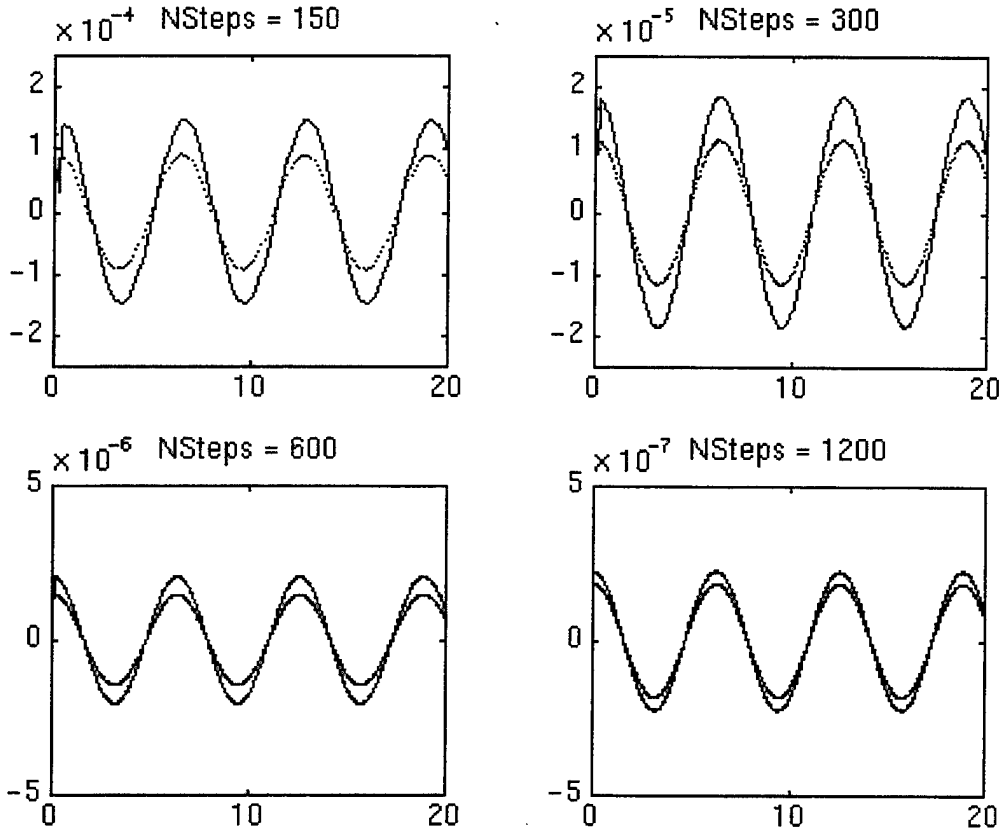



FIGURE 1. Butcher-Jackiewicz-Type, Fixed Step-Size Error Estimate.

TABLE 1. Butcher-Jackiewicz-Type, Fixed Step-Size Error Estimate.

| NSteps | % < 0.01 | % < 0.05 | % < 0.10 | % < 0.25 | % < 0.50 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 0.67 | 1.33 | 3.33 | 8.67 | 28.7 | 86.7 |
| 300 | 0.33 | 0.33 | 0.67 | 2.33 | 8.0 | 97.3 |
| 600 | 0.0 | 0.17 | 0.50 | 1.33 | 97.7 | 99.8 |
| 1200 | 0.33 | 0.92 | 1.50 | 87.1 | 98.8 | 99.5 |

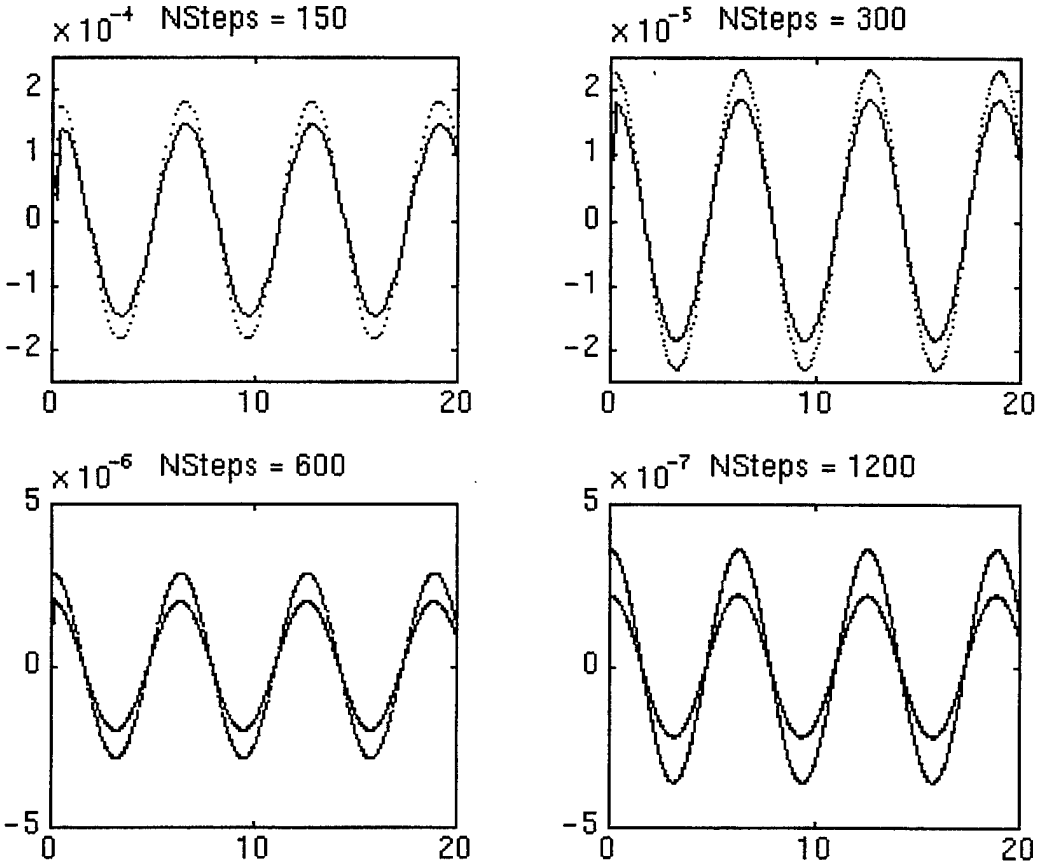
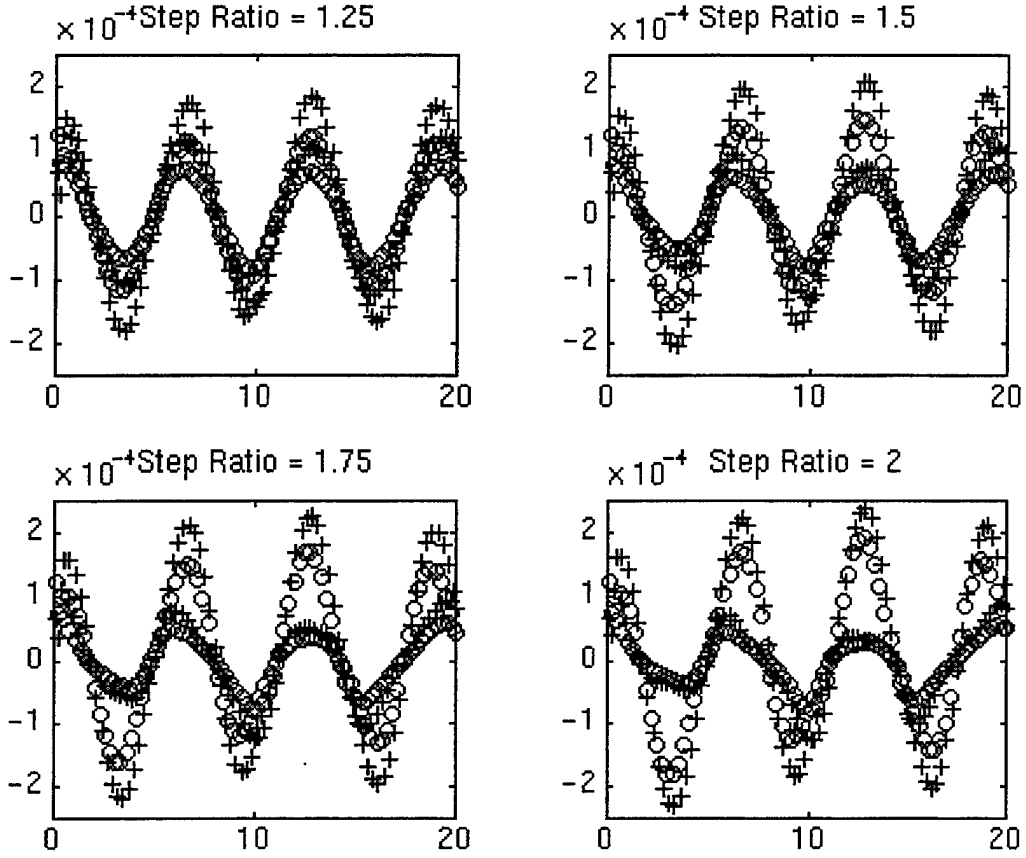



FIGURE 2. Companion Method Fixed Step-Size Error Estimate.

TABLE 2. Companion Method Fixed Step-Size Error Estimate.

| NSteps | % < 0.01 | % < 0.05 | % < 0.10 | % < 0.25 | % < 0.50 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 0.67 | 2.67 | 6.67 | 79.3 | 94.7 | 98.7 |
| 300 | 0.33 | 0.67 | 2.67 | 92.0 | 98.3 | 99.7 |
| 600 | 0 | 0.50 | 1.00 | 8.00 | 97.0 | 99.2 |
| 1200 | 0 | 0.25 | 0.33 | 1.75 | 96.9 | 99.7 |

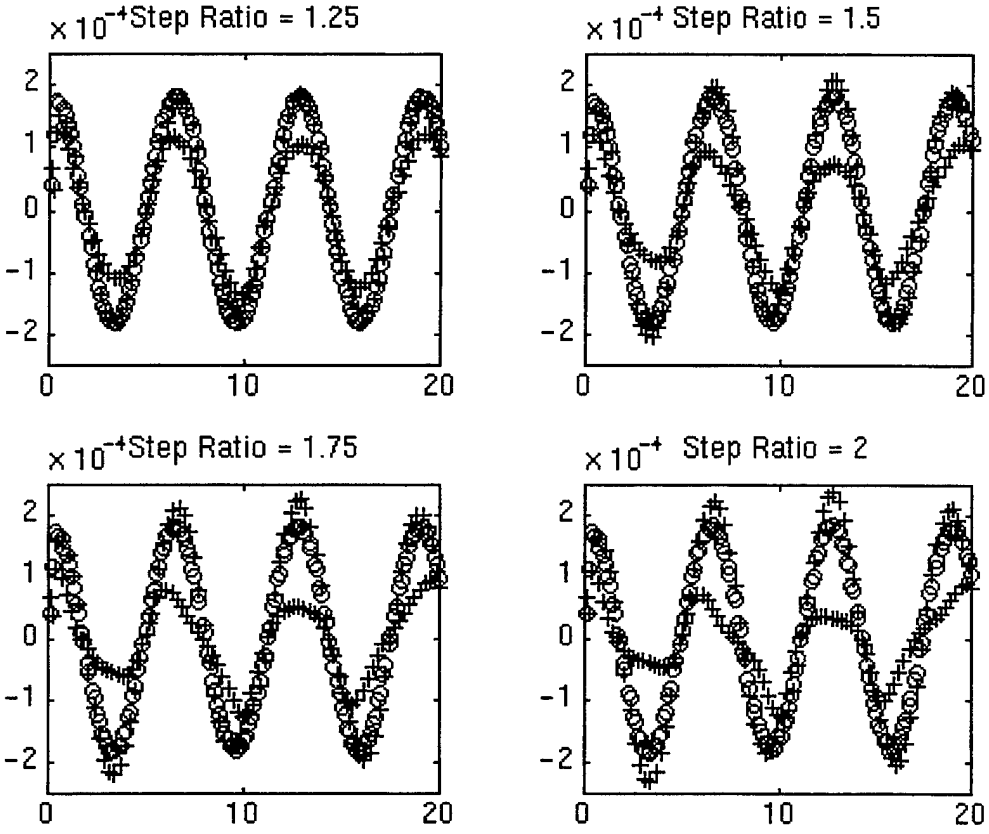



+ is local error, o is estimate

FIGURE 3. Butcher-Jackiewicz-Type Variable Step-Size Error Estimate.

TABLE 3. Butcher-Jackiewicz-Type Variable Step-Size Error Estimate.

| Ratio | % < 0.01 | % < 0.05 | % < 0.10 | % < 0.25 | % < 0.50 | % < 1 |
|---|---|---|---|---|---|---|
| 1.25 | 1.32 | 5.96 | 12.6 | 30.5 | 46.4 | 62.9 |
| 1.50 | 0 | 3.29 | 6.58 | 15.8 | 42.1 | 57.2 |
| 1.75 | 0 | 1.97 | 5.26 | 12.5 | 25.0 | 55.9 |
| 2.00 | 0 | 3.27 | 5.23 | 10.5 | 22.2 | 55.6 |




+ is local error, o is estimate

FIGURE 4. Companion Method Variable Step-Size Error Estimate.

TABLE 4. Companion Method Variable Step-Size Error Estimate.

| Ratio | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 1.25 | 3.31 | 18.54 | 29.1 | 51.7 | 94.0 | 98.7 |
| 1.50 | 2.63 | 11.2 | 22.4 | 46.1 | 72.4 | 99.3 |
| 1.75 | 0.66 | 6.58 | 15.1 | 42.1 | 65.1 | 99.3 |
| 2.00 | 1.31 | 1.31 | 9.80 | 34.0 | 62.7 | 98.0 |

It may be noted that only the designation "companion method error estimates" was used. This is because it was found that even with rescaling and variable step sizes, the same identical results were obtained regardless whether the third order method or the second order method was used as the companion method. In these tests the companion method error estimates with separate rescaling produced significantly better results.

The effect of stiffness on error estimation was also considered (Figures 5 through 8). This may be seen by comparing error estimation accuracy graphs for constant step size for values of $\lambda$ of -2, -10, -100 and -1000. The companion error estimate was used.

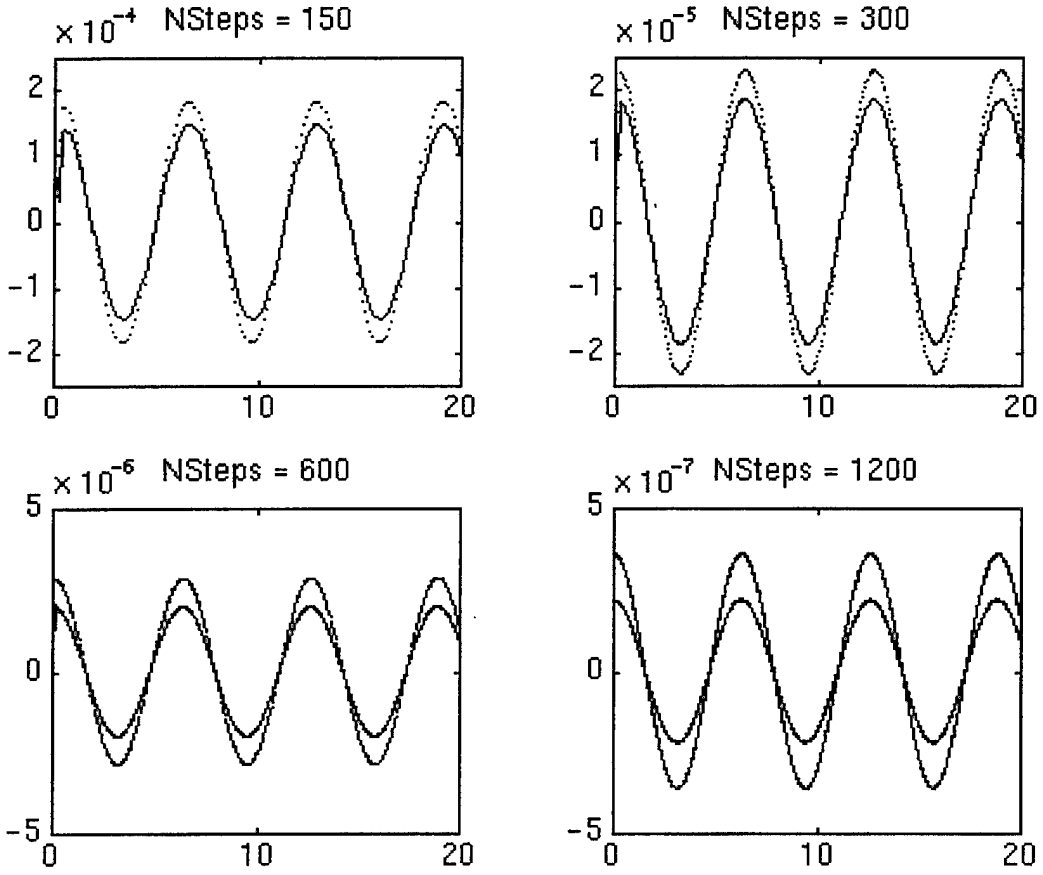



FIGURE 5. Fixed Step Size, $\lambda$ = -2.

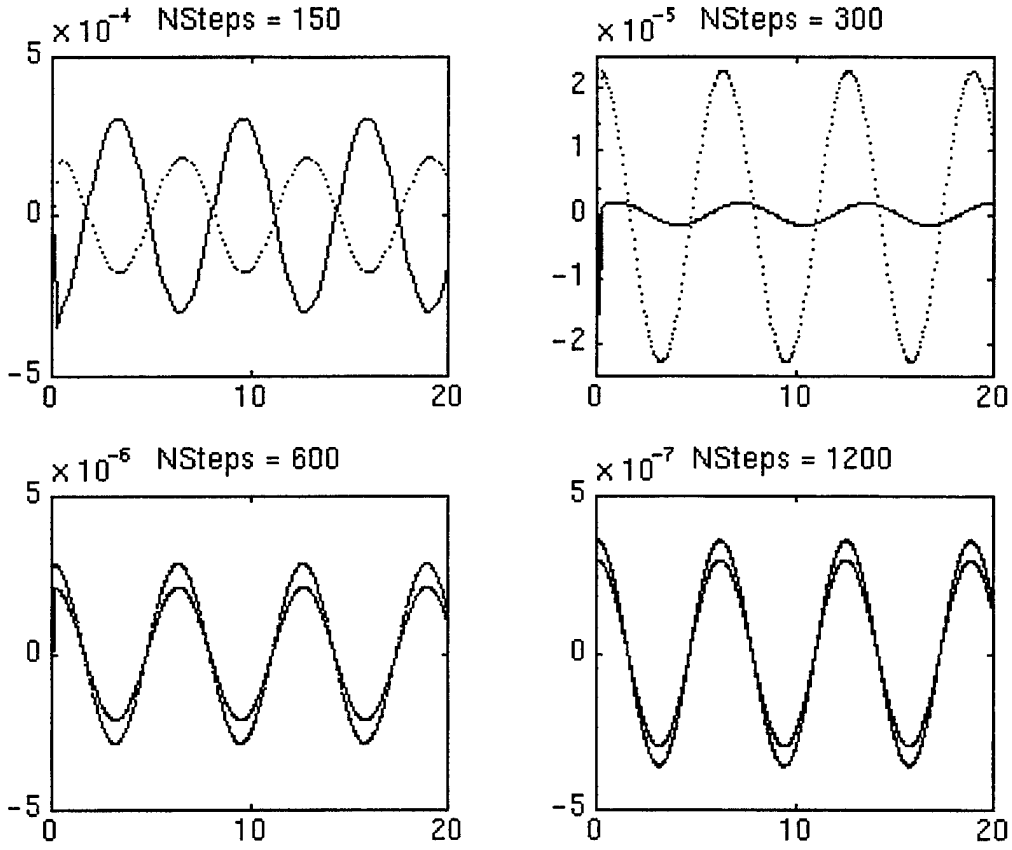



FIGURE 6. Fixed Step Size, $\lambda = -10$.

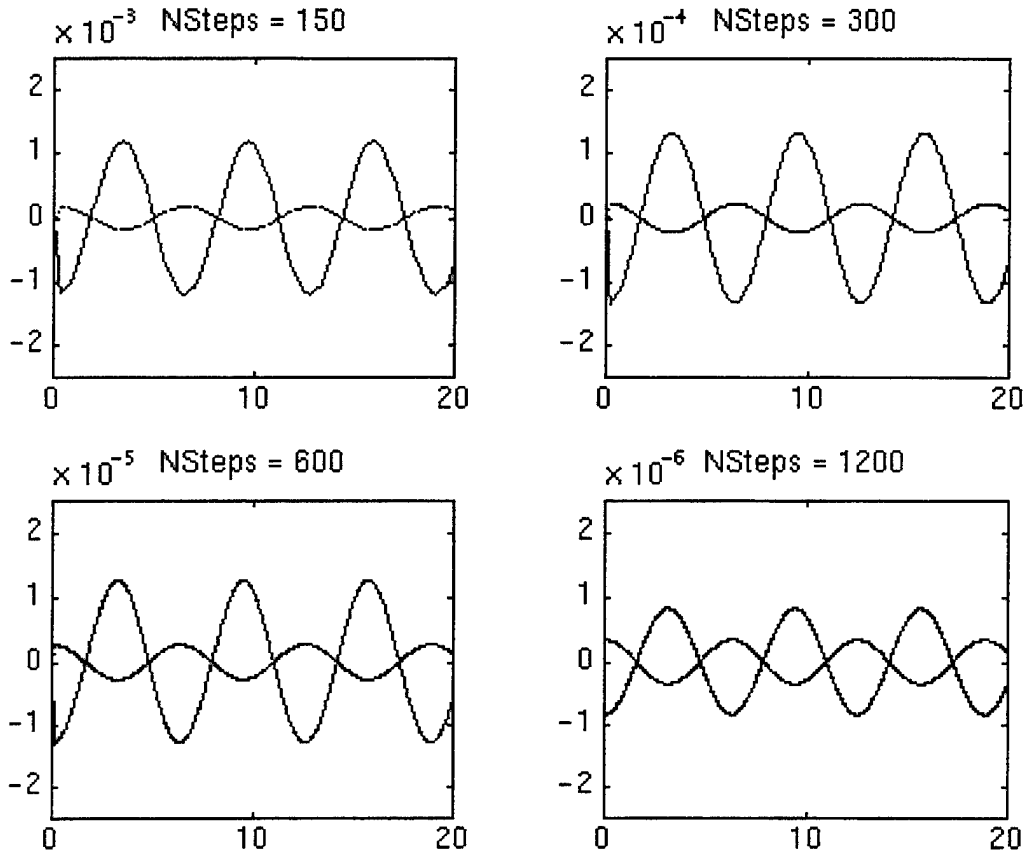



FIGURE 7. Fixed Step Size, $\lambda = -100$.

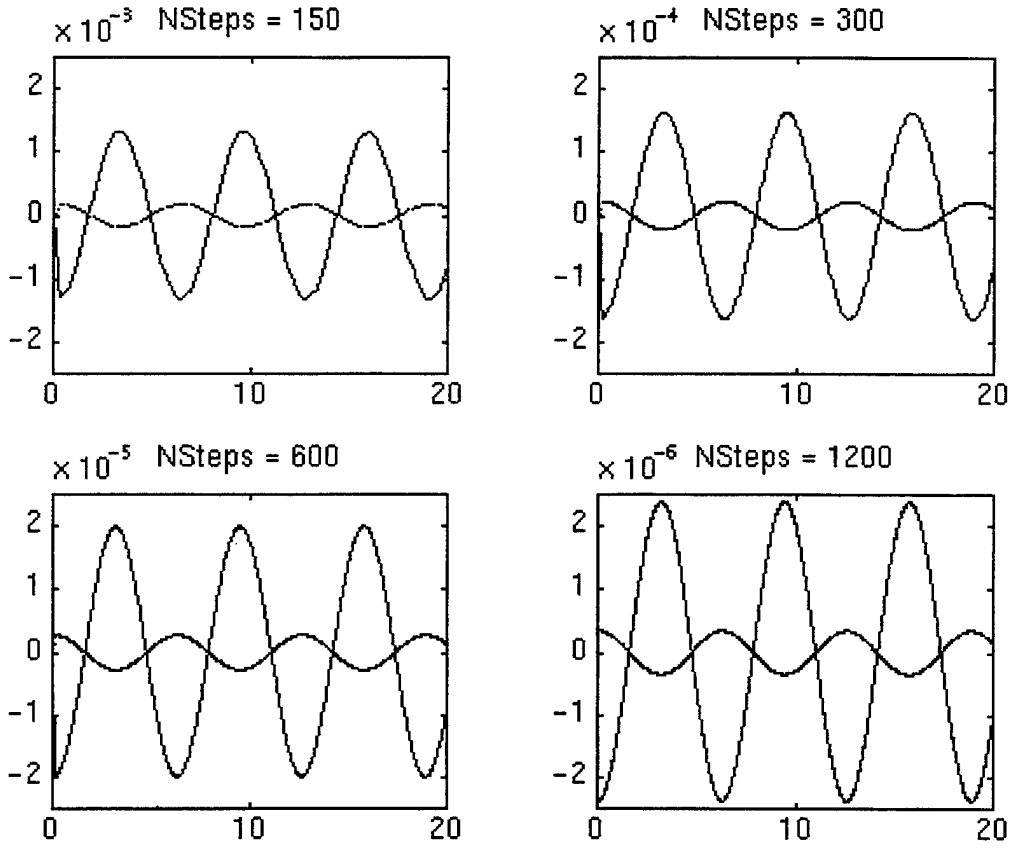



FIGURE 8. Fixed Step Size, $\lambda$ = -1000.

These results seem to indicate that despite final accuracy that improves with stiffness, error estimation deteriorates due probably to a dominance in the error of higher order terms. In fact the error changes sign as might be expected with a next higher term with a higher derivative proportional to a power of $\lambda$ and thus having opposite sign. The behavior for $\lambda$ = -10 seems to show that as the step size is reduced, the error does indeed first approach a point where the error in the next higher term seems to approximately cancel the error term of degree p + 1, and then the sign changes and the estimate and the true local error agree and asymptotically become equal.

## FIFTH ORDER ERROR ESTIMATION

The Type 2 fifth order DIMSIM described above was utilized. A companion method with a zero error constant has the same c vector, A matrix, and W matrix. Values for $v^T$ are chosen (not uniquely) to set the error constant $v^T\varphi_5 = 0$. The value

$$v = \begin{bmatrix} -1.772568719302804 \\ 1 \\ 1 \\ 1 \\ -.2274312806971961 \end{bmatrix}$$

was used in this case. $V = ev^T$ and $\tilde{V} = \hat{e}_1 v^T$. The B matrix for the companion method was then determined to be

$$\begin{bmatrix} 4.795023599486394 & -1.984105740930325 & .7671750153188124 & .3573344857080197 & -.2270751049413304 \\ 4.60412253612772 & -1.036197646359175 & -1.105030062712375 & 2.174678452628096 & -.899673301225274 \\ 4.754343324097641 & -1.978202649567453 & 1.345085911557982 & -1.199734505102299 & -.4055930140093798 \\ 4.75287149325526 & -2.52010855695574 & 3.885791847776441 & -5.72975871810922 & .4476923643319147 \\ 3.232051111474413 & 2.70845908114409 & -.6794204915582444 & -8.29439929396228 & .909495623606234 \end{bmatrix}$$

Finally, the matrix $\tilde{B}$ is calculated as

$$\begin{bmatrix} 4.795023599486394 & -1.984105740930325 & .7671750153188124 & .3573344857080197 & .0509787361951217 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & -\frac{16}{3} & 12 & -16 & \frac{25}{3} \\ \frac{44}{3} & -\frac{224}{3} & 152 & -\frac{416}{3} & \frac{140}{3} \\ 96 & -448 & 768 & -576 & 160 \\ 256 & -1024 & 1536 & -1024 & 256 \end{bmatrix}$$

It should be noted that although the error constant is 0 for steps after the first, the error constant for the first step is a fairly small nonzero number, 0.0000737847.

We now compare the error estimation resulting from using the companion method with the Butcher-Jackiewicz-type error estimate that we have derived. A test with constant step size yielded the results shown in Table 5 with increasing resolution.

TABLE 5. Butcher-Jackiewicz-Type Error Estimate For Fixed Step Size, Order 5.

| NPts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 0 | 0 | 0 | 2 | 3.33 | 99.3 |
| 300 | 0 | 0.333 | 0.333 | 1.33 | 4.33 | 99.3 |
| 300 | 0 | 0.333 | 0.333 | 1.17 | 80.3 | 99.5 |
| 1200 | 0.167 | 1.00 | 1.42 | 5.50 | 91.8 | 99.0 |

End point error for these tests are shown in Table 6.

TABLE 6. End Point Error.

| NPts | Error | Error/$h^5$ |
|---|---|---|
| 150 | 4.60e-09 | 1.09e-04 |
| 300 | 1.67e-10 | 1.27e-04 |
| 600 | 5.68e-12 | 1.38e-04 |
| 1200 | 1.65e-13 | 1.28e-04 |

Note that the method order clearly appears through the fixed ratio in this range of step size. A graph of error estimates (dotted line) versus local errors (solid line) are shown in Figure 9.

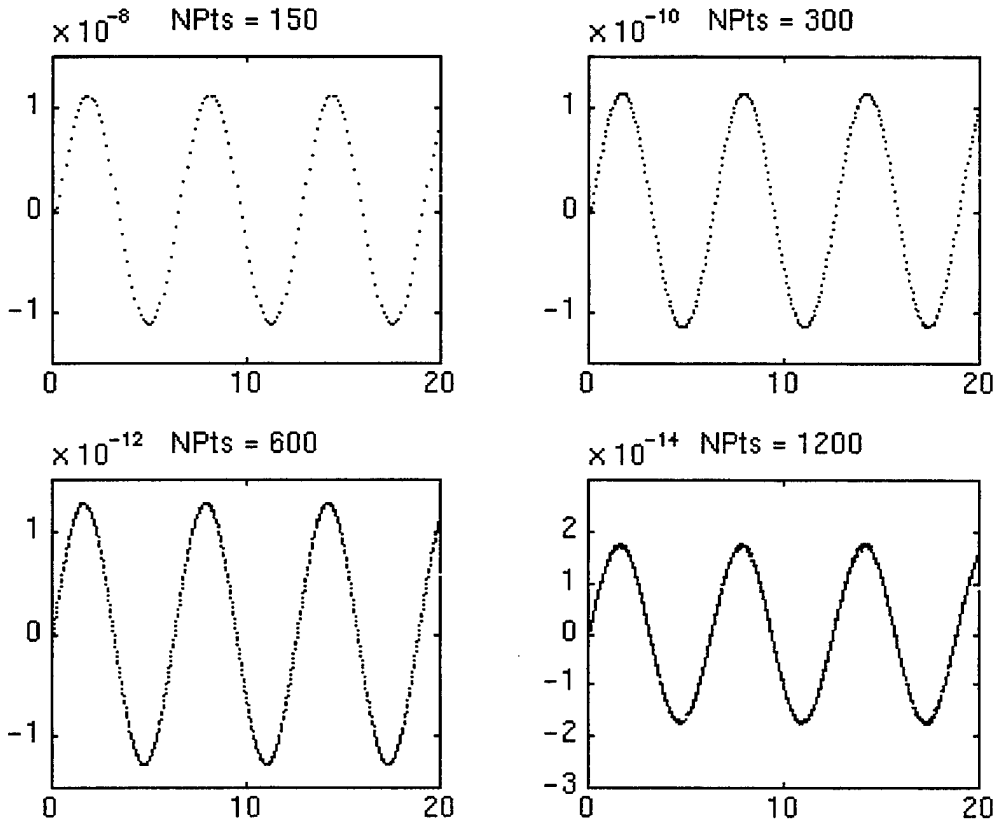



FIGURE 9. Butcher-Jackiewicz-Type Error Estimate For Fixed Step Size, Order 5.

The effect of variable step size also was tested. The problem with 600 points was used. The following data were obtained (Table 7).

TABLE 7. Butcher-Jackiewicz-Type Error Estimate For Variable Step Size, Order 5.

| Ratio | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 1.25 | 0.17 | 0.17 | 0.17 | 0.17 | 94.5 | 99.8 |
| 1.50 | 0.33 | 2.49 | 5.32 | 14.8 | 83.6 | 98.7 |
| 1.75 | 0.33 | 1.99 | 3.49 | 12.8 | 71.3 | 76.4 |
| 2.00 | 0.50 | 1.33 | 2.65 | 11.8 | 66.8 | 70.6 |

A graphical representation of the results appears in Figure 10.

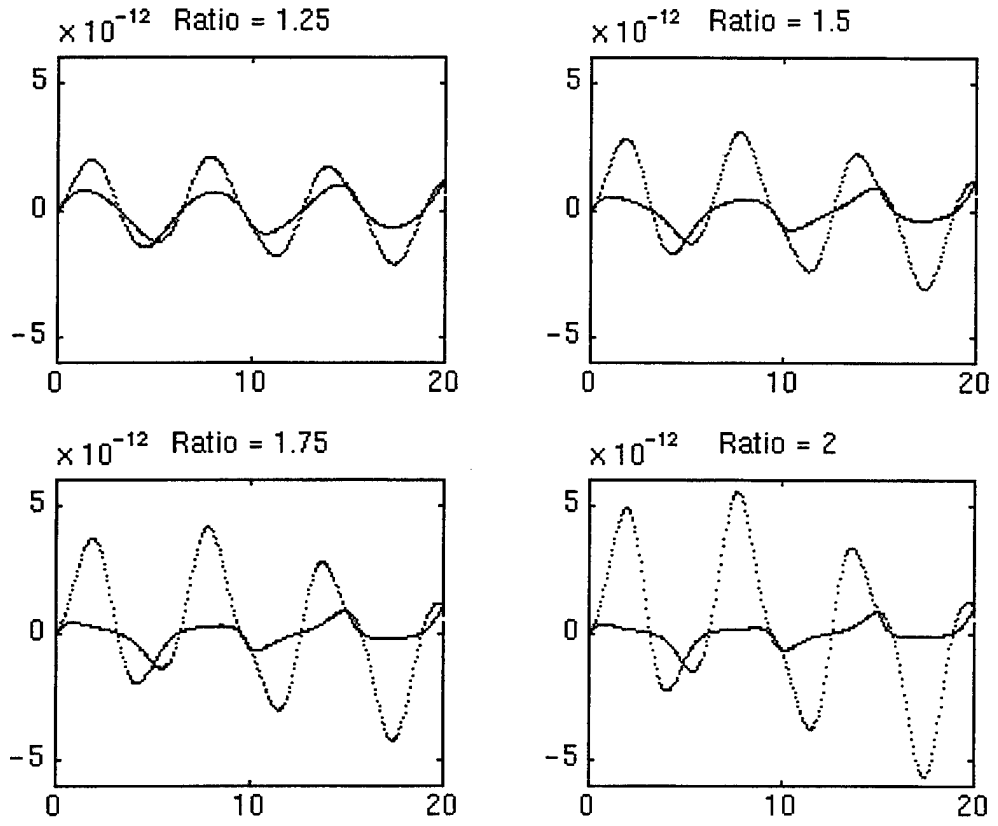



FIGURE 10. Butcher-Jackiewicz-Type Error Estimate For Variable Step Size, Order 5.

Although the picture for changing step size looks reasonably good, the number of points required to achieve the error estimation accuracy desired points to difficulty in efficient operation of a variable step-size solver.

The companion method was then used for purposes of comparison. The data were obtained for fixed step size shown in Table 8.

TABLE 8. Companion Method Error Estimate For Fixed Step Size, Order 5.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 0 | 1.33 | 1.33 | 3.33 | 7.33 | 97.3 |
| 300 | 1.33 | 6.33 | 13.0 | 36.3 | 72.0 | 92.3 |
| 600 | 0 | 0.50 | 1.66 | 4.5 | 9.33 | 35.2 |
| 1200 | 0.42 | 2.67 | 6.42 | 75.8 | 94.5 | 98.7 |

It is interesting to note in Figure 11 that the same pattern is seen as with stiffness earlier, and supports the interpretation that higher order term may more easily dominate the error when the error constant is small.

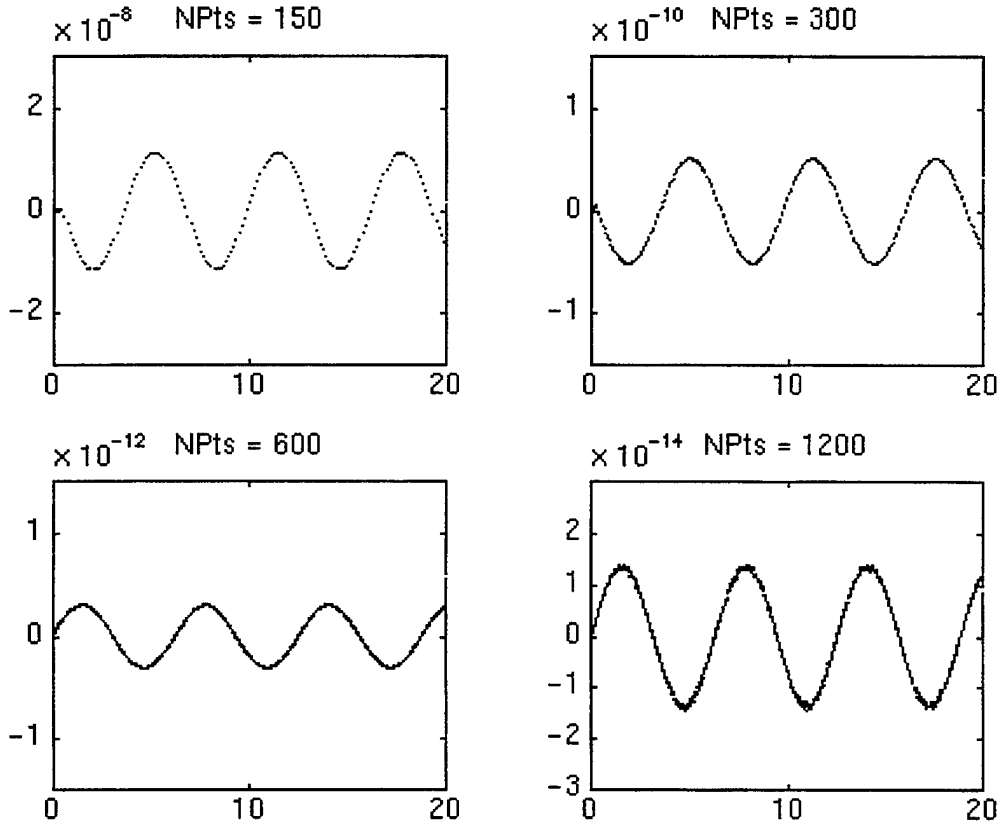



FIGURE 11. Companion Method Error Estimate For Fixed Step Size, Order 5.

Variable step size for 600 points was also explored. Note that the companion method produced statistically poor results for fixed step size at 600 points. The data in Table 9 and Figure 12 were obtained.

TABLE 9. Companion Method Error Estimate For Variable Step Size, Order 5.

| Ratio | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 1.25 | 0 | 3.16 | 7.15 | 23.23 | 41.9 | 63.1 |
| 1.50 | 0.17 | 1.83 | 4.32 | 11.1 | 40.7 | 83.1 |
| 1.75 | 0.17 | 1.50 | 3.16 | 13.2 | 45.3 | 88.0 |
| 2.00 | 0.33 | 1.49 | 3.98 | 27.2 | 47.1 | 90.2 |

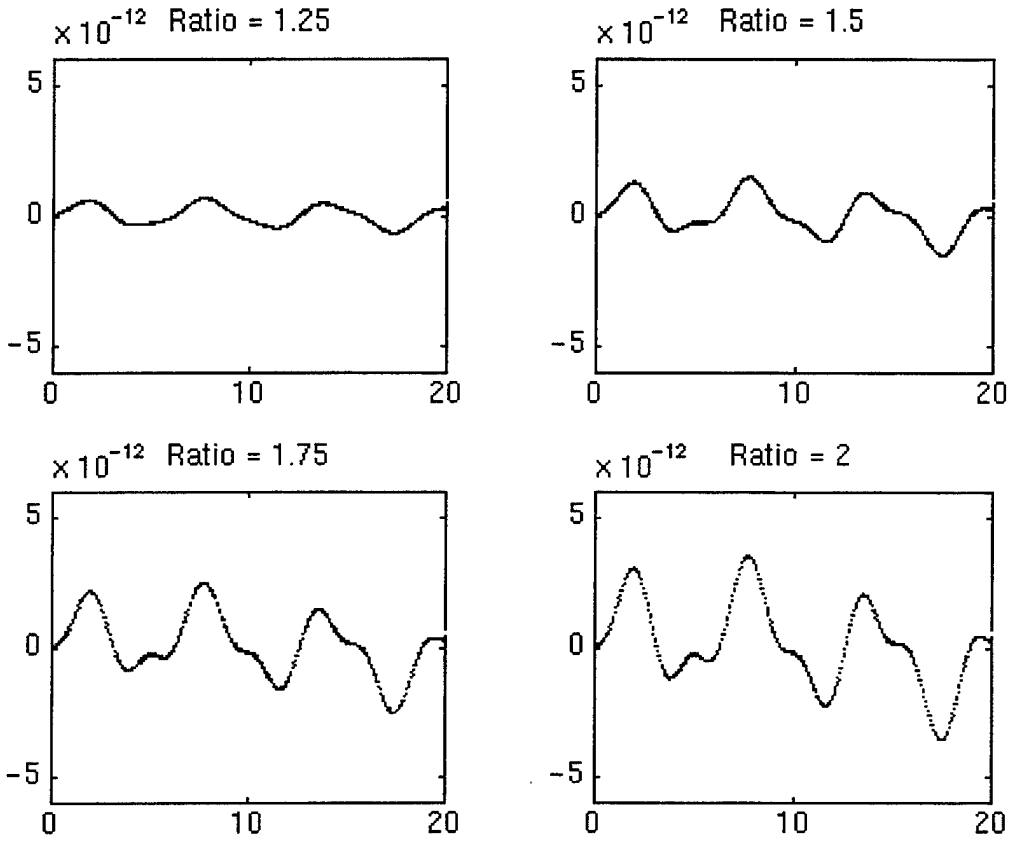



FIGURE 12. Companion Method Error Estimate For Variable Step Size, Order 5.

Astonishingly, the more the step-size changes, the more accurate the error estimation. This is probably to be explained in terms of the effect on step-size changing on the contribution to the local truncation error on leading versus higher order terms. The testing here is very limited, but the companion method error estimate clearly shows promise.

## RICHARDSON-TYPE ERROR ESTIMATION

The local stage order condition may also be used to supply an error estimate of the Richardson type. So long as the stage points are evenly spaced through the interval [0,1], the external stage vector for two steps of length h may be rescaled to provide the external stage input, and alternating internal stage values over two steps may be used to provide the approximate internal stages to yield a solution for step size 2h. We then have

$$y_h^n = h\tilde{B}F\left(Y_h^{[n]}\right) + v^T y_h^{[n-1]} = y(t_n) - 2Ch^{p+1}y^{(p+1)}(t_{n-2}) + O\left(h^{p+2}\right), \tag{43}$$

and

$$y_{2h}^n = h\tilde{B}F\left(Y_{2h}^{[n,n-1]}\right) + v^T y_{2h}^{[n-2]} = y(t_n) - C(2h)^{p+1}y^{(p+1)}(t_{n-2}) + O\left(h^{p+2}\right). \tag{44}$$

In the second equation the notation "n,n-1" for the internal stage vector is intended to indicate that every other stage point beginning with the first is used, the (p + 1)st derivative is expanded to enable use of the same time point, the references n,n-1,n-2 refer to step numbers using the step h, and 2h is used to indicate that a quantity is used with step size 2h. The difference then is given by

$$y_h^n - y_{2h}^n = \left(2^{p+1} - 2\right)Ch^{p+1}y^{p+1}(t_{n-2}) + O\left(h^{p+2}\right)$$

Then we may estimate the error as

$$err = \frac{\left(y_h^n - y_{2h}^n\right)}{2^{p+1} - 2} + O\left(h^{p+2}\right). \tag{45}$$

It should be noted that two steps must be completed to obtain an estimate. Thus if an integration step fails because tolerance is not met, it will be necessary to repeat two steps. And it is vital to keep constant step size over pairs of steps. Thus this approach provides a less sensitive adaptive integration and can only be competitive as a result of greater accuracy. Finally, because the larger step size is not repeated, it is expected that this approach may be used for explicit or predictor-corrector methods with limited stability regions since only the stability of the shorter step will be a factor.

### Testing Of Richardson Error Estimate

Richardson estimate tests were conducted (Table 10 and Figure 13) for fixed step sizes with the Prothero-Robinson problem (Reference 37), and with varying $\lambda$. For $\lambda$ values of smaller magnitude, results were excellent.

TABLE 10. Richardson Error Estimate, $\lambda = -2$, Order 2.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 1.33 | 8.67 | 17.33 | 52.67 | 80.67 | 92.67 |
| 300 | 0.67 | 3.33 | 9.67 | 81.00 | 95.00 | 98.33 |
| 600 | 7.00 | 91.17 | 96.00 | 98.67 | 99.33 | 99.67 |
| 1200 | 0.25 | 0.75 | 9.92 | 99.42 | 99.75 | 99.75 |

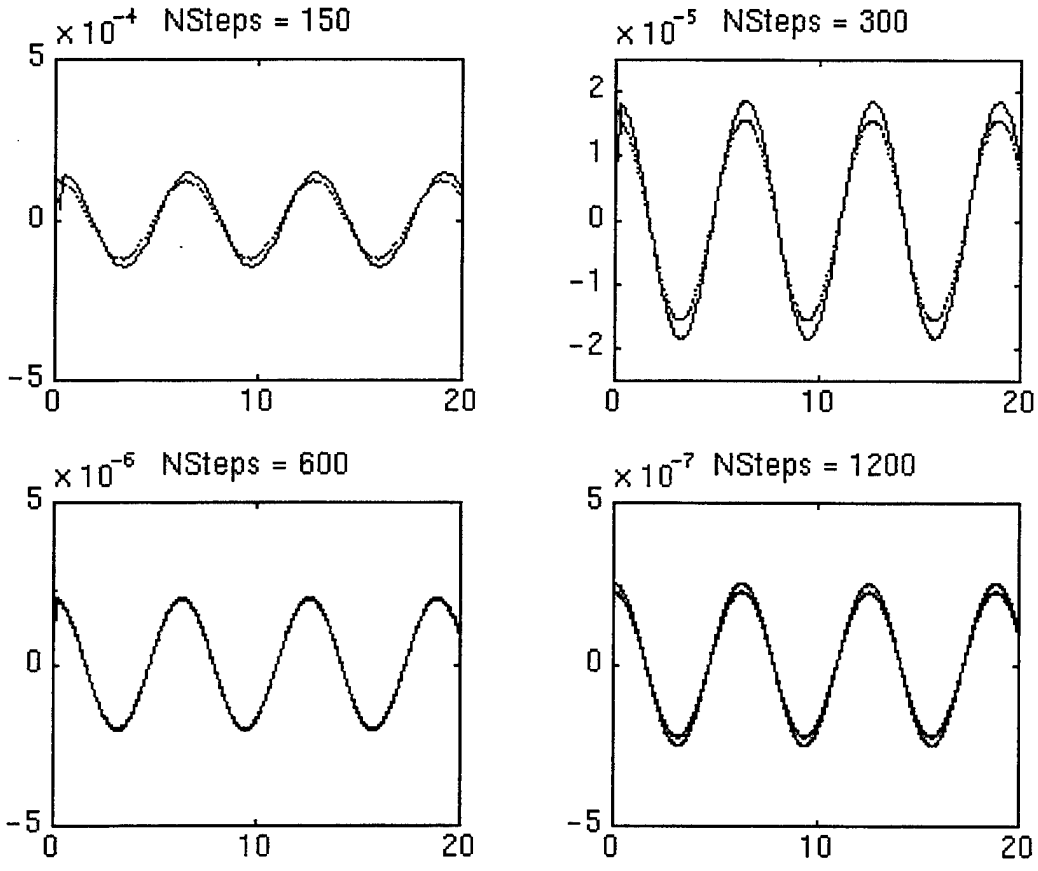



FIGURE 13. Richardson Error Estimate, $\lambda = -2$, Order 2.

As the magnitude of l increases, the same effects that were observed before are again seen in Table 11 and Figure 14.

TABLE 11. Richardson Error Estimate, $\lambda = -10$, Order 2.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 0.00 | 0.00 | 1.33 | 3.33 | 5.33 | 12.00 |
| 300 | 0.33 | 1.00 | 1.00 | 3.33 | 7.67 | 97.00 |
| 600 | 1.33 | 10.17 | 35.33 | 88.83 | 95.17 | 98.67 |
| 1200 | 0.25 | 0.75 | 2.00 | 87.00 | 98.42 | 99.50 |

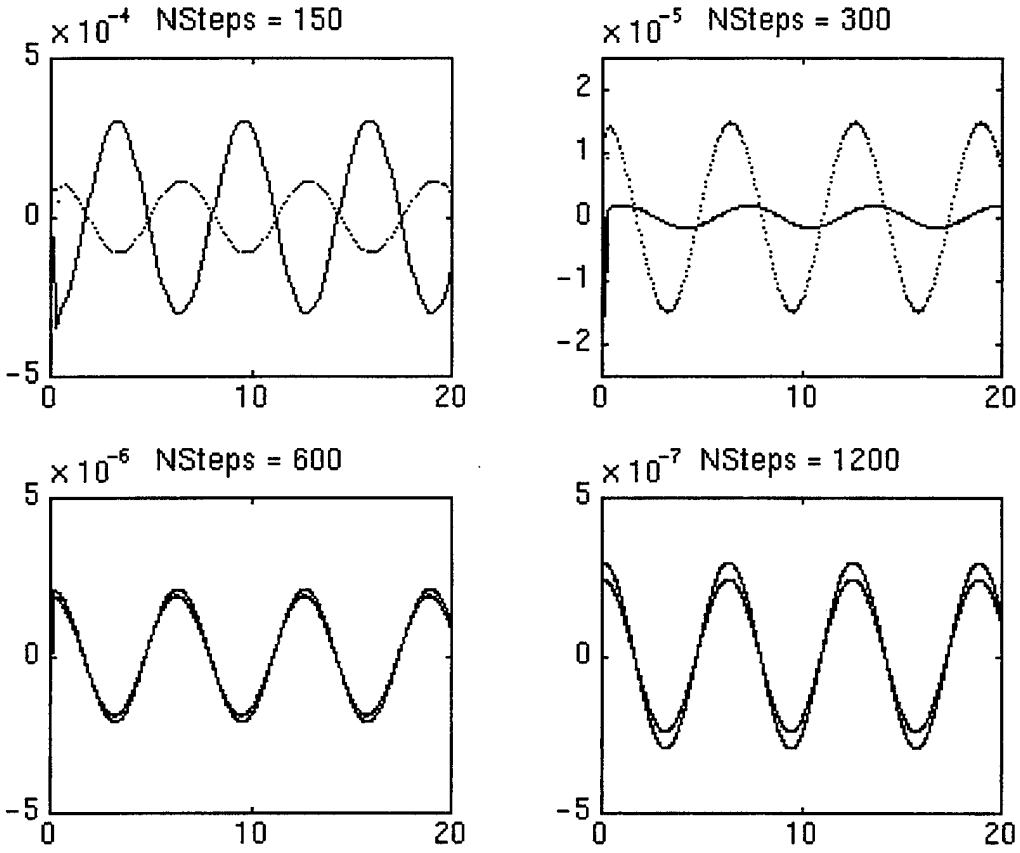



FIGURE 14. Richardson Error Estimate, $\lambda = -10$, Order 2.

The estimate was much worse for $\lambda = -100$.

Examination of the results pointed to a different heuristic formulation as the problem became increasingly stiff. It appeared that the result for the double step size was four times better than the result for the single step size. The following formula was then tested:

$$err = \frac{-2^p\left(y_h^n - y_{2h}^n\right)}{2^p - 1}. \tag{46}$$

The results as seen in Tables 12 through 16 and Figures 15 through 19 were very poor for small λ and smaller errors but became much better as λ and the errors became larger.

TABLE 12. Heuristic Richardson Error Estimate, λ = -10, Order 2.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 0.00 | 0.00 | 0.67 | 3.33 | 8.00 | 99.33 |
| 300 | 0.00 | 0.00 | 0.33 | 0.33 | 1.00 | 99.67 |
| 600 | 0.00 | 0.00 | 0.00 | 0.00 | 0.33 | 98.83 |
| 1200 | 0.00 | 0.00 | 0.00 | 0.172 | 0.33 | 100.00 |

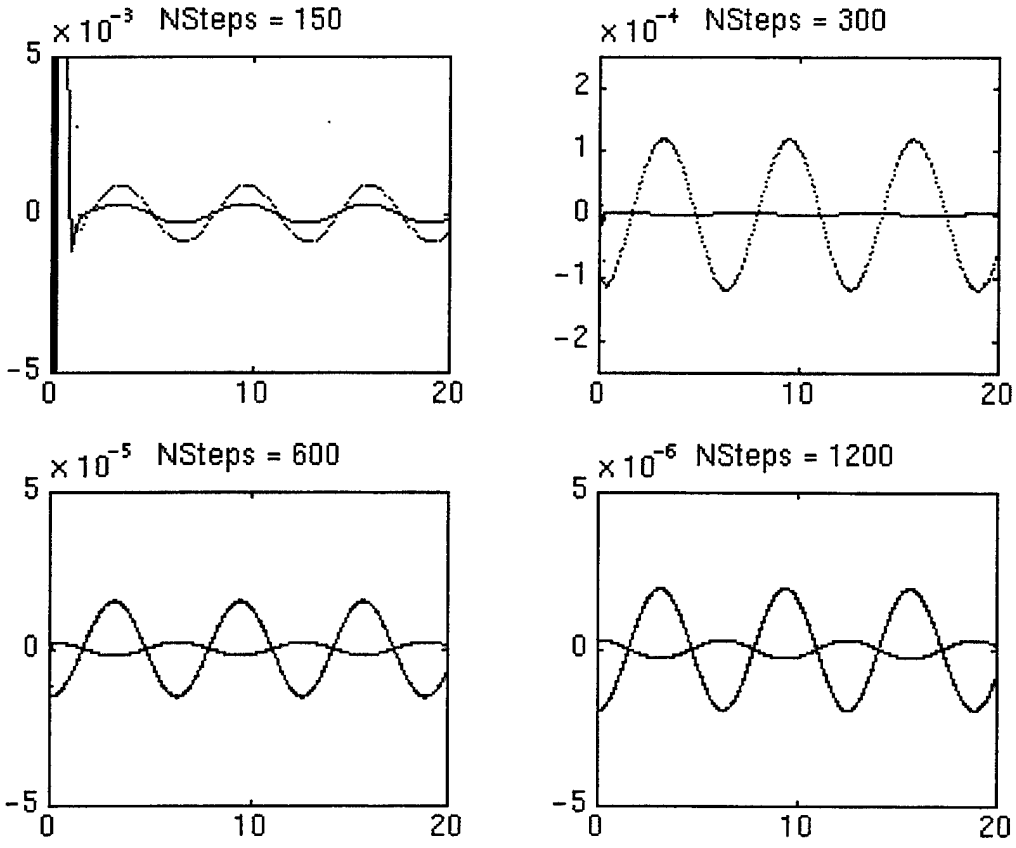



FIGURE 15. Heuristic Richardson Error Estimate, λ = -10, Order 2.

TABLE 13. Heuristic Richardson Error Estimate, $\lambda = -100$, Order 2.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 0.67 | 4.00 | 7.33 | 24.00 | 59.33 | 85.33 |
| 300 | 1.00 | 6.00 | 13.67 | 60.67 | 86.67 | 95.00 |
| 600 | 2.00 | 13.50 | 57.33 | 92.50 | 96.50 | 98.50 |
| 1200 | 0.33 | 0.33 | 0.58 | 0.92 | 6.50 | 99.75 |

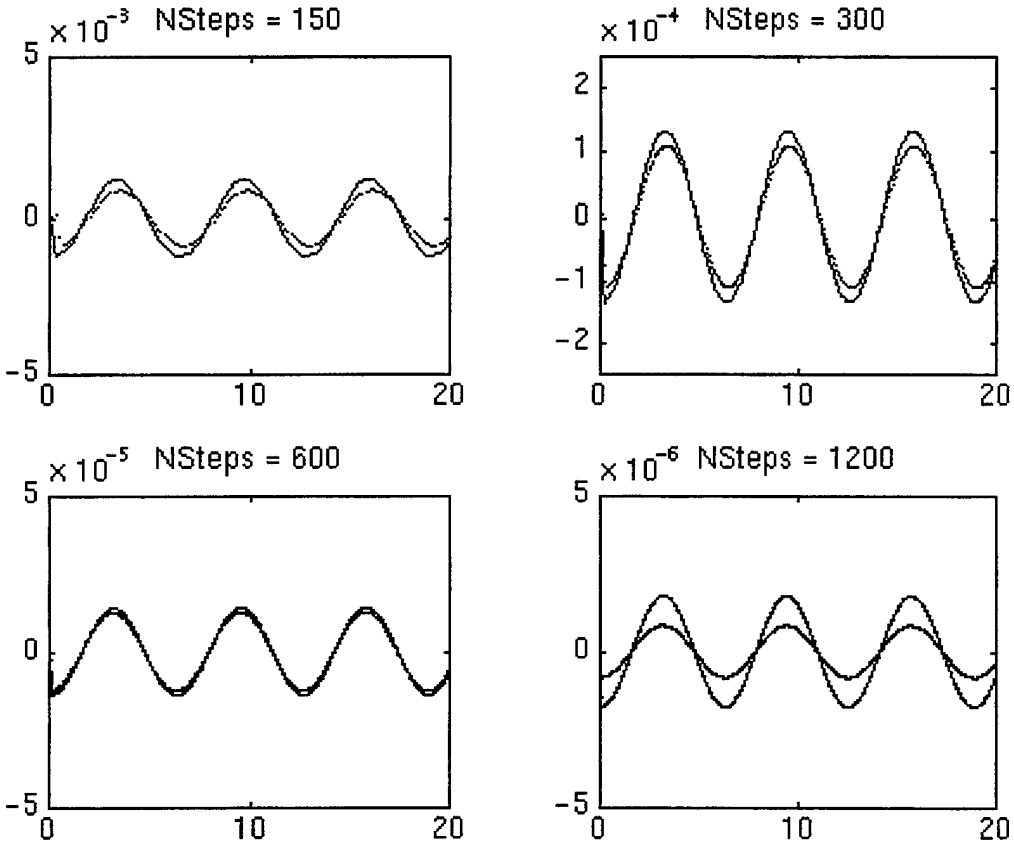



FIGURE 16. Heuristic Richardson Error Estimate, $\lambda = -100$, Order 2.

TABLE 14. Heuristic Richardson Error Estimate, $\lambda = -573$, Order 2.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 0.67 | 2.00 | 5.33 | 15.33 | 42.00 | 79.33 |
| 300 | 0.00 | 1.67 | 4.00 | 11.33 | 54.00 | 90.33 |
| 600 | 0.17 | 1.00 | 2.50 | 8.67 | 80.33 | 95.67 |
| 1200 | 0.25 | 0.92 | 2.00 | 16.00 | 95.00 | 98.33 |

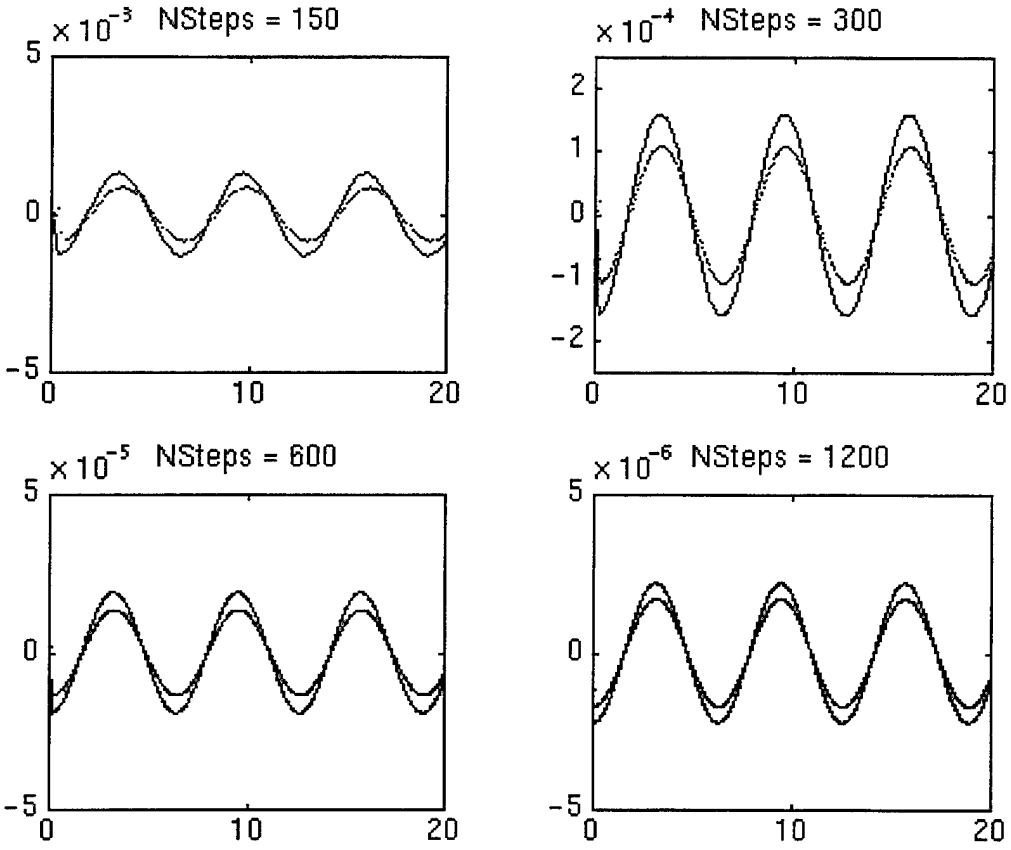



FIGURE 17. Heuristic Richardson Error Estimate, $\lambda = -573$, Order 2.

TABLE 15. Heuristic Richardson Error Estimate, $\lambda = -1{,}000$, Order 2.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 0.67 | 2.00 | 4.67 | 14.67 | 40.00 | 78.67 |
| 300 | 0.33 | 1.67 | 4.00 | 10.00 | 48.00 | 89.67 |
| 600 | 0.33 | 1.00 | 2.00 | 6.33 | 67.00 | 95.33 |
| 1200 | 0.17 | 0.50 | 1.25 | 5.00 | 90.92 | 98.00 |

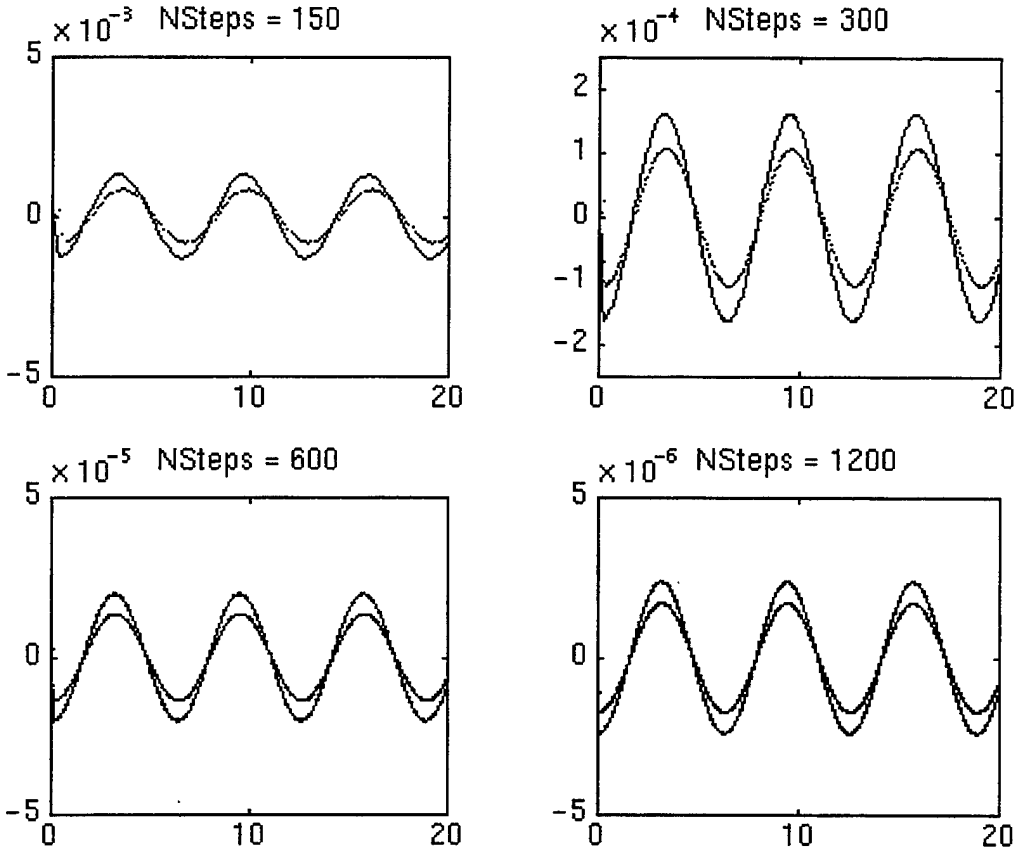



FIGURE 18. Heuristic Richardson Error Estimate $\lambda$ = -1,000, Order 2.

TABLE 16. Heuristic Richardson Error Estimate, $\lambda$ = -10,000, Order 2.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 150 | 0.67 | 2.67 | 6.00 | 14.67 | 39.33 | 78.00 |
| 300 | 1.00 | 1.67 | 3.67 | 9.33 | 41.00 | 88.33 |
| 600 | 0.17 | 1.00 | 1.67 | 5.00 | 38.50 | 94.33 |
| 1200 | 0.25 | 0.25 | 0.92 | 2.50 | 35.25 | 97.25 |

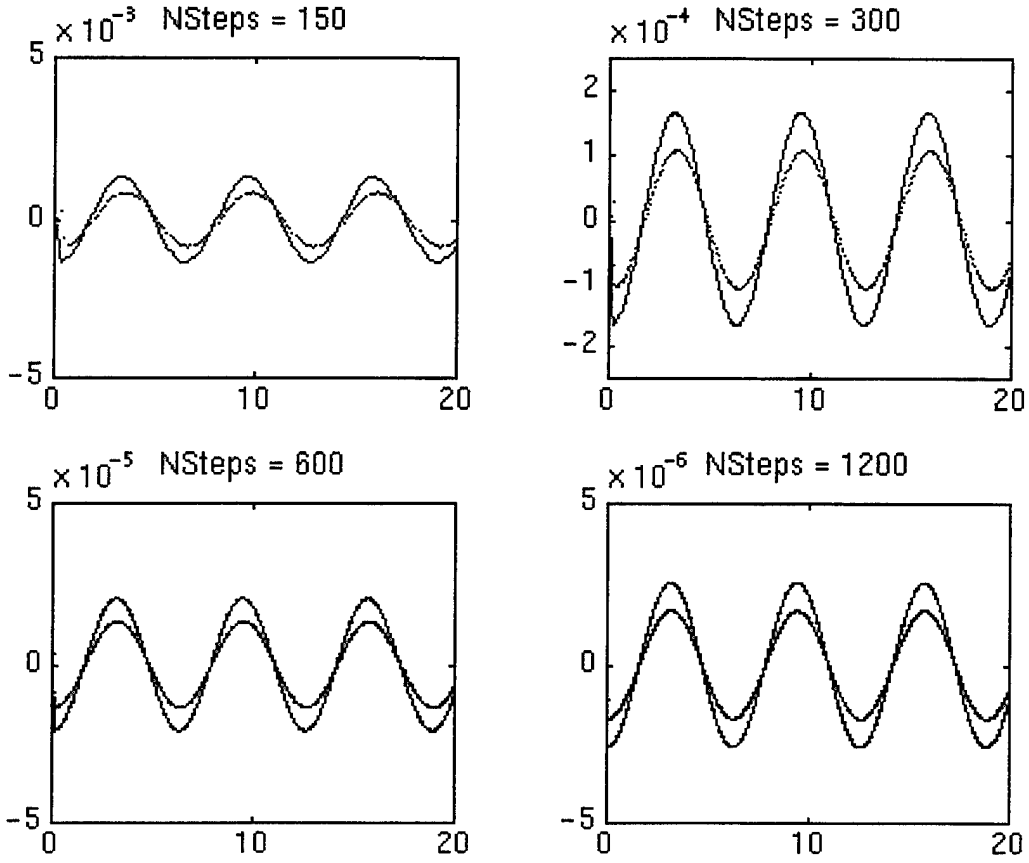



FIGURE 19. Heuristic Richardson Error Estimate, $\lambda$ = -10,000, Order 2.

We see that this formula seems to predict the error with adequate accuracy over a wide range of $\lambda$ values from -100 to -10,000. The case of $\lambda$ = -573 was used to eliminate the possibility of coincidental dependence on powers of 10.

These numerical experiments were repeated for fifth order. Results shown in Tables 17 through 19 and Figures 20 through 22 were excellent for $\lambda$ = -2, and began to deteriorate with increasing magnitude of $\lambda$ and lower accuracy.

TABLE 17. Richardson Error Estimate, $\lambda$ = -2, Order 5.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 75 | 1.33 | 2.67 | 4.00 | 12.00 | 46.67 | 94.67 |
| 150 | 0.67 | 2.67 | 6.67 | 24.67 | 80.00 | 96.67 |
| 300 | 0.67 | 4.00 | 8.33 | 57.33 | 91.00 | 97.67 |
| 600 | 0.67 | 4.12 | 10.83 | 86.17 | 95.50 | 98.83 |

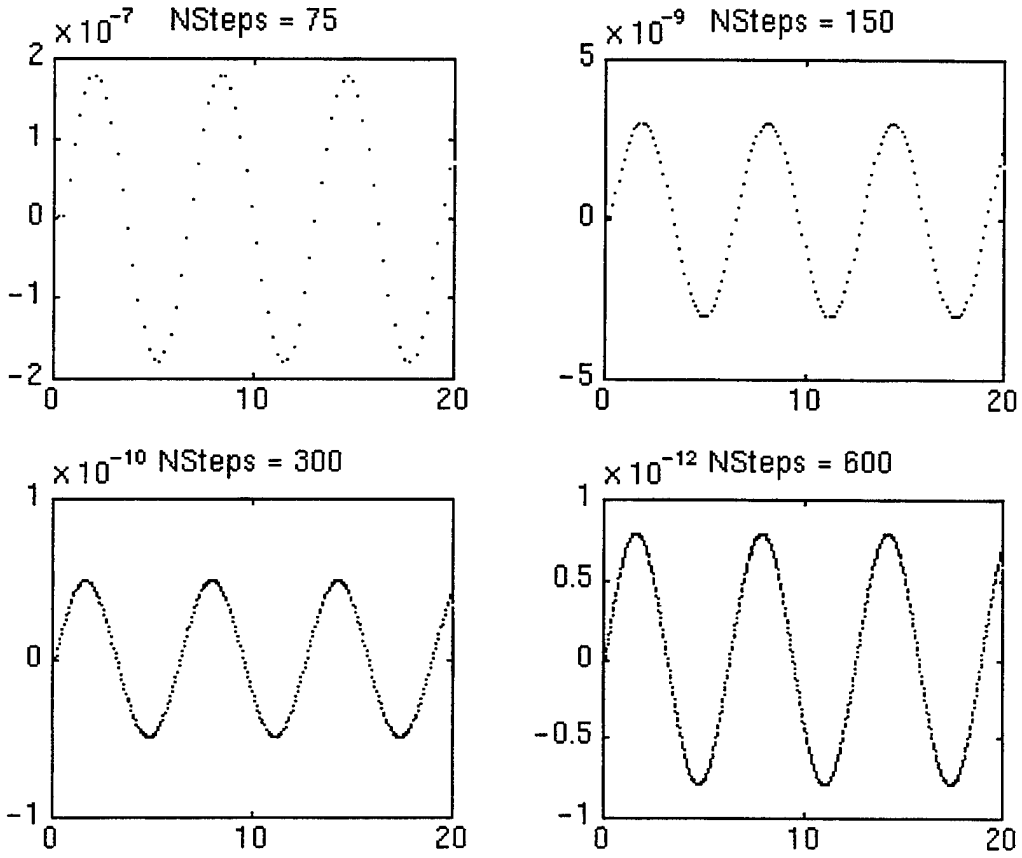



FIGURE 20. Richardson Error Estimate, $\lambda = -2$, Order 5.

TABLE 18. Richardson Error Estimate, $\lambda = -10$, Order 5.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 75 | 0.00 | 1.33 | 2.67 | 8.00 | 26.67 | 80.00 |
| 150 | 0.00 | 0.67 | 0.67 | 2.00 | 2.67 | 98.00 |
| 300 | 0.00 | 0.33 | 1.33 | 2.00 | 16.67 | 99.33 |
| 600 | 0.67 | 0.83 | 1.17 | 5.17 | 94.00 | 99.17 |

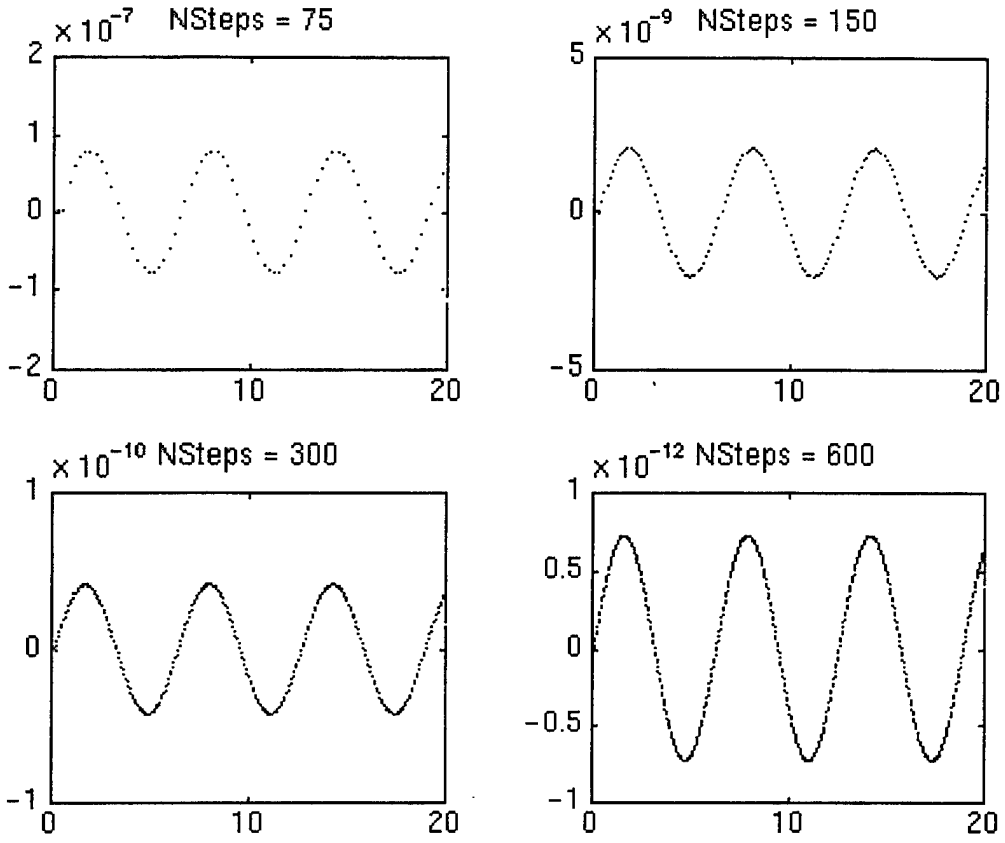



FIGURE 21. Richardson Error Estimate, $\lambda = -10$, Order 5.

TABLE 19. Richardson Error Estimate, $\lambda = -100$, Order 5

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 75 | 0.00 | 2.67 | 2.67 | 6.67 | 13.33 | 21.33 |
| 150 | 0.00 | 0.67 | 0.67 | 3.33 | 5.33 | 11.33 |
| 300 | 0.00 | 0.00 | 0.00 | 0.67 | 0.17 | 5.00 |
| 600 | 0.00 | 0.00 | 0.17 | 0.17 | 0.33 | 1.17 |

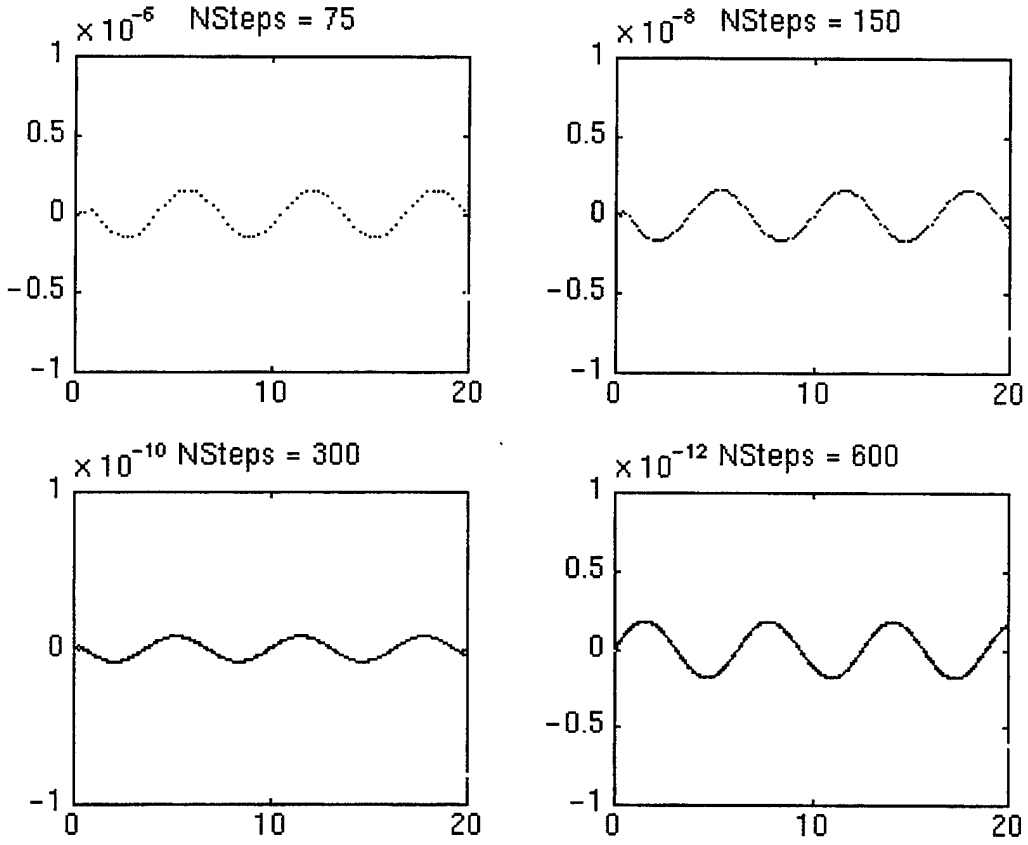



FIGURE 22. Richardson Error Estimate, $\lambda = -100$, Order 5.

The results at $\lambda = -100$ could only be described as very poor. Some experimentation resulted in a different heuristic estimate for this method that seemed to be effective with increasing stiffness and less accuracy, the formula

$$err = \frac{y_h^n - y_{2h}^n}{2^{p-1}}. \tag{47}$$

The following tests shown in Tables 20 through 22 and Figures 23 through 25 were conducted. Fewer points were used to avoid roundoff problems at high accuracy levels. The value $\lambda = -5732$ was chosen to eliminate coincidental dependence on powers of 10. Actual global convergence as shown by endpoint error ratios was sixth order for these problems.

TABLE 20. Heuristic Stiff Richardson Error Estimate, $\lambda = -1000$, Order = 5.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 75 | 1.33 | 5.33 | 9.33 | 22.67 | 41.33 | 72.00 |
| 150 | 2.00 | 8.00 | 15.33 | 35.33 | 59.33 | 82.00 |
| 300 | 3.33 | 15.00 | 28.00 | 56.33 | 76.33 | 90.33 |
| 600 | 4.50 | 22.50 | 37.00 | 69.83 | 85.00 | 94.17 |

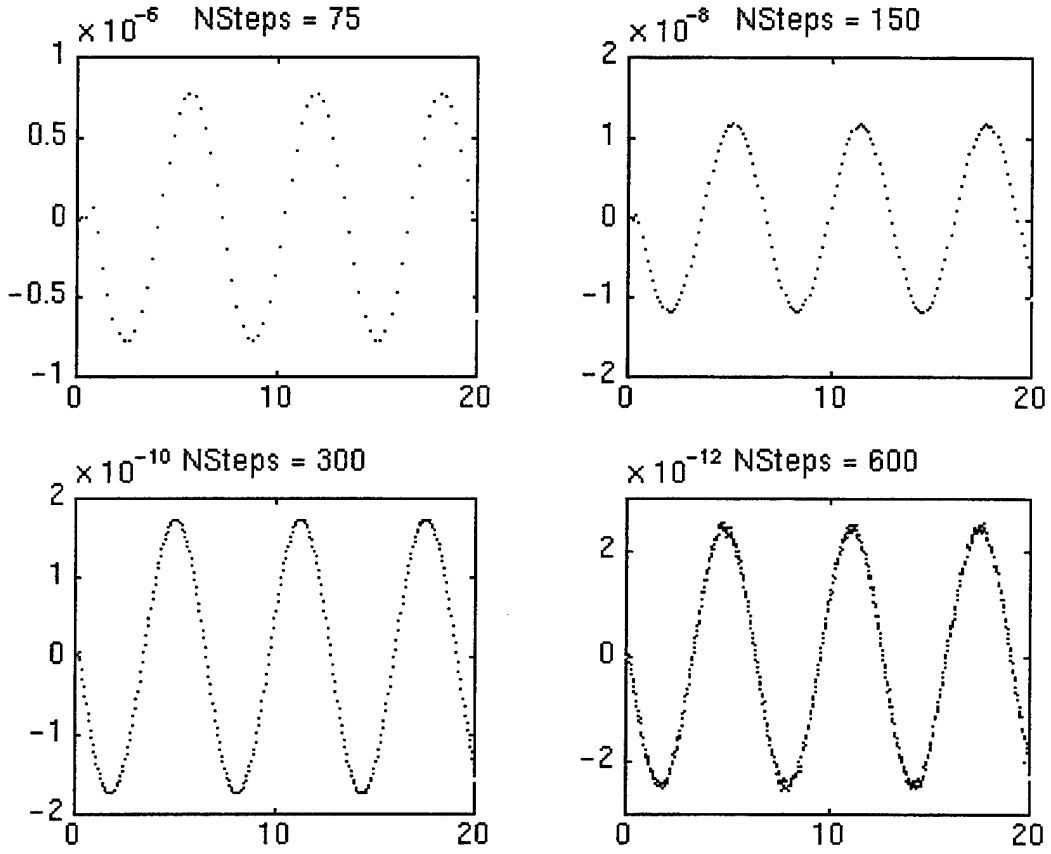



FIGURE 23. Heuristic Stiff Richardson Error Estimate, $\lambda = -1000$, Order 5.

TABLE 21. Heuristic Stiff Richardson Error Estimate, $\lambda = -5732$, Order = 5.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 75 | 1.33 | 6.67 | 9.33 | 25.33 | 41.33 | 72.00 |
| 150 | 1.33 | 8.00 | 15.33 | 37.33 | 60.67 | 83.33 |
| 300 | 2.33 | 15.00 | 29.67 | 57.67 | 78.33 | 91.00 |
| 600 | 2.33 | 12.50 | 22.33 | 51.17 | 79.83 | 93.67 |

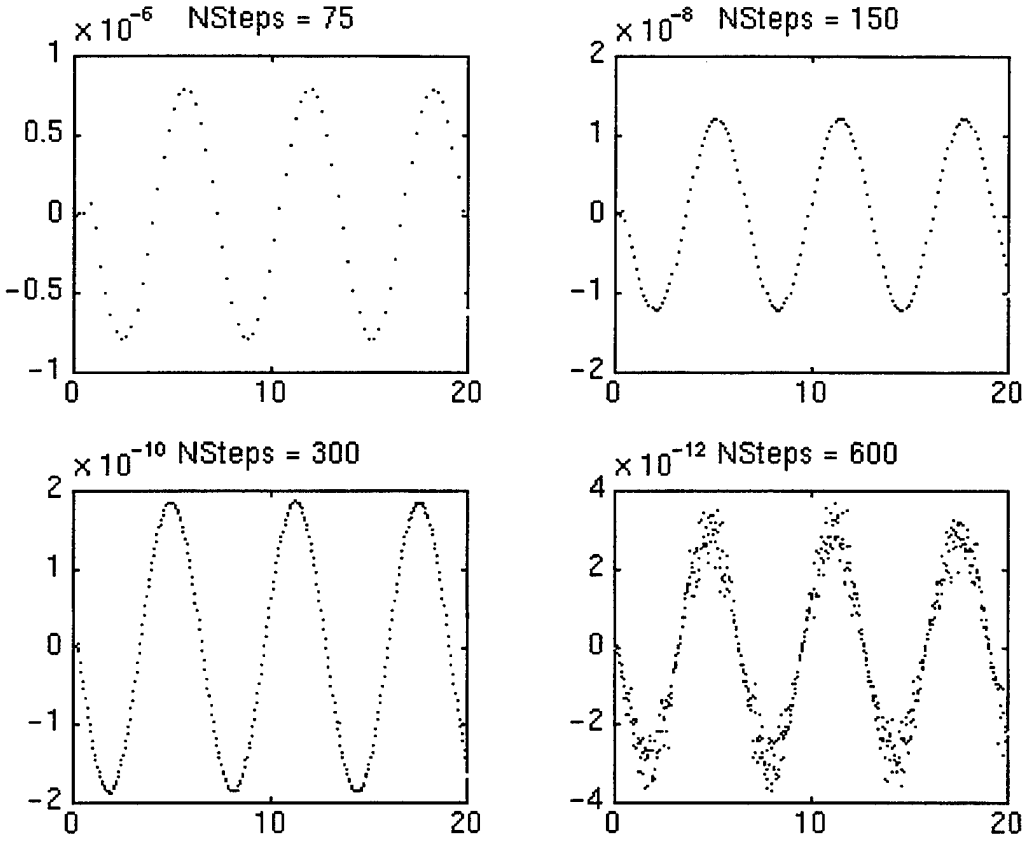



FIGURE 24. Heuristic Stiff Richardson Error Estimate, $\lambda$ = -5732, Order 5.

TABLE 22. Heuristic Stiff Richardson Error Estimate, $\lambda$ = -100, Order = 5.

| Npts | % < 0.01 | % < 0.05 | % < 0.1 | % < 0.25 | % < 0.5 | % < 1 |
|---|---|---|---|---|---|---|
| 75 | 0.00 | 4.00 | 8.00 | 21.33 | 38.67 | 69.33 |
| 150 | 2.00 | 5.33 | 10.67 | 25.33 | 45.33 | 70.67 |
| 300 | 0.00 | 1.00 | 2.67 | 6.67 | 12.67 | 29.00 |
| 600 | 2.00 | 14.33 | 60.50 | 92.50 | 96.83 | 98.50 |

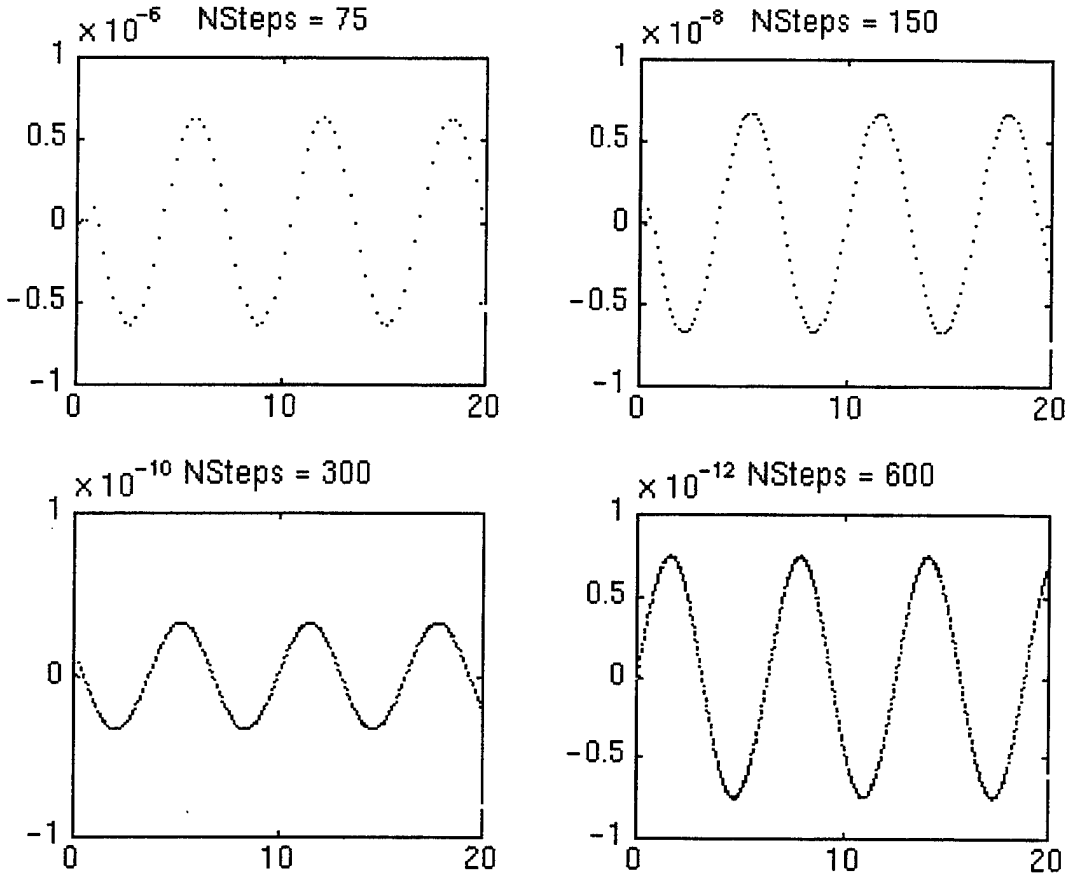



FIGURE 25. Heuristic Stiff Richardson Error Estimate, $\lambda = -100$, Order 5.

We see that there is a range within which neither the Richardson estimate nor the heuristic stiff Richardson estimate is effective, but both have a range of parameters for this problem for which high accuracy is attained, much better than that achieved with any other method. Because there is no significant computational cost, the Richardson approach must be considered as highly competitive for error estimation, especially for higher order where error estimation has suffered from less accuracy, and this despite the problem described earlier of the requirement of fixed step size across two steps. It must be said that the heuristic stiff estimates call for more extensive theoretical study and also testing with other problems with different derivative functions, including systems of equations. But the success for this standard problem provides encouragement that accurate error estimation for DIMSIM stiff solvers is possible.

# PREDICTOR-CORRECTOR IMPLEMENTATIONS

## ALTERNATIVE APPROACHES

Because of the high stage order of DIMSIMs, it is possible to provide a predicted internal stage value for implicit methods based on a Taylor series expansion using the Nordsieck vector, which is readily available without additional function evaluations. This prediction is invaluable in providing a starting point for either a modified Newton solver for stage values or for a function iteration approach. For a method with $c_1 = 0$ and $c_s = 1$, the final internal stage of the last step may instead be used as a predictor for the first stage. In the latter mode, which provides the basis for predictor-corrector methods, the internal stage formula is used to provide corrections in a fixed point iteration.

$$Y_{i,k}^{[n]} = h_n \sum_{j=1}^{i-1} a_{i,j} f\left(t_{n-1} + c_j h_n, Y_j^{[n]}\right) + h_n a_{i,i} f\left(t_{n-1} + c_i h_n, Y_{i,k-1}^{[n]}\right) + y_i^{[n-1]}. \tag{48}$$

Convergence testing here is based on the norm of the difference between successive stage value iterates.

There are a number of possible variations on the predictor-corrector idea that may be pursued. There are choices with the predictor. The predictor might include a term of degree $p + 1$ by using the error estimate from the previous step along with the Nordsieck vector. A FASAL implementation may be used to provide the first stage if $c_1 = 0$ and $c_s = 1$. Since it is the derivative function of the internal stage that is used, all types of iterations will terminate with an evaluation, but some fixed number m of corrector iterations might be used, including 0 for a PE (Predict-Evaluate) method, 1 for a PECE (Predict-Evaluate-Correct-Evaluate) method, and in general, $PE(CE)^m$ for any nonnegative integer m. And this fixed number might be different for each stage. Methods of this type are of course actually explicit methods. The family of $P(EC)^m$ methods, well-known with linear multistep methods, is not so relevant here because it is the derivative resulting from the evaluation step that is the goal of the stage calculations. On the other hand, iteration to convergence might be pursued, yielding true function iteration stage calculations for the implicit DIMSIM, but this would not be suitable for stiff problems due to difficulties in obtaining fixed point iteration convergence for truly stiff problems. With accurate prediction, the variable number of fixed point iterations required in any given calculation might be a very small number, however, perhaps in the range of 0 to 3. LSODE (Reference 7), which includes a typical implementation for predictor-corrector methods with iteration to convergence using linear multistep methods, allows only up to perhaps three iterations; and if convergence has not been achieved, the step size is shortened and the calculation for the step is repeated in the same way as is done when the error estimate indicates that required accuracy was not achieved. Note that L stability will not be an important criterion for the DIMSIM chosen for any of these approaches since they are suitable for no more than moderately stiff problems, but A-stability should be provided to minimize stability restrictions for the implementation.

A very significant variant to explore that has strong potential for reducing the number of function evaluations required is to predict the derivative function value instead of the

internal stage value itself, since the Taylor series terms of the Nordsieck vector may also be used to calculate a derivative, and this would result in what might be described as $P(CE)^m$ methods. Here a derivative is predicted and the stage formula provides a corrected stage value. The derivative function is applied first at this point in the calculation and convergence is determined from the norm of the difference between successive derivative values scaled with the step size. It should be noted that it is only the derivative function of the stage values that is used and not the stage values themselves, and so this provides not only an application of the stage formula before any evaluations but also a convergence measure that directly relates to the desired result.

If the stage values are calculated with sufficient accuracy, the DIMSIM formulation for error estimates, Nordsieck vectors, rescaling for step-size changes, and interpolation should remain valid. But the stability analysis must be modified to reflect the specific form of the predictor and iteration process used.

## RELATIONSHIPS FOR STABILITY ANALYSIS

Assuming a standard form for $\tilde{V}$ of $\hat{e}_1 v^T$, we have the Nordsieck vector given by

$$\hat{y}(t_{n-1},h) = h\tilde{B}F\left(Y^{[n-1]}\right) + \hat{e}_1 v^T y^{[n-2]},$$

where explicit reference to the dependency on h is included. Some simplification is achieved by eliminating the troublesome reference to $v^T y^{[n-2]}$ by using the relationship

$$y^{[n-1]} = hBF\left(Y^{[n-1]}\right) + ev^T y^{[n-2]}.$$

We may choose the first equation and write

$$v^T y^{[n-2]} = y_1^{[n-1]} - hB_1 F\left(Y^{[n-1]}\right),$$

where $B_1$ refers to the first row of the matrix B. Then

$$\hat{y}(t_{n-1},h) = h\tilde{B}F\left(Y^{[n-1]}\right) + e\left(y_1^{[n-1]} - hB_1 F\left(Y^{[n-1]}\right)\right).$$

Another formula is available. For if we write

$$Y^{[n-1]} = hAF\left(Y^{[n-1]}\right) + y^{[n-2]},$$

we see that

$$y^{[n-2]} = Y^{[n-1]} - hAF\left(Y^{[n-1]}\right).$$

If the error estimate is used, the external stage vector appears in a different form:

$$v^T le(t_{n-1}) = h\beta^T F\left(Y^{[n-1]}\right) + \gamma^T y^{[n-2]}.$$

To use standard general linear method notation, the vector representing the external stages must then include the internal stages, which are also carried along for the next step. The stability matrix will then be (2p)x(2p). If only the derivative is predicted, the analysis is immediately simplified since the first component of the Nordsieck vector is not used and this is the only component that directly depends on $y^{[n-2]}$.

## STABILITY ANALYSIS FOR A SECOND ORDER DIMSIM

A particular L-stable Type 2 second order DIMSIM with c = [0,1] was selected for analysis of stability with alternative approaches to prediction and varying numbers of corrections. The method has Butcher tableau

$$\begin{bmatrix} \frac{25}{24} & 0 & 1 & 0 \\ \frac{311}{324} & \frac{25}{24} & 0 & 1 \\ \frac{16477}{10368} & -\frac{75}{128} & \frac{225}{208} & -\frac{17}{208} \\ \frac{22093}{10368} & -\frac{11275}{10368} & \frac{225}{208} & -\frac{17}{208} \end{bmatrix}.$$

W is given by

$$W = \begin{bmatrix} 1 & -\frac{25}{24} & 0 \\ 1 & -\frac{649}{648} & -\frac{13}{24} \end{bmatrix},$$

and the rescaling matrices are given by

$$\tilde{B}=\begin{bmatrix} \frac{16477}{10368} & \frac{175}{384} \\ 0 & 1 \\ -1 & 1 \end{bmatrix} \text{ and } \tilde{V}=\begin{bmatrix} \frac{225}{208} & -\frac{17}{208} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

We note that for the rescaling matrices in this form, the second component of the Nordsieck vector is h times the derivative function applied to the second stage, while the third component is h times the difference between the two stages as the derivative function is applied. There is then no difference between using the second stage in predicting the derivative function of the first stage at the next step and using the Nordsieck vector. The error constant is -5/128.

1) Stage 1 P, Stage 2 PCE (1 Evaluation)

Here the derivative of stage 2 of the previous step is used to predict the derivative at the first step using the FASAL property. High accuracy is expected, but the FASAL implementation of the DIMSIM is not A-stable. The Nordsieck vector from the previous step is used to obtain a prediction for the derivative of the second stage. This is then used with the second stage formula to provide a correction. The derivative function is then applied with the new stage value. We thus compute

$$Y_1^{[n]} = Y_2^{[n-1]}$$
$$Y_2^{[n]} = ha_{21}F\left(Y_2^{[n-1]}\right) + a_{22}\left(\hat{y}_2^{[n-1]} + \hat{y}_2^{[n-1]}\right) + y_2^{[n-1]}.$$

We note that

$$\hat{y}^{[n-1]} = h\tilde{B}F\left(Y^{[n-1]}\right) + \tilde{V}y^{[n-2]},$$

so

$$\hat{y}_2^{[n-1]} + \hat{y}_3^{[n-1]} = h\left(\tilde{B}_2 + \tilde{B}_3\right)F\left(Y^{[n-1]}\right).$$

Note that single subscripts on matrices are used to denote row numbers. Then

$$Y_2^{[n]} = ha_{21}F\left(Y_2^{[n-1]}\right) + a_{22}h\left(\tilde{B}_2 + \tilde{B}_3\right)F\left(Y^{[n-1]}\right) + y_2^{[n-1]}.$$

We now use the test equation y' = λy and set z = hλ to yield

$$Y_2^{[n]} = a_{21} z Y_2^{[n-1]} + a_{22} z\left(\tilde{B}_2 + \tilde{B}_3\right) Y^{[n-1]} + y_2^{[n-1]}.$$

We then calculate

$$y^{[n]} = hBF\left(Y^{[n]}\right) + Vy^{[n-1]} = zBY^{[n]} + Vy^{[n-1]}.$$

We use the values calculated for the internal stages to rewrite this in terms of the previous external and internal stages:

$$y^{[n]} = zB\begin{bmatrix} Y_2^{[n-1]} \\ za_{21}Y_2^{[n-1]} + za_{22}\left(\tilde{B}_2 + \tilde{B}_3\right)Y^{[n-1]} + y_2^{[n-1]} \end{bmatrix} + Vy^{[n-1]}.$$

We now may assemble this into a matrix form expressing

$$\begin{bmatrix} Y_1^{[n]} \\ Y_2^{[n]} \\ y_1^{[n]} \\ y_2^{[n]} \end{bmatrix} = M \begin{bmatrix} Y_1^{[n-1]} \\ Y_2^{[n-1]} \\ y_1^{[n-1]} \\ y_2^{[n-1]} \end{bmatrix},$$

where M is the stability matrix for this predictor-corrector implementation. We identify M from the foregoing calculations as

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{22}\left(\tilde{B}_{21} + \tilde{B}_{31}\right)z & a_{21}z + a_{22}\left(\tilde{B}_{22} + \tilde{B}_{32}\right)z & 0 & 1 \\ z^2 b_{12} a_{22}\left(\tilde{B}_{21} + \tilde{B}_{31}\right) & b_{12}a_{21}z^2 + b_{11}z + z^2 b_{12}a_{22}\left(\tilde{B}_{22} + \tilde{B}_{32}\right) & v & b_{12}z + 1 - v \\ z^2 b_{22} a_{22}\left(\tilde{B}_{21} + \tilde{B}_{31}\right) & b_{22}a_{21}z^2 + b_{21}z + z^2 b_{22}a_{22}\left(\tilde{B}_{22} + \tilde{B}_{32}\right) & v & b_{22}z + 1 - v \end{bmatrix}.$$

For the method described above, we find that

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{25z}{24} & \frac{493z}{162} & 0 & 1 \\ \frac{625z^2}{1024} & \frac{32954z - 36975z^2}{20736} & \frac{225}{208} & -\frac{136+975z}{1664} \\ \frac{281875z^2}{248832} & \frac{3579066z - 5558575z^2}{1679616} & \frac{225}{208} & -\frac{11016+146575z}{134784} \end{bmatrix}. \quad (49)$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 - \left(1 + \tfrac{751z}{384}\right)w^3 + \tfrac{271z}{192}w^2 - \tfrac{175z}{384}w. \quad (50)$$

The stability region is calculated from the w roots of p. Note that 1 root is 0, leaving 3 nonzero roots. We establish a 150x150 grid of z values in the complex plane and for each z calculate the root w that has the maximum modulus. A contour plot was then obtained corresponding to a maximum modulus of 1. The same technique will be used for each of the following stability region calculations. For this case the stability region was obtained as shown in Figure 26.

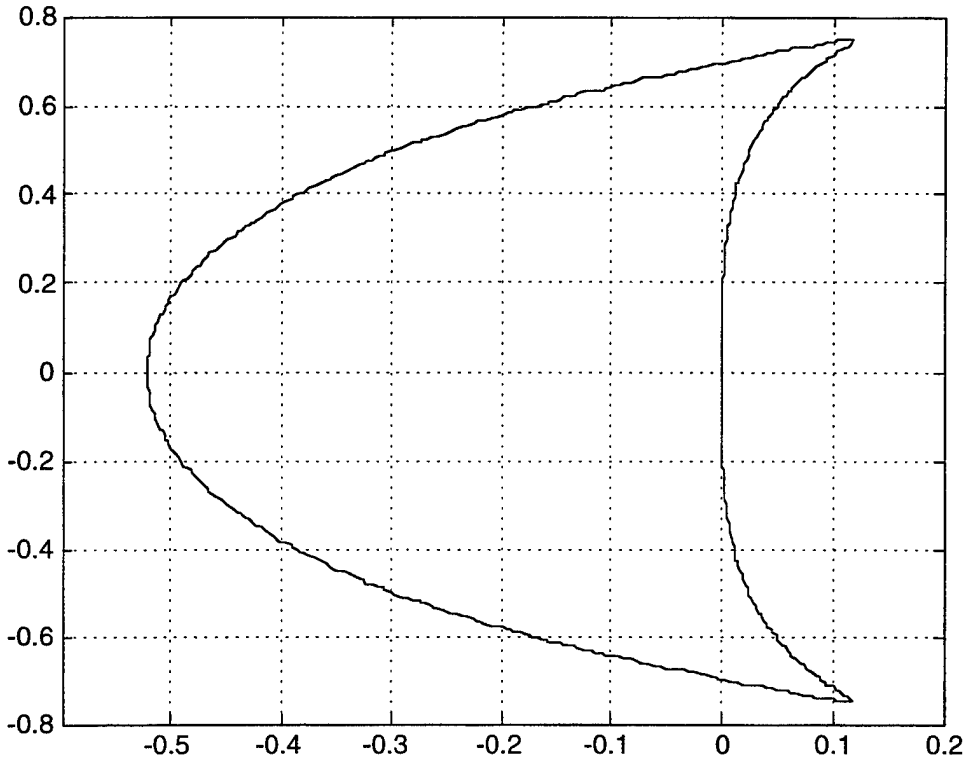



FIGURE 26. Stage 1 P, Stage 2 PCE.

The real axis is included to - 0.52.

2) Stage 1 PCE, Stage 2 PCE (2 Evaluations)

Here the derivative of stage 2 of the previous step is used to predict the derivative at the first step using the FASAL property. High accuracy is expected, and one iteration with the stage formula is used to provide extra stability from correction and evaluation.

The Nordsieck vector from the previous step is again used to obtain a prediction for the derivative of the second stage and the process is carried through as in implementation 1 described above. We thus compute

$$Y_1^{[n]} = a_{11} h\lambda Y_2^{[n-1]} + y_1^{[n-1]}$$

$$Y_2^{[n]} = h a_{21} F\left(Y_1^{[n]}\right) + a_{22}\left(\hat{y}_2^{[n-1]} + \hat{y}_2^{[n-1]}\right) + y_2^{[n-1]}.$$

We again note that

$$\hat{y}_2^{[n-1]} + \hat{y}_3^{[n-1]} = h\left(\tilde{B}_2 + \tilde{B}_3\right)F\left(Y^{[n-1]}\right).$$

Then

$$Y_2^{[n]} = ha_{21}F\left(Y_1^{[n]}\right) + a_{22}h\left(\tilde{B}_2 + \tilde{B}_3\right)F\left(Y^{[n-1]}\right) + y_2^{[n-1]}.$$

We now use the test equation y' = λy and the value of the first internal stage to yield, with z = hλ,

$$Y_2^{[n]} = za_{21}\left(a_{11}zY_2^{[n-1]} + y_1^{[n-1]}\right) + a_{22}z\left(\tilde{B}_2 + \tilde{B}_3\right)Y^{[n-1]} + y_2^{[n-1]}.$$

We then calculate

$$y^{[n]} = hBF\left(Y^{[n]}\right) + Vy^{[n-1]} = zBY^{[n]} + Vy^{[n-1]}.$$

We use the values calculated for the internal stages to rewrite this in terms of the previous external and internal stages:

$$y^{[n]} = zB\begin{bmatrix} a_{11}zY_1^{[n-1]} + y_1^{[n-1]} \\ za_{21}\left(a_{11}zY_1^{[n-1]} + y_1^{[n-1]}\right) + za_{22}\left(\tilde{B}_2 + \tilde{B}_3\right)Y^{[n-1]} + y_2^{[n-1]} \end{bmatrix} + Vy^{[n-1]}.$$

We again assemble this into a matrix form expressing

$$\begin{bmatrix} Y_1^{[n]} \\ Y_2^{[n]} \\ y_1^{[n]} \\ y_2^{[n]} \end{bmatrix} = M\begin{bmatrix} Y_1^{[n-1]} \\ Y_2^{[n-1]} \\ y_1^{[n-1]} \\ y_2^{[n-1]} \end{bmatrix},$$

where M is the stability matrix for this predictor-corrector implementation. We identify M from the foregoing calculations as

$$M = \begin{bmatrix} 0 & a_{11}z & 1 & 0 \\ a_{22}(\tilde{B}_{21}+\tilde{B}_{31})z & a_{21}a_{11}z^2 + a_{22}(\tilde{B}_{22}+\tilde{B}_{32})z & a_{21}z & 1 \\ z^2 b_{12}a_{22}(\tilde{B}_{21}+\tilde{B}_{31}) & b_{12}a_{21}a_{11}z^3 + b_{11}a_{11}z^2 + z^2 b_{12}a_{22}(\tilde{B}_{22}+\tilde{B}_{32}) & v + b_{11}z + b_{12}a_{21}z^2 & b_{12}z + 1 - v \\ z^2 b_{22}a_{22}(\tilde{B}_{21}+\tilde{B}_{31}) & b_{22}a_{21}a_{11}z^3 + b_{21}a_{11}z^2 + z^2 b_{22}a_{22}(\tilde{B}_{22}+\tilde{B}_{32}) & v + b_{21}z + b_{22}a_{21}z^2 & b_{22}z + 1 - v \end{bmatrix} \quad (51)$$

For the method described above, we find that

$$M = \begin{bmatrix} 0 & \frac{25z}{24} & 1 & 0 \\ -\frac{25z}{24} & \frac{25(648z+311z^2)}{7776} & \frac{311z}{324} & 1 \\ \frac{625z^2}{1024} & -\frac{25(-17308z^2+23325z^3)}{995328} & \frac{\begin{pmatrix}583200\\+856804z\\-303225z^2\end{pmatrix}}{539136} & -\frac{136+975z}{1664} \\ \frac{281875z^2}{248832} & -\frac{25(148068z^2+3506525z^3)}{80621568} & \frac{\begin{pmatrix}47239200\\+93055716z\\-45584825z^2\end{pmatrix}}{43670016} & -\frac{11016+146575z}{134784} \end{bmatrix}$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1 - \frac{1489z}{576} - \frac{54425z^2}{124416}\right)w^3 + \left(\frac{913z}{576} + \frac{127325z^2}{62208}\right)w^2 - \frac{65225z^2}{124416}w. \tag{52}$$

The stability region was obtained as shown in Figure 27 using the same technique described above.

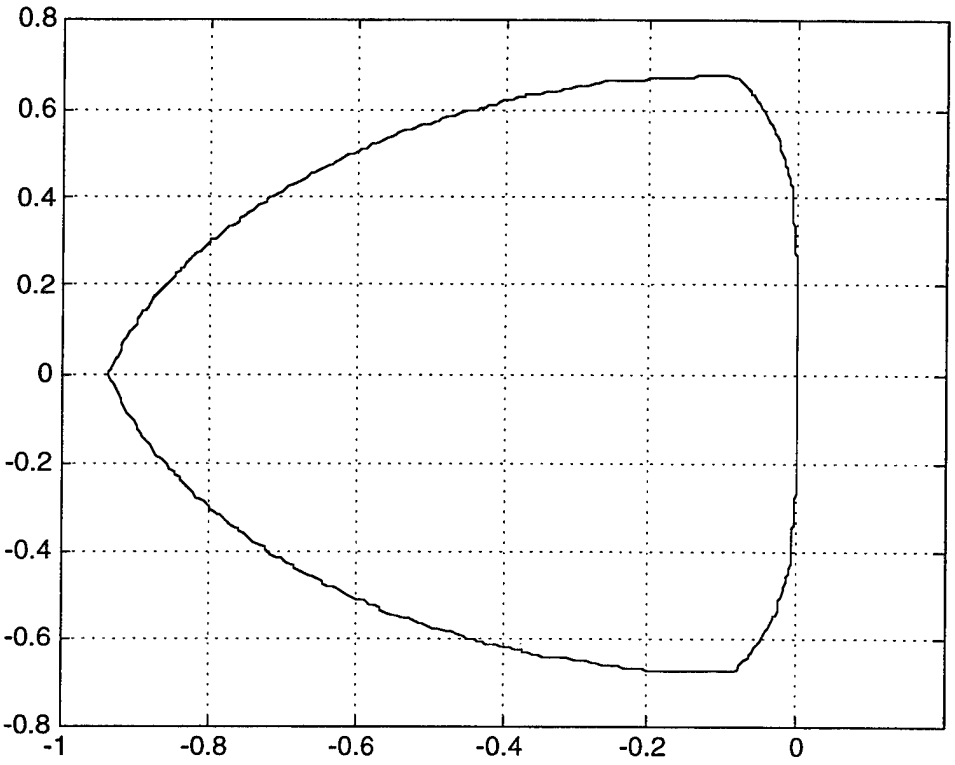



FIGURE 27. Stage 1 PCE, Stage 2 PCE.

This includes the negative real axis to - 0.93.

3) Stage 1 P, Stage 2 PECE (2 Evaluations)

Here we use the stage value derivative from the previous step for the first stage and use the Nordsieck vector from the previous step to predict the value of the second stage. The derivative function is applied and the stage formula is used to produce a corrected value. Its accuracy may be tested against the predicted value in an implementation where multiple corrections may be used. The derivative function is then applied for a second time to produce the value to be used in subsequent calculations. We thus compute

corrections may be used. The derivative function is then applied for a second time to produce the value to be used in subsequent calculations. We thus compute

$$Y_1^{[n]} = Y_2^{[n-1]}$$

$$Y_2^{[n]} = ha_{21}F\left(Y_2^{[n-1]}\right) + ha_{22}F\left(\hat{y}_1^{[n-1]} + \hat{y}_2^{[n-1]} + \tfrac{1}{2}\hat{y}_2^{[n-1]}\right) + y_2^{[n-1]}.$$

Since

$$\hat{y}^{[n-1]} = h\tilde{B}F\left(Y^{[n-1]}\right) + \tilde{V}y^{[n-2]},$$

we have

$$\hat{y}_1^{[n-1]} + \hat{y}_2^{[n-1]} + \tfrac{1}{2}\hat{y}_3^{[n-1]} = h\left(\tilde{B}_1 + \tilde{B}_2 + \tfrac{1}{2}\tilde{B}_3\right)F\left(Y^{[n-1]}\right) + v^T y^{[n-2]}.$$

We make a substitution to eliminate reference to $v^T y^{[n-2]}$. Then

$$\hat{y}_1^{[n-1]} + \hat{y}_2^{[n-1]} + \tfrac{1}{2}\hat{y}_3^{[n-1]} = h\left(\tilde{B}_1 + \tilde{B}_2 + \tfrac{1}{2}\tilde{B}_3\right)F\left(Y^{[n-1]}\right) + y_1^{[n-1]} - hB_1F\left(Y^{[n-1]}\right).$$

$$Y_2^{[n]} = ha_{21}F\left(Y_2^{[n-1]}\right) + a_{22}hF\left(\left(\tilde{B}_1 + \tilde{B}_2 + \tfrac{1}{2}\tilde{B}_3 - B_1\right)hF\left(Y^{[n-1]}\right) + y_1^{[n-1]}\right) + y_2^{[n-1]}.$$

We now use the test equation y' = λy and substitute z = hλ to yield

$$Y_2^{[n]} = za_{21}Y_2^{[n-1]} + a_{22}z^2\left(\tilde{B}_1 + \tilde{B}_2 + \tfrac{1}{2}\tilde{B}_3 - B_1\right)Y^{[n-1]} + a_{22}zy_1^{[n-1]} + y_2^{[n-1]}.$$

We then calculate

$$y^{[n]} = hBF\left(Y^{[n]}\right) + Vy^{[n-1]} = zBY^{[n]} + Vy^{[n-1]}.$$

We use the values calculated for the internal stages to rewrite this in terms of the previous external and internal stages:

$$y^{[n]} = zB\begin{bmatrix} Y_2^{[n-1]} \\ za_{21}Y_2^{[n-1]} + z^2a_{22}\left(\tilde{B}_1 + \tilde{B}_2 + \frac{1}{2}\tilde{B}_3 - B_1\right)Y^{[n-1]} + za_{22}y_1^{[n-1]} + y_2^{[n-1]} \end{bmatrix} + Vy^{[n-1]}.$$

We now may assemble this into a matrix form expressing

$$\begin{bmatrix} Y_1^{[n]} \\ Y_2^{[n]} \\ y_1^{[n]} \\ y_2^{[n]} \end{bmatrix} = M\begin{bmatrix} Y_1^{[n-1]} \\ Y_2^{[n-1]} \\ y_1^{[n-1]} \\ y_2^{[n-1]} \end{bmatrix},$$

where M is the stability matrix for this predictor-corrector implementation. We identify M from the foregoing calculations as

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ z^2a_{22}(\tilde{B}_{11} + \tilde{B}_{21} + \frac{1}{2}\tilde{B}_{31} - B_{11}) & a_{21}z + a_{22}(\tilde{B}_{12} + \tilde{B}_{22} + \frac{1}{2}\tilde{B}_{32} - B_{12})z^2 & a_{22}z & 1 \\ z^3b_{12}a_{22}(\tilde{B}_{11} + \tilde{B}_{21} + \frac{1}{2}\tilde{B}_{31} - B_{11}) & b_{12}a_{21}z^2 + b_{11}z + z^3b_{12}a_{22}(\tilde{B}_{12} + \tilde{B}_{22} + \frac{1}{2}\tilde{B}_{32} - B_{12}) & b_{12}a_{22}z^2 + v & b_{12}z + 1 - v \\ z^3b_{22}a_{22}(\tilde{B}_{11} + \tilde{B}_{21} + \frac{1}{2}\tilde{B}_{31} - B_{11}) & b_{22}a_{21}z^2 + b_{21}z + z^3b_{22}a_{22}(\tilde{B}_{12} + \tilde{B}_{22} + \frac{1}{2}\tilde{B}_{32} - B_{12}) & b_{22}a_{22}z^2 + v & b_{22}z + 1 - v \end{bmatrix}. \quad (53)$$

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{25z^2}{48} & \frac{4976z+13725z^2}{5184} & \frac{25z}{24} & 1 \\ \frac{625z^3}{2048} & \frac{(1054528z-373200z^2-1029375z^3)}{663552} & -\frac{25\left(-576+325z^2\right)}{13312} & -\frac{136+975z}{1664} \\ \frac{281875z^3}{497664} & \frac{(114530112z-56104400z^2-154749375z^3)}{53747712} & \frac{-25(-139968+146575z^2)}{3234816} & -\frac{11016+146575z}{134784} \end{bmatrix}.$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1+\frac{49z}{384}-\frac{18775z^2}{9216}\right)w^3 + \left(-\frac{329z}{192}+\frac{6775z^2}{4608}\right)w^2 + \left(\frac{75z}{128}-\frac{4375z^2}{9216}\right)w . \tag{54}$$

The stability region was obtained as shown in Figure 28.

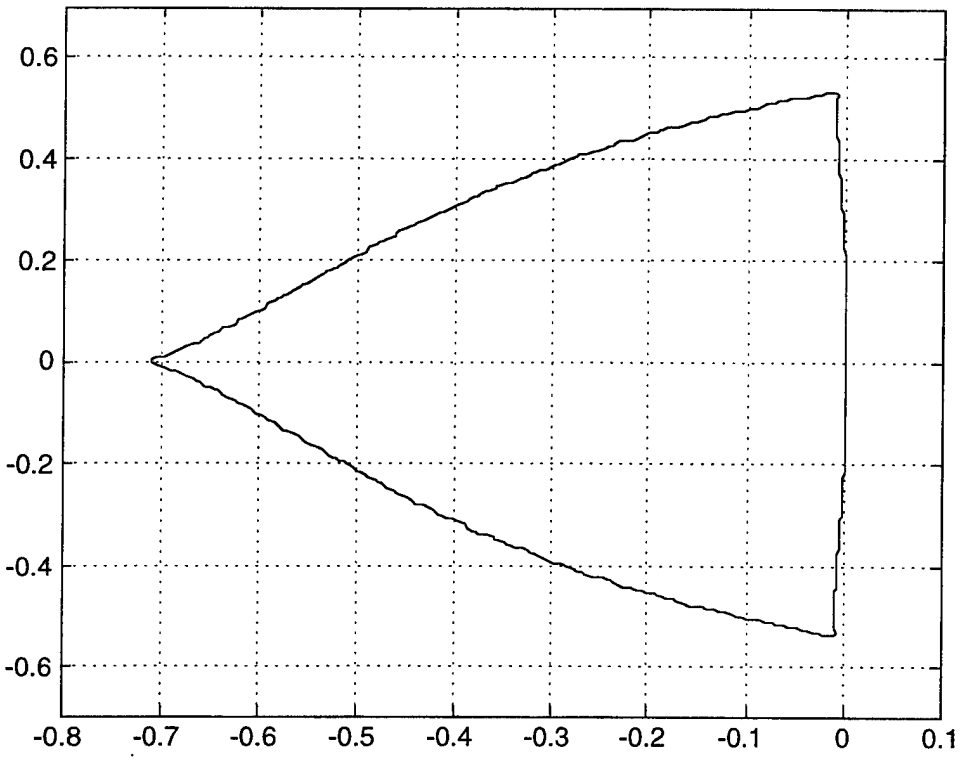



FIGURE 28. Stage 1 P, Stage 2 PECE.

It includes the real axis to -0.71.

4) Stage 1 P, Stage 2 PCECE (2 Evaluations)

The same approach is used as for approach 2, but an additional correction and evaluation is performed for the second stage. We thus compute

$$Y_1^{[n]} = Y_2^{[n-1]}$$

$$Y_{2,1}^{[n]} = ha_{21}F\left(Y_2^{[n-1]}\right) + a_{22}\left(\hat{y}_2^{[n-1]} + \hat{y}_2^{[n-1]}\right) + y_2^{[n-1]}.$$

We again note that

$$\hat{y}_2^{[n-1]} + \hat{y}_3^{[n-1]} = h\left(\tilde{B}_2 + \tilde{B}_3\right)F\left(Y^{[n-1]}\right).$$

Then

$$Y_{2,1}^{[n]} = ha_{21}F\left(Y_2^{[n-1]}\right) + a_{22}h\left(\tilde{B}_2 + \tilde{B}_3\right)F\left(Y^{[n-1]}\right) + y_2^{[n-1]},$$

and

$$Y_2^{[n]} = ha_{21}F\left(Y_2^{[n-1]}\right) + a_{22}hF\left(ha_{21}F\left(Y_2^{[n-1]}\right) + a_{22}h\left(\tilde{B}_2 + \tilde{B}_3\right)F\left(Y^{[n-1]}\right) + y_2^{[n-1]}\right) + y_2^{[n-1]}.$$

We now use the test equation y' = λy and substitute z = hλ to yield

$$Y_2^{[n]} = \left(za_{21} + z^2a_{21}a_{22}\right)Y_2^{[n-1]} + z^2a_{22}^2\left(\tilde{B}_2 + \tilde{B}_3\right)Y^{[n-1]} + \left(za_{22} + 1\right)y_2^{[n-1]}.$$

We then calculate

$$y^{[n]} = hBF\left(Y^{[n]}\right) + Vy^{[n-1]} = zBY^{[n]} + Vy^{[n-1]}.$$

We use the values calculated for the internal stages to rewrite this in terms of the previous external and internal stages:

$$y^{[n]} = zB\begin{bmatrix} Y_2^{[n-1]} \\ \left(za_{21} + z^2a_{21}a_{22}\right)Y_2^{[n-1]} + z^2a_{22}^2\left(\tilde{B}_2 + \tilde{B}_3\right)Y^{[n-1]} + \left(1 + za_{22}\right)y_2^{[n-1]} \end{bmatrix} + Vy^{[n-1]}.$$

We now may assemble this into a matrix form expressing

$$\begin{bmatrix} Y_1^{[n]} \\ Y_2^{[n]} \\ y_1^{[n]} \\ y_2^{[n]} \end{bmatrix} = M \begin{bmatrix} Y_1^{[n-1]} \\ Y_2^{[n-1]} \\ y_1^{[n-1]} \\ y_2^{[n-1]} \end{bmatrix},$$

where M is the stability matrix for this predictor-corrector implementation. We identify M from the foregoing calculations as

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ z^2 a_{22}^2\left(\tilde{B}_{21}+\tilde{B}_{31}\right) & \begin{matrix} a_{21}z + a_{21}a_{22}z^2 + \\ a_{22}^2\left(\tilde{B}_{22}+\tilde{B}_{32}\right)z^2 \end{matrix} & 0 & 1+a_{22}z \\ z^3 b_{12} a_{22}^2\left(\tilde{B}_{21}+\tilde{B}_{31}\right) & \begin{matrix} b_{12}a_{21}z^2 + b_{12}a_{21}a_{22}z^3 + b_{11}z \\ +z^3 b_{12}a_{22}^2\left(\tilde{B}_{22}+\tilde{B}_{32}\right) \end{matrix} & v & b_{12}z(1+a_{22}z)+1-v \\ z^3 b_{22} a_{22}^2\left(\tilde{B}_{21}+\tilde{B}_{31}\right) & \begin{matrix} b_{22}a_{21}z^2 + b_{22}a_{21}a_{22}z^3 + b_{21}z \\ +z^3 b_{22}a_{22}^2\left(\tilde{B}_{22}+\tilde{B}_{32}\right) \end{matrix} & v & b_{22}z(1+a_{22}z)+1-v \end{bmatrix}. \quad (55)$$

For the method described above, we find that

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{625z^2}{576} & \frac{3732z+12325z^2}{3888} & 0 & \frac{24+25z}{24} \\ \frac{15625z^3}{24576} & \frac{(263632z-93300z^2-308125z^3)}{165888} & \frac{225}{208} & \frac{-(1088+7800z+8125z^2)}{13312} \\ \frac{7046875z^3}{5971968} & \frac{(85897584z-42078300z^2-138964375z^3)}{40310784} & \frac{225}{208} & \frac{-(264384+3517800z+3664375z^2)}{3234816} \end{bmatrix}$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1+\frac{49z}{384}-\frac{18775z^2}{9216}\right)w^3 + \left(-\frac{329z}{192}+\frac{6775z^2}{4608}\right)w^2 + \left(\frac{75z}{128}-\frac{4375z^2}{9216}\right)w\,. \tag{56}$$

It is identical to the stability polynomial for the preceding implementation (stage 1 P, stage 2 PECE) and the stability region is of course then also identical. The stability region is then obtained as shown in Figure 29.

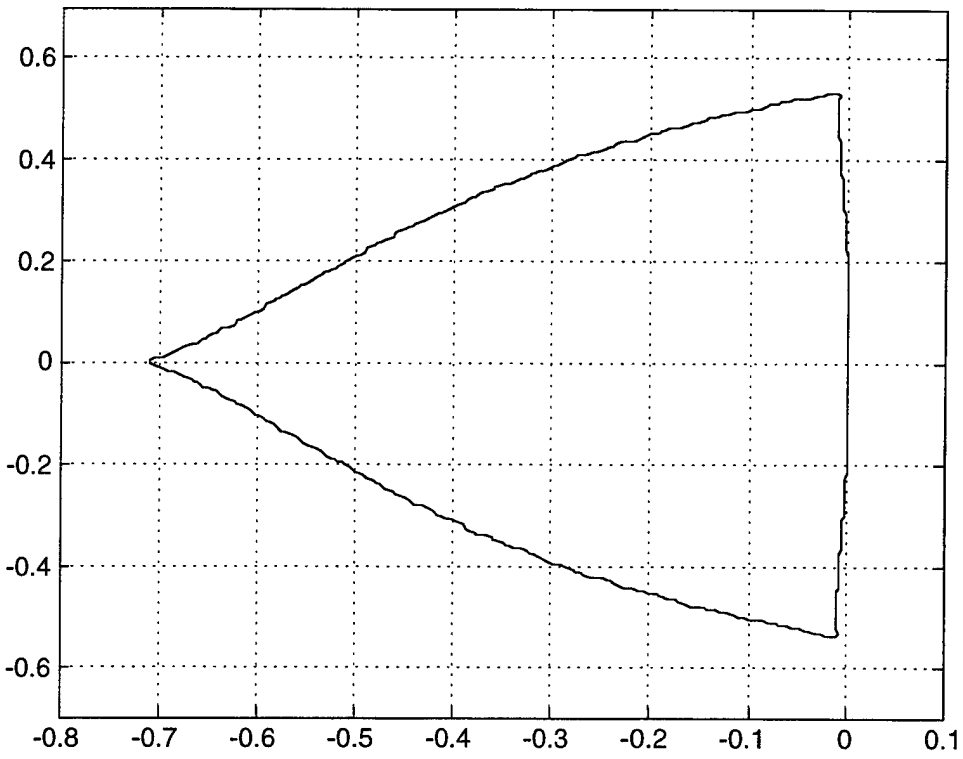



FIGURE 29. Stage 1 P, Stage 2 PCECE.

It includes the real axis to - 0.71.

5) Stage 1 PCE, Stage 2 $P(CE)^2$ (3 Evaluations)

Here the derivative of stage 2 of the previous step is used to predict the derivative at the first step using the FASAL property. High accuracy is expected, and one iteration with the stage formula is used to provide extra stability from correction and evaluation.

The Nordsieck vector from the previous step is again used to obtain a prediction for the derivative of the second stage. The stage formula is used to provide a corrected second stage value and the derivative function is applied to this result. The stage formula is used to provide one more correction to yield the final value for $Y_2$, and the derivative function is applied to this result in computing the external stages. We thus compute

$$Y_1^{[n]} = a_{11} h\lambda Y_2^{[n-1]} + y_1^{[n-1]}$$

$$Y_{2,1}^{[n]} = ha_{21}F\left(Y_1^{[n]}\right) + a_{22}\left(\hat{y}_2^{[n-1]} + \hat{y}_2^{[n-1]}\right) + y_2^{[n-1]}$$

$$= ha_{21}F\left(ha_{11}F\left(Y_2^{[n-1]}\right) + y_1^{[n-1]}\right) + ha_{22}\left(\tilde{B}_2 + \tilde{B}_3\right)F\left(Y^{[n-1]}\right) + y_2^{[n-1]}$$

$$Y_2^{[n]} = ha_{21}F\left(Y_1^{[n]}\right) + a_{22}F\left(Y_{2,1}^{[n]}\right) + y_2^{[n-1]}$$

$$= ha_{21}F\left(ha_{11}F\left(Y_2^{[n-1]}\right) + y_1^{[n-1]}\right)$$

$$+ ha_{22}F\left(ha_{21}F\left(ha_{11}F\left(Y_2^{[n-1]}\right) + y_1^{[n-1]}\right) + ha_{22}\left(\tilde{B}_2 + \tilde{B}_3\right)F\left(Y^{[n-1]}\right) + y_2^{[n-1]}\right)$$

$$+ y_2^{[n-1]}$$

We now use the test equation y' = λy and the value of the first internal stage and set z = hλ to yield

$$Y_1^{[n]} = za_{11}Y_2^{[n-1]} + y_1^{[n-1]}$$

$$Y_2^{[n]} = \left(z^2a_{21}a_{11} + z^3a_{22}a_{21}a_{11}\right)Y_2^{[n-1]} + z^2a_{22}^2\left(\tilde{B}_2 + \tilde{B}_3\right)Y^{[n-1]}$$

$$+ \left(za_{21} + z^2a_{22}a_{21}\right)y_1^{[n-1]} + \left(1 + za_{22}\right)y_2^{[n-1]}$$

We then calculate

$$y^{[n]} = hBF\left(Y^{[n]}\right) + Vy^{[n-1]} = hB\lambda Y^{[n]} + Vy^{[n-1]}$$

We use the values calculated for the internal stages to rewrite this in terms of the previous external and internal stages:

$$y^{[n]} = zB\begin{bmatrix} a_{11}zY_2^{[n-1]} + y_1^{[n-1]} \\ \left(z^2a_{21}a_{11} + z^3a_{22}a_{21}a_{11}\right)Y_2^{[n-1]} + z^2a_{22}^2\left(\tilde{B}_2 + \tilde{B}_3\right)Y^{[n-1]} \\ +\left(za_{21} + z^2a_{22}a_{21}\right)y_1^{[n-1]} + \left(1 + za_{22}\right)y_2^{[n-1]} \end{bmatrix} + Vy^{[n-1]}$$

We again assemble this into a matrix form expressing

$$\begin{bmatrix} Y_1^{[n]} \\ Y_2^{[n]} \\ y_1^{[n]} \\ y_2^{[n]} \end{bmatrix} = M \begin{bmatrix} Y_1^{[n-1]} \\ Y_2^{[n-1]} \\ y_1^{[n-1]} \\ y_2^{[n-1]} \end{bmatrix}$$

where M is the stability matrix for this predictor-corrector implementation. We identify M from the foregoing calculations as

$$M = \begin{bmatrix} 0 & a_{11}z & 1 & 0 \\ & a_{21}a_{11}z^2 + a_{22}a_{21}a_{11}z^3 & & \\ a_{22}^2(\tilde{B}_{21} + \tilde{B}_{31})z^2 & +a_{22}^2(\tilde{B}_{22} + \tilde{B}_{32})z^2 & a_{21}z(1 + za_{22}) & 1 + za_{22} \\ & b_{12}a_{21}a_{11}z^3 + b_{11}a_{11}z^2 & & \\ z^3 b_{12}a_{22}^2(\tilde{B}_{21} + \tilde{B}_{31}) & +z^3 b_{12}a_{22}^2(\tilde{B}_{22} + \tilde{B}_{32}) & v + b_{11}z & b_{12}z(1 + za_{22}) \\ & +b_{12}a_{22}a_{21}a_{11}z^4 & +b_{12}a_{21}z^2(1 + za_{22}) & +1 - v \\ & b_{22}a_{21}a_{11}z^3 + b_{21}a_{11}z^2 & & \\ z^3 b_{22}a_{22}^2(\tilde{B}_{21} + \tilde{B}_{31}) & +z^3 b_{22}a_{22}^2(\tilde{B}_{22} + \tilde{B}_{32}) & v + b_{21}z & b_{22}z(1 + za_{22}) \\ & +b_{22}a_{22}a_{21}a_{11}z^4 & +b_{22}a_{21}z^2(1 + za_{22}) & +1 - v \end{bmatrix} \quad (57)$$

For the method described above, we find that

$$M = \begin{bmatrix} 0 & \frac{25z}{24} & 1 & 0 \\ -\frac{625z^2}{576} & \frac{25(23664z^2+7775z^3)}{186624} & \frac{311z(24+25z)}{7776} & \frac{24+25z}{24} \\ \frac{15625z^3}{24576} & -\frac{25(-527264z^2+591600z^3+194375z^4)}{7962624} & \frac{(4665600+6854432z-2425800z^2-2526875z^3)}{4313088} & \frac{-(1088+7800z+8125z^2)}{13312} \\ \frac{7046875z^3}{5971968} & -\frac{25(-171795168z^2+266811600z^3+87663125z^4)}{1934917632} & \frac{(1133740800+2233337184z-1094035800z^2-1139620625z^3)}{1048080384} & \frac{-(264384+3517800z+3664375z^2)}{3234816} \end{bmatrix}$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1 - \frac{289z}{576} - \frac{40775z^2}{27648} - \frac{1360625z^3}{2985984}\right)w^3 + \left(-\frac{287z}{576} - \frac{925z^2}{6912} + \frac{3183125z^3}{1492992}\right)w^2 + \left(\frac{625z^2}{1024} - \frac{1630625z^3}{2985984}\right)w \tag{58}$$

The stability region was obtained as shown in Figure 30.

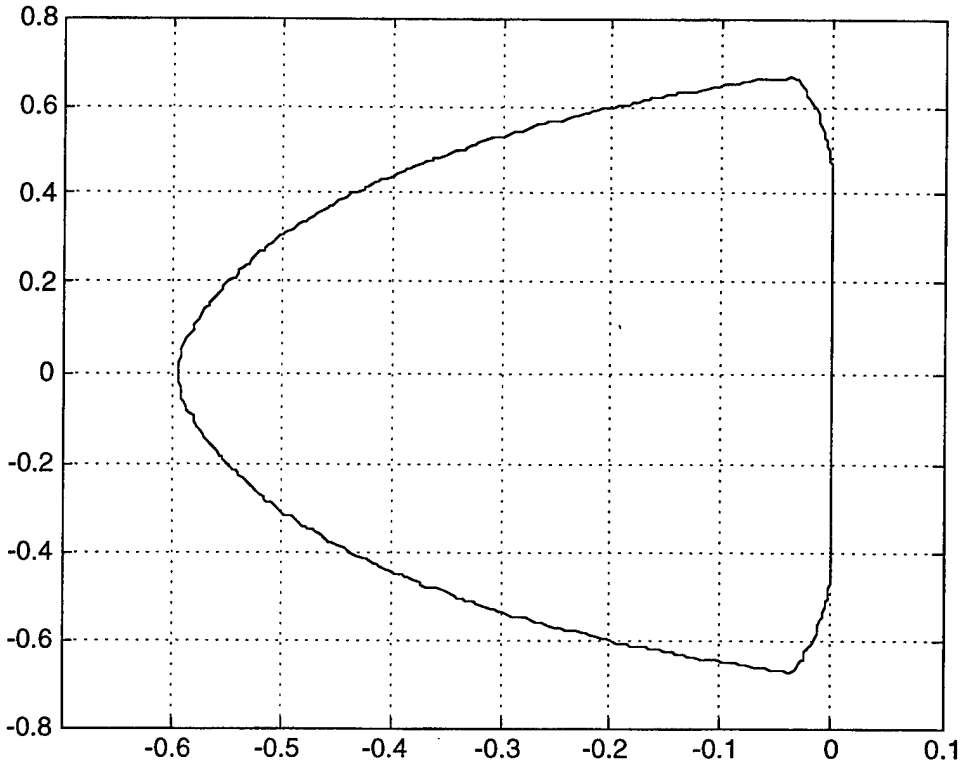



FIGURE 30. Stage 1 PCE, Stage 2 PCECE.

This includes the negative real axis to - 0.59.

6) Stage 1 PCE, Stage 2 PECE (3 Evaluations)

Here we use the stage value derivative from the previous step for the first stage to predict the stage value derivative for the first stage and then carry out a correction and evaluation using the first stage formula. The size of the change may be evaluated as a convergence test. We then use the Nordsieck vector from the previous step to predict the value of the second stage. The derivative function is applied and the stage formula is used to produce a corrected value. Its accuracy may be tested against the predicted value in an implementation where multiple corrections may be used. The derivative function is then applied for a second time to produce the value to be used in subsequent calculations. We thus compute

$$Y_1^{[n]} = ha_{11}F\left(Y_2^{[n-1]}\right) + y_1^{[n-1]}$$

$$Y_2^{[n]} = ha_{21}F\left(Y_1^{[n]}\right) + a_{22}hF\left(\hat{y}_1^{[n-1]} + \hat{y}_2^{[n-1]} + \frac{1}{2}\hat{y}_2^{[n-1]}\right) + y_2^{[n-1]}$$

Since

$$\hat{y}^{[n-1]} = h\tilde{B}F\left(Y^{[n-1]}\right) + \tilde{V}y^{[n-2]}$$

we have

$$\hat{y}_1^{[n-1]} + \hat{y}_2^{[n-1]} + \tfrac{1}{2}\hat{y}_3^{[n-1]} = h\left(\tilde{B}_1 + \tilde{B}_2 + \tfrac{1}{2}\tilde{B}_3\right)F\left(Y^{[n-1]}\right) + v^T y^{[n-2]}$$

We make a substitution to eliminate reference to $v^T y^{[n-2]}$. Then

$$\hat{y}_1^{[n-1]} + \hat{y}_2^{[n-1]} + \tfrac{1}{2}\hat{y}_3^{[n-1]} = h\left(\tilde{B}_1 + \tilde{B}_2 + \tfrac{1}{2}\tilde{B}_3\right)F\left(Y^{[n-1]}\right) + y_1^{[n-1]} - hB_1F\left(Y^{[n-1]}\right)$$

$$Y_2^{[n]} = ha_{21}F\left(ha_{11}F\left(Y_2^{[n-1]}\right) + y_1^{[n-1]}\right) + a_{22}hF\left(\left(\tilde{B}_1 + \tilde{B}_2 + \tfrac{1}{2}\tilde{B}_3 - B_1\right)hF\left(Y^{[n-1]}\right) + y_1^{[n-1]}\right) + y_2^{[n-1]}.$$

We now use the test equation y' = λy and substitute z = hλ to yield

$$Y_2^{[n]} = z^2 a_{21}a_{11}Y_2^{[n-1]} + a_{22}z^2\left(\tilde{B}_1 + \tilde{B}_2 + \tfrac{1}{2}\tilde{B}_3 - B_1\right)Y^{[n-1]} + \left(a_{21} + a_{22}\right)zy_1^{[n-1]} + y_2^{[n-1]}$$

We then calculate

$$y^{[n]} = hBF\left(Y^{[n]}\right) + Vy^{[n-1]} = zBY^{[n]} + Vy^{[n-1]}$$

We use the values calculated for the internal stages to rewrite this in terms of the previous external and internal stages:

$$y^{[n]} = zB\begin{bmatrix} za_{11}Y_2^{[n-1]} + y_1^{[n-1]} \\ z^2a_{11}a_{21}Y_2^{[n-1]} + z^2a_{22}\left(\tilde{B}_1 + \tilde{B}_2 + \frac{1}{2}\tilde{B}_3 - B_1\right)Y^{[n-1]} \\ + z(a_{21} + a_{22})y_1^{[n-1]} + y_2^{[n-1]} \end{bmatrix} + Vy^{[n-1]}$$

We now may assemble this into a matrix form expressing

$$\begin{bmatrix} Y_1^{[n]} \\ Y_2^{[n]} \\ y_1^{[n]} \\ y_2^{[n]} \end{bmatrix} = M\begin{bmatrix} Y_1^{[n-1]} \\ Y_2^{[n-1]} \\ y_1^{[n-1]} \\ y_2^{[n-1]} \end{bmatrix}$$

where M is the stability matrix for this predictor-corrector implementation. We identify M from the foregoing calculations as

$$M = \begin{bmatrix} 0 & a_{11}z & 1 & 0 \\ z^2a_{22}(\tilde{B}_{11} + \tilde{B}_{21} + \frac{1}{2}\tilde{B}_{31} - B_{11}) & a_{21}a_{11}z^2 + a_{22}(\tilde{B}_{12} + \tilde{B}_{22} + \frac{1}{2}\tilde{B}_{32} - B_{12})z^2 & (a_{21} + a_{22})z & 1 \\ z^3b_{12}a_{22}(\tilde{B}_{11} + \tilde{B}_{21} + \frac{1}{2}\tilde{B}_{31} - B_{11}) & b_{12}a_{21}a_{11}z^3 + b_{11}a_{11}z^2 + z^3b_{12}a_{22}(\tilde{B}_{12} + \tilde{B}_{22} + \frac{1}{2}\tilde{B}_{32} - B_{12}) & b_{11}z + b_{12}(a_{21} + a_{22})z^2 + v & b_{12}z + 1 - v \\ z^3b_{22}a_{22}(\tilde{B}_{11} + \tilde{B}_{21} + \frac{1}{2}\tilde{B}_{31} - B_{11}) & b_{22}a_{21}a_{11}z^3 + b_{21}a_{11}z^2 + z^3b_{22}a_{22}(\tilde{B}_{12} + \tilde{B}_{22} + \frac{1}{2}\tilde{B}_{32} - B_{12}) & b_{21}z + b_{22}(a_{21} + a_{22})z^2 + v & b_{22}z + 1 - v \end{bmatrix} \quad (59)$$

For the method described above, we find that

$$M = \begin{bmatrix} 0 & \frac{25z}{24} & 1 & 0 \\ -\frac{25z^2}{48} & \frac{56725z^2}{15552} & \frac{1297z}{648} & 1 \\ \frac{625z^3}{2048} & -\frac{25(-131816z^2+170175z^3)}{1990656} & \frac{-(1166400+1713608z-1264575z^2)}{1078272} & -\frac{136+975z}{1664} \\ \frac{281875z^3}{497664} & -\frac{25(-14316264z^2+25582975z^3)}{161243136} & \frac{(94478400+186111432z-190107775z^2)}{87340032} & -\frac{11016+146575z}{134784} \end{bmatrix}$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1 - \frac{289z}{576} - \frac{615775z^2}{248832}\right)w^3 + \left(-\frac{287z}{576} + \frac{53875z^2}{62208} + \frac{112825z^3}{41472}\right)w^2 + \left(\frac{625z^2}{1024} - \frac{36725z^3}{62208}\right)w \tag{60}$$

The stability region was obtained as shown in Figure 31.

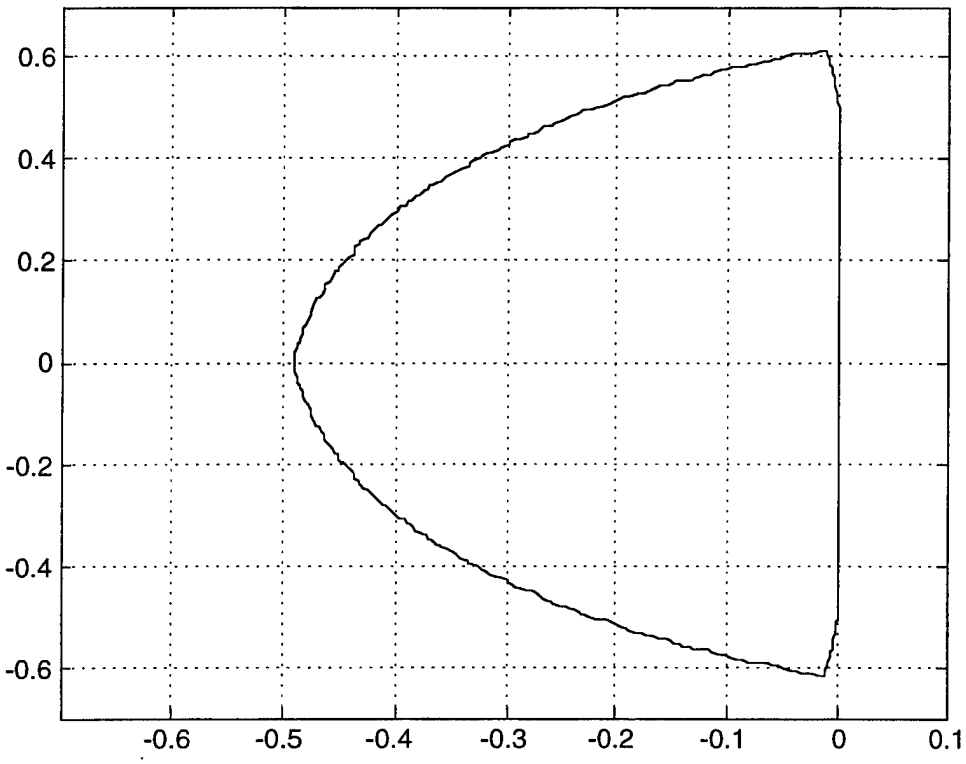



FIGURE 31. Stage 1 PCE, Stage 2 PECE.

This includes the negative real axis to - 0.48.

## STABILITY USING AN A-STABLE FASAL METHOD

The stability regions were recalculated using the following method for which the FASAL implementation is A-stable to get some idea of the value of A-stability for the FASAL implementation in a predictor-corrector setting. The stage point vector c = [0,1], the error constant is $-\frac{65}{768} \approx -0.085$, and the Butcher tableau is given by

$$\begin{bmatrix} \frac{257}{512} & 0 & 1 & 0 \\ \frac{64513}{64768} & \frac{257}{512} & 0 & 1 \\ \frac{387841}{388608} & -\frac{1}{1536} & \frac{1}{3} & \frac{2}{3} \\ \frac{48575}{48576} & \frac{259}{194304} & \frac{1}{3} & \frac{2}{3} \end{bmatrix}.$$

The matrix W is given by

$$W = \begin{bmatrix} 1 & -\frac{257}{512} & 0 \\ 1 & -\frac{64511}{129536} & -\frac{1}{512} \end{bmatrix}$$

and the rescaling matrices are given by

$$\tilde{B} = \begin{bmatrix} \frac{387841}{388608} & \frac{385}{768} \\ 0 & 1 \\ -1 & 1 \end{bmatrix}, \quad \tilde{V} = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} .$$

1) Predict Stage 1, PCE Stage 2 (1 Evaluation)

Equation 48 applies as above, but with this method we now have

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{257z}{512} & \frac{64767z}{32384} & 0 & 1 \\ \frac{257z^2}{786432} & \frac{49643648z - 64767z^2}{49741824} & \frac{1}{3} & \frac{1024 - z}{1536} \\ -\frac{66563z^2}{99483648} & \frac{6292211200z + 16774653z^2}{6292340736} & \frac{1}{3} & \frac{129536 + 259z}{194304} \end{bmatrix} .$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1 - \frac{1537z}{768}\right)w^3 + \frac{577z}{384}w^2 - \frac{385z}{768}w \quad . \tag{61}$$

This gives the stability region shown in Figure 32.

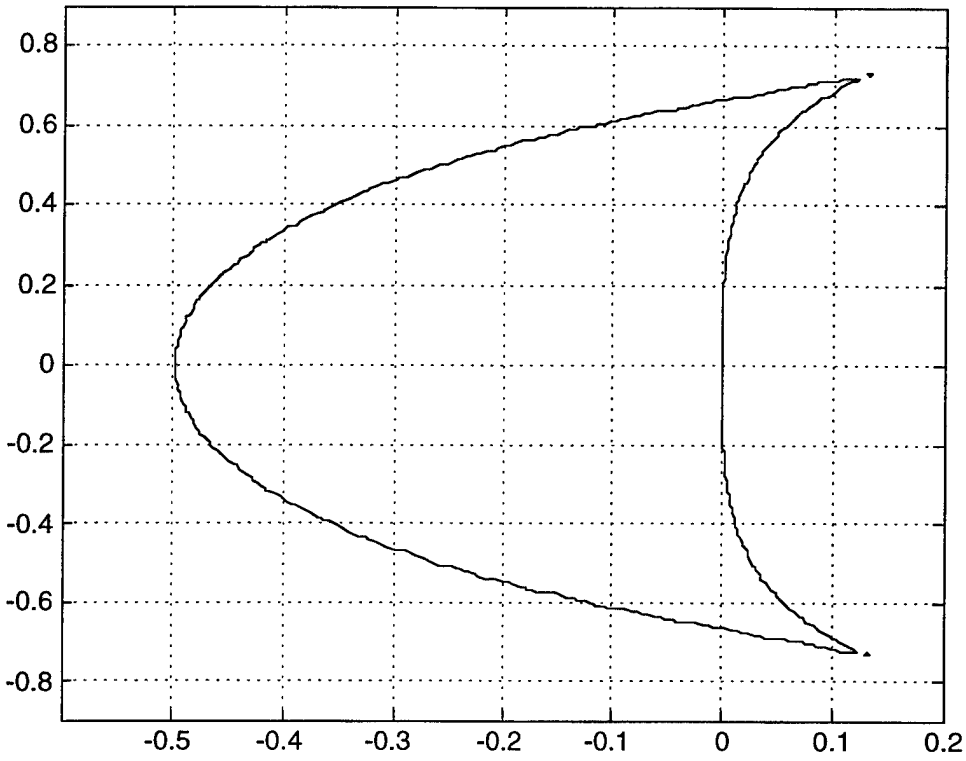



FIGURE 32. FASAL A-Stable, Stage 1 P, Stage 2 PCE.

This includes the negative real axis to - 0.499.

2) Stage 1 PCE, Stage 2 PCE (2 Evaluations)

For the method described above, we find from Equation 50 that

$$M = \begin{bmatrix} 0 & \frac{257z}{512} & 1 & 0 \\ -\frac{257z}{512} & \frac{257(129536z+64513z^2)}{33161216} & \frac{64513z}{64768} & 1 \\ \frac{257z^2}{786432} & \frac{-257(-99157760z^2+64513z^3)}{50935627776} & \frac{(33161216+99287296z-64513z^2)}{99483648} & \frac{1024-z}{1536} \\ -\frac{66563z^2}{99483648} & \frac{257(12617972224z^2+16708867z^3)}{6443356913664} & \frac{(4194893824+12584422400z+16708867z^2)}{12584681472} & \frac{129536+259z}{194304} \end{bmatrix}$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1 - \frac{259495z}{129536} - \frac{2257955z^2}{4521984}\right)w^3 + \left(\frac{129959z}{129536} + \frac{31144673z^2}{24870912}\right)w^2 - \frac{4139499z^2}{16580608}w\,. \tag{62}$$

The stability region was obtained as shown in Figure 33.

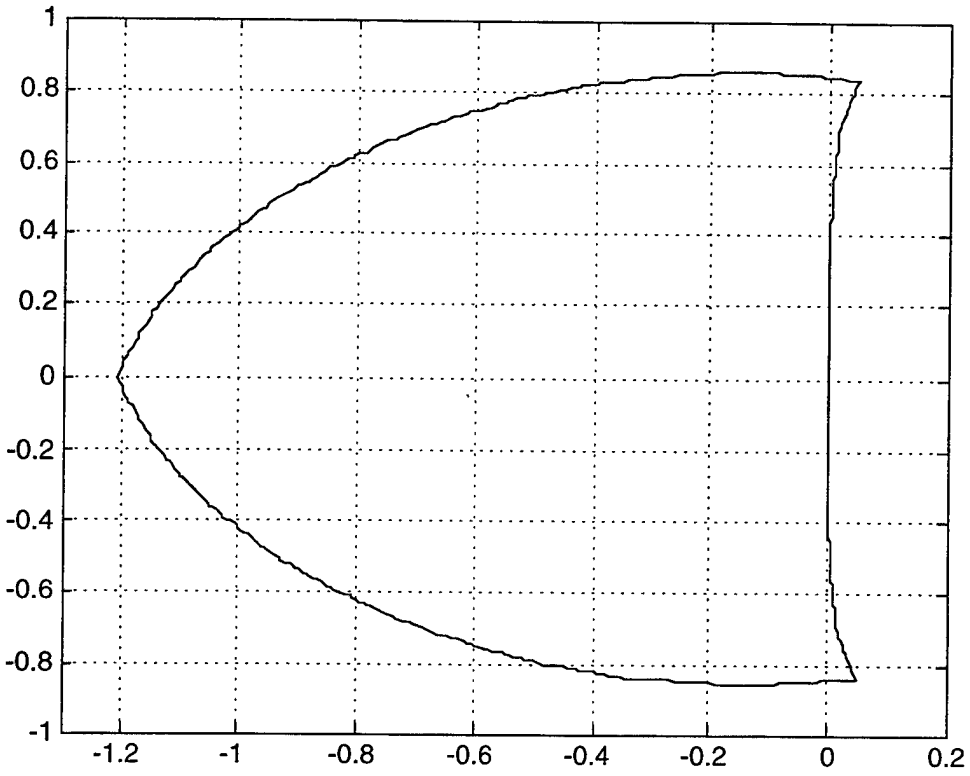



FIGURE 33. FASAL A-Stable, Stage 1 PCE, Stage 2 PCE.

This includes the negative real axis to -1.20.

3) Stage 1 P, Stage 2 PECE (2 Evaluations)

For the method described above, we find using Equation 52 that

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{257z^2}{1024} & \frac{66061312z + 66646525z^2}{66322432} & \frac{257z}{512} & 1 \\ \frac{257z^3}{1572864} & \frac{(101670191104z - 66061312z^2 - 66646525z^3)}{101871255552} & \frac{262144 - 257z^2}{786432} & \frac{1024 - z}{1536} \\ -\frac{66563z^3}{198967296} & \frac{(12886448537600z + 17109879808z^2 + 17261449975z^3)}{12886713827328} & \frac{33161216 + 66563z^2}{99483648} & \frac{129536 + 259z}{194304} \end{bmatrix}$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1 - \tfrac{383z}{384} - \tfrac{395009z^2}{393216}\right)w^3 + \left(-\tfrac{5z}{1536} + \tfrac{148289z^2}{196608}\right)w^2 + \left(\frac{z}{1536} - \tfrac{98945z^2}{393216}\right)w \quad . \tag{63}$$

The stability region was obtained as shown in Figure 34.

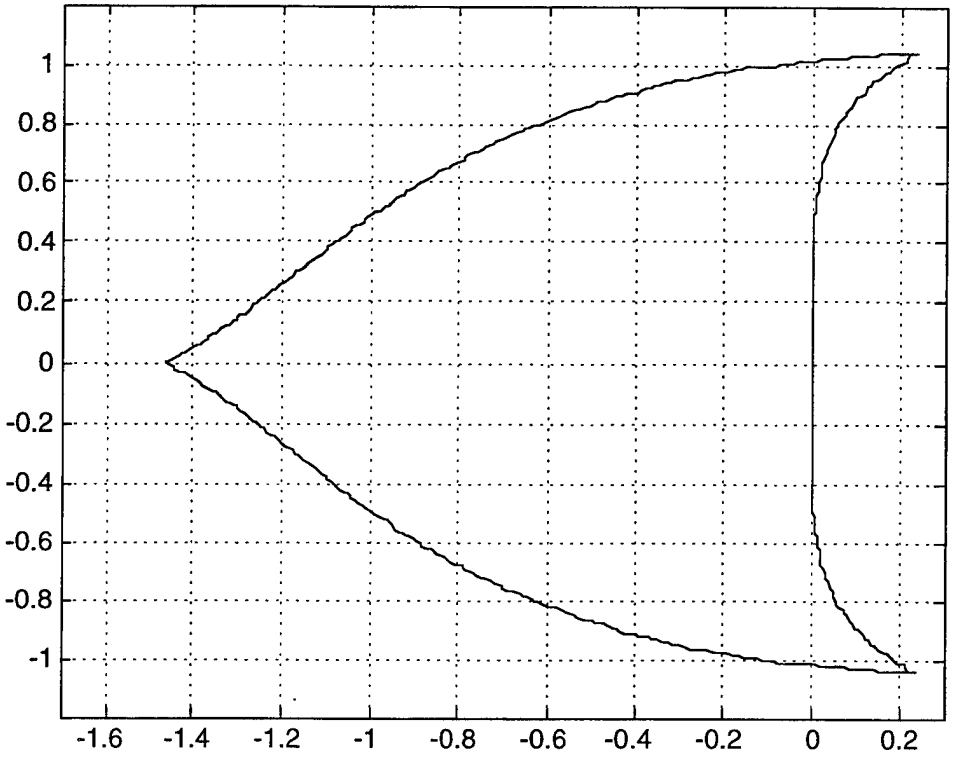



FIGURE 34. FASAL A-Stable, Stage 1 P, Stage 2 PECE.

The negative real axis to -1.46 is included.

4) Stage 1 P, Stage 2 PCECE (2 Evaluations)

We evaluate for this second method, using Equation 54,

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{66049z^2}{262144} & -\frac{16515328z+16645119z^2}{16580608} & 0 & \frac{512+257z}{512} \\ \frac{66049z^3}{402653184} & \frac{(25417547776z-16515328z^2-16645119z^3)}{25467813888} & \frac{1}{3} & \frac{524288-512z-257z^2}{786432} \\ -\frac{17106691z^3}{50935627776} & \frac{(3221612134400z+4277469952z^2+4311085821z^3)}{3221678456832} & \frac{1}{3} & \frac{(66322432+132608z+66563z^2)}{99483648} \end{bmatrix}$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1-\frac{383z}{384}-\frac{395009z^2}{393216}\right)w^3 + \left(-\frac{5z}{1536}+\frac{148289z^2}{196608}\right)w^2 + \left(\frac{z}{1536}-\frac{98945z^2}{393216}\right)w. \tag{64}$$

The stability region was obtained as shown in Figure 35.

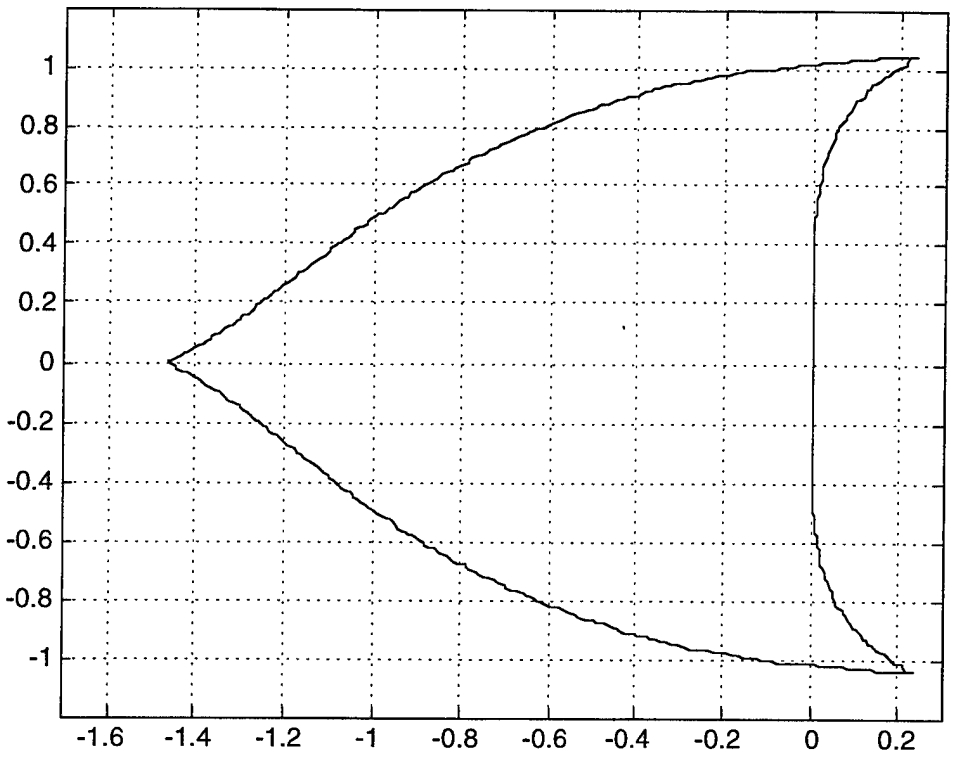



FIGURE 35. FASAL A-Stable, Stage 1 P, Stage 2 PCECE.

The negative real axis to -1.46 is included.

5) Stage 1 PCE, Stage 2 $P(CE)^2$ (3 Evaluations)

For the second method described, using Equation 56, we find that

$$M = \begin{bmatrix} 0 & \frac{257z}{512} & 1 & 0 \\ -\frac{66049z^2}{262144} & \frac{257(66321408z^2+16579841z^3)}{16978542592} & \frac{64513z(512+257z)}{33161216} & \frac{(512+257z)}{512} \\ \frac{66049z^3}{402653184} & \frac{-257(-50835095552z^2+66321408z^3+16579841z^4)}{26079041421312} & \frac{(16978542592+50835095552z-33030656z^2-16579841z^3)}{3298998739795968} & \frac{(54288-512z-257z^2)}{786432} \\ -\frac{17106691z^3}{50935627776} & \frac{257(6443224268800z^2+17177244672z^3+4294178819z^4)}{3298998739795968} & \frac{(2147785637888+6443224268800z+8554939904z^2+4294178819z^3)}{6443356913664} & \frac{(66322432+132608z+66563z^2)}{99483648} \end{bmatrix}.$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1 - \frac{129453z}{129536} - \frac{24968191z^2}{24870912} - \frac{580294435z^3}{2315255808}\right)w^3 + \left(-\frac{83z}{129536} + \frac{33356457z^2}{66322432} + \frac{8004180961z^3}{12733906944}\right)w^2 + \left(\frac{257z^2}{786432} - \frac{1063851243z^3}{8489271296}\right)w. \quad (65)$$

The stability region was obtained as shown in Figure 36.

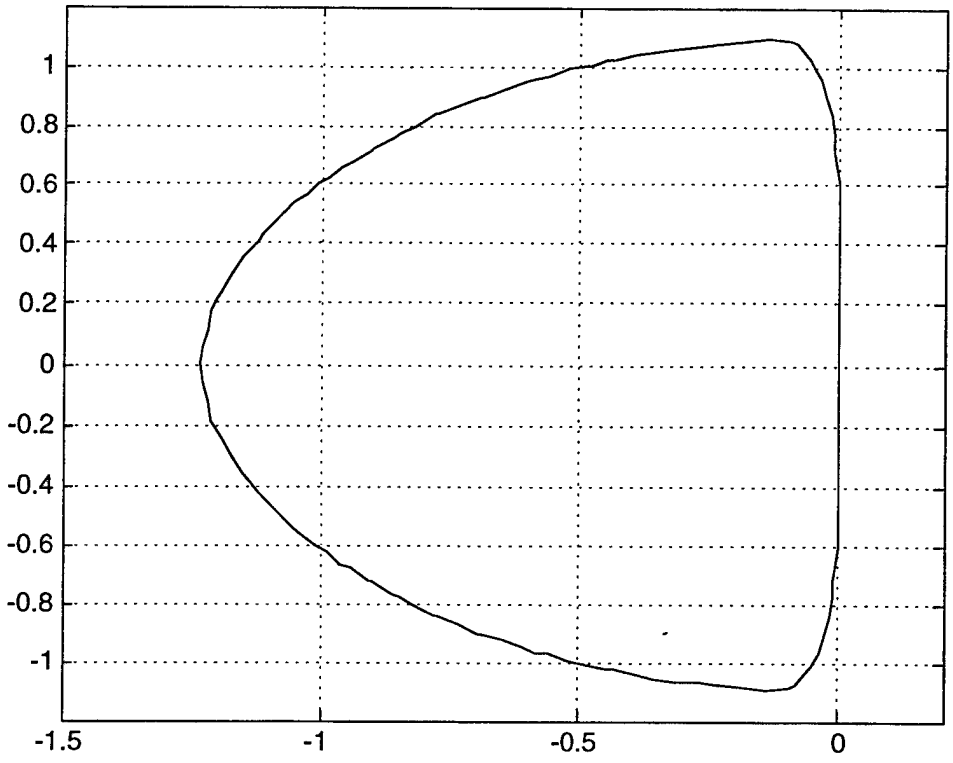



FIGURE 36. FASAL A-Stable, Stage 1 PCE, Stage 2 PCECE.

The negative real axis to -1.23 is included.

6) Stage 1 PCE, Stage 2 PECE (3 Evaluations)

For this method, using Equation 60, we obtain

$$M = \begin{bmatrix} 0 & \frac{257z}{512} & 1 & 0 \\ -\frac{257z^2}{1024} & \frac{99806207z^2}{66322432} & \frac{194047z}{129536} & 1 \\ \frac{257z^3}{1572864} & \frac{-257(-198574592z^2+388351z^3)}{101871255552} & \frac{(66322432+198574592z-194047z^2)}{198967296} & \frac{1024-z}{1536} \\ -\frac{66563z^3}{198967296} & \frac{257(25168844800z^2+100582909z^3)}{12886713827328} & \frac{(8389787648+25168844800z+50258173z^2)}{25169362944} & \frac{129536+259z}{194304} \end{bmatrix}$$

This corresponds to the stability polynomial

$$p(w,z) = w^4 + \left(-1 - \frac{129453z}{129536} - \frac{13601117z^2}{9043968}\right)w^3 + \left(-\frac{83z}{129536} + \frac{66516139z^2}{66322432} + \frac{349091581z^3}{397934592}\right)w^2 + \left(\frac{257z^2}{786432} - \frac{49608967z^3}{397934592}\right)w . \quad (66)$$

The corresponding stability region is as shown in Figure 37.

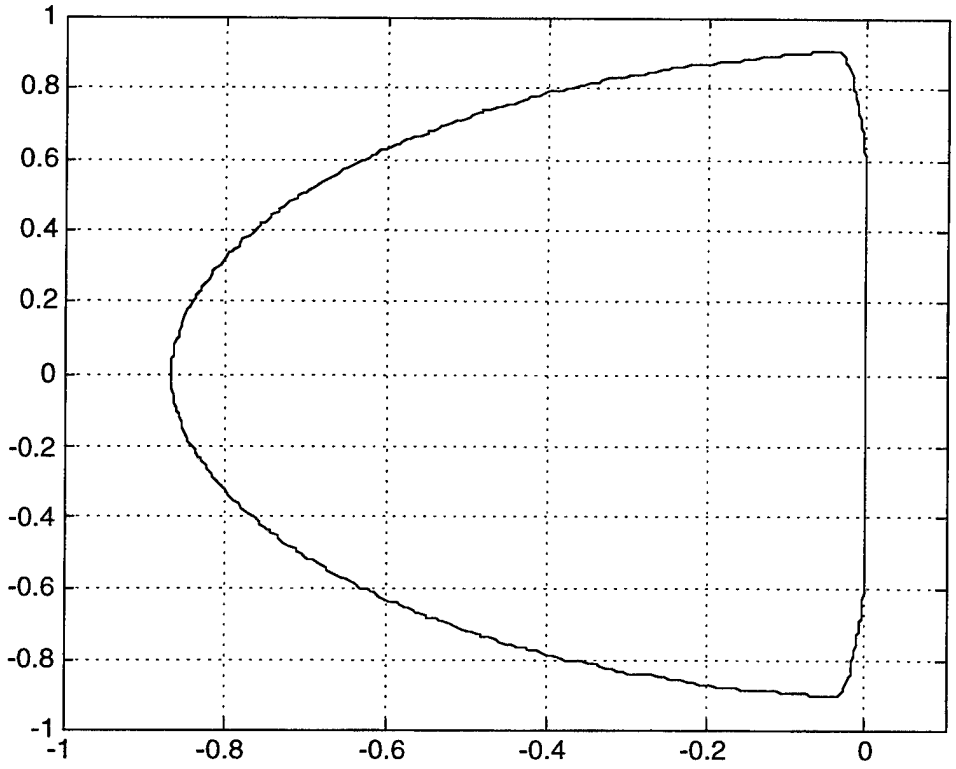



FIGURE 37. FASAL A-Stable, Stage 1 PCE, Stage 2 PECE.

The negative real axis is included to - 0.86.

## CONCLUSIONS

Although only two second order DIMSIMs were studied, some directions for more general predictor-corrector implementations of DIMSIMs are suggested. The size of the stability matrices (2p x 2p) and the number of alternatives make this kind of study more difficult for high order.

The value of a method with an A-stable FASAL implementation was evident. The stability regions for the second method were significantly larger. And although a correction with the first stage formula improved the stability region as compared with only one correction iteration with the second stage, an additional iteration could better be performed for the second stage. But two evaluations per step seem to be required to obtain a robust stability region, making this approach roughly comparable to PECE implementations for linear multistep methods. Iteration to convergence might be tried with success to produce a competitive method.

The stability regions for the first method were disappointingly small. Certainly for this method and perhaps in general for methods that do not have an A-stable FASAL

implementation, iteration to convergence is the only predictor-corrector implementation that might lead to a competitive solver. But the fact that the stability region for the first implementation was comparable to that obtained for the A-stable FASAL method is a surprising result that keeps this a yet somewhat open question.

No stability advantage was seen for providing an accurate prediction of the internal stage value with the full Nordsieck vector as compared with predicting the internal stage derivative at the second stage, and in comparing the stability regions for the Stage 1-PCE, Stage 2-PECE and PCECE methods; the second, corresponding to predicting the internal stage derivative, had the larger stability region. For iteration to convergence the advantage of concluding with an evaluation and test that are directly related to further computations would make this mode seem to be the preferred one.

Finally, it may be possible to design special methods with large stability regions for predictor-corrector implementations. This issue must be deferred for future study.

# THE PROTOTYPE DIMSTIFF FAMILY OF STIFF ODE SOLVERS

## INTRODUCTION

L-stable implicit DIMSIMs of orders 2 and 5 have been implemented to develop prototype versions of the computer codes DIMSTIFF2 and DIMSTIFF5 to solve stiff systems of ordinary differential equations using adaptive step-size selection. A modified Newton iteration with Gaussian elimination is used to solve the nonlinear systems that arise in the calculation of internal stages. The user is asked to provide a Jacobian subroutine. Calculation of numerical approximations for Jacobians and other refinements, several of which are discussed below, are left for future work.

## SOME DESIGN DECISIONS FOR A STIFF SOLVER

Some helpful recommendations made by Hairer and Wanner (Reference 2) for implementation of implicit Runge-Kutta methods are followed. First, because stiff problems are associated with derivative functions having large Lipschitz constants, errors due to both roundoff and especially the stage-by-stage iterative process for internal stages may result in very large errors. To avoid these problems, we work with a quantity Z as follows:

$$Z_i^{[n]} = Y_i^{[n]} - \hat{y}_i^{[n-1]} \quad . \tag{67}$$

We may then write

$$Z_i^{[n]} = h_n \lambda f\left(t_{n-1} + c_i h_n, Z_i^{[n]} + \hat{y}_i^{[n-1]}\right) + \sum_{k=1}^{i-1} h_n a_{ik} f\left(t_{n-1} + c_k h_n, Z_k^{[n]} + \hat{y}_k^{[n-1]}\right), \tag{68}$$

or in matrix form (for a one-dimensional system) with some shorthand notation,

$$Z^{[n]} = h_n A F\left(Z^{[n]}\right) \quad . \tag{69}$$

We then note that A is diagonal and nonsingular, so

$$h_n F\left(Z^{[n]}\right) = A^{-1} Z^{[n]} \quad . \tag{70}$$

We use the internal stage derivative values calculated in this way for rescaling, Nordsieck vector computation, and error estimation instead of doing the extra work of applying the derivative function, which would have a large Lipschitz constant and could induce large errors. For a diagonally implicit method, the process of using these relationships in computing the internal stage values works out in a step-by-step manner. Derivative function evaluations are only performed as part of the Newton iteration for computation of internal stage values. For example, with a second order method and an ODE of arbitrary dimension, we first solve

$$Z_1^{[n]} = h_n \lambda f\left(t_{n-1} + c_1 h_n, Z_1^{[n]} + \hat{y}_1^{[n-1]}\right) \quad . \tag{71}$$

We then set

$$h_n f\left(t_{n-1} + c_1 h_n, Z_1^{[n]} + \hat{y}_1^{[n-1]}\right) = \frac{1}{\lambda} Z_1^{[n]} \tag{72}$$

and then solve

$$Z_2^{[n]} = h_n \lambda f\left(t_{n-1} + c_2 h_n, Z_2^{[n]} + \hat{y}_2^{[n-1]}\right) + h_n a_{21} f\left(t_{n-1} + c_1 h_n, Z_1^{[n]} + \hat{y}_1^{[n-1]}\right)$$

$$= h_n \lambda f\left(t_{n-1} + c_2 h_n, Z_2^{[n]} + \hat{y}_2^{[n-1]}\right) + \frac{a_{21}}{\lambda} Z_1^{[n]}. \tag{73}$$

The efficiency of an implementation depends on the frequency with which Jacobians are evaluated and matrix factorization is carried out. Clearly this should be done as seldom as possible, but also as often as necessary for convergence. This must be optimized using a heuristic approach based on extensive testing, which is a subject for future work. As an initial approach awaiting future refinement some ideas were borrowed from Hairer and Wanner (Reference 2) and their discussion of implementing implicit Runge-Kutta methods, and from Enenkel and Jackson (Reference 25) who took an algorithm that has been used successfully with the code VODE (Reference 8) and modified it to reflect differences between backwards differentiation formula (BDF) methods and the general linear methods they were using. Further modifications are made because of the differences between DIMSIMs and implicit Runge-Kutta methods or DIMSEMs, the latter differs most notably for implementation purposes in that they were designed to have all stage points coincide at the left grid point. Thus the end point solution for the previous step serves as a predictor for all internal stages, which would have similar convergence properties. But DIMSIMs typically have nonconfluent stage points (no two points are the same). Methods appropriate for waveform relaxation have these stage points distributed in some fashion across the interval [0,1], and for methods derived thus far these are usually equally spaced.

Newton iteration is modified here to produce a "chord method" in which the Jacobian is not changed during the iteration process. Although convergence is linear rather than quadratic, performance of chord methods for stiff problems is considered satisfactory and is commonly used (see for example Reference 4). A new solution value is provided at the end of a fully successful step, so we re-evaluate Jacobians and carry out fresh matrix factorizations only at the end of successful steps, rather than to do this extra work in the middle of steps that may fail. At this time Jacobians are re-evaluated at a set interval, but this may be changed to include all times when matrix factorization is carried out. Factorization is currently performed when step size changes or at a maximum spacing, whichever occurs first.

For methods for which $c_1 = 0$ and $c_p = 1$, the predictor for a 0 stage point may be either the final internal stage of the last step or the first component of the Nordsieck vector. A final choice will depend on extensive testing, but at this time, for each internal stage, the Nordsieck vector is used to predict the internal stage value. This is somewhat different from predictor-corrector methods where prediction of the derivative may result in fewer function evaluations. This is not expected to be effective for stiff problems because function iteration tends to diverge. For the DIMSTIFF codes, the Nordsieck vector is used to predict all stage values.

A more accurate prediction for an internal stage value should result in fewer Newton iterations. If convergence has not occurred within the maximum allowable number of iterations, either it is time to re-evaluate the Jacobian and do a fresh matrix factorization or, alternatively, the step size should be shortened. At the same time, accuracy of the Taylor

series for the Nordsieck predictor would be expected to deteriorate away from the left grid point where the Nordsieck vector is provided. Also, it would seem important to quickly determine that a step must be repeated with shorter step size or that fresh Jacobian evaluation and matrix factorization is needed before too much work is done for a step, while after most of the stages have been successfully calculated extra effort might well be made to calculate the final stage or stages. Thus for DIMSIMs, the number of iterations allowed for calculation of an internal stage should depend on how far the stage point is from the left grid point. At this time, for second order, the maximum allowable number of iterations before nonconvergence is determined is set at 1 for the left end point and 6 for the right end point. For fifth order the maximum allowable number of iterations is currently set to the same number for all steps after the first, in which only 1 is allowed. However Hairer and Wanner (Reference 2) indicate that greater efficiency may result through allowing more iterations. Finding optimal numbers for each stage is a subject for future work.

Change of step size is an important design issue. Error estimation is used to predict the accuracy of a step and to evaluate its success in producing a result within tolerance. Careful adaptation through incremental lengthening and shortening of the step size is usually desirable. But the matrix to be factored is

$$G(t,y,h_n) = I - h_n \lambda J(t,y) \tag{74}$$

where $\lambda$ is the diagonal element of the DIMSIM coefficient matrix A and J is the Jacobian of the derivative function. This matrix is strongly dependent on the step size, and so steps must be varied with caution to avoid a great deal of extra work. It may not be necessary to re-evaluate J, but appropriate factorization may change significantly and this is a very expensive operation. As a result, for an initial approach, the conservative strategy is followed that when step size reduction is called for, the step is reduced by a factor of a power of 2, and step size increase is not carried out unless error estimation indicates the step should be at least doubled, and then a maximum factor of 2 is used. Also, step size is not increased for a minimum of 3 steps after Jacobian evaluation and factorization are carried out.

Special consideration must be given to the strategy to be followed after a step fails. These failures may be due either to lack of convergence or failure of the subsequent error estimate to be within tolerance. Actions may include step-size reduction, fresh factorization, Jacobian re-evaluation, or even an error message and termination. If Newton convergence takes place but tolerance is not met, the approach taken here is to shrink the step size. If step size had just increased by a factor of 2, a return made to the original step size might be contemplated since factorization could be preserved and little additional work is needed, but this has not been implemented and requires substantial additional storage. A total of seven retries are allowed, and step size is shrunk by a factor of 2 each time. Fresh matrix factorization is required for each retry. Jacobian evaluation is carried out as well only if the Jacobian is not current through the end of the previous step. An error message is provided and processing terminates if seven retries do not suffice. If the Newton iteration does not converge for any stage, the step must begin again with a new step size, which is shrunk by a factor of 4. The Jacobian is updated if it is not current. Factorization is performed with each retry. A maximum of 10 retries are allowed before an error message is provided and processing is terminated.

Because the penalties for step failure can be so draconian, a proactive strategy for periodic updating of Jacobians and matrix factorizations is utilized. A factorization is carried out at least every 20 steps. If a factorization is to be performed and the Jacobian has not been updated for at least 40 steps, the Jacobian is recalculated first.

Richardson error estimation is currently carried out. A logical "stiff" provides a means to chose either the nonstiff Richardson estimator or a heuristic Richardson-type stiff estimator, which differs from the nonstiff estimator only by a constant factor. Since two successive steps with the same step size are needed to carry out a Richardson error estimation, some special logic is required so that after a step-size change, no estimate is calculated. Also, if the second step fails, the first step becomes suspect as well since it was not successfully tested. Thus processing must continue at the starting point of the previous step.

## TESTING THE DIMSTIFF CODES

Because of project time constraints, testing was limited to the Prothero-Robinson problem (Equation 40) with the differential equation parameter $\lambda$ assuming negative values of large magnitude. It was chosen especially because for this stiff problem there has been some success in developing heuristic Richardson-type error estimation. The starting point (0,1/2) was utilized. To verify that the solver had the characteristics of a stiff solver, comparison was made with LSODE (Reference 7). A comparison is provided here (Table 23) first between the stiff (BDF) and nonstiff (predictor-corrector Adams-Moulton) options to illustrate the difference between stiff and nonstiff solvers. The parameter $\lambda$ takes on a wide range of values, with stiffness increasing with the magnitude of $\lambda$. The analytical Jacobian was provided. Entries are the number of steps required to integrate from 0 to 20 with absolute tolerance of $10^{-9}$.

TABLE 23. Stiff Versus Nonstiff Solvers on the Prothero-Robinson Problem.

| $\lambda$ | AM2 | BDF2 | AM5 | BDF5 |
|---|---|---|---|---|
| -2 | 8757 | 13452 | 465 | 542 |
| -1000 | 35,608 | 15410 | 34444 | 1199 |
| -10,000 | 386,122 | 15402 | 384855 | 1367 |

It is evident that for nonstiff problems, the nonstiff solver is more efficient. However, as the problem becomes increasingly stiff, the amount of work with a nonstiff solver increases drastically, while the work remains approximately the same for the stiff solver.

The same problem was used to test the DIMSTIFF prototype codes. Statistics are provided in Table 24 on the number of steps required, number of function calls, ratio of the number of function calls to the number of steps times, number of internal stages per step, number of tolerance misses, number of convergence failures, number of times the error estimator was deceived (that is, error was greater than the tolerance but the step was

accepted), number of Jacobian evaluations, number of requests for matrix factorization, and relative error at the end point. Of course the Jacobian is simply the number $\lambda$, and the matrix is also a single number. This simply provides an indication of the operation of the program, which mandates that these operations be carried out with a minimum frequency as well as in the case of tolerance or convergence misses and step-size changes. The number of steps required for the LSODE BDF solver of the same order is also provided.

TABLE 24. DIMSTIFF Solvers on the Prothero-Robinson Problem.

| | Order = 2 | | | Order = 5 | | |
|---|---|---|---|---|---|---|
| $\lambda$ | -2 | -1000 | -10,000 | -2 | -1000 | -10,000 |
| # steps | 10780 | 24031 | 21916 | 356 | 694 | 1161 |
| # BDF steps | 13452 | 15410 | 15402 | 542 | 1199 | 1367 |
| # func calls | 32353 | 51250 | 59671 | 1970 | 4215 | 6605 |
| Func/stage | 1.50 | 1.07 | 1.36 | 1.11 | 1.21 | 1.14 |
| # tol misses | 5 | 1 | 126 | 0 | 18 | 23 |
| # conv fails | 0 | 10 | 108 | 0 | 2 | 28 |
| # deceived | 1 | 6 | 120 | 0 | 9 | 36 |
| # Jac eval | 270 | 601 | 548 | 9 | 18 | 30 |
| # factor | 668 | 1370 | 1709 | 44 | 97 | 240 |
| End relerr | 6e-8 | 2e-10 | 2e-10 | 6e-10 | 2e-10 | 4e-12 |

We see that both DIMSTIFF2 and DIMSTIFF5 clearly behaved as stiff solvers. The Richardson error estimator worked very well for the nonstiff problem with $\lambda$ = -2, and the heuristic Richardson-type stiff estimators performed well enough for the other problems. The number of function evaluations was generally just a bit higher than the optimal 1 per stage, and more work may be done to improve this performance. It should be noted that additional function evaluations are required when multiple iterations are needed for convergence, and when steps must be repeated because iterations fail to converge or tolerances are missed. The low ratio indicates that prediction using the Nordsieck vector is working very well, but the linearity of the problem makes this an easy test for the solver. The fifth order solver showed the expected large improvement over the second order solver and actually required fewer steps than LSODE. It should be noted that the fifth order BDF method requires only one-fifth the work in nonlinear system solving, however. It is hoped that accurate error estimation and efficient step-size control will significantly reduce the number of DIMSIM steps required. But the greatest advantage of implicit DIMSIMs will be seen in stiff oscillatory problems (the linearized problem has eigenvalues in the left half-plane but with large imaginary parts), where the stability regions of BDF methods with order above 3 become very restrictive, and in the potential for solvers of order above 5. The starting method worked well for second order, but more work in automatically selecting the $h_0$ used in computing initial derivatives for fifth order must be done. A value of $h_0 = 10^{-6}$ was chosen arbitrarily and provided adequate results while the previously used algorithm generated an $h_0$ that was much too large.

We may note that the problem starts out nonstiff during a transient period in which the true solution rapidly converges to the sine function. Here step size is determined by

accuracy and not stability. We would expect the nonstiff error estimator to work well during this region. This is followed by the bulk of the integration interval, in which the problem is clearly stiff and the stiff formula works well. Ideally error estimators might be switched back and forth, depending on stiffness. Developing stiffness detection for DIMSIMs is a topic for future work, however. Also, it should be noted that there is a transition period between stiff and nonstiff regions in which both types of error estimation work poorly.

These experiments clearly demonstrate the potential of using implicit DIMSIMs for solving stiff problems. It is evident that more work is needed on stiff error estimation before a generally useful stiff solver can be developed. Of course, optimization is needed with implementation details, such as selection of the $h_0$ value, determination of when to recompute Jacobians and refactor, and also with step-size control. We plan to pursue these after the larger error estimation problem is adequately resolved. Although only second and fifth order experimental solvers are described here, an L-stable eighth order method has been derived (Reference 30) and this could also be developed into a solver that would have exceptionally high order. The work of developing DIMSIM solvers to reach their powerful potential is only just beginning.

## REFERENCES


1. Naval Air Warfare Center Weapons Division. *Using Diagonally Implicit Multistage Integration Methods for Solving Ordinary Differential Equations. Part 1: Introduction and Explicit Methods*, by Jack Van Wieren. China Lake, Calif., NAWCWPNS, January 1997. 136 pp. (NAWCWPNS TP 8340, publication UNCLASSIFIED.)

2. E. Hairer and G. Wanner. *Solving Ordinary Differential Equations, Vol. II: Stiff and Differential-Algebraic Problems.* New York, Springer-Verlag, 1991.

3. C. F. Curtiss and J. O. Hirschfelder. "Integration of Stiff Equations," *Proc. Nat. Acad. Sci.*, Vol. 38 (1952), pp. 235- 243.

4. L. F. Shampine. *Numerical Solution of Ordinary Differential Equations.* New York, Chapman and Hall, 1994.

5. C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations.* Englewood Cliffs, New Jersey, Prentice-Hall, 1971.

6. G. D. Byrne and A. C. Hindmarsh. "A Polyalgorithm for the Numerical Solution of Ordinary Differential Equations." *ACM Trans. Math. Software*, Vol. 1 (1975), pp. 71-96.

7. K. Radhakrishnan and A. Hindmarsh. "Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations," *NASA Reference Publication 1327*, Lawrence Livermore National Laboratory Report UCRL-ID-113855, December 1993.

8. P. N. Brown, G. D. Byrne, and A. C. Hindmarsh. "VODE, a Variable Coefficient ODE Solver," *SIAM J. Sci. Stat. Comput.*, Vol. 10 (1989), pp. 1038-1051.

9. J. C. Butcher. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods.* New York, John Wiley and Sons, 1987.

10. K. Burrage. "A Special Family of Runge-Kutta Methods for Solving Stiff Differential Equations," *BIT*, Vol. 18 (1978), pp. 22-41.

11. J. C. Butcher. "A Transformed Implicit Runge-Kutta Method," *J. Assoc. Comput. Mach.*, Vol 26 (1979), pp. 731-738.

12. K. Burrage, J. C. Butcher, and F. H. Chipman. "An Implementation of Singly-Implicit Runge-Kutta Methods," *BIT*, Vol. 20 (1980), pp. 326-340.

13. J. C. Butcher, J. R. Cash, and M. T. Diamantakis. "DESI Methods for Stiff Initial Value Problems," *ACM Trans. Math. Software*, Vol. 22 (1996), pp. 401-422.

14. K. Burrage and J. C. Butcher. "Nonlinear Stability of a Class of a General Class of Differential Equation Methods," *BIT*, Vol. 20 (1980), pp. 185-203.

15. R. D. Skeel and A. K. Kong. "Blended Linear Multistep Methods." *ACM Trans. Math. Software*, Vol. 3 (1977), pp. 326-345.

16. C. A. Addison. "Implementing a Stiff Method Based upon the Second Derivative Formulas." Dept. of Comput. Sc., University of Toronto, Canada, 1979 (Technical Report 130/79).

17. W. H. Enright. "Second Derivative Methods for Stiff Ordinary Differential Equations," *SIAM J. Numer. Anal.*, Vol. 11 (1974), pp. 321-331.

18. J. R. Cash. "On the Integration of Stiff Systems of ODEs Using Extended Backward Differentiation Formulae," *Numer. Math.*, Vol. 34 (1980), pp. 235-246.

19. J. R. Cash. "The Integration of Stiff Initial Value Problems in ODEs Using Modified Extended Bacward Differentiation Formulae," *Comp. and Maths. with Appls.*, Vol. 9 (1983), pp. 645-657.

20. J. R. Cash and S. Considine. "An MEBDF Code for Stiff Initial Value Problems," *ACM Trans. Math. Software*, Vol. 18 (1992), pp. 142-155.

21. Z. Jackiewicz and S. Tracogna. "A General Class of Two-Step Runge-Kutta Methods for Ordinary Differential Equations," *SIAM J. Numer. Anal.* 32 (1995), pp. 1390-1427.

22. S. Tracogna. "Implementation of Two-Step Runge-Kutta Methods for Ordinary Differential Equations," to appear in *J. Comp. Appl. Math.*

23. J. C. Butcher. "Diagonally Implicit Multi-Stage Integration Methods," *Appl. Numer. Math.*, Vol. 11 (1993), pp. 347-63.

24. R. Enenkel. "DIMSEMs-Diagonally Implicit Single-Eigenvalue Methods for the Numerical Solution of Stiff Ordinary Differential Equations on Parallel Computers." Ph.D. dissertation, University of Toronto, 1996.

25. R. Enenkel and K. Jackson, "DIMSEMs-Diagonally Implicit Single-Eigenvalue Methods for the Numerical Solution of Stiff ODEs on Parallel Computers," to appear in *Adv. Comp. Math.*

26. J. C. Butcher and Z. Jackiewicz. "Diagonally Implicit General Linear Methods for Ordinary Differential Equations," *BIT*, Vol. 33 (1993), pp. 452-472.

27. J. C. Butcher and Z. Jackiewicz. "Implementation of Diagonally Implicit Multistage Integration Methods for Ordinary Differential Equations," to appear in *SIAM J. Num. Anal.*

28. Z. Jackiewicz, R. Vermiglio, and M. Zennaro. "Variable Stepsize Diagonally Implicit Multistage Integration Methods for Ordinary Differential Equations," *Appl. Numer. Math.*, Vol. 16 (1995), pp. 343-67.

29. Z. Jackiewicz, R. Vermiglio, and M. Zennaro. "Regularity Properties of Multistage Integration Methods," submitted to *J. Comp. Appl. Math.*

30. J. C. Butcher, Z. Jackiewicz, and H. Mittelmann. "A Nonlinear Optimization Approach to the Construction of General Linear Methods of High Order," to appear in *J. Comp. Appl. Math.*

31. J. C. Butcher and Z. Jackiewicz. "Construction of Diagonally Implicit General Linear Methods of Type 1 and 2 for Ordinary Differential Equations," to appear in *Appl. Numer. Math.*

32. J. C. Butcher and Z. Jackiewicz. "Construction of High Order DIMSIMs for Ordinary Differential Equations," submitted to *IMA J. Num. Anal.*

33. J. D. Lambert. *Computational Methods in Ordinary Differential Equations.* London, Wiley and Sons, 1973.

34. K. Burrage. *Parallel and Sequential Methods for Ordinary Differential Equations.* Oxford, Clarendon Press, 1995.

35. P. Chartier. "Parallelisme dans la resolution numerique des problemes de valeur initiale pour les equations differentielles ordinaires et algebriques." Ph.D. dissertation, University of Rennes, France, 1993.

36. J. D. Lambert. *Numerical Methods for Ordinary Differential Systems: the Initial Value Problem.* New York, John Wiley and Sons, 1991.

37. A. Prothero and A. Robinson. "On the Stability and Accuracy of One-Step Methods for Solving Stiff Systems of Ordinary Differential Equations," *Math. Comp.*, Vol. 28 (1974), pp. 145-62.

## INITIAL DISTRIBUTION

1 Commander in Chief, U. S. Pacific Fleet, Pearl Harbor (Code 325)
1 Naval War College, Newport
1 Headquarters, 497 IG/INT, Falls Church (OUWG Chairman)
2 Defense Technical Information Center, Fort Belvoir
1 Center for Naval Analyses, Alexandria, VA (Technical Library)

---

ON-SITE DISTRIBUTION

4 Code 4BL000D (3 plus Archives copy)
1 Code 4B0000D
17 Code 4B4000D
S. Chesnut (1)
C. Schwartz (1)
J. VanWieren (15)
1 Code 472000D
1 Code 473000D