Stiffness in Numerical Initial-Value Problems: A and L-Stability of Numerical Methods

Riccardo Fazio

Department of Mathematics, University of Messina
Salita Sperone 31, 98166 Messina, Italy
email: rfazio@dipmat.unime.it

May 15, 2000 and in revised form November 28, 2000

Abstract. The main aim of this note is to help the students to get an insight into the asymptotic stability concept by means of visual representation of the stability regions of different numerical methods. This facilitates understanding of the meaning of stability for constant step sizes and the related concept of stiffness in numerical initial-value problems. Moreover, the distinction between A and L-stability of numerical methods can be more easily understood.

1 Introduction.

The first quoted paper on the numerical solution of stiff problems is due to Curtiss and Hirschfelder [5]. In 1952 they used the simple initial value problem

$$\frac{dy}{dx} = -50(y - \cos\ x) \ , \quad y(0) = 0 \ , \quad x \in [0, x_{max}] \ ,$$

to explain stiffness and introduced the Backward Difference Formulae or BDF methods to cope with stiff problems. By that time stiff problems had arisen in most of the applied sciences. The book by Hairer and Wanner [9, pp. 2-12], provides examples of stiff problems from chemical reaction, electrical circuits, mechanics, and methods of lines applied to diffusion problems. However, what does stiffness mean within numerical initial value problems? A

mathematical definition of the stiffness concept cannot be stated in simple terms. Lambert in his recent text book made four different "statements" on and gave a heuristic "definition" of the stiffness concept [12, pp. 216-224]. These statements and definition lead to the following assertions pertinent to stiffness:
1) "Explicit methods don't work" quoted from Hairer and Wanner [9, p. 2];
2) Stability is more a constraint on the step-size than accuracy;
3) The problem and the numerical method used contains widely different evolving scales.
For a precise mathematical characterization of the stiffness concept the interested reader is referred to the recent paper by Brugnano and Trigiante [2]. The aforementioned characterization is given in a more general context, namely for general ordinary differential equations, and therefore for simplicity reasons will be omitted here.

Starting with the classical book by Henrici [10], the fundamental questions of consistency, order of local accuracy, zero-stability and convergence of numerical methods for initial value problems have been treated in many books on the subject. However, the convergence theorem is meaningless within the stiffness context because it is concerned with the numerical solution behaviour as the step-size goes to zero whereas the stiffness phenomenon is related to fixed step-sizes. The stability of numerical methods for constant step sizes is treated in several books [1, 3, 4, 7, 9, 11, 12, 18]. However, generally no mention is made on how to plot the stability regions of numerical methods. The author's point of view is that we need not only to teach the theory, but also we have to provide a laboratory experience to the students to understand the theory. Hence, to this end the author has used the MATLAB plotting capabilities. All the written MATLAB files are available on internet from the anonymous ftp area at the URL: `ftp://ftp.mathworks.com/pub/contrib/v5/teaching/ALstab` and its mirror sites.

MATLAB has been used effectively to teach particular topics [15, 14] as well as entire courses within numerical analysis [16, 17], scientific computing [13] and computational physics [6].

2 Numerical methods.

For simplicity we consider the scalar problem

$$\frac{dy}{dx} = f(x,y) \ , \quad y(x_0) = y_0 \ , \quad x \in [x_0, x_{max}] \ .$$

Let us introduce the constant mesh-size $h = (x_{max} - x_0)/N$ and grid-points $x_n = x_0 + nh$ for $n = 0, 1, 2, \ldots, N$, and denote by y_n the numerical approximation of the exact value $y(x_n)$. Moreover, we use the simple notation $f_s = f(x_s, y_s)$. The following numerical methods will be considered: the 1-st oder forward Euler

$$y_{n+1} = y_n + hf_n \quad \text{for} \ \ n = 0, 1, \ldots, N-1 \ ; \tag{2.1}$$

the 1st order backward Euler

$$y_{n+1} = y_n + hf_{n+1} \quad \text{for} \ \ n = 0, 1, \ldots, N-1 \ ; \tag{2.2}$$

the 2-nd order Taylor

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2}\left[\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y}\right]_n \quad \text{for} \ \ n = 0, 1, \ldots, N-1 \ , \tag{2.3}$$

and within the class of consistent and zero-stable k-steps Linear Multistep Methods (LMMs)

$$\sum_{j=0}^{k} \alpha_j y_{n+1-j} = h \sum_{j=0}^{k} \beta_j f_{n+1-j} \quad \text{for} \ \ n = k-1, k, \ldots, N-1 \ , \tag{2.4}$$

where α_j, β_j for $j = 0, 1, \ldots, k$ are constants with $\alpha_0 = 1$ and $|\alpha_k| + |\beta_k| \neq 0$, the 2-nd order: Adams-Bashforth (AB)

$$y_{n+1} = y_n + \frac{h}{2}\left[3f_n - f_{n-1}\right] \quad \text{for} \ \ n = 1, 2, \ldots, N-1 \ , \tag{2.5}$$

Adams-Moulton (AM)

$$y_{n+1} = y_n + \frac{h}{2}\left[f_n + f_{n+1}\right] \quad \text{for} \ \ n = 0, 1, \ldots, N-1 \ , \tag{2.6}$$

and Backward Difference Formula (BDF)

$$y_{n+1} = \frac{4}{3}y_n - \frac{1}{3}y_{n-1} + \frac{2}{3}hf_{n+1} \quad \text{for} \ \ n = 1, 2, \ldots, N-1 \ . \tag{2.7}$$

Note that the Taylor and the AM are one-step methods whereas the AB and BDF are two-steps methods: to apply the two-steps methods we use as an approximation of $y(x_1)$ the value obtained by a one-step method.

3 Stability for constant step sizes.

The exact mathematical concept is the asymptotic stability (A-stability) of numerical methods defined within Dalquist's theory. In particular, applying a method to Dalquist's test problem

$$\frac{dy}{dx} = \lambda y \ , \quad y(0) = 1 \ , \quad x \in [0, \infty) \ , \tag{3.1}$$

where $\lambda < 0$, we require that the numerical solution verifies the asymptotic condition $\lim_{n\to\infty} y_n = 0$. Note that the above request is motivated by the asymptotic behaviour of the exact solution: (being $\lambda < 0$) $\lim\limits_{x\to\infty} y(x) = 0$.

It is a simple matter to verify that the numerical solution $y_k = (1 + h\lambda)^k$ of the forward Euler method $y_0 = 1, y_{k+1} = y_k + h\lambda y_k$ is asymptotically stable if and only if $|1 + h\lambda| < 1$, so that $h < -2/\lambda$. Hence the forward Euler method is conditionally stable. On the other hand the numerical solution $y_k = (1 - h\lambda)^{-k}$ of the backward Euler method $y_0 = 1, y_{k+1} = y_k + h\lambda y_{k+1}$ is always asymptotically stable (for all $h > 0$). The backward Euler method is unconditionally stable.

Exercise. *Use the two MATLAB script files* `fEtest.m` *(forward Euler method) and* `bEtest.m` *(backward Euler method) to verify the behaviour of the numerical solution.*

3.1 A stiff problem.

Let us consider an interesting example. For several values of the step size, we apply the four 2-nd order methods introduced in Section 2 to the stiff problem

$$\frac{dy}{dx} = -100 \left(y - e^{-x} \right) - e^{-x} \ , \quad y(0) = 1 \ , \quad x \in [0, 1] \ . \tag{3.2}$$

In Table 1 we list the numerical approximation for $y(1)$ obtained by the four 2-nd order methods for several values of the step size. Figure 1 shows, within a semi-log_{10} scale, the results of Table 1.

To grasp the meaning of the obtained numerical results we have to study the stability properties of the four 2-nd order methods.

Table 1: Computed values of y_N.

h	Taylor	AB	AM	BDF
0.1	5.44E + 10	−3.46E + 06	0.3658	0.3679
0.0667	5.98E + 12	1.41E + 09	0.3679	0.3679
0.05	1.06E + 13	−1.06E + 11	0.3679	0.3679
0.04	7.79E + 11	2.03E + 12	0.3679	0.3679
0.0333	4.77E + 09	−1.23E + 13	0.3679	0.3679
...				
0.0133	0.3679	−5.36E + 03	0.3679	0.3679
0.0125	0.3679	1.83E + 02	0.3679	0.3679
0.0118	0.3679	−1.1296	0.3679	0.3679
0.0111	0.3679	0.3743	0.3679	0.3679
0.0105	0.3679	0.3679	0.3679	0.3679
0.0100	0.3679	0.3679	0.3679	0.3679

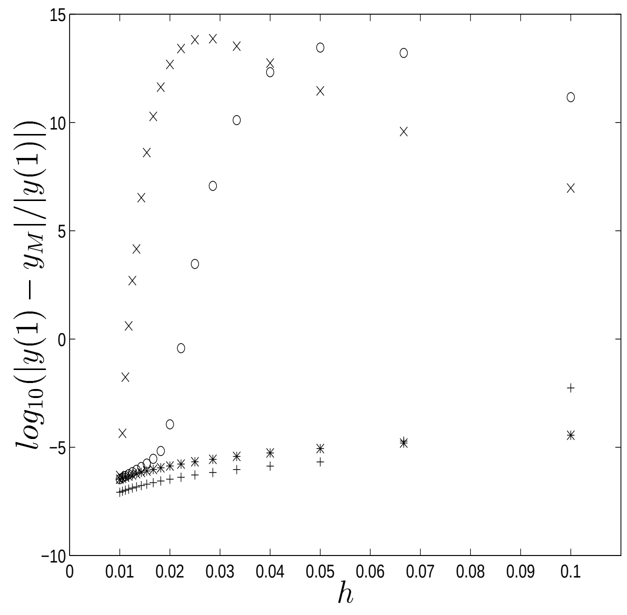

Figure 1: Relative error in a semi-log_{10} scale for Taylor o, AB $\times$, AM + and BDF $*$ 2-nd order methods. The MATLAB script is `meth2nd.m`. Type `help meth2nd` at the MATLAB prompt line to get information on the script file.

3.2 A-stability of the 2-nd order Taylor method.

A Taylor method applied to the differential equation of the Dalquist's test problem results in the formula

$$y_{n+1} = \mathrm{R}(\mu)\ y_n\ , \tag{3.3}$$

where $\mathrm{R}(\mu)$ is the Taylor expansion of e^{μ} to the same order as the Taylor method and $\mu = \lambda h$. Since $y_0 = 1$, by induction we get $y_n = [\mathrm{R}(\mu)]^n$ and consequently for the A-stability of the method we must require that $|\mathrm{R}(\mu)| < 1$. Note that for the 2-nd order Taylor method $\mathrm{R}(\mu) = 1+\mu+\mu^2/2$. If μ is real the above inequality gives us the limitation for the step-size $h < 2/|\lambda|$. By letting the value of λ be complex we can consider the region of the complex plane of the variable μ, called the A-stability region R_A of the method, where $|\mathrm{R}(\mu)| < 1$.

To get the A-stability region of a Taylor method we can proceed in two different ways. The simplest one is to follow Lambert [12, p.202] and use the "scanning technique" to find out the interior of the A-stability region. To that end, let $x + iy = \mu$ (not $\mathrm{R}(\mu)$, there is a misprint in Lambert's book), we scan for $-Y \leq y \leq Y$ $(Y > 0)$ the line $x = X$ (usually $X < 0$) and mark a point (x, y) if and only if $|\mathrm{R}(\mu)| < 1$; then increment the X value and repeat the scanning procedure. The second way, described by Gear [8, p. 41], is known as the "boundary locus technique". In the boundary locus technique we concentrate on the boundary ∂R_A of the A-stability region. For explicit one-step methods we can set $\mathrm{R}(\mu) = e^{i\theta}$ and determine $\mu(\theta)$ (where $\mu(\theta) = x(\theta) + iy(\theta)$) by any polynomial root-finder (see again Gear [8, p. 41]). For the 2-nd order Taylor method we get the nonlinear system

$$\tfrac{1}{2}\,(x^2 - y^2) + x + 1 - cos\theta = 0\ , \quad xy + y - sin\theta = 0\ .$$

Butcher [4, p. 239] indicates the need to take $\theta \in [0, 2\pi\ s]$ for an explicit Runge-Kutta method of stages and order $s \leq 4$, hence of the Taylor's method of corresponding order, to get its complete stability region.

The Figure 2 shows the result of the described techniques applied to the 2-nd order Taylor method.

Exercise. *Use the MATLAB scripts named* `a1_tayl2.m` *and* `a2_tayl2.m` *to get separate figures.*

The A-stability of the 2-nd order AB, AM, and BDF methods is discussed below within the general class of LMMs.

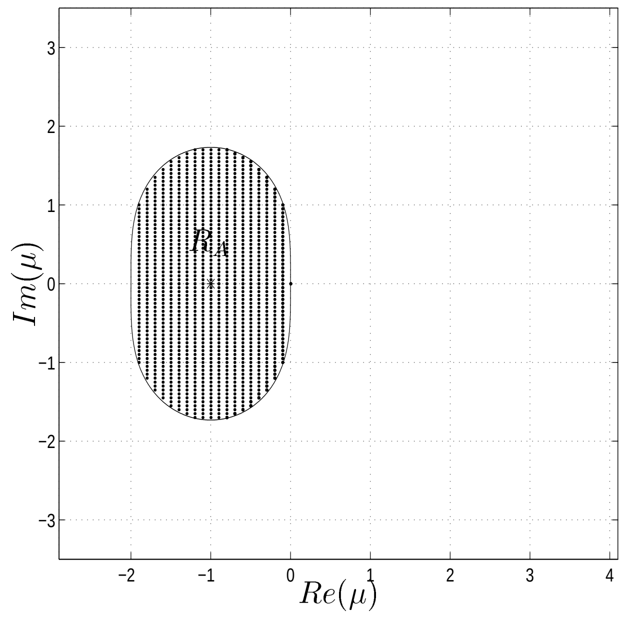

Figure 2: A-stability region for the 2-nd order Taylor method: scanning technique points marked by a dot and boundary locus technique by a solid line. MATLAB script file `a_Tayl2.m`.

3.3 A-stability of LMMs.

An alternative representation of the LMMs (2.4) is given by

$$\rho(\mathrm{E})y_{n+1-k} = h\sigma(\mathrm{E})f_{n+1-k} \ ,$$

where

$$\rho(\mathrm{E}) = \sum_{j=0}^{k} \alpha_j \mathrm{E}^{k-j} \ , \qquad \sigma(\mathrm{E}) = \sum_{j=0}^{k} \beta_j \mathrm{E}^{k-j} \ ,$$

are, respectively, the first and second characteristic polynomials and E is the shift operator defined by $\mathrm{E}y_n = y_{n+1}$ ($\mathrm{E}^j = \mathrm{E}^{j-1}\mathrm{E}$).

By applying a k-steps LMM to Dalquist's test problem we get the finite difference equation

$$[\rho(\mathrm{E}) - \mu\sigma(\mathrm{E})]y_{n+1-k} = 0 \ ,$$

where $\mu = h\lambda$. For the solutions of this difference equation we apply the method of Lagrange: by setting $y_s = r^s$ and dividing by the common factor r^{n+1} we get

$$\rho(r) - \mu\sigma(r) = \sum_{j=0}^{k}(\alpha_j - \mu\beta_j)r^{k-j} = 0 \ .$$

The k roots of this polynomial will be denoted by $r_s(\mu)$ for $s = 1, \ldots, k$. Let us denote by R_A the following region of the complex plane

$$R_A = \left\{ \ \mu \ : \ |r_s(\mu)| \leq 1 \ , \quad \text{for} \quad s = 1, \ldots, k \quad \text{if} \quad |r_j(\mu)| = 1 \ , \quad \text{then} \quad \rho\prime(r_j) \neq 0 \ \right\}$$

and by ∂R_A its boundary. A method is said to be A-stable if

$$\{\mu : Re(\mu) < 0\} \subset R_A \ .$$

Remark. The zero-stability that is involved in Dalquist's convergence theorem is a particular case of the above definition: by setting $\mu = 0$ we consider the roots $r_s(0), s = 1, \ldots, k$ of the first characteristic polynomial $\rho(r)$. Note that all the considered numerical methods are consistent and zero-stable so that they are convergent.

In order to get ∂R_A with the boundary locus technique, we note that for a value of $\mu \in \partial R_A$ at least one of the roots should have module equal to one, that is $r = e^{i\theta}$ for $\theta \in [0, 2\pi)$. Therefore, ∂R_A can be drawn by $\mu = \rho(e^{i\theta})/\sigma(e^{i\theta})$ for $\theta \in [0, 2\pi)$. Let us note here that for a consistent and zero stable method the origin of the complex plane belongs to ∂R_A. To ascertain that, take $\theta = 0$ that means $\mu = \rho(1)/\sigma(1) = 0$, since $\sigma(1) \neq 0$. In fact, for a consistent method we have that $\rho(1) = 0$ and $\rho\prime(1) - \sigma(1) = 0$, and for the method to be zero-stable the first characteristic polynomial must have a simple root at one, hence $\rho\prime(1) \neq 0$ that implies $\sigma(1) \neq 0$.

Remark. The geometric interpretation of the boundary locus technique is simple. We can consider $\rho(r) - \mu\sigma(r) = 0$ as a mapping from the complex plane of the r variable to the complex plane of the μ variable. Our interest is to define the range of values of μ coming from all r in the unit circle with center the origin. For all r on the circumference of that unit circle there exists a value of $\theta \in [0, 2\pi)$ such that $r = e^{i\theta}$, this defines ∂R_A.

To apply the boundary locus technique to a LMM we have to perform some simple algebra. Given a LMM, the two complex numbers $a + ib = \rho(e^{i\theta})$ and $c + id = \sigma(e^{i\theta})$ are known, whereas $x + iy = \mu$ in unknown. Moreover, by

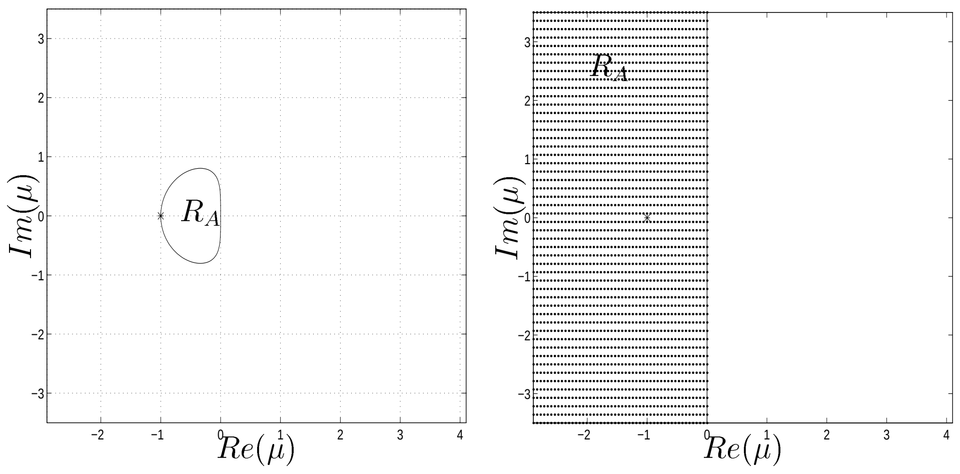

Figure 3: A-stability regions for the 2-nd order AB (on the left) and AM (right) methods. MATLAB files `a_AB2.m` and `a_AM2.m`.

setting $\mu = \rho(e^{i\theta})/\sigma(e^{i\theta})$ we get the complex equation $(c+id)(x+iy) = a+ib$ which is equivalent to the real linear system

$$cx - dy = a \ , \quad dx + cy = b \ ,$$

where, since $\sigma(1) \neq 0$, the matrix of the coefficients has determinant $\Delta = c^2 + d^2 \neq 0$, so that for each value of θ there exists a unique solution of the above system given by $x = (ac + bd)/\Delta$ and $y = (cb - da)/\Delta$. The reasoning reported above can be used to define an algorithm for plotting the A-stability region of a given LMM. The corresponding MATLAB script file named `a_LMM.m` is available from the MathWorks anonymous `ftp` site mentioned above. As input data for `a_LMM.m` the user has to provide the number of steps k and the coefficients α_j, β_j for $j = 0, \ldots, k$ of the first and second characteristic polynomials.

Figure 3 shows the result of the application of the boundary locus technique described above to the 2-nd order AB and AM methods.

The boundary locus technique can also be applied to BDF methods.

Exercise. *Use the MATLAB script named* `a_LMMs.m` *to reproduce the Figure 3 and to find out the A-stability region of the BDF2 method (compare the obtained plot with that from the MATLAB script file* `a_BDF2.m`*.)*

4 L-stability.

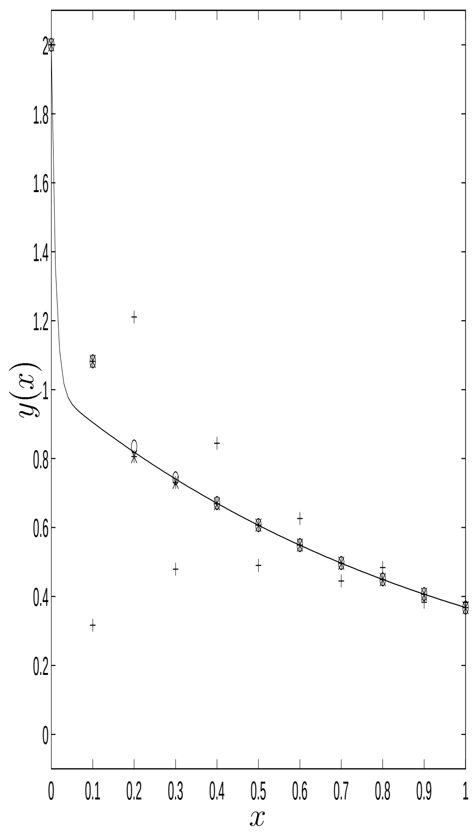

Figure 4: Solutions of (4.1). The exact solution $y(x) = e^{-100x} + e^{-x}$ is drawn by a solid line. Numerical solutions with 11 mesh-points, $+$ AM, $*$ backward Euler and o BDF methods. Plot obtained with the MATLAB script `methtra.m`.

As we have found out in Section 3, the 2-nd order AM and BDF methods are both A-stable. However, we cannot say that the two considered implicit methods are equivalent. A method is called L-stable if the roots r_j of the second characteristic polynomial $\sigma(r)$ are strictly inside the unit circle, that is $|r_j| < 1$ for all j. For a L-stable method the line at infinity in the μ plane has to belong to R_A. Let us propose a simple variation of the test problem proposed in Section 3.1, in order to explain the meaning of the L-stability concept,

$$\frac{dy}{dx} = -100\left(y - e^{-x}\right) - e^{-x} \ , \quad y(0) = 2 \ , \quad x \in [0,1] \ . \qquad (4.1)$$

Note that with respect to (3.2) we have only changed the initial condition. However, that results in the presence of a fast transient in the exact solution as illustrated by Figure 4. The same Figure shows the numerical results. For the chosen step-size the AM method is unable to resolve the fast transient. It is really a surprise for the students to verify that the 1st order backward Euler method gives a qualitative numerical solution more reliable than that provided by the 2nd order AM method. The three numerical methods used herein are all A-stable. However, the backward Euler and BDF methods are also L-stable, whereas the AM method is not. For a suitable numerical solution of problems belonging to the class of oscillatory problems the imaginary axes in the μ complex plane has to belong to the A-stability region of the method used [1, p. 66]. In this context the BDF method is more reliable than the AM method.

5 Final remarks.

Let us consider first the general solution of the governing differential equation involved in (3.2) and (4.1): $y(x) = ce^{-100x} + e^{-x}$, where c is an arbitrary constant. The initial condition $y(0) = 1$ in (3.2) requires $c = 0$ so that we have the smooth solution $y(x) = e^{-x}$. All other solutions approach this smooth one after a "rapid transient".
Exercise. *Use the MATLAB script* `trajsti.m` *to get a plot of the family of solutions.*

As we have mentioned above the test problem (4.1) differs from problem (3.2) because of the enforced initial condition (corresponding to $c = 1$ in the general solution). So that the particular solution for problem (4.1) is given by $y(x) = e^{-100x} + e^{-x}$.

For those readers interested in the implementation of numerical methods within MATLAB as well as in the numerical solution of problems of practical interest we can quote the recent MATLAB ODE suite developed by Shampine and Reichelt [19]. The suite contains a collection of M-files for solving initial value problems and is freely available on internet from the anonymous ftp area at the URL:
`ftp://ftp.mathworks.com/pub/mathworks/toolbox/matlab/funfun`.

References

[1] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations.* SIAM, Philadelphia, 1998.

[2] L. Brugnano and D. Trigiante. On the characterization of stiffness for odes. *Dynamics of Continuous, Discrete and Impulsive Systems*, 2:317–335, 1996.

[3] L. Brugnano and D. Trigiante. *Solving Differential Problems by Multistep Initial and Boundary Methods.* Gordon & Breach, Amsterdam, 1998.

[4] J. C. Butcher. *The Numerical Analysis of Ordinary Differential Equations Runge-Kutta and General Linear Methods.* Wiley, Chichester, 1987.

[5] C. F. Curtiss and J. O. Hirschfelder. Integration of stiff equations. *Proc. Nat. Acad. Sci. USA*, 38:235–243, 1952.

[6] A. L. Garcia. *Numerical Methods for Physics.* Prentice Hall, Englewood Cliffs NJ, 2000. 2nd edition.

[7] W. Gautschi. *Numerical Analysis. An Introduction.* Birkhauser, Boston, 1997.

[8] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice Hall, Englewood Cliffs NJ, 1971.

[9] E. Hairer and G. Wanner. *Solving Differential Equations II: Stiff and Differential-Algebraic Problems.* Springer, Berlin, 1991.

[10] P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations.* Wiley, New York, 1962.

[11] A. Iserles. *A First Course in the Numerical Analysis of Differential Equations.* Cambridge University Press, Cambridge, 1996.

[12] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems.* Wiley, Chichester, 1991.

[13] C. F. Van Loan. *Introduction to scientific Computing, a Matrix-Vector Approach Using MATLAB.* Prentice Hall, Upper Saddle River, 1997.

[14] J. H. Mathews. Using MATLAB to obtain both numerical and graphical solutions to hyperbolic p.d.e.'s. *Comp. Ed. J.*, IV:58–60, 1994.

[15] J. H. Mathews and K. D. Fink. Using MATLAB as a programming language for numerical analysis. *Int. J. Math. Educ. Sci. Technol.*, 25:481–490, 1994.

[16] J. H. Mathews and K. D. Fink. *Numerical Mesthods Using MATLAB.* Prentice Hall, Upper Saddle River, 3rd edition, 1999.

[17] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics.* Springer, Berlin, 2000.

[18] L. F. Shampine. *Numerical Solution of Ordinary Differential Equations.* Chapman & Hall, New York, 1994.

[19] L. F. Shampine and M. W. Reichelt. The MATLAB ODE suite. *SIAM J. Sci. Comput.*, 18:1–22, 1997.