# ENGR545

Term-Paper

## Stiff ODE Methods in Neuronal Modeling

Daniel Nikovski

## 1 Introduction

A system of differential equations is stiff when there are two or more very different scales of the independent variable. Different steps of integration correspond to each of these scales, and if the system is solved through traditional methods and accuracy and stability have to be maintained, the smallest of all steps should be chosen. This might lead to very slow and inefficient integration.

Alternative methods have been investigated in which the step of integration can be much larger and the integration much fater. Most of the methods proposed are generalizations of traditional techniques like Runge-Kutta schemes, the Bulirsch-Stoer method and predictor-corrector algorithms. Typically, implicit integration is chosen, because explicit methods often suffer stability problems in stiff sets of equations. Furthermore, elaborate step adaptation schemes are provided to accomodate all possible scales of the independent variable.

## 2 Overview of methods

Three major classes of methods for the solution of stiff ODEs are of primary interest [7]:

1. Rosenbrock-Wanner (ROW) methods [4] that are in practice a generalization of the Runge-Kutta method.
2. Predictor-corrector methods, based on Gear's backward differentiation method [3].
3. Generalizations of the Bulirsch-Stoer method. The semi-implicit extrapolation method due to Bader and Deuflhard will be discussed [5].

## 2.1 Rosenbrock-Wanner methods

The solution for the moment $x_0 + h$ is of the form

$$\mathbf{y}(x_0 + h) = \mathbf{y}_0 + \sum_{i=1}^{s} c_i \mathbf{k}_i$$

The corrections $\mathbf{k}_i$ are found by solving $s$ linear equations:

$$(\mathbf{I} - \gamma h \mathbf{f}') \cdot \mathbf{k}_i = h\mathbf{f}\left(\mathbf{y}_0 + \sum_{j=1}^{i-1} \alpha_{ij}\mathbf{k}_j\right) + h\mathbf{f}' \cdot \sum_{j=1}^{i-1} \gamma_{ij}\mathbf{k}_j, \quad i = 1, \ldots, s$$

The Jacobian of the right-hand side $\mathbf{f}$ is denoted by $\mathbf{f}'$. The coefficients $\gamma, c_i, \alpha_{ij}$, and $\gamma_{ij}$ are fixed constants. Kaps and Rentrop gave two different sets of values for these parameters [4], and another set is used in [7]. Runge-Kutta methods are obtained for $\gamma = \gamma_{ij} = 0$; if, in addition to that, $\alpha_{ij} = 0$, the Euler forward method is obtained.

The step adjustment is implemented through a Runge-Kutta-Fehlberg embedded scheme. Two values for $\mathbf{y}(x_0 + h)$ are computed for two different orders of the method - usually the third and fourth order. The difference between these two values is used as an estimate $EST$ of the truncation error. A new step size is computed based on the the truncation error estimate and the desired accuracy criterion $TOL$:

$$h_{new} = 0.9 h_{old} \left(\frac{TOL}{EST}\right)^{\frac{1}{4}}$$

The new step is accepted only if it differs sufficiently from the old one:

If $h_{new} > 1.5 h_{old}$, then $h_{new} = 1.5 h_{old}$.

If $h_{new} < 0.5 h_{old}$, then $h_{new} = 0.5 h_{old}$.

## 2.2 Gear's methods

Gear proposed a series of methods with first- to seventh-order accuracy. His methods are predictor-corrector methods based on the Adams linear- multistep techniques with explicit prediction and implicit iterative correction [3]. These methods are enhanced with step control

strategies, so that stiff sets of equations can be solved. In addition to that, the methods proposed by Gear can change the order of the method being used during the course of integration.

The Adams-Bashforth $p$th order predictor equation for the differential equation $\mathbf{y}' = \mathbf{f}(\mathbf{y}, x)$ is

$$\mathbf{y}_{n,(0)} = y_{n-1} + \beta_1 h \mathbf{y}'_{n-1} + \cdots + \beta_p h \mathbf{y}'_{n-p},$$

where $\mathbf{y}_i = \mathbf{y}(x_i)$, $x_i = ih$. The coefficients $\beta_i$ are given in the texbook [6]. The approximation $\mathbf{y}_{n,(0)}$ is used as first approximation in the Adams-Moulton corrector formula of $p$th order, given by

$$\mathbf{y}_{n,(m+1)} = y_{n-1} + \beta_0^* h \mathbf{f}(\mathbf{y}_{n,(m)}, x_n) + \beta_1^* h \mathbf{y}'_{n-1} + \cdots + \beta_{p-1}^* h \mathbf{y}'_{n-p+1}$$

The corrector equation is iterated until it converges to $\mathbf{y}_n$. The method for stiff equations is similar. The predictor formula is

$$\mathbf{y}_{n,(0)} = \alpha_1 \mathbf{y}_{n-1} + \cdots + \alpha_p \mathbf{y}_{n-p} + \eta_1 h \mathbf{y}'_{n-1}$$

.

The corrector formula is

$$\mathbf{y}_{n,(m+1)} = \alpha_1^* y_{n-1} + \cdots + \alpha_p^* \mathbf{y}_{n-p} + \eta_0^* h \mathbf{f}(\mathbf{y}_{n,(m)}, x_n)$$

The predictor-corrector equations can be cast into matrix form for ease of implementation.

Step size control in Gear's method is more involved than the approach in ROW methods, because no embedded scheme is available to provide a truncation error estimate. Instead, Gear's method estimates the truncation error as proportional to the newly computed values of the dependent variables and the $p$th power of the step size:

$$ERROR = C_{p+1} h^{p+1} \mathbf{y}^{(p+1)}(x_n)$$

The coefficient $C_{p+1}$ depends on the method being used. If $YMAX$ is a vector, containing the maximum values for the individual dependent variables, the requirement for the step to be accepted is

$$D = \sum_{i=1}^{N} \left( \frac{ERROR_i}{YMAX_i} \right)^2 \leq E$$

where $E$ is a quantity derived from the desired accuracy and dependent on the order of predictor-corrector currently used. If the order is to be retained the new step is calculated as $h/PR2$ with

$$PR2 = 1.2 \left( \frac{D}{E} \right)^{\frac{1}{2(p+1)}}$$

Similarly, estimates of the step size are computed for the case when the order of the method would be lowered (PR1) or increased (PR3). The smallest step size of the three is chosen and the order is changed accordingly.

Three final heuristic controls are used:

1. If a step increase is less than 1.1, it is not taken.
2. A change in step or order is not permitted for $p + 1$ steps after the last change unless a step fails.
3. If, after $p + 1$ successful steps, no step increase is possible, further testing is not attemped for ten more steps in order to reduce the overhead of trying to increase the step during all immediately following steps.

## 2.3 Semi-implicit Extrapolation Method

Bader and Deuflhardt [5] proposed a semi-implicit discretization, based on the Bulirsch-Stoer method. Their method works very well for stiff sets of ODEs, unlike the original Bulirsch-Stoer method. The midpoint rule used in Bulirsch-Stoer is rewritten in implicit form:

$$\mathbf{y}_{n+1} - \mathbf{y}_{n-1} = 2h\mathbf{f} \left( \frac{\mathbf{y}_{n+1} + \mathbf{y}_{n-1}}{2} \right)$$

This can be converted to semi-implicit form by linearizing the right-hand side about $\mathbf{f}(\mathbf{y}_n)$. The result is the semi-implicit midpoint rule:

$$\left[\mathbf{I} - h\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right] \cdot \mathbf{y}_{n+1} = \left[\mathbf{I} - h\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right] \cdot \mathbf{y}_{n-1} + 2h\left[\mathbf{f}(\mathbf{y}_n) - \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \cdot \mathbf{y}_n\right]$$

The first step is the semi-implicit Euler step:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\left[\mathbf{I} - h\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right]^{-1} \cdot \mathbf{f}(\mathbf{y}_n)$$

The method uses a special last step in which the last $\mathbf{y}_n$ is replaced by

$$\bar{\mathbf{y}}_n \equiv \frac{1}{2}(\mathbf{y}_{n+1} + \mathbf{y}_{n-1})$$

Since this method is based on Richardson extrapolation like Bulirsch-Stoer, the step control mechanism of the latter (proposed also by Bader and Deuflhard) can be used for stiff equations as well.

# 3 Applications to neuronal modeling

Computer simulation of single neurons and whole networks of neurons is currently a field of very intensive research. There are many levels of neuronal modeling, ranging from simulation of abstract computing units to detailed, biologically realistic simulation of neuronal cells. Simulation of the latter type has become possible only recently due to the advent of accessible and inexpensive supercomputing power [1].

The excitation of the membrane of a neuronal cell is described by the Hogkin-Huxley set of partial differential equations (PDEs) [2]. Under certain conditions, the PDEs can be reduced to the folowing system of ODEs:

$$C\frac{dV}{dt} = -\bar{g}_{Na}m^3h(V - E_{Na}) - \bar{g}_K m^3 h(V - E_K) - \bar{g}_{leak}m^3h(V - E_{leak})$$

$$\frac{dm}{dt} = (1 - m)\alpha_m(V) - m\beta_m(V)$$

$$\frac{dh}{dt} = (1-h)\alpha_h(V) - h\beta_h(V)$$

$$\frac{dn}{dt} = (1-n)\alpha_n(V) - n\beta_n(V)$$

Hodgkin and Huxley provided expressions for the functions $\alpha_m(V)$, $\alpha_h(V)$, $\alpha_n(V)$, $\beta_m(V)$, $\beta_h(V)$, $\beta_n(V)$ based on their experiments with axons of the giant squid *Loligo*.

The general Hodgkin-Huxley set of equations is not particularly stiff [2]; however, if a compartmental model of a neuron is employed, the resulting system of equations is typically stiff.

A compartmental model is obtained if the PDE set of Hodgkin-Huxley equations is discretized along the $x$-axis. The voltage within a single segment is assumed to be the same. In case only one large segment exists, the set of PDEs reduces to the above set of ODEs. The latter condition is equivalent to inserting a silver electrode along the axis of the axon, which effectively equilizes the potentials along the axis.

If, however, the number of segments is more than one, the PDEs get reduced to a large set of equations, namely four for each segment. This is equivalent to applying the *method of lines* for transforming PDEs to ODEs. The resulting set of ODEs is typically stiff.

The stiffness of a ODE problem can be defined quantitatively as the ratio between the largest to smallest time scale of the problem. The system of ODEs can be linearized :

$$\frac{d\vec{V}}{dt} = \mathbf{A}\vec{V}$$

For this problem the matrix $\mathbf{A}$ is tridiagonal Toepitz matrix. The solution of the linearized system is a sum of exponential functions of the form $e^{\lambda t}$, where $\lambda$ is an eigenvalue of the matrix $\mathbf{A}$. Hence the stiffness of the system can be inferred from the ratio of the largest to smallest eigenvalue of the matrix. This ratio is also the $l_2$-norm condition number of the matrix.

The stiffness of the system then can be computed as

$$\frac{\sin^2\left(\frac{(N-1)\pi}{2N}\right) + \rho}{\sin^2\left(\frac{\pi}{2N}\right) + \rho}$$

where $\rho$ is the dissipation of the cable model used in terms of the membrane time constant $RC$.

The stiffness of the system increases with the number of the compartments, so if enough spatial detail is to be obtained, methods for the solution of stiff ODEs have to be employed. For this set of equations, Gear's method has been used [2].

# References


[1] Wilson, Matthew A., James M. Bower (1989). The Simulation of Large-Scale Neural Networks , in Christoph Koch and Idan Segev, (eds.), *Methods in Neuronal Modeling*, pp.291-333, Cambridge: MIT Press.

[2] Mascagni, Michael V. (1989). Numerical Methods for Neuronal Modeling, in Christoph Koch and Idan Segev, (eds.), *Methods in Neuronal Modeling*, pp.439-484, Cambridge: MIT Press.

[3] Gear, C.W. (1971). The Automatic Integration of Ordinary Differential Equations. *Communications of ACM* vol 14, pp.176-179

[4] Kaps, P., and Rentrop, P. (1979). Generalized Runge-Kutta Methods of Order Four with Stepsize Control for Stiff Ordinary Differential Equations. *Numerische Mathematik*, vol.33, pp.55-68.

[5] Bader, G. and Deuflhard, P. (1983). A Semi-Implicit Mid-Point Rule for Stiff Systems of Differential Equations. *Numerische Mathematik*, vol.41, pp.373-398.

[6] Hoffman, Joe D. (1992). *Numerical Methods for Engineers and Scientists*, New York: McGraw-Hill.

[7] Press, W.H et al. (1992) *Numerical Recipes in C*, Cambridge University Press.