# Numerical methods for extremely stiff systems of ordinary differential equations

T. D. Bui

*Department of Computer Science, Sir George Williams Campus, Concordia University, Montréal, Québec, Canada*

T. R. Bui

*Department of Applied Sciences, Université du Québec, Chicoutimi, Quebec, Canada*
*(Received October 1978; revised April 1979)*

Computer simulation of dynamic systems very often leads to the solution of a set of stiff ordinary differential equations. The solution of this set of equations involves the eigenvalues of its Jacobian matrix. The greater the spread in eigenvalues, the more time consuming the solutions become when existing numerical methods are employed. Extremely stiff differential equations can become a very serious problem for some systems, rendering accurate numerical solutions completely uneconomic. In this paper, we propose new techniques for solving extremely stiff systems of differential equations. These algorithms are based on a class of implicit Runge-Kutta procedure with complete error estimate. The new techniques are applied to solving mathematical models of the relaxation problem behind blast waves.

## Introduction

Stiff differential equations arise in physical equations due to the existence of greatly differing time constants. Time constant is a term used by engineers and physicists to refer to the rate of decay. For example, the equation $dy/dt = \lambda y$ has the solution $y = c\,e^{\lambda t}$. If $\lambda$ is negative, then $y$ decays by a factor of $1/e$ in time $T = 1/\lambda$. $T$ is called the time constant for a system described by the above differential equation. The larger the value of $\lambda$, the smaller the time constant. In a system of differential equations, different state variables decay at different rates. For the system:

$$\frac{dY}{dt} = F(Y) \tag{1}$$

where:

$$Y = (y_1, y_2, \ldots, y_n)^T$$
$$F = (f_1, f_2, \ldots, f_n)^T$$

the decay rates may be related locally to the eigenvalues of the Jacobian matrix of the system (i.e. $\partial F/\partial Y$). If some of the decay rates are slow and others are fast, the fast ones will control the stability of the numerical method used to solve the system of equations (1).

Consider, for example, the motion of a damped harmonic oscillator which is described by the pair of equations:

$$\begin{aligned} \frac{dy}{dt} &= v \qquad && y = \text{displacement} \\ \frac{dv}{dt} &= -\gamma v - w^2 y \qquad && v = \text{velocity} \end{aligned} \tag{2}$$

Assuming that: $\gamma = 1000.1$; $w = 1$; $y(0) = 0$; $v(0) = 999.9$.

To a good approximation, the solutions are:

$$\begin{aligned} y &= -e^{-1000t} + e^{-0.1t} \\ v &= 1000\,e^{-1000t} - 0.1\,e^{-0.1t} \end{aligned} \tag{3}$$

0307-904X/79/050355-04/$02.00

In this system, the eigenvalues of the Jacobian matrix are $\lambda_1 = -1000, \lambda_2 = -0.1$. Both $y$ and $v$ have a rapidly decaying component corresponding to $\lambda_1$, which very quickly becomes insignificant. However, for a stable numerical solution, most explicit methods require that $|h\lambda_1|$ and $|h\lambda_2|$ be bounded by a small region. This region is called the region of absolute stability.

The stability of the numerical integration of the system of equations (2) is governed by $|-1000h|$. If the well-known Runge-Kutta fourth-order method is employed, then we must have $|-1000h| \leqslant 2.8$ giving the maximum stable step-size $h = 0.0028$. Thus, nearly 100 000 integration steps would be required to reach the point where the slowly varying term is one-tenth of the maximum value.

## New developments

We are concerned here with the numerical solutions of the system of differential equations:

$$\frac{dy}{dt} = f(y), t > t_0$$
$$y(t_0) = y_0 \qquad (4)$$

where $t \in R, y \in R^n, y_0 \in R^n$ and $f: R^n \to R^n$ is assumed analytic in the neighbourhood of $y_0$. The system of differential equations (4) is called stiff in an interval $I$, if for $\forall t \in I$ the eigenvalues $\lambda_i(t)$ of the Jacobian matrix $\partial f/\partial y$ satisfy the following conditions:[1]

$$\text{Real}\,\lambda_i(t) < 0, i = 1, 2, \ldots, n \qquad \text{(i)}$$
$$\max_t |\text{Real}\,\lambda_i(t)| \geqslant \min_t |\text{Real}\,\lambda_i(t)| \qquad \text{(ii)}$$

New developments for stiff differential equations are based upon one of the following two approaches. The first approach is to build into the computer program logic that notes the fact that some terms in the solution are no longer important and then modifies or eliminates certain equations from the system. An example of this first approach is a program called NEST developed by Emanuel[2] for large systems of equations arising in chemical kinetics. The second approach aims at developing methods having a large region of stability on the complex $h\lambda$-plane (where $h$ is the time step-length and $\lambda$ is the largest eigenvalue of the Jacobian matrix of the system). Examples of this latter category include work done by others.[3–7] The region of stability for a method is defined as the domain on the $\lambda h$-complex plane, in which we obtain $|y_{m+1}/y_m| = c < 1$, when the method is applied to the test equation:

$$\frac{dy}{dt} = \lambda y, y(t_0) = y_0 \qquad (5)$$

where $\lambda$ is a complex constant with negative real part.

*Definition 1*

A method is stiffly-stable if its region of stability contains a region $R = R_1 \cup R_2$ where:

$$R_1 = \{\text{Real}(\lambda h) \leqslant D < 0\}$$
$$R_2 = \{D < \text{Real}(\lambda h) < \alpha, |\text{Im}(\lambda h)| < 0\}$$

*Definition 2*

A method is called $A$-stable in the sense of Dahlquist if the region of stability associated with that method contains the open left half-plane.

*Definition 3*

A method is called $L$-stable if it is $A$-stable and $c \to 0$ as $\lambda h \to -\infty$.

The concept of $L$-stability is a very important one, since in general $A$-stable methods which are not damped maximally as $\lambda h \to -\infty$ are unsatisfactory. This undesirable asymptotic behaviour often results in oscillatory solutions for very stiff systems. These oscillatory solutions may be meaningless but are considered suitable as long as the oscillations gradually die out. Oscillatory solutions are associated with, for example, the $A$-stable implicit trapezoidal rule since the asymptotic behaviour for this method is:

$$\left|\frac{y_{m+1}}{y_m}\right| \to 1 \text{ as } \lambda h \to -\infty$$

For extremely stiff systems, it is desirable to develop $L$-stable methods rather than non-maximally damped $A$-stable methods or stiffly stable methods. The development of $L$-stable methods has not received adequate attention in the past. Most available packages for stiff differential equations are based on methods which are not $A$-stable or methods which are $A$-stable but not damped maximally as $\lambda h \to -\infty$. The aim of this work is to develop methods with variable orders which are $L$-stable and to implement these methods into a package for handling extremely stiff systems.

## Development and implementation of $L$-stable methods

We will derive and implement some $L$-stable methods based on a class of Runge-Kutta methods known as the Rosenbrock procedure. The conditions of $L$-stability require that the methods must be implicit in nature. Fully implicit Runge-Kutta as well as semi-implicit Runge-Kutta methods are known to satisfy the condition of strong stability.[6,8] However, these methods require to solve a system of nonlinear equations at each step. The Rosenbrock procedure requires only the solution of a system of linear equations at each step, a much simpler task compared to fully implicit or semi-implicit approaches.[9]

The Rosenbrock procedure for solving the system of equations (4) is defined by:

$$y_{m+1} = y_m + h\sum_{i=1}^{s} w_i k_i$$
$$k_i = f\left(y_m + h\sum_{j=1}^{i-1} a_{ij}k_j\right) + dh\frac{\partial f}{\partial y}\left(y_m + \sum_{j=1}^{i-1} b_{ij}k_j\right) k_i \qquad (6)$$

where $w_i, a_{ij}, b_{ij}$ and $d$ are constants to be defined so that the condition of $L$-stability is satisfied.

*Two-stage second-order method*

Rosenbrock[10] has proved that the following two-stage, second-order scheme is $L$-stable: $w_1 = 0, w_2 = 1$, $a_{21} = (\sqrt{2}-1)/2, b_{21} = 0$, and $d = 1 - \sqrt{2}/2$. The method is defined as follows:

$$y_{m+1} = y_m + hk_2$$
$$k_1 = A(y_m)^{-1} f(y_m) \qquad (7)$$
$$k_2 = A(y_m)^{-1} f(y_m + ha_{21}k_1)$$

where:

$$A(y_m) = \left[I - dh\frac{\partial f}{\partial y}(y_m)\right]$$

*Three-stage third-order method*

For $s = 3$, we have proved that the following scheme is $L$-stable: $w_1 = 0, w_2 = w_3 = \frac{1}{2}, a_{21} = -0.509\,643\,6824$, $a_{31} = 0.327\,025\,8661, a_{32} = 0.310\,884\,7731$, $d = 0.435\,866\,5216$. The method is defined by:

$$\begin{aligned} y_{m+1} &= y_m + \tfrac{1}{2}h(k_2 + k_3) \\ k_1 &= A(y_m)^{-1} f(y_m) \\ k_2 &= A(y_m)^{-1} f(y_m + ha_{21}k_1) \\ k_3 &= A(y_m)^{-1} f(y_m + ha_{31}k_1 + ha_{32}k_2) \end{aligned} \tag{8}$$

where $b_{ij} = 0$ for all $i$ and $j$.

*Four-stage fourth-order method*

A four-stage, fourth-order $L$-stable method based on procedure (6) was developed in one of our previous papers.[11,12] The parameters are defined by:

$w_1 = 0.945\,156\,478\,6$ $\quad w_2 = 0.341\,323\,172$

$w_3 = 0.565\,513\,957\,5$ $\quad w_4 = -0.851\,993\,608\,1$

$a_{21} = -0.5$ $\quad a_{31} = -0.101\,223\,611\,5$

$a_{32} = 0.976\,223\,611\,5$ $\quad a_{41} = -0.392\,209\,676\,3$

$a_{42} = 0.715\,114\,025\,1$ $\quad a_{43} = 0.143\,037\,162\,5$

$d = 0.572\,816\,062\,5$

$$\begin{aligned} y_{m+1} &= y_m + h\sum_{i=1}^{4} w_i k_i \\ k_1 &= A(y_m)^{-1} f(y_m) \\ k_2 &= A(y_m)^{-1} f(y_m + ha_{21}k_1) \\ k_3 &= A(y_m)^{-1} f(y_m + h(a_{31}k_1 + a_{32}k_2)) \\ k_4 &= A(y_m)^{-1} f(y_m + h(a_{41}k_1 + a_{42}k_2 + a_{43}k_3)) \end{aligned} \tag{9}$$

The LSTIFF program works with a fixed formula selected by the user at the start of the integration from among the above three formulae, designated by number of stages and order:

| | |
|---|---|
| LSTIFF (2, 2) | Equation (7) |
| LSTIFF (3, 3) | Equation (8) |
| LSTIFF (4, 4) | Equation (9) |

This program was implemented on the CDC Cyber 172 at the Computing Center of Concordia University. The step-size is controlled by estimating local error using Richardson's extrapolation scheme. First a step-size $h$ is taken from $y_m$ to compute $y_{m+1}$. Next, the value of $y^*_{m+1}$ is recomputed using twice the step-size $h/2$. The estimate of the local truncation error is taken to be:

$$E_{m+1} = \frac{\|y^*_{m+1} - y_{m+1}\|}{2^s - 1}$$

where $s$ is the order of the method, $\|\cdot\|$ is the weighted RMS norm:

$$\|y\| = \left[\frac{1}{n}\sum_{i=1}^{n}(y^i/y^i_{\max})^2\right]^{1/2}$$

and $y^i_{\max}$ is the maximum modulus of the $i$th component so far in the integration. This is motivated by the idea that, if the user specifies a local error tolerance $\epsilon$, then errors of order $\epsilon \cdot y^i_{\max}$ are allowed when $y^i$ is near $y^i_{\max}$, and it is not in general useful to require later errors to be smaller than that. The following strategy is employed:[13]

(i) If $E_{m+1} > \epsilon$, the step is rejected and $h$ is reduced to make the expected error $\simeq \epsilon/5$.

(ii) If $3\epsilon/4 < E_{m+1} \leqslant \epsilon$, the step is accepted, but $h$ is reduced to make the expected error on the next step $\sim\epsilon/5$.

(iii) If $\epsilon/10 < E_{m+1} \leqslant 3\epsilon/4$, the step is accepted and the same $h$ is used for the next step.

(iv) If $E_{m+1} \leqslant \epsilon/10$, the step is accepted and $h$ is increased to make an expected error on the next step $\sim\epsilon/2$.

The matrices $[A(y_m)]$ are evaluated from the Jacobian supplied by the user. Equation (9) of this package was tested on the system of two stiff ordinary differential equations:

$$\frac{dx}{dt} = -1000\,04x + 10\,000y^4$$

$$\frac{dy}{dt} = x - y - y^4$$

with initial conditions $x(0) = y(0) = 1.0$. The stiffness ratio for this problem is $10^4$. Some comparisons with other $A$-stable methods were published elsewhere.[12] It was concluded that $L$-stable methods are very important for extremely stiff systems.

## Applications

The package described in this paper was employed to solve the relaxation equations for the two modes of vibrational coupling of the $CO_2$ molecules in a $CO_2$–$N_2$–He mixture behind blast waves.[14] The mathematical modelling of this physical system produces two relaxation equations for the vibrational energies $e_{v1}$ and $e_{v2}$:

$$\begin{aligned} \frac{de_{v1}(T_{v1})}{dt} &= \frac{1}{\tau_1}\{e_{v1}(T) - e_{v1}(T_{v1})\} \\ \frac{de_{v2}(T_{v2})}{dt} &= \frac{1}{\tau_2}\{e_{v2}(T) - e_{v2}(T_{v2})\} \end{aligned} \tag{10}$$

where $T_{v1}$ and $T_{v2}$ are the vibrational temperatures for the the two modes:

$$\begin{aligned} e_{v1} &= \frac{X_{CO_2}W_{CO_2}}{W}R_{CO_2}\left[\frac{h\nu_1}{k}(e^{h\nu_1/kT_{v1}} - 1)^{-1} + \frac{2h\nu_2}{k}(e^{h\nu_2/kT_{v1}} - 1)^{-1}\right] \\ e_{v2} &= \frac{X_{CO_2}W_{CO_2}}{W}R_{CO_2}\frac{h\nu_3}{k}(e^{h\nu_3/kT_{v2}} - 1)^{-1} + \frac{X_{N_2}W_{N_2}}{W}R_{N_2}\frac{h\nu}{k}(e^{h\nu/kT_{v2}} - 1)^{-1} \end{aligned}$$

where $X_i$, $W_i$ and $R_i$ are the mole fraction, molecular weight and the gas constant of species $i$ respectively, $W = \Sigma_i X_i W_i$; $\nu_1, \nu_2, \nu_3$ denote the vibrational frequencies of the symmetric, bending and asymmetric modes for $CO_2$ respectively, $\nu$ denotes the vibrational frequency of $N_2$, $h$ and $k$ are the Planck and Boltzmann constants respectively.

$T$ denotes the translational temperature, and the relaxation times $\tau_1$ and $\tau_2$ are as given as:

$$\tau_1 = \tau_c$$

$$\tau_2 = (X_{CO_2} + X_{N_2})\,(X_{CO_2}\tau_a^{-1} + X_{N_2}\tau_b^{-1})^{-1}$$

where $\tau_a$, $\tau_b$ and $\tau_c$ refer to the relaxation times which are averages of the detailed $CO_2$–$CO_2$, $CO_2$–$N_2$, $CO_2$–He, $N_2$–$N_2$ and $N_2$–He collisions, i.e.:

$$\tau_a^{-1} = \frac{X_{CO_2}}{(\tau_a)\ CO_2\text{–}CO_2} + \frac{X_{N_2}}{(\tau_a)\ CO_2\text{–}N_2} + \frac{X_{He}}{(\tau_a)\ CO_2\text{–}He}$$

$$\tau_b^{-1} = \frac{X_{CO_2}}{(\tau_b)\ N_2\text{–}CO_2} + \frac{X_{N_2}}{(\tau_b)\ N_2\text{–}N_2} + \frac{X_{He}}{(\tau_b)\ N_2\text{–}He}$$

$$\tau_c^{-1} = \frac{X_{CO_2}}{(\tau_c)\ CO_2\text{–}CO_2} + \frac{X_{N_2}}{(\tau_c)\ CO_2\text{–}N_2} + \frac{X_{He}}{(\tau_c)\ CO_2\text{–}He}$$

The values $(\tau)_{i-j}$ are taken from the compilation by Taylor and Bitterman.[15] The program LSTIFF was used to solve the system of equations (10) for $T_{v1}$ and $T_{v2}$. These vibrational temperatures can be used to calculate the particle populations of the upper and lower laser levels behind the shock front. Due to the very fast quenching rate of the translational temperature (i.e. $T$) behind the blast front, the system of equations (10) becomes very stiff. The well-known Runge-Kutta method of order four requires a time step-size in the order of $10^{-12}$ to obtain stable solutions. Our package allows the use of large integration step-sizes thus reducing computing time by a significant factor while maintaining small truncation errors.

The coupling between the gas dynamics of the blast wave and the chemical system is an extremely nonlinear problem. Due to the differences between the reaction rates, the stiffness ratio for this problem can be as high as $10^{12}$. Therefore, packages based on methods which are only stiffly stable such as Gear's DIFSUB will not be able to solve problems of this kind.

## Conclusions

While most other techniques for integrating stiff differential equations are not $A$-stable, or are $A$-stable but not damped as $\lambda h \to -\infty$, the methods developed in this paper are all $L$-stable. The package LSTIFF combines the desirable condition of strong stability ($L$-stable) with a reasonable order of accuracy while still maintaining computational efficiency by preserving the one-step nature of the method.

## Acknowledgement

This work was supported in part by the National Research Council of Canada through its grant A-9265.

## References

1 Lambert, J. D. 'Computational methods in ordinary differential equations', Wiley, New York, 1973
2 Emanuel, G. Rep. TDR-269(4230-20)-3, The Aerospace Corporation, El Segundo, California, 1964
3 Liniger, W. and Willoughby, R. A. *SIAM J. Numer. Anal.*, 1970, 7, 47
4 Treanor, C. E. *Math. Comp.*, 1966, 20, 39
5 Gear, C. W. 'Numerical initial value problems in ordinary differential equations', Prentice-Hall, Englewood Cliffs, New Jersey, 1971
6 Butcher, J. C. *Math. Comp.*, 1964, 18, 50
7 Butcher, J. C. *BIT*, 1976, 16, 237
8 Nørsett, S. P. Math. and Computation Rep. 6/74. Dept of Math., Univ. of Trondheim, Norway, 1974
9 Bui, T. D. *Math. Comp.*, 1979, 33, No. 147
10 Rosenbrock, H. H. *Comput. J.*, 1963, 5, 329
11 Bui, T. D. *Inf. Process. Lett.*, 1977, 6, 158
12 Bui, T. D. *J. A.C.M.*, 1979, 27 (3), 483
13 Alexander, R. *SIAM J. Numer. Anal.*, 1977, 14, 1006
14 Bui, T. D. *et al. Acta Astron.*, 1974, 1, 1085
15 Taylor, R. L. and Bitterman, S. *Rev. Mod. Phys.*, 1969, 41, 26