NEUMAIER'S METHOD FOR THE SOLUTION OF INITIAL VALUE PROBLEMS FOR STIFF ORDINARY DIFFERENTIAL EQUATIONS

by

Annie Hsiao Chen Yuk

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Neumaier's Method For The Solution Of Initial Value Problems For Stiff Ordinary Differential Equations

Annie Hsiao Chen Yuk

Master of Science

Graduate Department of Computer Science

University of Toronto

2005

Compared with standard numerical methods for initial value problems (IVPs) for ordinary differential equations (ODEs), validated methods not only compute a numerical solution to a problem, but also generate a guaranteed bound on the global error associated with the numerical solution. There have been significant developments in the field of validated numerical methods of IVPs over the past few decades. However, none of the validated methods developed to date are suitable for stiff IVPs for ODEs.

This thesis investigates the potential of Neumaier's validated method for stiff IVPs for ODEs. We show that Neuamier's result is a special case of Dahlquist's result. Our findings show that this method has promise as an effective validated method for stiff IVPs for ODEs, for problems where there is no wrapping effect.

Acknowledgements

I would like to express my special thanks to my supervisor, Professor Ken Jackson for his patience, guidance, and support during my MSc program. I want to thank Professor Wayne Enright for his helpful suggestions and advice. I would also like to give special thanks to Dr. Markus Neher of University of Karlsruhe, Germany, who provided his MAPLE code for Neumaier's enclosure method. His valuable advice, comments and suggestions contributed much to this research. Finally, I would like to thank Dominic for his patience, and support.

Contents

Chapter 1

Introduction

Consider the initial value problem (IVP) for an ordinary differential equation (ODE)

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad t \in [t_0, T] \tag{1.1}$$

where $y \in \mathbb{R}^n$ and $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$. Assume that f is smooth and that there exists a unique solution to (1.1) on $[t_0, T]$.

The purpose of this thesis is to investigate the potential of Neumaier's enclosure method [16] for the solution of the IVP (1.1) and to provide an insight into how this method behaves in practice.

1.1 Validated Methods

Validated numerical methods (also called interval methods) for approximating the solution of an IVP for an ODE differ from standard numerical methods which compute an approximation to the solution of the IVP and may also attempt to keep the error associated with the approximation within a user-specified tolerance. Validated methods, not only compute a numerical solution, but also generate a guaranteed bound on the global error associated with the numerical solution. As a side benefit, validated methods also prove that the solution to the IVP actually exists.

Over the past few decades, Moore [11], Eijgenraam [2], Lohner [8] and others, have produced significant developments in the field of validated numerical methods for IVPs. Most methods are based on Taylor series and use interval arithmetic to generate guaranteed global error bounds. The attractiveness of Taylor series stems from the fact that the Taylor series coefficients can be readily generated by automatic differentiation, the stepsize can be easily changed without doing extra work in recomputing the Taylor series coefficients, the order of the method can be changed easily by varying the number of terms in the Taylor series, and a bound on the local error in the step can be computed easily from the Taylor series remainder term.

However, none of the validated methods developed to date are suitable for *stiff* problems, since they all suffer from a severe step size restriction on this class of problems. For further discussions of validated methods of IVPs, see [4].

1.1.1 Stiff Problems

An IVP is considered *stiff* if some components of the solution decay rapidly compared to the time-scale of the problem. To illustrate, consider a system of equations of the form (1.1), where the behavior of its solution $y(t)$ near a particular solution $g(t)$ of the ODE $y' = f(t, y)$ can be approximated well by a Taylor series expansion of the form

$$\begin{aligned} y' &= f(t, y(t)) \\ &= f(t, y(t)) - f(t, g(t)) + f(t, g(t)) \\ &\approx J(t, g(t))(y - g(t)) + f(t, g(t)) \\ &= J(t, g(t))(y - g(t)) + g'(t) \end{aligned} \tag{1.2}$$

where $J(t, g(t) = f_y(t, g(t))$ is the Jacobian matrix associated with $f(t, g(t))$. Assume that $J(t, g(t))$ is slowly varying in t and can be approximated locally by a constant matrix A. If A is diagonalizable, then the systems of equations

$$y' = A(y(t) - g(t)) + g'(t) \tag{1.3}$$

can be uncoupled giving rise to a set of IVPs

$$x_i' = \lambda_i(x_i - p_i(t)) + p_i'(t), \quad x_i(t_0) = x_{i_0}, \tag{1.4}$$

for $i = 1, .., n$, where λ_i is an eigenvalue of A. The analytical solution of (1.4) is

$$x_i(t) = (x_{i_0} - p_i(t_0)) \exp(\lambda_i(t - t_0)) + p_i(t). \tag{1.5}$$

If λ_i is large and negative, the solution $x_i(t)$ decays very quickly to $p_i(t)$ as t increases. This is a key characteristic of a stiff problem.

A problem of the form (1.4) is stiff if at least one eigenvalue satisfies $Re(\lambda_i(T - t_0)) << 0$, and where no other eigenvalue satisfies $Re(\lambda_j(t - t_0)) >> 0$. A nonlinear problem may be stiff for some intervals of the independent variable, but not for others. Furthermore, it is generally the case that a problem is not stiff in an initial transient region where the solution $y(t)$ to (1.1) changes rapidly. For more examples and discussion of stiff problems, see [17].

1.1.2 Validated Methods for Stiff Problems

Consider the simple test problem

$$y' = \lambda y, \quad y(0) = y_0 \in [y_0] \tag{1.6}$$

where $\lambda \in \mathbb{R}$, $\lambda < 0$ and $[y_0]$ is an interval = $[\underline{y}_0, \overline{y}_0]$ with $\underline{y}_0 \leq \overline{y}_0$.

The set of intervals on the real line $\mathbb{R}$ is defined by

$$\mathbb{IR} = \{[a] = [\underline{a}, \overline{a}] \mid \underline{a}, \overline{a} \in \mathbb{R}, \ \underline{a} \leq \overline{a}\}.$$

If $\underline{a} = \overline{a}$, then $[a]$ is a *thin* interval; if $\underline{a} \geq 0$, then $[a]$ is *non-negative* ($[a] \geq 0$); and if $\underline{a} = -\overline{a}$, then $[a]$ is *symmetric*. An *interval vector* is a vector with interval components and we denote the set of n-dimensional real interval vectors by $\mathbb{IR}^n$. Details of the interval-arithmetic operations involving scalar components and interval components can be found in [12].

As noted above, most validated methods are based on Taylor series and use interval arithmetic to generate guaranteed global error bounds. The Taylor series methods are explicit and,

as is well-know, they are not suitable for stiff problems. It was observed by Nedialkov [13], that, even if a formula such as the Hermite-Obreschkoff formula which is suitable for stiff problems when implemented in a standard numerical method is used, a validated numerical method based on the same formula may be unsuitable for stiff problems. This is due to a term in the formula for the associated global error bound, which "blows up" when $|h\lambda|$ is large, causing severe step size restrictions. As a result, all traditional validated methods that we are aware of are unsuitable for stiff problems. For further details and discussion on why traditional validated methods are unsuitable for stiff problems, see [13].

1.2 Neumaier's Validated Method

Neumier's validated method uses a different approach when computing error bounds for the solution to a stiff problem. We believe that this method is suitable for stiff problems in the sense that it does not suffer from a severe step size restrictions when applied to stiff problems.

To briefly illustrate Neumaier's method, let the grid points $\{t_k\}$ satisfy $t_0 < t_1 < \cdots < t_K = T$, and denote the step size on the k^{th} step from t_{k-1} to t_k by $h_k = t_k - t_{k-1}$. Let $y(t)$ be the true solution of (1.1) and let $p(t)$ be a piecewise smooth approximation to $y(t)$. Then the global error associated with the piecewise smooth function $p(t)$ is

$$e(t) = p(t) - y(t), \;\; t \in [t_0, T],$$

and the associated defect is

$$\delta(t) = p'(t) - f(t, p(t)), \;\; t \in [t_0, T].$$

Let $S \in \mathbb{R}^{n\times n}$ be invertible. Neumaier's validated method is based on a theorem that uses properties of the *logarithmic norm* (reviewed in Chapter 2) to show that, if we take $t_0 = 0$ and $t_1 = \overline{t}$, $||S^{-1}e(0)|| \leq \alpha$, $||S^{-1}\delta(t)|| \leq \epsilon$ for all $t \in [0, \overline{t}]$, the *logarithmic norm* of

$S^{-1}f_y(t,y)S \leq \mu$ for all $t \in [0,\overline{t}]$ and for all y in a suitable domain, then

$$||S^{-1}e(t)|| \leq \begin{cases} \alpha e^{\mu t} + \frac{\epsilon}{\mu}(e^{\mu t} - 1), & \text{if } \mu \neq 0 \\ \alpha + \epsilon t, & \text{if } \mu = 0 \end{cases}$$

for all $t \in [0,\overline{t}]$.

One might assume that, when the differential equation (1.1) with $(t_k = 0)$ satisfies the *uniform dissipation condition* $\mu < 0$, rigorous and realistic global error bounds of approximate solutions for stiff systems can be obtained for all times. However, in practice, it was found that, for our simple version of Neumaier's method, this is not the case for some stiff problems, due to a wrapping effect (discussed in Chapter 3).

1.3 Thesis Outline

A brief outline of this thesis follows. Chapter 2 contains the background material needed for this thesis. In particular, we review vector norms, matrix norms, the logarithmic norm and generation of Taylor expansions, as well as some relevant results of Dahlquist and Neumaier. In Chapter 3, we discuss the implementation of Neumaier's enclosure method, including the choices that we made for various parameters in his method. Chapter 4 presents numerical results of seven simple stiff IVPs. The results obtained as well as some shortcomings that they reveal in our implementation are discussed. Conclusions and directions for further research are laid out in Chapter 5.

Chapter 2

Preliminaries

This chapter reviews some mathematical background that is needed later in this thesis, including vector norms, matrix norms, the logarithmic norm, the mean value theorem for functions of several variables and the generation of Taylor expansions, as well as the enclosure methods of Dahlquist and Neumaier.

2.1 Norms

2.1.1 Vector Norm

Given a n-dimensional vector $x = (x_1, x_2..., x_n)^T \in \mathbb{R}^n$, a general vector norm is a mapping from $\mathbb{R}^n$ to $\{x|x \in \mathbb{R}, x \geq 0\}$ that satisfies the three conditions:

$$||x|| \leq 0 \text{ and } ||x|| = 0 \text{ iff } x = 0, \tag{2.1}$$

$$||\alpha x|| = |\alpha| \cdot ||x||, \text{ for all } \alpha \in \mathbb{R}, \tag{2.2}$$

$$||x + y|| \leq ||x|| + ||y||. \tag{2.3}$$

A frequently used class of vector norms, called the p-norms, is defined by

$$||x||_p = (\sum_i |x_i|^p)^{1/p}, \tag{2.4}$$

and the most commonly encountered p-norms are

$$||x||_1 = \sum_i |x_i|, \tag{2.5}$$

$$||x||_2 = (\sum_i |x_i|^2)^{1/2} = (x^T x)^{1/2}, \tag{2.6}$$

$$||x||_\infty = \max_{1 \le i \le n} |x_i|. \tag{2.7}$$

2.1.2 Matrix Norm

Note that we are only interested in matrix norms that are subordinate to a vector norm. The matrix norm $||A||$ subordinate to a vector norm is defined as

$$||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||}, \text{ where } A \in \mathbb{R}^{n \times n} \tag{2.8}$$

and it satisfies three conditions similar to (2.1), (2.2), (2.3) and condition

$$||AB|| \le ||A||||B||, \text{ where } A \text{ and } B \in \mathbb{R}^{n \times n}. \tag{2.9}$$

Furthermore,

$$||Ax|| \le ||A||||x|| \tag{2.10}$$

follows immediately from (2.8).

The matrix p-norms, for p = 1,2 and ∞, are

$$||A||_1 = \sup_{x \neq 0} \frac{||Ax||_1}{||x||_1} = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|, \tag{2.11}$$

$$||A||_2 = \sup_{x \neq 0} \frac{||Ax||_2}{||x||_2} = \max(\sqrt{\lambda} : \lambda \text{ is an eigenvalue of } A^T A), \tag{2.12}$$

$$||A||_\infty = \sup_{x \neq 0} \frac{||Ax||_\infty}{||x||_\infty} = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|. \tag{2.13}$$

2.1.3 Logarithmic Norm

The logarithmic norm (also known as the logarithmic derivative or the measure of a matrix) was introduced by Dahlquist [1] and Lozinskij [9]. It is often used to study the growth of errors associated with the numerical solutions to ODEs. Let A $\in \mathbb{R}^{n\times n}$ and let $||\cdot||$ be any matrix norm subordinate to a vector norm, the logarithmic norm of A is defined as

$$\mu(A) = \lim_{h\to+0} \frac{||I+hA|| - 1}{h}. \tag{2.14}$$

The properties of the logarithmic norm include

$$\alpha(A) \leq \mu(A) \leq ||A|| \tag{2.15}$$

$$\mu(A+B) \leq \mu(A) + \mu(B) \tag{2.16}$$

$$\mu(cA) = c\mu(A), \text{ for any real } c \geq 0 \tag{2.17}$$

$$||e^{At}|| \leq e^{\mu(A)t}, \quad t \geq 0 \tag{2.18}$$

$$||\cdot||_1 \Rightarrow \mu_1(A) = \max_{1\leq j\leq n}(a_{jj} + \sum_{i\neq j, i=1}^{n} |a_{ij}|) \tag{2.19}$$

$$||\cdot||_2 \Rightarrow \mu_2(A) = \alpha(\frac{1}{2}(A+A^T)) \tag{2.20}$$

$$||\cdot||_\infty \Rightarrow \mu_\infty(A) = \max_{1\leq i\leq n}(a_{ii} + \sum_{j\neq i, j=1}^{n} |a_{ij}|) \tag{2.21}$$

where A=(a_{ij}) and $\alpha(A)$ = max { a: $\lambda = a+ib$ is an eigenvalue of A } is the "spectral abscissa" of A. These and other properties of the logaritmic norm can be found in Ström [18]. Note that $\mu(A)$ can be negative.

2.2 Mean Value Theorem for Functions of Several Variables

Let $F : D \subset \mathbb{R}^n \to \mathbb{R}^m$ be differentiable at every point in a convex set D. Then for any two points $x \in D$ and $y \in D$

$$\begin{aligned} F(y) - F(x) &= \int_0^1 F'(y - t(y-x))(y-x)dt \\ &= J(y-x), \end{aligned}$$

where $J = \int_0^1 F'(y - t(y - x))dt$ is an "averaged" Jacobian of F.

Proof:

$$\begin{aligned}\int_0^1 F'(y - t(y - x))(y - x)dt &= -\int_0^1 (\frac{d}{dt}F(y - t(y - x)))dt \\ &= -F(y - t(y - x))|_{t=0}^{t=1} \\ &= F(y) - F(x).\end{aligned}$$

2.3 Generation of the Taylor Expansion

Since we need to generate the Taylor expansion of a function about t_0, we briefly describe two ways of generating such an expansion: automatic differentiation and symbolic differentiation.

2.3.1 Automatic Differentiation

Here we briefly describe the automatic generation of point (as opposed to interval) Taylor coefficients. Denote the ith Taylor coefficient of $u(t)$ evaluated at some point t_j by

$$(u_j)_i = \frac{u^{(i)}(t_j)}{i!}.$$

where $u^{(i)}$ is the ith derivative of $u(t)$. Let $(u_j)_i$ and $(v_j)_i$ be the ith Taylor coefficients of $u(t)$ and $v(t)$ at t_j respectively, then it is easily shown,

$$(u_j \pm v_j)_i = (u_j)_i \pm (v_j)_i \tag{2.22}$$

$$(u_j v_j)_i = \sum_{r=0}^{i} (u_j)_r (v_j)_{i-r} \tag{2.23}$$

$$(\frac{u_j}{v_j})_i = \frac{1}{v_j}\{(u_j)_i - \sum_{r=1}^{i} (v_j)_r (\frac{u_j}{v_j})_{i-r}\}. \tag{2.24}$$

Consider the autonomous differential system

$$y'(t) = f(y), \quad y(t_j) = y_j. \tag{2.25}$$

Introduce the following sequence of functions

$$f^{[0]}(y) = y, \tag{2.26}$$

$$f^{[i]}(y) = \frac{1}{i}(\frac{\partial f^{[i-1]}}{\partial y} f)(y), \text{ for } i \geq 1. \tag{2.27}$$

Using (2.25) – (2.27), the Taylor coefficients of $y(t)$ at t_j satisfy

$$(y_j)_0 = f^{[0]}(y_j), \tag{2.28}$$

and

$$\begin{aligned} (y_j)_i &= f^{[i]}(y_j) = \frac{1}{i}(\frac{\partial f^{[i-1]}}{\partial y} f)(y_j) \\ &= \frac{1}{i}(f(y_j))_{i-1}, \text{ for } i \geq 1, \end{aligned} \tag{2.29}$$

where $(f(y_j))_{i-1}$ is the $(i-1)$st Taylor coefficient of $f(y(t))$ evaluated at y_j. If $f(y)$ contains the simple arithmetic operations, $+$, $-$, $*$, $/$, we can recursively evaluate $(y_j)_i$ by using (2.22) – (2.24) and (2.28). For a more detailed discussion of automatic differentiation, see ([12], [10], [6]).

2.3.2 Symbolic Differentiation

MAPLE (the software package we used for our implementation of Neumaier's enclosure method), has a built-in function to solve an IVP or a system of IVPs symbolically, outputing the solution in the form of a Taylor series expansion.

To compute the Taylor series expansion, MAPLE uses one of the following three methods:
• The first method is a Newton iteration described by Geddes [5]. This paper considers the problem of computing the solution of an initial value problem for a first-order implicit differential equation

$$G(y, y') = 0, \quad y(0) = \alpha_0, \tag{2.30}$$

where $G(y, y')$ is a polynomial in y (scalar) and y' with coefficients which are power series in t. The method computes the solution in the power series form

$$y = y(t) = \sum_{k=0}^{\infty} y_k t^k,$$

by applying a Newton iteration. The Newton iteration formula for the problem (2.30) is obtained by considering the bivariate Taylor series expansion of the function $G(y, y')$. If y_k is a Taylor power series approximation to y, then expanding $G(y, y')$ about the "point" (y_k, y_k') gives

$$G(y, y') = G(y_k, y_k') + (y - y_k)G_y(y_k, y_k') + (y' - y_k')G_{y'}(y_k, y_k') + \cdots,$$

where G_y and $G_{y'}$ are the partial derivatives of G with respect to y and y' respectively. If $G(y, y') = 0$, then by ignoring terms beyond the first degree, we get

$$G_{y'}(y' - y_k') + G_y(y - y_k) \approx -G,$$

where functions G, G_y and $G_{y'}$ are evaluated at y_k and y_k'. Thus, if the linear ODE

$$G_{y'}e_k' + G_y e_k = -G$$

is solved for the "correction term" e_k as a power series, then

$$y_{k+1} = y_k + e_k$$

yields a new higher-order approximation y_{k+1} to y.

• The second method involves a direct substitution to generate a system of equations. The exact details of this method are unclear from the MAPLE documentation.

• The third method is the method of Frobenius for n^{th}-order linear differential equations. The method of Frobenius can be used to obtain general solution in the form of series. For the purpose of our discussion, we will assume that $t = 0$ of the ODE,

$$P(t)y'' + Q(t)y' + R(t)y = 0. \tag{2.31}$$

To find the solutions of (2.31) by the method of Frobenius, assume a solution of the form

$$y = t^k \sum_{n=0}^{\infty} a_n t^n = \sum_{n=0}^{\infty} a_n t^{n+k}, \tag{2.32}$$

exists. Hence,

$$y' = \sum_{n=0}^{\infty} a_n (n+k) t^{k+n-1},$$

$$y'' = \sum_{n=0}^{\infty} a_n (n+k)(n+k-1) t^{k+n-2}.$$

Substitute y back into (2.31), group the coefficients by power to obtain a recursion relation for the coefficient a_n, and then write the series expansion in terms of these a_ns. Equating the a_0 term to 0 yields the indical equation (also known as the characteristic equation), which gives the allowed values of k in the series expansion. For further details, see ([7], [3]).

It is unclear from the MAPLE documentation how it chooses which of these methods to use to obtain the series expansion. This lack of clarity in the documentation is one of the shortcomings we encounterd with MAPLE.

Symbolic differentiation, compared to the automatic differentiation, can be considerably more expensive with respect to computing time and memory space. However, the goal of this research is to get an idea of the potential of Neumaier's method for stiff problems, thus the cost of the differentiation process was not an important consideration in our choice of method.

2.4 Dahlquist's Results

Dahlquist showed how the global error associated with the numerical solution of a stiff IVP for an ODE can be bounded using the logarithmic norm. The following is Dahlquist's enclosure formula for the global error associated with an approximate solution to an IVP for a stiff ODE.

THEOREM 1 (DAHLQUIST) *Let $f : [0, \overline{t}] \times \mathbb{R}^n \to \mathbb{R}^n$ and let $p : [0, \overline{t}] \to \mathbb{R}^n$ be an approximate solution of the IVP*

$$y'(t) = f(t, y(t)), \quad y(0) = y_0, \quad t \in [0, \overline{t}] \tag{2.33}$$

in the sense that

$$||e(0)|| \leq \alpha, \tag{2.34}$$

$$||\delta(t)|| \leq \rho(t), \quad \textit{for all } \ t \in [0, \overline{t}], \tag{2.35}$$

where $e(t) = p(t) - y(t)$ *and* $\delta(t) = p'(t) - f(t, p(t))$. *If*

$$\mu(f_y(t, p(t) - s(p(t) - y(t)))) \leq c(t), \quad \textit{for all } \ t \in [0, \overline{t}] \quad \textit{and for all } \ s \in [0, 1], \tag{2.36}$$

then (2.33) has an unique solution $y : [0, \overline{t}] \to \mathbb{R}^n$ *and* $p(t)$ *satisfies*

$$||e(t)|| \leq \alpha e^{\int_0^t c(s)ds} + e^{\int_0^t c(s)ds} \int_0^t \rho(s) e^{-\int_0^s c(\eta)d\eta} ds, \quad \textit{for all } \ t \in [0, \overline{t}]. \tag{2.37}$$

The proof of this theorem can be found in [1].

COROLLARY 1 *Let* $f : [0, \overline{t}] \times \mathbb{R}^n \to \mathbb{R}^n$ *and let* $p : [0, \overline{t}] \to \mathbb{R}^n$ *be an approximate solution of the IVP (2.33) in the sense that*

$$||e(0)|| \leq \alpha, \tag{2.38}$$

$$||\delta(t)|| \leq \epsilon, \quad \textit{for all } \ t \in [0, \overline{t}], \tag{2.39}$$

where $e(t) = p(t) - y(t)$ *and* $\delta(t) = p'(t) - f(t, p(t))$. *If*

$$\mu(f_y(t, p(t) - s(p(t) - y(t)))) \leq \mu, \quad \textit{for all } \ t \in [0, \overline{t}] \quad \textit{and for all } \ s \in [0, 1], \tag{2.40}$$

then (2.33) has an unique solution $y : [0, \overline{t}] \to \mathbb{R}^n$ *and* $p(t)$ *satisfies*

$$||e(t)|| \leq \alpha e^{\mu t} + \epsilon e^{\mu t} \int_0^t e^{-\mu s} ds = \begin{cases} \alpha e^{\mu t} + \frac{\epsilon}{\mu}(e^{\mu t} - 1), & \textit{if } \ \mu \neq 0 \\ \alpha + \epsilon t, & \textit{if } \ \mu = 0 \end{cases} \tag{2.41}$$

2.5 Neumaier's Results

Neumaier's enclosure formula for an approximate solution to an IVP for a stiff ODE is similar to Dahlquist's. He also uses the logarithmic norm to obtain global enclosures for a class of ODEs containing those satisfying a uniform dissipation condition.

THEOREM 2 (NEUMAIER) *Let $f : [0,\overline{t}] \times \mathbb{R}^n \to \mathbb{R}^n$. Let $S \in \mathbb{R}^{n\times n}$ be invertible, and let $p : [0,\overline{t}] \to \mathbb{R}^n$ be an approximate solution of the IVP*

$$y'(t) = f(t, y(t)), \quad y(0) = y_0, \quad t \in [0,\overline{t}] \tag{2.42}$$

in the sense that

$$||S^{-1}e(0)|| \leq \alpha, \tag{2.43}$$

$$||S^{-1}\delta(t)|| \leq \epsilon, \quad \textit{for all } \ t \in [0,\overline{t}], \tag{2.44}$$

where $e(t) = p(t) - y(t)$ and $\delta(t) = p'(t) - f(t, p(t))$. If

$$\mu(S^{-1}f_y(t, y)S) \leq \mu, \quad \textit{for all } \ t \in [0,\overline{t}] \quad \textit{and for all } \ y \in \mathbb{R}^n, \tag{2.45}$$

then (2.42) has an unique solution $y : [0,\overline{t}] \to \mathbb{R}^n$ and $p(t)$ satisfies

$$||S^{-1}e(t)|| \leq \phi(t) := \alpha e^{\mu t} + \epsilon t \exp_1(\mu t), \quad \textit{for } \ t \in [0,\overline{t}], \tag{2.46}$$

where

$$\exp_1(\xi) := \begin{cases} \frac{(e^{\xi}-1)}{\xi}, & \xi \neq 0 \\ 1, & \xi = 0 \end{cases}$$

The proof of this theorem can be found in [16].

COROLLARY 2 *Let $f : [0,\overline{t}] \times \mathbb{R}^n \to \mathbb{R}^n$, let $S = I$, and let $p : [0,\overline{t}] \to \mathbb{R}^n$ be an approximate solution of the IVP (2.42) in the sense that*

$$||e(0)|| \leq \alpha, \tag{2.47}$$

$$||\delta(t)|| \leq \epsilon, \quad \textit{for all } \ t \in [0,\overline{t}], \tag{2.48}$$

$$\mu(f_y(t, y)) \leq \mu, \quad \textit{for all } \ t \in [0,\overline{t}], \quad \textit{for all } \ y \in \mathbb{R}^n, \tag{2.49}$$

then (2.42) has an unique solution $y : [0,\overline{t}] \to \mathbb{R}^n$ and $p(t)$ satisfies

$$||e(t)|| \leq \begin{cases} \alpha e^{\mu t} + \frac{\epsilon}{\mu}(e^{\mu t} - 1), & \textit{if } \ \mu \neq 0 \\ \alpha + \epsilon t, & \textit{if } \ \mu = 0 \end{cases} \tag{2.50}$$

This result follows immediately from Corollary 1.

2.6 Combining Dahlquist's and Neumaier's Results

Neumaier's result (Theorem 2) is really a special case of Dahlquist's result (Theorem 1). To illustrate, we state the following generalization of Dahlquist's theorem.

THEOREM 3 *Let $f : [0, \overline{t}] \times \mathbb{R}^n \to \mathbb{R}^n$. Let $S \in \mathbb{R}^{n\times n}$ be invertible, and let $p : [0, \overline{t}] \to \mathbb{R}^n$ be an approximate solution of the IVP*

$$y'(t) = f(t, y(t)), \quad y(0) = y_0, \quad t \in [0, \overline{t}] \tag{2.51}$$

in the sense that

$$||S^{-1}e(0)|| \leq \alpha, \tag{2.52}$$

$$||S^{-1}\delta(t)|| \leq \rho(t), \quad \textit{for all } \ t \in [0, \overline{t}], \tag{2.53}$$

where $e(t) = p(t) - y(t)$ and $\delta(t) = p'(t) - f(t, p(t))$. If

$$\mu(S^{-1}f_y(t, p(t) - s(p(t) - y(t)))S) \leq c(t), \quad \textit{for all } \ t \in [0, \overline{t}] \quad \textit{and for all } \ s \in [0, 1], \tag{2.54}$$

then (2.51) has an unique solution $y : [0, \overline{t}] \to \mathbb{R}^n$ and $p(t)$ satisfies

$$||S^{-1}e(t)|| \leq \alpha e^{\int_0^t c(s)ds} + e^{\int_0^t c(s)ds} \int_0^t \rho(s)e^{-\int_0^s c(\eta)d\eta}ds, \quad \textit{for all } \ t \in [0, \overline{t}]. \tag{2.55}$$

Neumaier results, Theorem 2, is a special use of Theorem 3. We now derive Theorem 3 from Dahlquist's result, Theorem 1.

Since S is an invertible matrix, we may define

$$\begin{aligned} y &= Su, \\ \Leftrightarrow u &= S^{-1}y. \end{aligned} \tag{2.56}$$

Substituting this change of variables into (2.51) yields

$$\begin{aligned} Su' &= f(t, Su), \\ \Leftrightarrow u' &= S^{-1}f(t, Su) \\ &= F(t, u), \end{aligned} \tag{2.57}$$

resulting in the following IVP

$$u' = F(t,u), \quad u(0) = S^{-1}y(0) = S^{-1}y_0. \tag{2.58}$$

Let $p(t)$ be an approximate solution to (2.51). Then

$$\begin{aligned} \delta(t) &= p'(t) - f(t,p(t)), \\ \Leftrightarrow p'(t) &= f(t,p(t)) + \delta(t). \end{aligned}$$

Let

$$\begin{aligned} v &= S^{-1}p, \\ \Leftrightarrow p &= Sv. \end{aligned} \tag{2.59}$$

Then

$$\begin{aligned} Sv' &= f(t,Sv) + \delta(t), \\ \Leftrightarrow v' &= S^{-1}f(t,Sv) + S^{-1}\delta(t) \\ &= F(t,v) + \triangle(t), \end{aligned} \tag{2.60}$$

where $\triangle(t) = S^{-1}\delta(t)$ is the defect associated with the approximate solution $v(t)$ to the IVP (2.58).

Now assume that

$$||E(0)|| \leq \alpha \tag{2.61}$$

$$||\triangle(t)|| \leq \rho(t), \quad \text{for all } \ t \in [0,\overline{t}], \tag{2.62}$$

$$\mu(F_u(t, v(t) - s(v(t) - u(t)))) \leq c(t) \quad \text{for all } \ t \in [0,\overline{t}] \quad \text{and for all } \ s \in [0,1], \tag{2.63}$$

where $E(t) = v(t) - u(t)$ and $\triangle(t) = v'(t) - F(t,v)$. Then, by Theorem 1, (2.58) has an unique solution $u : [0,\overline{t}] \to \mathbb{R}^n$ satisfying

$$||E(t)|| \leq \alpha e^{\int_0^t c(s)ds} + e^{\int_0^t c(s)ds} \int_0^t \rho(s) e^{-\int_0^s c(\eta)d\eta} ds, \quad \text{for all } \ t \in [0,\overline{t}]. \tag{2.64}$$

Since

$$\begin{aligned} E(t) &= v(t) - u(t) \\ &= S^{-1}(p(t) - y(t)) \\ &= S^{-1}e(t), \end{aligned}$$

and

$$E(0) = S^{-1}e(0),$$

(2.61) is equivalent to (2.52):

$$\begin{aligned} ||E(0)|| &\leq \alpha \\ \Leftrightarrow ||S^{-1}e(0)|| &\leq \alpha. \end{aligned} \tag{2.65}$$

Since $\triangle(t) = S^{-1}\delta(t)$, (2.62) is equivalent to (2.53):

$$\begin{aligned} ||\triangle(t)|| &\leq \rho(t), \quad \text{for all } \ t \in [0, \overline{t}] \\ \Leftrightarrow ||S^{-1}\delta(t)|| &\leq \rho(t) \quad \text{for all } \ t \in [0, \overline{t}]. \end{aligned}$$

Finally, (2.63) is equivalent to (2.54):

$$\begin{aligned} \mu(F_u(t, v(t) - s(v(t) - u(t))) &\leq c(t) \quad \text{for all } \ t \in [0, \overline{t}] \quad \text{and for all } \ s \in [0, 1] \\ \Leftrightarrow \mu(S^{-1}f_u(t, Sv(t) - s(Sv(t) - Su(t)))) &\leq c(t) \quad \text{for all } \ t \in [0, \overline{t}] \quad \text{and for all } \ s \in [0, 1] \\ \Leftrightarrow \mu(S^{-1}f_y(t, p(t) - s(p(t) - y(t)))S) &\leq c(t) \quad \text{for all } \ t \in [0, \overline{t}] \quad \text{and for all } \ s \in [0, 1]. \end{aligned}$$

Hence, (2.52), (2.53), (2.54) imply that (2.51) has a unique solution $y : [0, \overline{t}] \to \mathbb{R}^n$ satisfying

$$||S^{-1}e(t)|| \leq \alpha e^{\int_0^t c(s)ds} + e^{\int_0^t c(s)ds} \int_0^t \rho(s) e^{-\int_0^s c(\eta)d\eta} ds, \quad \text{for all } \ t \in [0, \overline{t}].$$

That is, Theorem 3 holds.

COROLLARY 3 *Let $f : [0, \overline{t}] \times \mathbb{R}^n \to \mathbb{R}^n$. Let $S \in \mathbb{R}^{n \times n}$ be invertible, and let $p : [0, \overline{t}] \to \mathbb{R}^n$ be an approximate solution of the IVP (2.51) in the sense that*

$$||S^{-1}e(0)|| \leq \alpha, \tag{2.66}$$

$$||S^{-1}\delta(t)|| \leq \epsilon, \quad \textit{for all } \ t \in [0, \overline{t}], \tag{2.67}$$

where $e(t) = p(t) - y(t)$ *and* $\delta(t) = p'(t) - f(t, p(t))$. *If*

$$\mu(S^{-1} f_y(t, p(t) - s(p(t) - y(t)))S) \leq \mu, \quad \textit{for all } \ t \in [0, \overline{t}] \quad \textit{and for all } \ s \in [0, 1], \tag{2.68}$$

then (2.51) has an unique solution $y : [0, \overline{t}] \to \mathbb{R}^n$ *and* $p(t)$ *satisfies*

$$||S^{-1}e(t)|| \leq \phi(t) := \alpha e^{\mu t} + \epsilon e^{\mu t} \int_0^t e^{-\mu s} ds = \begin{cases} \alpha e^{\mu t} + \frac{\epsilon}{\mu}(e^{\mu t} - 1), & \textit{if } \ \mu \neq 0 \\ \alpha + \epsilon t, & \textit{if } \ \mu = 0 \end{cases} \tag{2.69}$$

This result follows immediately from Theorem 3.

We use Corollary 3 in our implementation of Neumaier's enclosure method. Note that the difference between Corollary 3 and Neumaier's Corollary 2 is how μ is obtained. Corollary 3 considers $y \in p(t) - s(p(t) - y(t))$ where $s \in [0, 1]$ in (2.68), while Corollary 2 considers all y $\in \mathbb{R}^n$ in (2.45). Since we may be able to obtain much better bounds for μ over a small set enclosing y and p, than over all y $\in \mathbb{R}^n$, we use Corollary 3 in our implementation of Neumaier's enclosure method.

Chapter 3

Implementation of Neumaier's Enclosure Method

In this chapter, we discuss our implementation of Neumaier's enclosure method (Corollary 3), including the choices that we made for various parameters in his method, as well as a simple stepsize control strategy. Our implementation is non-rigorous, in the sense that our methods of estimating some of these parameters are not rigorous, as discussed in more detail below. The purpose of the non-rigorous implementation is to investigate the potential of Neumaier's enclosure method for stiff problems and to provide insight into how this method behaves in practice. A rigorous implementation would have taken considerably longer to develop. The advantage of working in MAPLE is that it has a built-in differential equations tools package (`DEtools`), and so an implementation of Neumaier's method is easier for us to develop than it would be if we did not have these tools available.

Consider the initial value problem (IVP) for an ordinary differential equation (ODE)

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad t \in [t_0, T] \tag{3.1}$$

where y is a system of n components, $y \in \mathbb{R}^n$ and $f : [t_0, T] \times \mathbb{R}^n \to \mathbb{R}^n$. Let the grid points $\{t_k\}$ satisfy $t_0 < t_1 < \cdots < t_K = T$, and denote the stepsize on the kth integration timestep from t_{k-1} to t_k by $h_k = t_k - t_{k-1}$. Let $y(t)$ be the true solution of (3.1) and let $p(t)$ be a

piecewise smooth approximation to $y(t)$. The global error associated with $p(t)$ is

$$e(t) = p(t) - y(t), \quad t \in [t_0, T], \tag{3.2}$$

and the associated defect is

$$\delta(t) = p'(t) - f(t, p(t)), \quad t \in [t_0, T], \tag{3.3}$$

where we assume that t is not one of the grid points associated with p.

Let $S \in \mathbb{R}^{n\times n}$ be invertible. Note that S is consistant with each step $[t_k, t_{k+1}]$, but S may change between steps. From Corollary 3, we take $t_k = 0$, $t_{k+1} = \overline{t}$ and assume,

$$||S^{-1}e(0)|| \leq \alpha, \tag{3.4}$$

$$||S^{-1}\delta(t)|| \leq \epsilon, \quad \text{for all } \ t \in [0, \overline{t}], \tag{3.5}$$

$$\mu(S^{-1}f_y(t, p(t) - s(p(t) - y(t)))S) \leq \mu, \quad \text{for all } \ t \in [0, \overline{t}] \quad \text{and for all } \ s \in [0, 1], \tag{3.6}$$

then

$$||S^{-1}e(t)|| \leq \phi(t) := \begin{cases} \alpha e^{\mu t} + \frac{\epsilon}{\mu}(e^{\mu t} - 1), & \text{if } \ \mu \neq 0 \\ \alpha + \epsilon t, & \text{if } \ \mu = 0 \end{cases} \tag{3.7}$$

for all $t \in [0, \overline{t}]$. Hence, we can advance a global error bound at t_k in (3.4), to a global error bound at t_{k+1} in (3.7).

Note, in our implementation, we consider t to be implicitly re-initialized to $t = 0$ at every integration timestep k so that we can apply the result above starting from t_k (t = 0) and ending at t_{k+1} ($t = \overline{t}$). Note also, for the rest of this thesis, we take $||\cdot||$ to be $||\cdot||_2$. In addition, we assume throughout the rest of this thesis that $\mu < 0$.

We now discuss how $p(t)$, S, μ, ϵ and α are computed in our implementation of Neumaier's method, as well as a simple stepsize control strategy based on controlling $\phi(t)$.

3.1 Choice of $p(t)$

The function $p(t)$ is a piecewise smooth approximation of $y(t)$ on interval $[t_0, T]$. Two natural choices for this piecewise smooth approximation are a piecewise polynomial and a piecewise

rational function. We chose the latter for our implementation, since some preliminary testing by Neher [14] suggests that this choice is better for stiff problems. More specifically, we use a piecewise rational Pade approximation described below.

3.1.1 Taylor Expansion

To obtain a Pade rational approximation for each component of vector $y(t)$ on the interval $I_k = [t_k, t_{k+1}]$, we first compute the Taylor expansion of the exact local solution to the IVP $y' = f(t, y)$, $y(t_k) = y_k$ for each component. This is computed symbolically in MAPLE, using a built-in function to construct the Taylor series expansion. Since the goal of this research is to get an idea of the potential of Neumaier's method for stiff problems, this method of generation is sufficient for our purpose, although it would not be recommended for an efficient implementation.

For each component of vector $y(t)$, the output from MAPLE is a Taylor polynomial of the form

$$\hat{y}_r(t) = \sum_{j=0}^{N} y_j (t - t_k)^j, \quad \text{for } r = 1, \ldots, n \text{ and } n \text{ is the number of components in } y(t), \tag{3.8}$$

where

$$y_j = \frac{1}{j!} y^{(j)}(t_k). \tag{3.9}$$

The order N of the polynomial $\hat{y}_r$ is set to 20 in our implementation.

3.1.2 Pade Rational Approximation

We use MAPLE's built-in function to convert each polynomial $\hat{y}_r(t)$ in (3.8), where $r = 1, \ldots, n$ and n is the number of components in $y(t)$, into a Pade approximation of the form

$$\frac{a_r(t)}{b_r(t)}, \tag{3.10}$$

where

$$a_r(t) = \sum_{j=0}^{\hat{m}} a_j (t - t_k)^j$$

is a polynomial of degree $\hat{m}$ (set to 9 in our implementation), and

$$b_r(t) = 1 + \sum_{j=1}^{\hat{n}} b_j (t - t_k)^j$$

is a polynomial of degree $\hat{n}$ (set to 10 in our implementation). Note that further investigation is required in choosing these degrees. Also, further work is required in investigating other approximation methods that might be more suitable.

3.2 Choice of S

Recall from (3.6) that we need to choose a non-singular S and find a bound μ such that

$$\mu(S^{-1} f_y(t, y) S) \leq \mu, \quad \text{for all } \ t \in [0, \overline{t}], \tag{3.11}$$

where $y \in p(t) - s(p(t) - y(t))$ and $s \in [0, 1]$. We follow Neumaier's [16] suggestion and compute the eigenvectors of the Jacobian matrix

$$H := \frac{\partial f}{\partial y}(0, p(0)) \tag{3.12}$$

and use the real and imaginary parts of a full set of eigenvectors of H for the columns of S, assuming that H has a full set of eigenvectors. If H does not have a full set of eigenvectors, then it is defective. Since the H matrices in our examples are not defective, we did not consider this case, as the goal of this research is to get an idea of the potential of Neumaier's enclosure method for stiff problems, not to develop a robust code. For complex eigenvectors, we place the real part in one column of S and the imaginary part in the next. If S is ill-conditioned, Neumaier [16] suggests replacing the eigenvectors by independent basis vectors of an invariant subspace of H. However, we never found ill-conditioning to be a problem in our examples and so did not implement this alternative.

Note that in (3.12) we took $t = 0$ and $y = p(0)$ instead of taking the interval $[0, \overline{t}]$ and $y \in \{\, p(t) - s(p(t) - y(t)) : s \in [0, 1] \,\}$ in our implementation. It would require considerable further work to extend our implementation to take $t \in [0, \overline{t}]$ and $y \in \{\, p(t) - s(p(t) - y(t)) : s \in$

$[0,1]\ \}$ into account. One consequence of this is that, for our simple choice (3.12), $S^{-1}f_y(t,y)S$ is block diagonal (see below) in most cases, while this may not be the case over the full range $t \in [0,\overline{t}]$ and $y \in \{\ p(t) - s(p(t) - y(t)) : s \in [0,1]\ \}$. As the goal of this research, as noted above, is to get an idea of the potential of Neumaier's enclosure method for stiff problems, the restriction $t = 0$ and $y = p(0)$ is sufficient for our purpose.

Since we have assumed that $f_y(0, p(0))$ has a full set of eigenvectors, it is diagonalizable. The advantage of diagonalization is that it results in a block-diagonal matrix $M = S^{-1}HS$ with either a real eigenvalue λ on the diagonal or a 2×2 block of the form

$$\begin{bmatrix} Re\ \lambda & Im\ \lambda \\ -Im\ \lambda & Re\ \lambda \end{bmatrix}$$

on the diagonal. Hence, $\frac{M+M^T}{2}$ is a diagonal matrix with $Re\ \lambda$ on the diagonal, where λ is a complex eigenvalue of H, giving rise to $\mu(M) \leq \mu(H)$. Note that this is in some sense optimal in that no matrix $\hat{M}$ that is similar to H can have $\mu(\hat{M}) < \mu(M)$. If $f_y(0,y)$ is not diagonalizable, then μ may be larger than $Re\ \lambda$, since the entries along the superdiagonal of M are not all equal to 0.

An important decision in our implementation is how often we should re-evaluate S. Two possible options, along with their advantages and disadvantages, are outlined below.

The first option is to compute S at the initial time t_0 and leave it fixed for all integration timesteps. The advantages of keeping S fixed are that the implementation requires less work and there is no *wrapping* effect (discussed in §3.6.2). The disadvantage is that μ may not be optimal, since it can happen that

$$\mu(S_0^{-1}H_kS_0) > \mu(H_k).$$

The second option is to re-compute S at every integration timestep k. The advantage is that the optimum μ can be estimated at each integration timestep k:

$$\mu(S_k^{-1}H_kS_k) = \mu(H_k),$$

assuming that H_k is diagonalizable. The disadvantage is that this can lead to a *wrapping* effect, as discussed in §3.6.2.

There are many other possibilities for the choice of S_k and strategies for determining how often to re-evaluate S_k. For this thesis, we chose to implement the second option in our tests.

Before we discuss the classical wrapping effect (covered in §3.6.2) and how the wrapping effect is reflected in S (covered in §3.6.3), we first finish our discussion of how we compute μ, ϵ and α in our implementation.

3.3 Estimation of μ

To achieve rigorous and realistic error bounds, we must determine a tight upper bound μ_k on the *uniform dissipation condition* associated with step k,

$$\sup \{ \mu(S^{-1} f_y(t,y) S) \ : \ t \in [t_k, t_{k+1}],\ y \in (p(t) - s(p(t) - y(t))), s \in [0,1] \} \leq \mu_k.$$

In our implementation, S is re-computed at each integration timestep k, which results in the optimum μ being redefined for each integration timestep k. However, we use a non-rigorous method to estimate μ_k at each integration timestep k as follows:

$$\hat{\mu}_k := \mu(S_k^{-1} H_k S_k), \tag{3.13}$$

where

$$H_k = f_y(0, p(0))$$

at integration timestep k.

As a result of this simplification, it may happen that

$$\hat{\mu}_k < \sup \{ \mu(S^{-1} f_y(t,y) S) \ : \ t \in [t_k, t_{k+1}],\ y \in (p(t) - s(p(t) - y(t))), s \in [0,1] \}.$$

As noted earlier, this is one of several reasons why our implementation is not rigorous.

3.4 Estimation of ϵ

Recall from (3.5) that ϵ must satisfy

$$||S^{-1}(p'(t) - f(t, p(t))|| \leq \epsilon, \quad \text{for all } \ t \in [0, \overline{t}].$$

and that we re-compute S at each integration timestep k. That is, at each integration timestep k, ϵ_k should ideally satisfy

$$\epsilon_k := \max_{t \in [t_k, t_{k+1}]} ||S_k^{-1}(p_k'(t) - f(t, p_k(t)))||. \tag{3.14}$$

However, we use a non-rigorous method to estimate ϵ_k. In our implementation, we sample the defect at 21 evenly spaced points in the interval $I_k := [t_k, t_{k+1}]$ and approximate ϵ_k by

$$\hat{\epsilon}_k := max\{ \ ||S_k^{-1}(p_k'(t_{k_j}) - f(t_{k_j} - p_k(t_{k_j}))|| \ \ : \ \ t_{k_j} = t_k + j\frac{h_k}{20} \ , \ j = 0, \ldots, 20\}.$$

3.5 Estimation of α

Recall from (3.4) that α must satisfy

$$||S^{-1}(p(0) - y(0))|| \leq \alpha,$$

and that we re-compute S at each integration timestep k. We estimate α_k at each integration timestep k as follows.

Since the initial error $(p_k(0) - y_k(0))$ at integration timestep $k > 1$ is the error $p_{k-1}(t_k) - y(t_k)$ at the endpoint of the previous timestep $(k-1)$, and we know that

$$||S_{k-1}^{-1}(p_{k-1}(t_k) - y_{k-1}(t_k))|| \leq \phi_{k-1},$$

it follows that

$$\begin{aligned} ||S_k^{-1}(p_k(0) - y_k(0))|| &= ||S_k^{-1}(S_{k-1}S_{k-1}^{-1})(p_k(0) - y(0))|| \\ &\leq ||S_k^{-1}S_{k-1}||\ ||S_{k-1}^{-1}(p_k(0) - y(0))|| \\ &\leq ||S_k^{-1}S_{k-1}||\phi_{k-1}, \end{aligned}$$

where $\phi(t)$ is defined in (3.7). Hence we estimate α by

$$\alpha_k := ||S_k^{-1} S_{k-1}||\phi_{k-1}, \tag{3.15}$$

at each integration timestep $k > 1$. At $k = 1$, α is set to zero, since we know the initial condition $y(t_0)$ and we set $p(0) = y(t_0)$. Note that $S_k^{-1} S_{k-1}$ is computationally expensive, and thus it would not be recommended for an efficient implementation.

3.6 A Simple Stepsize Control Strategy

As noted above, we assume throughout this Chapter that $\mu < 0$. Recall from (3.7) that the global error bound ϕ is defined as

$$||S^{-1}(p(t) - y(t))|| \leq \phi(t) := \alpha e^{\mu t} + \frac{\epsilon}{\mu}(e^{\mu t} - 1), \quad \text{for all } t \in [0, T]$$

and that we re-compute S at each integration timestep k. We estimate the global error bound ϕ_k at the endpoint of each integration timestep k by

$$\phi_k(h_k) := \alpha_k e^{\mu_k h_k} + \frac{\epsilon_k}{\mu_k}(e^{\mu_k h_k} - 1),$$

where h_k is the stepsize at step k.

Given a user-specified global error bound τ, we try to control the global error by requiring

$$\alpha_k e^{\mu_k h_k} + \frac{\epsilon_k}{\mu_k}(e^{\mu_k h_k} - 1) \leq \tau \tag{3.16}$$

on each step k.

The idea behind this simple stepsize control strategy is to continue the integration while the defect is small compared to the propagated initial error in the integration timestep k.

3.6.1 Predicting a Stepsize

Suppose that the $(k-1)$th integration timestep is accepted. We attempt to choose a stepsize h_k at the next timestep satisfying both (3.16) and

$$\frac{h_{k-1}}{20^l} \leq h_k \leq 10 h_{k-1}, \tag{3.17}$$

where $l = 0, \ldots, 10$. In other words, the stepsize h_k at integration timestep k can increase within bounds as long as the condition (3.16) holds.

First we try to satisfy (3.16) with l = 0 in (3.17). If this fails, that is if $\phi_k(h_k) > \tau$, then $\phi_k(h_k)$ is re-evaluated with a smaller $h_k = \frac{h_{k-1}}{20^l}$, where l = 1 and this process is continued until the condition (3.16) is satisfied. If the condition (3.16) continues to fail with increasing l, then the integration is considered to have failed and the computation is terminated.

As long as there is no *wrapping* effect (discussed in the next subsection), the integration can always continue (in an idealized setting without roundoff error etc.) using this simple stepsize control strategy. To illustrate, recall that we estimate α_k at each integration timestep $k > 1$ as the initial global error at step k propagated from the endpoint of global error at timestep $(k-1)$. In other words, at timestep k, $\alpha_k \approx \tau$. Let $\mu_k = -x$. Recall $\mu_k < 0$, so $x > 0$. Note that

$$\begin{aligned} e^{\mu_k h_k} &= e^{-x h_k} \\ &= 1 - x h_k + O(h_k^2). \end{aligned}$$

Using this together with $\alpha_k = \tau$ in (3.16), we get

$$\tau(1 - xh + O(h^2)) - \frac{\epsilon}{x}((1 - xh + O(h^2)) - 1) \leq \tau,$$

$$\Leftrightarrow \frac{\epsilon}{x}(xh - O(h^2)) \leq \tau(xh - O(h^2)),$$

$$\Leftrightarrow \frac{\epsilon}{x}(1 - \frac{1}{x}O(h)) \leq \tau(1 - \frac{1}{x}O(h)),$$

$$\Leftrightarrow \frac{\epsilon}{x} - \frac{\epsilon}{x^2}O(h) + \frac{\tau}{x}O(h) \leq \tau,$$

$$\Leftrightarrow \frac{\epsilon}{x} + O(h) \leq \tau.$$

Since ϵ decreases to zero with h, where as $x = -\mu_k$ does not, this last inequality can be satisfied (in an idealized setting without roundoff error etc.) for h_k sufficiently small and so the integration can continue using this simple stepsize control strategy.

It was found that, for some problems, the integration failed to continue because the initial ϕ_k exceeded τ, though the condition (3.16) held at timestep $(k-1)$. This is due to a *wrapping* effect, as is explained below.

3.6.2 The Classical Wrapping Effect

To illustrated the classical *wrapping* effect, we consider Moore's example [12],

$$y_1' = y_2, \quad y_2' = -y_1. \tag{3.18}$$

The solution of (3.18) with an intial condition y_0 is given by $y(t) = A(t)y_0$, where

$$A(t) = \begin{bmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{bmatrix}.$$

Let $y_0 \in [y_0]$, where interval vector[1] $[y_0] \in \mathbb{IR}^2$ can be viewed as a rectangle in the (y_1, y_2) plane. As shown in Figure 3.1, $A(t_1)$ maps $[y_0]$ into a rectangle of the same size at $t_1 > t_0$. To enclose this rectangle in an interval vector, we have to wrap it with another rectangle with sides parallel to the y_1 and y_2 axes. On the next step, this larger rectangle is rotated and it is enclosed by a still larger rectangle. As a result, at each step, the enclosing rectangles become larger and larger. However, the set $\{A(t)y_0 \mid y_0 \in [y_0],\ t > t_0\}$ remains a rectangle of the original size. It was shown by Moore that, as the stepsize approaches zero, at $t = 2\pi$, the interval inclusion is inflated by a factor of $e^{2\pi} \approx 535$.

[1] An explaination of interval vectors can be found in §1.1.2

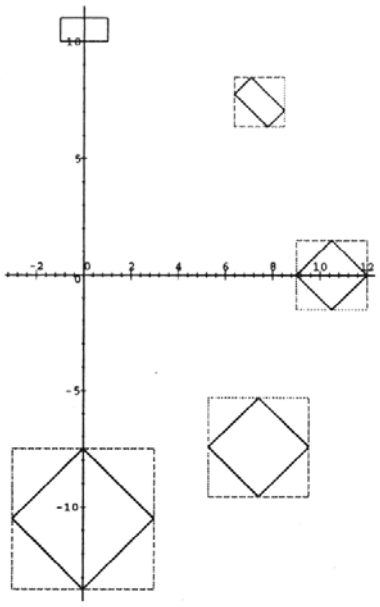


Figure 3.1: Wrapping of a rectangle specified by the interval vector $([-1,1],[10,11])^T$. The rotated rectangle is wrapped at $t = \frac{\pi}{4}n$, where $n = 1,\ldots,4$.

3.6.3 The Wrapping Effect in S

To see how a wrapping effect occurs in our chosen method of computing S and as a consequence, how the integration can fail to continue using this simple stepsize control strategy outlined above, recall from (3.15) that

$$\alpha_k := ||S_k^{-1} S_{k-1}|| \phi_{k-1}.$$

Let

$$C = ||S_k^{-1} S_{k-1}||. \tag{3.19}$$

If the eigenvectors of an IVP are time independent, then in our implementation, $S_k^{-1} S_{k-1}$ is the identity matrix. This implies that $C = 1$ and α_k is the value of ϕ at the endpoint of the $(k-1)$st integration timestep. Thus, for problems with $\mu < 0$ and with fixed eigenvectors, we can always (in theory) choose ϵ_k small enough so that the integration can continue while keeping the global error $\leq \tau$.

If the eigenvectors of an IVP are time dependent, then in our implementation, the matrix S_k is also time dependent. Because S_k changes from step to step, it is possible for $C > 1$ and as a result, α at integration timestep k may be greater than the value of ϕ at the endpoint of integration timestep $(k-1)$. In other words, depending on the value of C, the initial value of ϕ could exceed the maximum tolerated error τ, even if the value of ϕ at the endpoint of the previous integration timestep was below τ. Thus, due to the wrapping effect in S, the integration can fail to continue using the simple stepsize control strategy outlined above.

In Chapter 4, we present two example problems that illustrate the wrapping effect discussed here. Alternative stepsize control strategies are also presented and discussed there.

Chapter 4

Numerical Results and Discussion

In this chapter, seven 2D initial-value problems (IVPs) are presented. The numerical results obtained for these test problems and the shortcomings that they reveal in our implementation of Neumaier's enclosure method (Corollary 3) are discussed. In addition, we breifly discuss some alternative stepsize control strategies that alleviated some of the shortcomings.

4.1 Test Problems

Listed below are the seven 2D test problems, each of the form

$$y'(t) = A(t)y(t) + b(t),$$

where $t \in [0, T]$. These problems can be found in Neher ([15], [14]). The global error tolerance τ is set to 10^{-6}, a representative value for this parameter.

The initial condition for each test problem is

$$\begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Problem 1: Linear ODE with constant coefficients and zero $b(t)$, $t \in [0, 100]$, $\mu_k = \hat{\mu}_k = -1$ for all k:

$$\begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1000 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.1}$$

The true solution is $y_1(t) = e^{-t}$ and $y_2(t) = e^{-10^3 t}$.

Problem 2: Linear ODE with constant coefficients and constant non-zero $b(t)$, $t \in [0, 100]$, $\mu_k = \hat{\mu}_k = -1$ for all k:

$$\begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1000 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.2}$$

The true solution is $y_1(t) = 1$ and $y_2(t) = \frac{1}{10^3} + \frac{999}{10^3} e^{-10^3 t}$.

Problem 3: Linear ODE with constant coefficients and time dependant $b(t)$, $t \in [0, 100]$, $\mu_k = \hat{\mu}_k = -1$ for all k:

$$\begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1000 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} t \\ t \end{bmatrix} \tag{4.3}$$

The true solution is $y_1(t) = t - 1 + 2e^{-t}$ and $y_2(t) = \frac{1}{10^3} t - \frac{1}{10^6} + \frac{1000001}{10^6} e^{-10^3 t}$.

Problem 4: Linear ODE with time dependant coefficients and $b(t)$, $t \in [0, 20\pi]$, $\mu_k = \hat{\mu}_k = -1$ for all k:

$$\begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} = \begin{bmatrix} -5.5 + 4.5\cos(2t) & -4.5\sin(2t) \\ -4.5\sin(2t) & -5.5 - 4.5\cos(2t) \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} t \\ t \end{bmatrix} \tag{4.4}$$

Problem 5: Linear ODE with time dependant coefficients and $b(t)$, $t \in [0, 10\pi]$, μ_k and $\hat{\mu}_k \in [-10, -10 + \frac{14}{\sqrt{2}}] \approx [-10, -0.1]$ for all k:

$$\begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} = \begin{bmatrix} -10 & 14\cos(t) \\ 14\sin(t) & -10 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} e^{3\sin(t)} \\ t \end{bmatrix} \tag{4.5}$$

Problem 6: Linear ODE with constant $A(t)$ and time dependent $b(t)$ having a spike at $t = 1$, $t \in [0, 2]$, $\mu_k = \hat{\mu}_k = -1$ for all k:

$$\begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} \frac{(t-2)^2+0.00001^2-1}{((t-1)^2+0.00001^2)^2} \\ 0 \end{bmatrix} \tag{4.6}$$

The true solution is $y_1(t) = \frac{10^{10}}{10^{10}t^2-2\times10^{10}t+10000000001} + \frac{1}{10000000001}e^{-t}$ and $y_2(t) = e^{-t}$.

Problem 7: Linear ODE with time dependant coefficients and $b(t)$, $t \in [0, 10\pi]$, $\mu_k \in [-1, \frac{-1001+\sqrt{1978101}}{2}] \approx [-1, 202.7]$ and $\hat{\mu}_k = -1$ for all k:

$$\begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} = \begin{bmatrix} -1 & \frac{-99\cos(t)}{\sin^2(t)+0.1} \\ 0 & -1000 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} e^{3\sin(t)} \\ t \end{bmatrix} \tag{4.7}$$

4.2 Numerical Results and Discussion

The number of significant digits of precision that MAPLE uses can be set using the command `digits`. In our implementation, we set `digits` to 15, with the exception of a couple of example problems discussed below where setting `digits` to a higher value (such as 30) gave better results compared to `digits` = 15. This issue as well as the shortcomings encountered when using MAPLE are discussed at the end of this chapter.

As mentioned in §3.6, given a user-specified global error bound τ, we choose the largest stepsize at each timestep, h_k such that

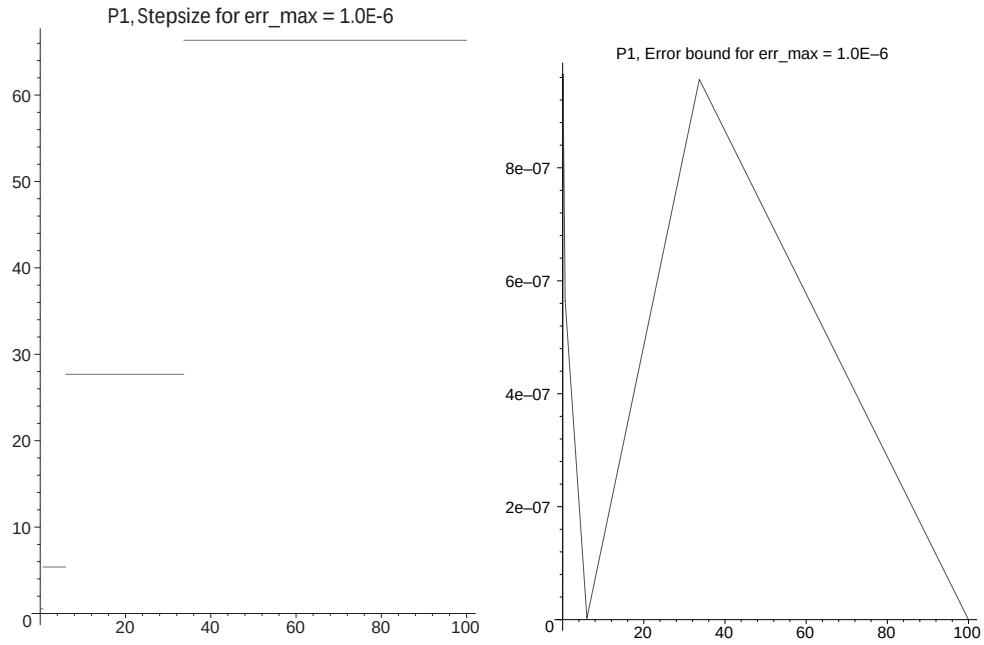
$$\phi_k(h_k) := \alpha_k e^{\hat{\mu}_k h_k} + \frac{\hat{\epsilon}_k}{\hat{\mu}_k}(e^{\hat{\mu}_k h_k} - 1) \leq \tau \tag{4.8}$$

and

$$\frac{h_{k-1}}{20^l} \leq h_k \leq 10h_{k-1}, \quad \text{for } l = 0, \ldots, 10,$$

are satisfied.

The following are the graphs obtained for problems 1, 2, 3, 4 and 6, using the above stepsize control strategy. The stepsizes taken for problems 1, 2, 3 and 6 are presented in the Table 4.1, 4.2, 4.3 and 4.4 respectively, since the number of integration steps taken for these problems are small. Problems 5 and 7 are discussed later.



(a) Stepsize vs Time. (b) Error bound vs Time.

Figure 4.1: Problem 1: $\mu = \hat{\mu}_k = -1$, digits = 15, integration steps taken = 6.

Table 4.1: Stepsize for each integration step for Problem 1.

Integration step k	initial timestep t_{k-1}	final timestep t_k	stepsize taken at k
1	0.0	0.0125	0.0125
2	0.0125	0.06625	0.05375
3	0.06625	0.60375	0.5375
4	0.60375	5.97875	5.375
5	5.97875	33.66	27.68125
6	33.66	100.0	66.34

Table 4.2: Stepsize for each integration step for Problem 2.

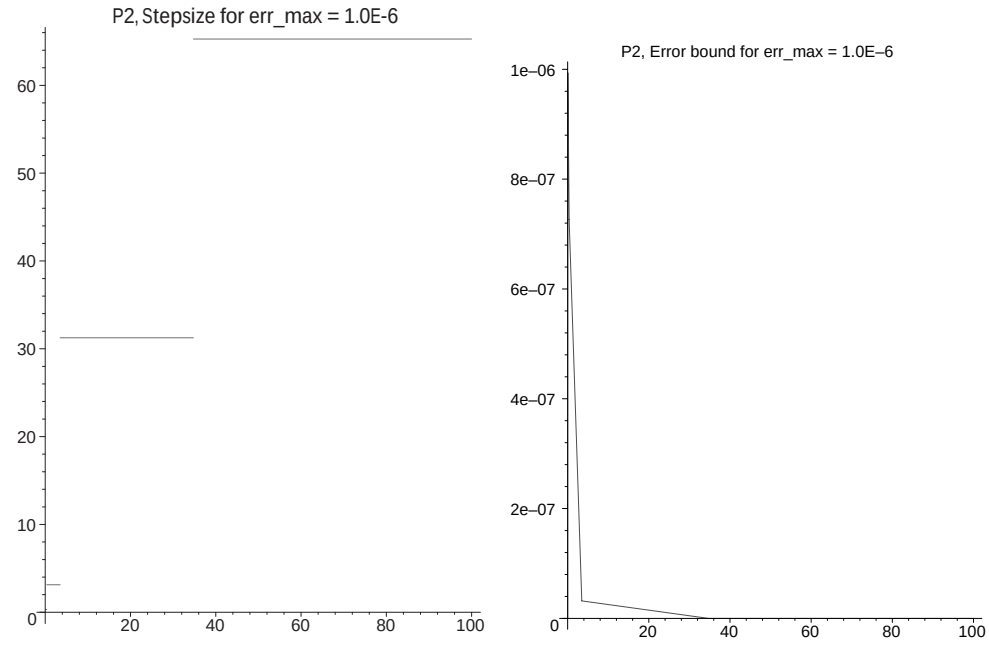
Integration step k	initial timestep t_{k-1}	final timestep t_k	stepsize taken at k
1	0.0	0.0125	0.0125
2	0.0125	0.04375	0.03125
3	0.04375	0.35625	0.3125
4	0.35625	3.48125	3.125
5	3.48125	34.73125	31.25
6	34.73125	100.0	65.26875

Table 4.3: Stepsize for each integration step for Problem 3.

Integration step k	initial timestep t_{k-1}	final timestep t_k	stepsize taken at k
1	0.0	0.0125	0.0125
2	0.0125	0.040625	0.028125
3	0.040625	0.321875	0.28125
4	0.321875	2.3046875	1.9828125
5	2.3046875	3.8909375	1.58625
6	3.8909375	16.818875	12.9279375
7	16.818875	76.2873875	59.4685125
8	76.2873875	100.0	23.7126125

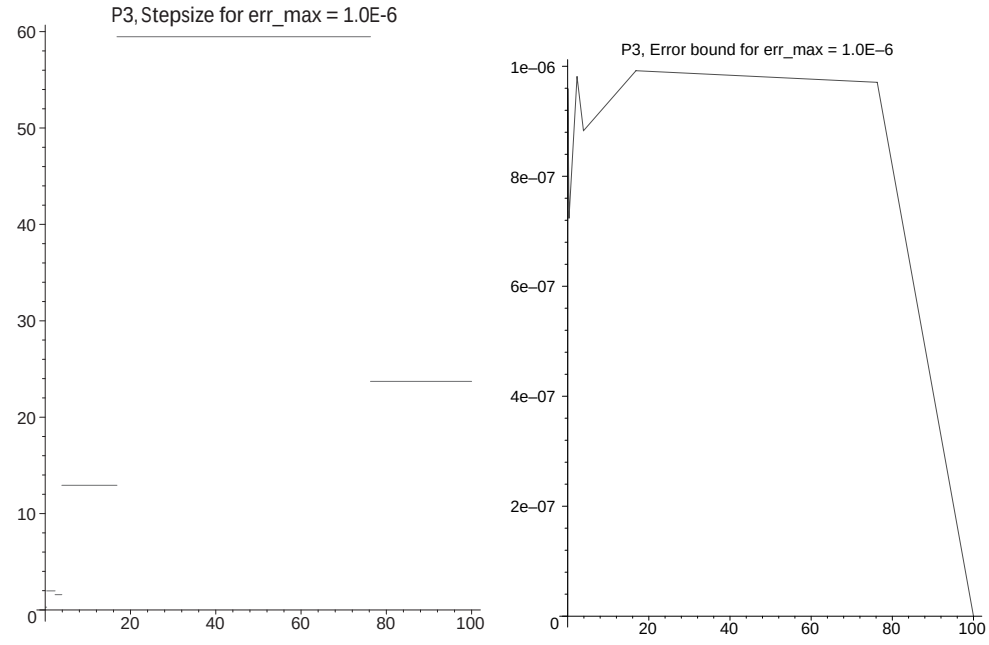
Table 4.4: Stepsize for each integration step for Problem 6.

Integration step k	initial timestep t_{k-1}	final timestep t_k	stepsize taken at k
1	0.0	0.1	0.1
2	0.1	1.1	1.0
3	1.1	2.0	0.9



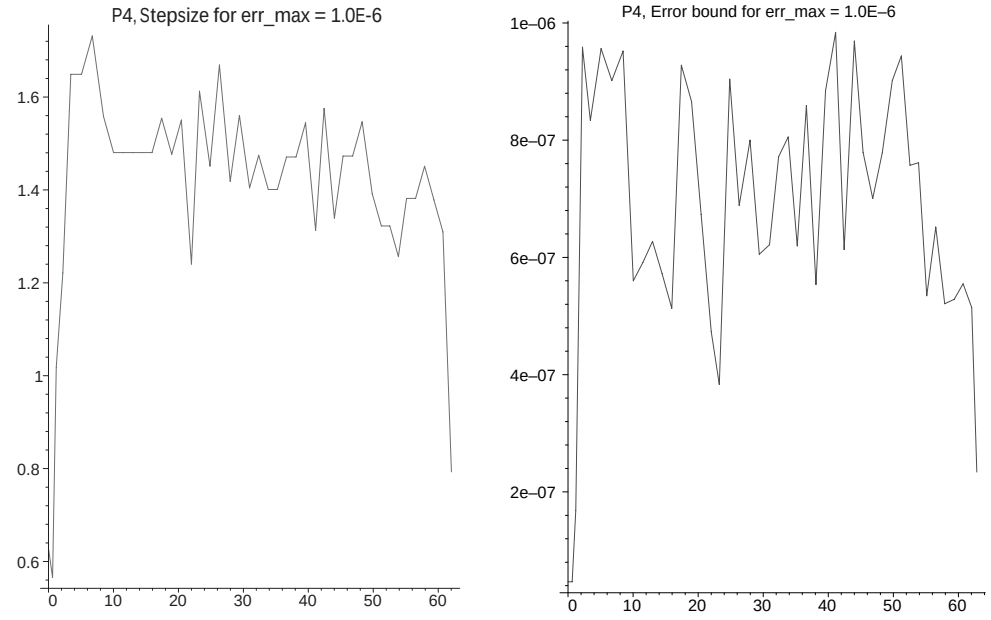
(a) Stepsize vs Time. (b) Error bound vs Time.

Figure 4.2: Problem 2: $\mu = \hat{\mu}_k = -1$, digits = 15, integration steps taken = 6.



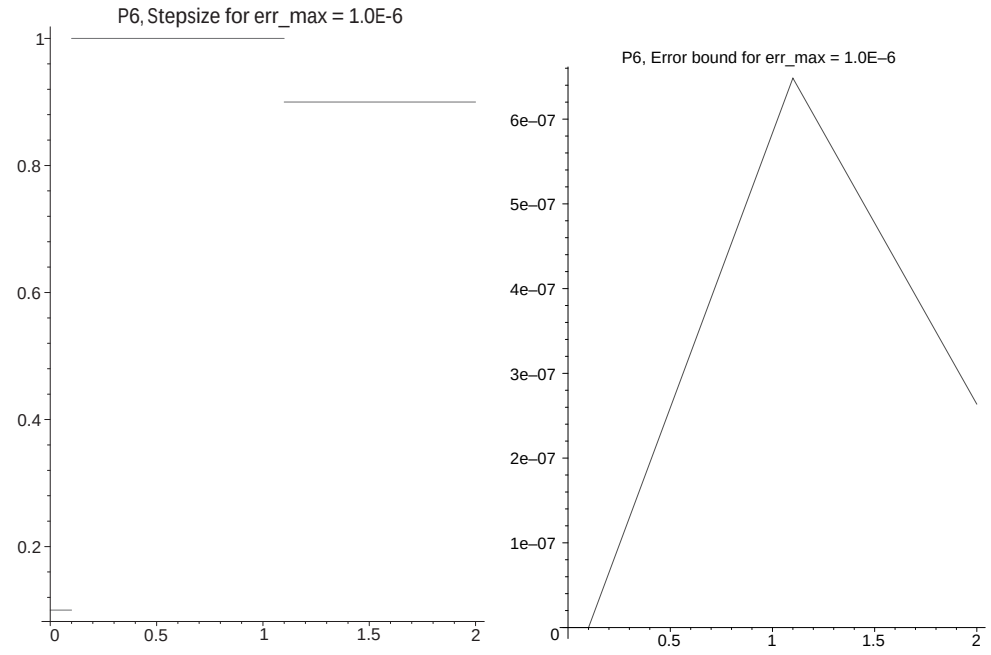
(a) Stepsize vs Time. (b) Error bound vs Time.

Figure 4.3: Problem 3: $\mu = \hat{\mu}_k = -1$, digits = 15, integration steps taken = 8.



(a) Stepsize vs Time. (b) Error bound vs Time.

Figure 4.4: Problem 4: $\mu = \hat{\mu}_k = -1$, digits = 15, integration steps taken = 45.



(a) Stepsize vs Time. (b) Error bound vs Time.

Figure 4.5: Problem 6: $\mu = \hat{\mu}_k = -1$, digits = 30, integration steps taken = 3.

Note that at the last integration step, the stepsize chosen can be smaller than if the function is to be evaluated over the time interval $t \in [0, \overline{T}]$, where $\overline{T} > T$. Thus, by taking a smaller stepsize at the last integration step results in a smaller error bound at the last integration step.

As explained in §3.5, we estimate α_k at each integration timestep $k > 1$ by

$$\alpha_k = ||S_k^{-1} S_{k-1}|| \phi_{k-1},$$

where S_k is the eigenvector matrix associated with A_k. Let

$$C = ||S_k^{-1} S_{k-1}||. \tag{4.9}$$

For problems 1, 2, 3 and 6, the matrix A_k is fixed and $\mu_k = -1$ for all k. As was mentioned in §3.6, this implies that C = 1, since $S_k = S$ for all k and so $S_k^{-1} S_{k-1}$ is the identity matrix. Hence the value of α_k for integration timestep $k > 1$ is the value of ϕ_{k-1} at the endpoint of integration timestep $(k-1)$. This is true for any fixed matrix A, since C is always equal to 1. Thus for problems with $\mu_k < 0$ and with fixed eigenvectors, the integration can always be continued (in theory) while keeping the global error $\leq \tau$.

For problems 4, 5 and 7, the matrix A_k is time dependent and this implies that the matrix S_k may change with k. It was noted that, although the value of ϕ_{k-1} at the endpoint of integration timestep $(k-1)$ may be below the prescribed error bound, the initial value of ϕ_k at integration timestep k may exceed the prescribed error bound. This occurs because the value of C may exceed 1 on some steps, due to S_k changing from step to step. As a result, the value of α_k at integration step k may exceed the final value of ϕ_{k-1} at integration step $(k-1)$. Thus, if the the eigenvectors are changing from step to step, there is a possiblity that the integration error will exceed the maximum tolerated error τ. In other words, depending on the value of C, the initial value of ϕ_k could exceed τ, even if the final value of ϕ_{k-1} in the previous integration timestep is below τ. This occured in problems 5 and 7.

For problem 5, the following output was observed:

• At integration time step k = 1,

$$\phi_{final} = 0.914080293343665 \times 10^{-6}.$$

• At integration time step k = 2,

$$C = 1.46096064526068,$$

$$\phi_{initial} = 1.28496606987386 \times 10^{-6},$$

which is greater than the prescribed error bound.

For problem 7, the following output was observed:

• At integration time step k = 3,

$$\phi_{final} = 0.953685844674828 \times 10^{-6}.$$

• At integration time step k = 4,

$$C = 1.05871506753380,$$

$$\phi_{initial} = 1.00398217157301 \times 10^{-6},$$

which is greater than the prescribed error bound.

The value of C for problem 5 ranges between 0.5 and 1.5, and, for problem 7, C ranges between 1 and 4.6. While the eigenvector matrix S_k changes from step to step in problem 4, S_k is an orthogonal matrix for all t. Thus the matrix $S_k^{-1}S_{k-1}$ is also orthogonal. Since the 2-norm of an orthogonal matrix is 1, the value of C is 1. So, for problem 4, the integration can always be continued (in theory) while keeping the global error $\leq \tau$.

4.2.1 Stepsize Control Strategy 1

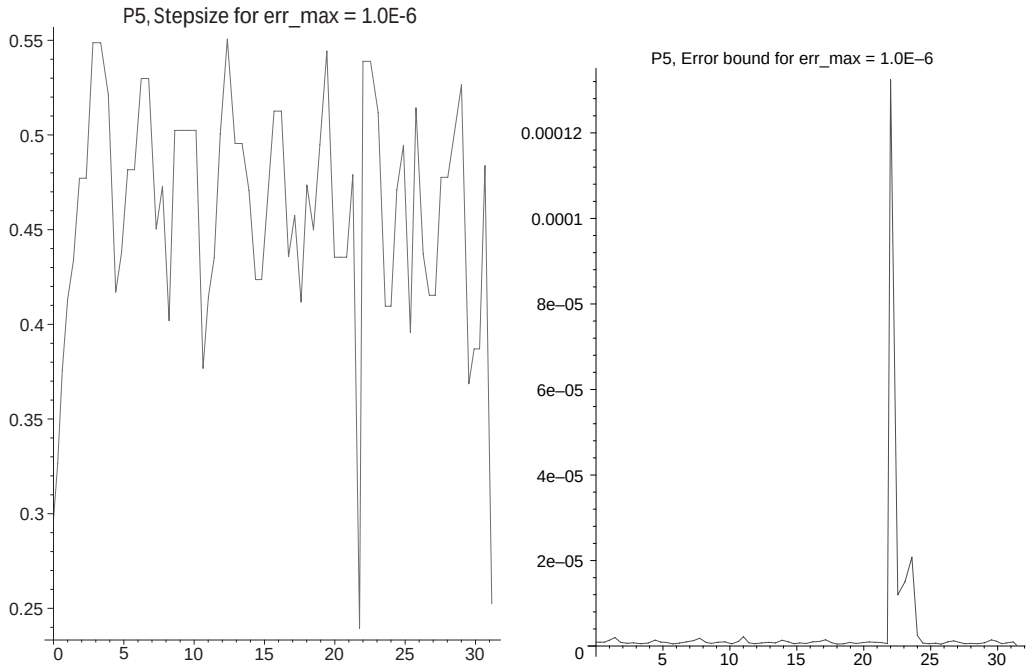
When the integration fails to continue using the stepsize control strategy described in §3.6, Neher [15] proposed an alternate condition for the stepsize control in order for the integration to continue. This alternate condition allows the global error to exceed the prescribed tolerance τ, but by an amount such that the global error would not grow too fast. More precisely, it allows the global error bound at the end of the integration step k to grow by at most x times the damped initial error at the end of integration step k, where $1 \leq x \leq 1\frac{1}{3}$.

That is, the integration can continue provided that

$$(\phi_k(h_k) \leq \tau) \quad \text{or} \quad (|\frac{\hat{\epsilon}_k}{\hat{\mu}_k}(e^{\hat{\mu}_k h_k} - 1)| < |\frac{\alpha_k e^{\hat{\mu}_k h_k}}{c}|), \tag{4.10}$$

where c is a positive integer. If c = 3, then the global error bound at the end of integration step k is at most $1\frac{1}{3}$ times the damped initial error bound at integration step k.

The following graphs show the stepsize changes and the error bounds for problems 5 and 7, respectively, using this stepsize control strategy, with c = 3 in (4.10). Note that $\hat{\mu}_k$ and $\mu_k \in [-10, -10 + \frac{14}{\sqrt{2}}] \approx [-10, -0.1]$ in problem 5, and $\hat{\mu}_k = -1$ while $\mu_k \in [-1, \frac{-1001+\sqrt{1978101}}{2}] \approx [-1, 202.7]$ in problem 7.



(a) Stepsize vs Time.

(b) Error bound vs Time.

Figure 4.6: Problem 5: c = 3, digits = 15, integration steps taken = 69.

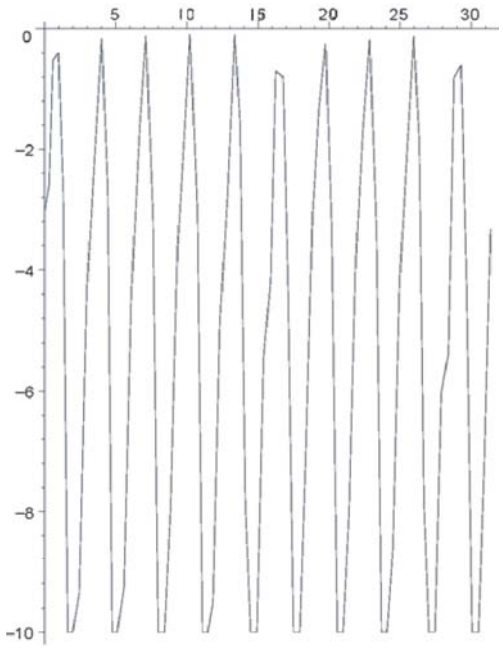
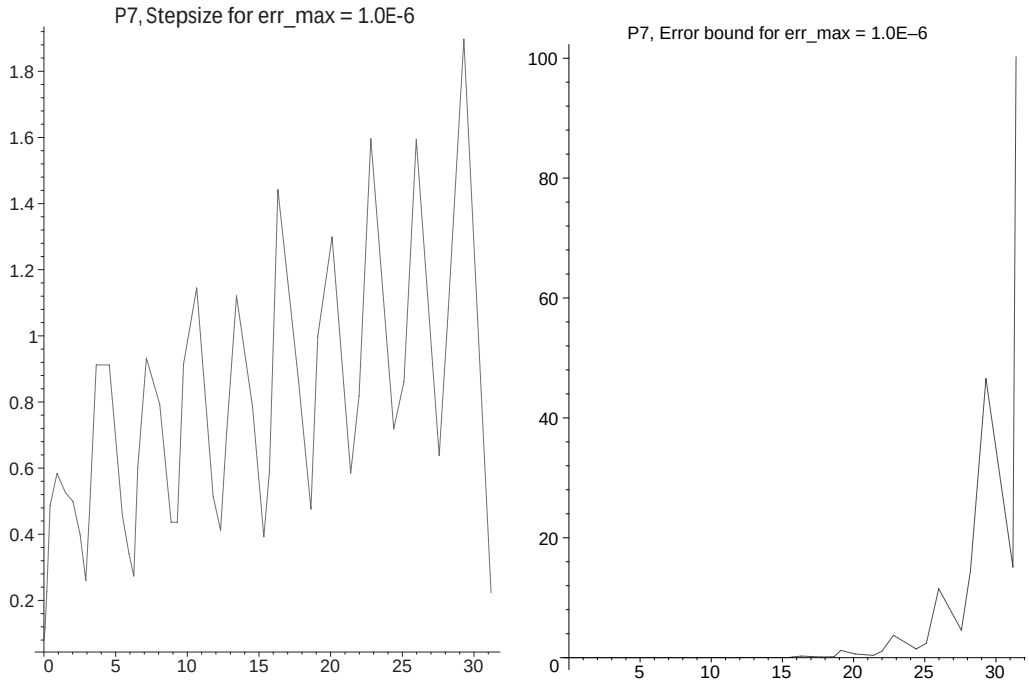


Figure 4.7: Problem 5: $\hat{\mu}$ vs Time, c = 3, digits = 15, integration steps taken = 69.

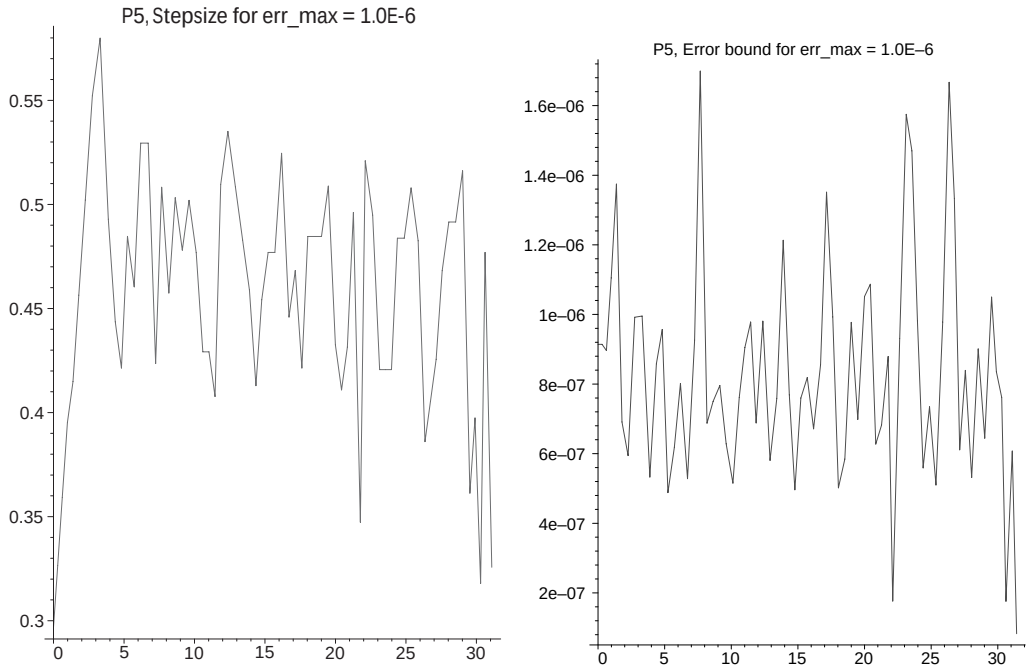


(a) Stepsize vs Time.

(b) Error bound vs Time.

Figure 4.8: Problem 7: c = 3, digits = 30, integration steps taken = 44.

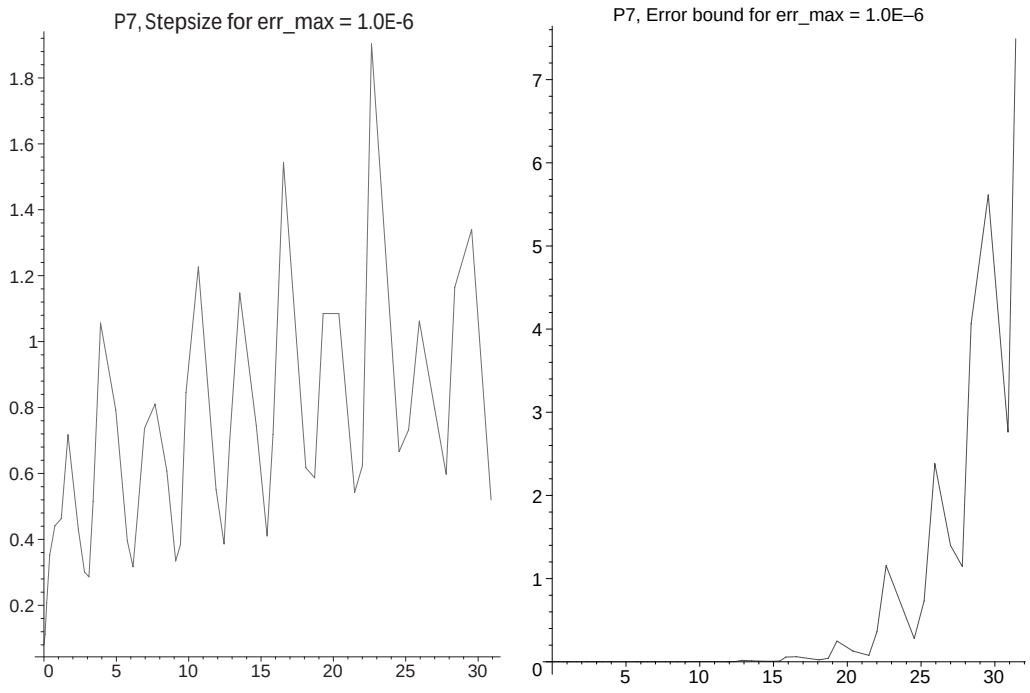
If c is set to 5 instead of 3 in (4.10), the results obtained for the error bounds of problems 5 and 7 are better.



(a) Stepsize vs Time.

(b) Error bound vs Time.

Figure 4.9: Problem 5: c = 5, digits = 15, integration steps taken = 69.



(a) Stepsize vs Time.

(b) Error bound vs Time.

Figure 4.10: Problem 7: c = 5, digits = 15, integration steps taken = 46.

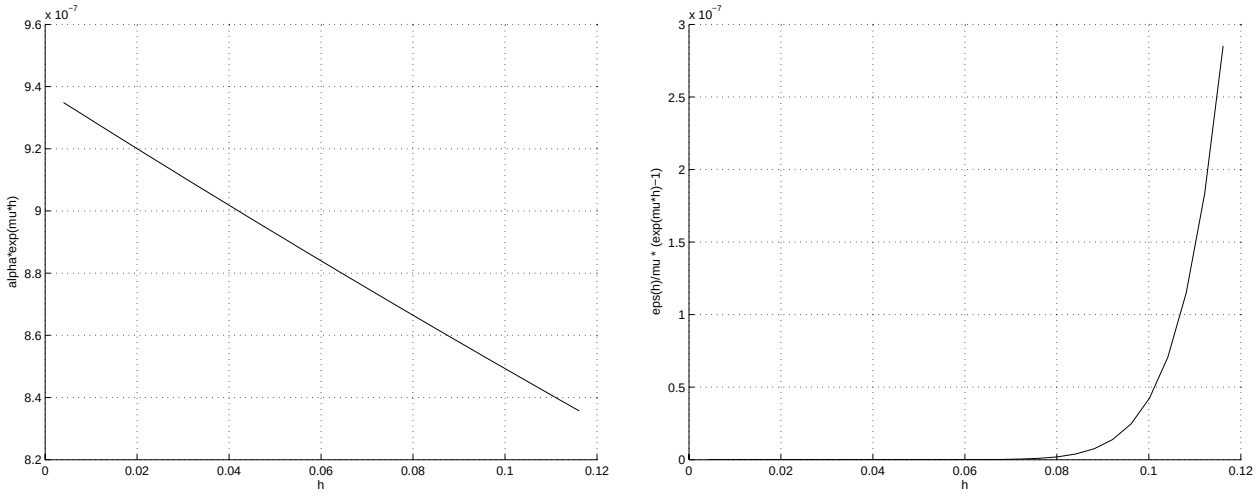
4.2.2 Step Size Control Strategy 2

Recall (4.8) and drop the subscript k. Let

$$A = \alpha e^{\hat{\mu}h},$$

$$B = \frac{\hat{\epsilon}(h)}{\hat{\mu}}(e^{\hat{\mu}h} - 1).$$

Let $\hat{\mu}_k < 0$ for all k. Using Neher's proposed stepsize control strategy 1 (§4.2.1) the following are the graphs of A, B and $\phi(h)$ with increasing h for problem 7, taken at integration timestep $k = 2$, where α = $0.938634288722006 \times 10^{-6}$ and $\hat{\mu}_k = -1$.



(a) A vs h.

(b) B vs h.

Figure 4.11: Problem 7 at integration step k = 2, c = 3, digits = 15.

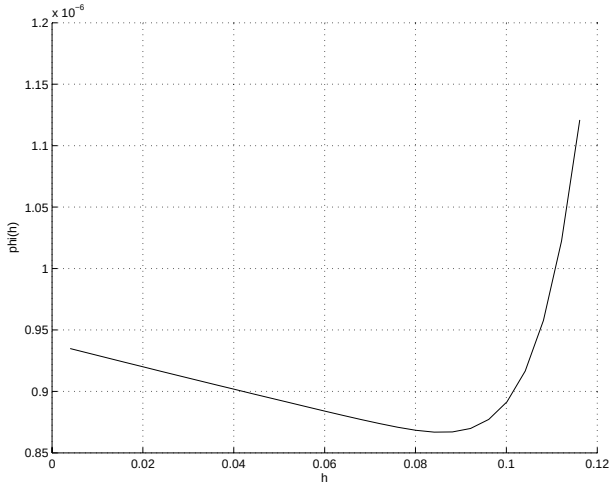


Figure 4.12: $\phi(h)$ for problem 7, c = 3, digits = 15.

Note, if h is increased further in Figure 4.11(b), B will clearly blow up. This indicates that the defect is large and suggests that replacing the Pade approximation to $y(t)$ by another choice of approximation method could lead to more robust results.

If we assume $\hat{\epsilon}(h) = ch^p$ with constant c and p for simplicity, then

$$\phi(h) = \alpha e^{\hat{\mu}h} + \frac{ch^p}{\hat{\mu}}(e^{\hat{\mu}h} - 1).$$

Thus,

$$\begin{aligned} \phi'(h) &= \hat{\mu}\alpha e^{\hat{\mu}h} + \frac{pch^{p-1}}{\hat{\mu}}(e^{\hat{\mu}h} - 1) + ch^p e^{\hat{\mu}h} \\ &= (\hat{\mu}\alpha + \frac{pch^{p-1}}{\hat{\mu}} + ch^p)e^{\hat{\mu}h} - \frac{pch^{p-1}}{\hat{\mu}}. \end{aligned}$$

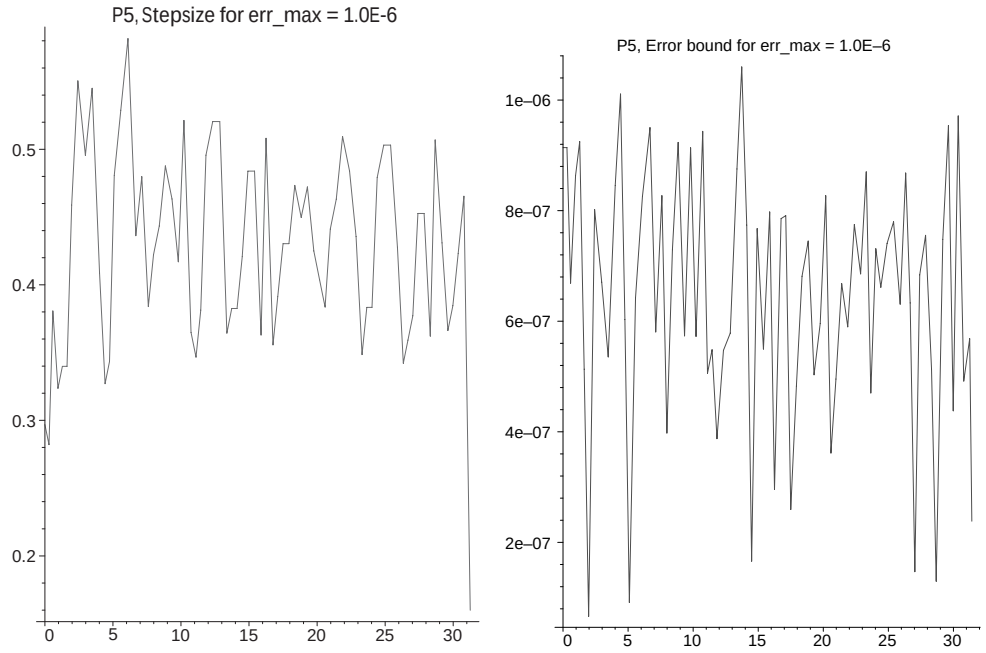
For $h = 0$, $\phi'(h) = \hat{\mu}\alpha < 0$, and as h increases, $\phi'(h)$ becomes > 0. Hence, the graph of $\phi(h)$ is in general similar to that shown in Figure 4.12 for problem 7.

Noting the behaviour of $\phi(h)$ as h increases (at integration step k), we propose an alternative stepsize control strategy 2 as follows.

• If $\phi_{initial} < \tau$ (at integration step k), allow h to increase as long as condition (4.8) is satisfied and take the maximum step size h such that (4.8) holds.

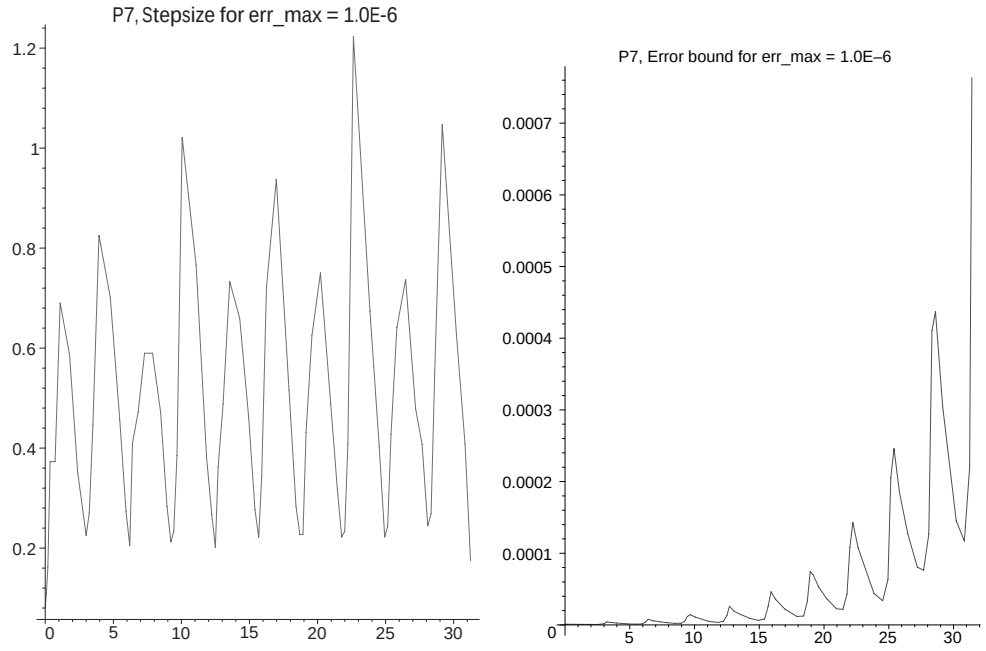
• If $\phi_{initial} > \tau$ (at integration step k), allow h to increase as long as (4.10) is satisfied and take the step size h that minimizes $\phi(h)$.

Using this stepsize control strategy, the results obtained for problems 5 and 7 are better than those presented earlier for Neher's stepsize control strategy given in §4.2.1. The following are the graphs obtained for problems 5 and 7, using the above stepsize control strategy. Note that the graphs obtained for $c = 3$ and $c = 5$ for problems 5 and 7 are the same in this case.



(a) Stepsize vs Time. (b) Error bound vs Time.

Figure 4.13: Problem 5: c = 3 or 5, digits = 15, integration steps taken = 74.



(a) Stepsize vs Time. (b) Error bound vs Time.

Figure 4.14: Problem 7: c = 3 or 5, digits = 15, integration steps taken = 71.

4.3 Problems Encountered

The problems encountered in testing our implementation of Neumaier's enclosure method using MAPLE version 8 are discussed below.

4.3.1 Memory Allocation

It is well-known that MAPLE sometimes requires a significant amount of memory and this can sometimes cause the computation to abort. In order to re-execute a problem in MAPLE, there is a `restart` command which clears all values and memory for re-use. However, this command does not always free all the values and memory for re-use. This was the case for problem 4.

In this problem, it was found that, if the "restart" command is used before re-solving the problem, the results obtained for the first run are different than the results obtained for the second run. However, if the problem is run once, followed by exiting MAPLE and re-opening MAPLE, then re-executing the problem for a second run, the output is identical to the output from the first run. This behaviour indicates that MAPLE's memory allocation is inconsistent, producing varying results.

4.3.2 Output Problems

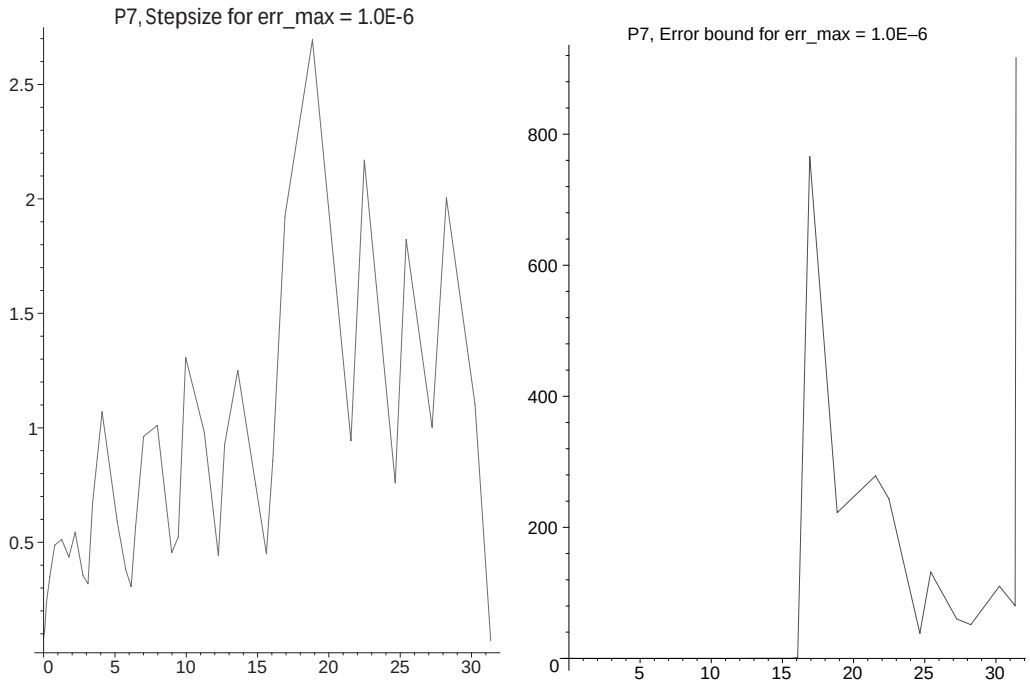
When solving ODEs with MAPLE, the output of its built-in function `dsolve` is an unsorted list. Similary, when calculating the eigenvalues and their corresponding eigenvectors of a matrix in MAPLE, the output of its built-in function `Eigenvectors` would occasionally be in a different order. However, our program requires that these lists be kept in a consistent order. Thus, in both cases, steps were taken to correct the order of the output.

When calculating the eigenvalues and their corresponding eigenvectors of a matrix in MAPLE, the output is always complex even if the imaginary parts of the output are zero. Steps were taken to distinguish between real and non-real eigenvalues and eigenvectors.

4.3.3 Significant Digits of Precision

As was mentioned earlier, the number of significant decimal digits of precision that MAPLE uses can be set using the command `digits`. We used `digits` = 15 in most cases and for these problems we found that increasing the number of digits has an insignificant effect on the output. However, it was found that for problem 7, increasing the number of digits gave varying results.

In problem 7, the results shown in Figure 4.8 used `digits` = 30, with Neher's stepsize control strategy, based on (4.10) with $c = 3$. Compare this with the following results, also using Neher's stepsize control strategy with $c = 3$, but using `digits` = 15.



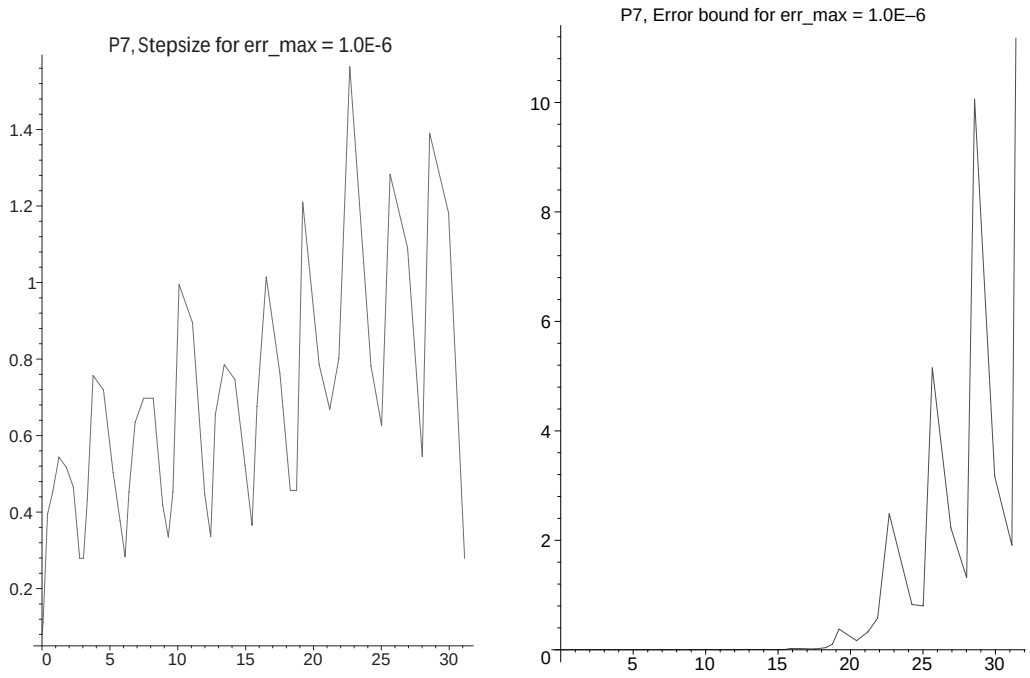
(a) Step size vs Time.

(b) Error bound vs Time.

Figure 4.15: Problem 7: c = 3, digits = 15, integration steps taken = 38.

Notice in Figure 4.15, the error bounds obtained with `digits` = 15 is at least 8 times larger than the error bounds obtained with `digits` = 30, shown in Figure 4.8. Also the stepsize changes with `digits` = 15 are slightly bigger than the stepsize changes with `digits` = 30.

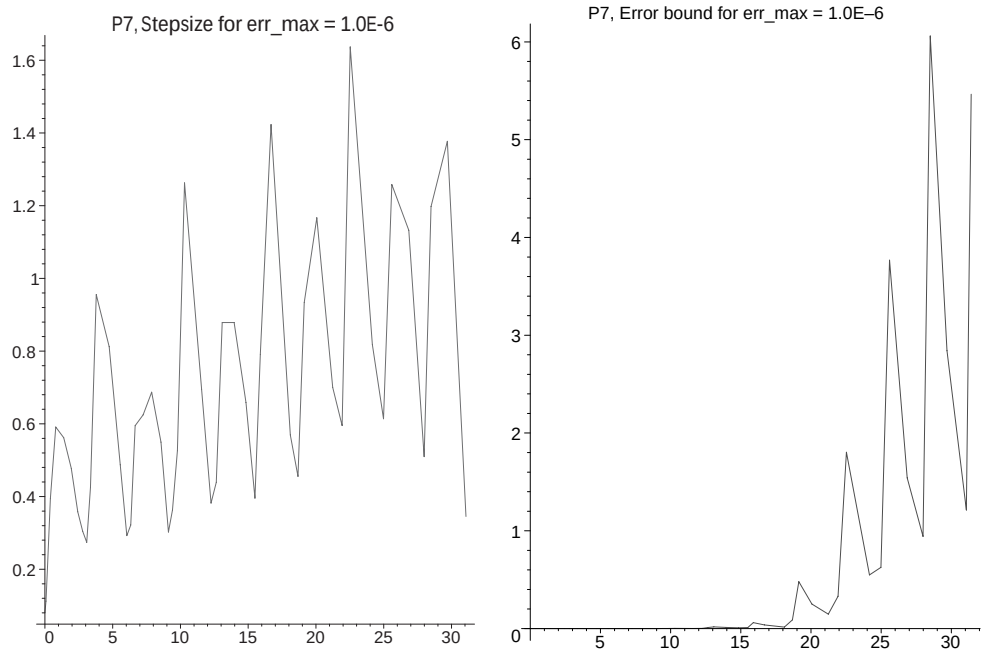
The results shown in Figure 4.10 used `digits` = 15, with Neher's stepsize control strategy, that is condition (4.10) with $c = 5$. Compare this with the following results, also using Neher's stepsize control strategy with $c = 5$, but using `digits` = 30, 40 and 50 respectively.



(a) Stepsize vs Time.

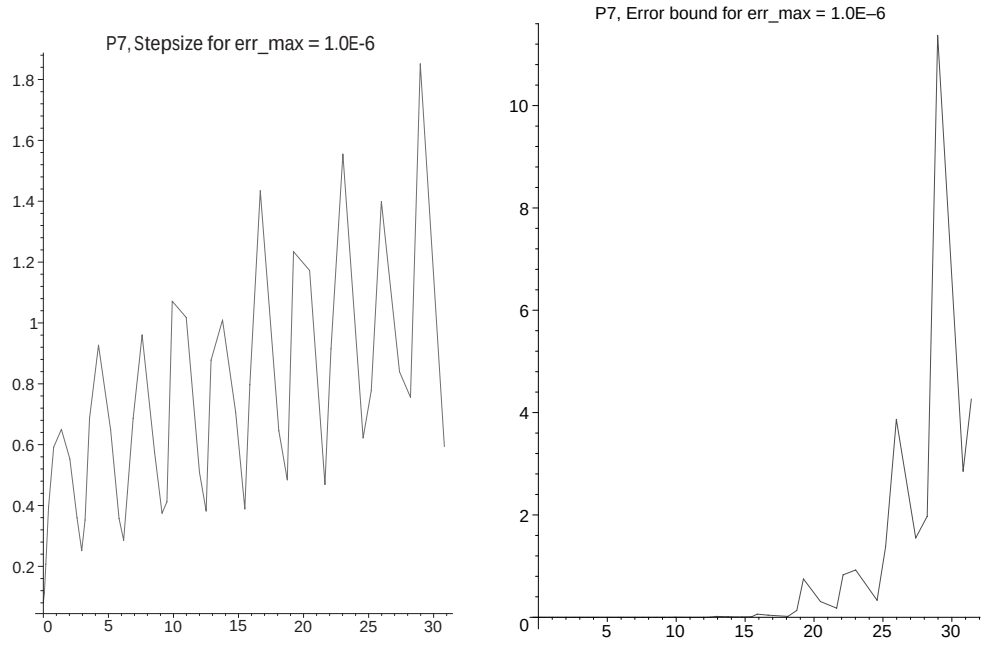
(b) Error bound vs Time.

Figure 4.16: Problem 7: c = 5, digits = 30, integration steps taken = 50.



(a) Stepsize vs Time. (b) Error bound vs Time.

Figure 4.17: Problem 7: c = 5, digits = 40, integration steps taken = 48.



(a) Stepsize vs Time. (b) Error bound vs Time.

Figure 4.18: Problem 7: c = 5, digits = 50, integration steps taken = 45.

Notice that, as we increase the number of digits, the results obtained for the stepsize changes and the error bounds vary. In going from `digits` = 15 to 30, the stepsize changes became smaller but the error bounds became worse. In going from `digits` = 30 to 40, the stepsize changes became bigger and the error bounds became better. In going from `digits` = 40 to 50, the stepsize changes became bigger and the error bounds became worse.

Recall that μ_k oscillates between -1 and $\frac{-1001+\sqrt{1978101}}{2} \approx 202.7$ in problem 7. This supports our belief that this problem is ill-conditioned. This is consistent with the numerical results discussed above.

Chapter 5

Conclusions and Future Work

Our goal in this thesis was to investigate the potential of Neumaier's enclosure method for stiff problems and to provide insight into how this method behaves in practice. In Chapter 2, we reviewed the logarithmic norms, the enclosure methods of Dahlquist and Neumaier, and showed that Neumaier's result is a special case of Dahlquist's result. In Chapter 3, we discussed our implementation of Neumaier's enclosure method based on the combination of these two results.

We believe that this method can be the basis for a validated method for stiff IVPs for ODEs, in cases when there is no wrapping effect. However, further investigations are required to find the best method of computing the various parameters in Neumaier's enclosure method. In particular, how we should calculate the invertible matrix S and how often we should evaluate it in order to obtain a near optimal μ while keeping the wrapping effect under control. Further work is required in investigating other approximation methods and the degrees of freedom in approximating $y(t)$, which might lead to more robust results. Improvements also need to be made in our simple stepsize control strategy. Ideally, we would like to achieve smooth gradual stepsize changes as well as keep the global error below the maximum tolerated error at all times. Given that the purpose of our implementation of Neumaier's enclosure method was to explore the potential of the method and was non-rigorous, we would need to estimate μ for all

y in a tube containing both the true solution y and the approximate solution p, and to estimate the defect, ϵ, for all $t \in [0, \overline{t}]$ in order to achieve rigorous bounds for this method. We would also need to take $t \in [0, \overline{t}]$ and $y \in \{\ p(t) - s(p(t) - y(t)) : s \in [0, 1]\ \}$ into account when computing the eigenvectors of the Jacobian matrix to achieve rigorous bounds for this method. We leave these tasks for future work.

Bibliography

[1] G. Dahlquist. Stability and Error Bounds in the Numerical Integration of Ordinary Differential Equations. *Trans. Royal Inst. Tech., Stockhom, Sweden*, No. 130, 1959.

[2] P. Eijgenraam. The Solution of Initial Value Problems using Interval Arithmetic. *Math. Center Tracts 144, Amsterdam*, 1981.

[3] A. R. Forsyth. *Theory of Differential Equations, Vol IV*. New York: Dover, 1959.

[4] K. R. Jackson G. F. Corliss and N. S. Nedialkov. Validated Solutions of Initial Value Problems for Ordinary Differential Equations. *Appl. Math. Comp.*, No. 105:21–68, 1999.

[5] K. Geddes. Convergence Behaviour of the Newton Iteration for First Order Differential Equations. *Proceedings of EUROSAM*, pages 78–79, 1979.

[6] A. Griewank and G. F. Corliss. Computational Differentiation: Theory, Implementation and Application. *SIAM*, Philadelphia, Penn., 1991.

[7] E. L. Ince. *Ordinary Differential Equations*. New York: Dover, 1944.

[8] R. Lohner. Enclosing the Solution of Ordinary Initial- and Boundary-Value Problems. *Computer Arithmetic, Teubner, Stuttgart*, pages 255–289, 1987.

[9] S. M. Loziinskij. Error Estimate for Numerical Integration of Ordinary Differential Equations. *Part I. Izv. Vyss. Ucehn. Zaved. Mathematika*, No. 6:52–90, 1958.

[10] C. Bischof M. Berz and G. F. Corliss. Computational Differentiation: Techniques, Applications, and Tools. *SIAM*, Philadelphia, Penn., 1996.

[11] R. E. Moore. Interval Arithmetic and Automatic Error Analysis in Digital Computing. *Ph. D. Thesis, Appl. Math. Statist. Lab. Rep. 25, Standford University*, 1962.

[12] R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1966.

[13] N. S. Nedialkov. Computing Rigorous Bounds on the Solutions of an Initial Value Problem for an Ordinary Differential Equation. *Ph. D. Thesis. Department of Computer Science, University of Toronto*, 1999.

[14] M. Neher. Private communications.

[15] M. Neher. On Neumaier's Enclosure Method for the Solution of Dissipative ODEs. *Third Int. Workshop on Taylor Methods, Miami Beach.*, Dec, 2004.

[16] A. Neumaier. Global, Rigorous and Realistic Bounds for the Solution of Dissipative Differential Equations. Part I: Theory. *Computing*, Volume 52, No. 4:315–336, Feb. 1994.

[17] L. F. Shampine and C. W. Gear. A User's View of Solving Stiff Ordinary Differential Equations. *SIAM Review*, Volume 21, No. 1:1–17, Feb. 1979.

[18] T. Strom.