

A comparison of non-standard solvers for ODEs describing cellular reactions in the heart

Mary C. MacLachlan[†], Joakim Sundnes^{†‡}, Raymond J. Spiteri[§]

(September 28, 2005)

Mathematical models for the electrical activity in cardiac cells are normally formulated as systems of ordinary differential equations (ODEs). The equations are nonlinear and describe processes occurring on a wide range of time scales. Under normal accuracy requirements, this makes the systems stiff and therefore challenging to solve numerically. Because standard implicit solvers are difficult to implement, explicit solvers such as the forward Euler method are commonly used, despite their poor efficiency. Non-standard formulations of the forward Euler method, derived from the analytical solution of linear ODEs, can give significantly improved performance while maintaining simplicity of implementation. In this paper we study the performance of three non-standard methods on two different cell models with comparable complexity but very different stiffness characteristics.

Keywords: cardiac cell models, non-standard finite difference methods, numerical methods for initial-value problems

1 Introduction

Since the first Purkinje cell model was introduced in 1962 by Noble et al. [1], a large number of models have been developed for different types of cardiac cells. Important contributions include [1, 2] describing Purkinje cells, [3–5] describing ventricular cells, and [6, 7] describing atrial cells. The models are typically formulated as systems of nonlinear ordinary differential equations (ODEs). These models are being continually developed to give an increasingly detailed and accurate description of cellular physiology. However, this development also tends to increase the complexity of the models. The newer models attempt to capture processes on a wider range of time scales. This range of time scales is normally a source of stiffness, and hence the equations become more challenging to solve numerically. Explicit numerical methods are forced to take steps on the fastest time scale due to stability restrictions.

[†] Simula Research Laboratory, P.O. Box 134, N-1325 Lysaker, Norway; email: mary@simula.no; email: sundnes@simula.no; phone: +47 67 82 82 00; fax: +47 67 82 82 01

[‡] Department of Informatics, University of Oslo, P.O. Box 1080, N-0316 Oslo, Norway

[§] Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, S7N 5C9, Canada; email: spiteri@cs.usask.ca; phone: (306)-966-2909; fax: (306)-966-4884

These small steps then translate into inefficient simulations and yield solutions that are much more accurate than normal accuracy requirements would necessitate.

The fact that most advanced models for cardiac cells are stiff is one of the considerable challenges when attempting to solve these equations efficiently. A wide range of solvers have been developed for stiff ODE systems, see e.g., Hairer and Wanner [8], but these are normally implicit methods that are difficult to implement in a robust manner. When simulating the behaviour of a single cell, i.e., a single ODE system, over short time intervals, the efficiency of the numerical method is hardly noticeable. Both simple explicit methods and various available libraries of solvers for stiff problems may be suitable solutions. However, both the poor efficiency of explicit solvers and the complexity of implicit solvers become important issues when the cell models are used in simulations of heart tissue or the complete heart muscle. In this case the cell model ODEs are coupled to partial differential equations (PDEs), e.g., the bidomain model [9, 10], that describes the variations in the electrical potentials throughout the tissue. The bidomain model is a system of two partial differential equations, and the coupling of the cell model ODEs to this model typically makes it more difficult to utilize standard explicit solvers or available libraries for solving the ODEs. Furthermore, when the PDEs of the bidomain model are discretized in space, we typically need to solve one cell model system for each node in the computational grid, which for simulations of a complete heart may easily exceed 10 million [11]. In this case the efficiency of the ODE solver becomes a very important issue.

Because of the complexity involved in utilizing implicit ODE solvers in simulations based on the bidomain model, the simple forward Euler method is a popular choice among researchers in this field [12, 13]. However, this explicit method is known to behave poorly for stiff problems, and for some recent cell models, such as the 1999 canine model by Winslow et al. [5], its poor performance renders it infeasible for use in practice, even for single cell simulations. A clever alternative to the forward Euler method was derived in the 1970s by Rush and Larsen [14]; it has gained a wide popularity in the heart cell simulation research community [15–17]. The method utilizes the fact that although the cell model ODE systems are nonlinear, most of the ODEs are linear when some variables are assumed to be constant. For these equations it is therefore possible to derive an update formula based on the analytical solution of linear ODEs. The remaining equations in the system are nonlinear even when all other variables are assumed constant, and in the original method these are treated with the forward Euler method.

Although formally only first-order accurate, the non-standard finite-difference (NSFD) ODE solver introduced by Rush and Larsen has proved to be a significant improvement over the standard forward Euler method. Numerical studies have shown that for a wide range of cell models both stability and accuracy properties are substantially improved. Although the method may not be as efficient or robust as general-purpose stiff ODE solvers such as BDF methods or implicit Runge-Kutta methods, see e.g., [18], the method is effectively as easy to implement as the forward

Euler method, and this is probably the main reason for its popularity.

The efficiency of the Rush–Larsen NSFD solver for the Luo–Rudy [3] cell model ODE system was investigated in [19]. The purpose of the present paper is to evaluate the performance of NSFD methods for a pair of fairly recent cell models. One is an atrial model by Courtemanche et al. [7], and the other is a ventricular model by Winslow et al. [5]. These models are ODE systems consisting of 21 and 31 equations, respectively, and the models are of comparable complexity. However, the characteristics of the two models in terms of stiffness are very different, and we investigate how this affects the efficiency of the NSFD solvers. The numerical experiments presented here are for single cell simulations only, but the results regarding computational efficiency are directly transferable to the case of full-scale simulations of the bidomain model based on operator splitting [20].

The organization of this paper is as follows. Section 2 introduces the two ODE systems considered. Section 3 describes different versions of the NSFD solvers. Section 4 describes numerical experiments with the different solvers and compares their efficiency for solving the two model systems. Finally, Section 5 summarizes our work and gives recommendations on the applicability of the various solvers.

2 Mathematical models

2.1 The atrial cell model

The atrial cell model of Courtemanche et al. includes 15 ionic and pump currents, including the handling of intracellular calcium by the sarcoplasmic reticulum (SR). The intracellular concentrations of calcium, sodium, and potassium are also included in the model. The transmembrane potential, V , satisfies

$$\frac{dV}{dt} = -\frac{1}{C_m} I_{\text{ion}},$$

where I_{ion} is defined as

$$\begin{aligned} I_{\text{ion}} = & I_{\text{Na}} + I_{\text{K1}} + I_{\text{to}} + I_{\text{Kur}} + I_{\text{Kr}} + I_{\text{Ks}} \\ & + I_{\text{Ca,L}} + I_{\text{p,Ca}} + I_{\text{NaK}} + I_{\text{NaCa}} + I_{\text{b,Na}} + I_{\text{b,Ca}}. \end{aligned}$$

Twelve of the 15 currents included in the model are ionic currents, 7 of which are controlled by the action of gating variables described by ODEs of the form

$$\frac{dy}{dt} = \frac{y_{\infty} - y}{\tau_y}, \quad (1)$$

where y is the gating variable in question, and the terms y_∞ and τ_y are defined as

$$y_\infty = \frac{\alpha_y}{\alpha_y + \beta_y},$$

$$\tau_y = \frac{1}{\alpha_y + \beta_y},$$

with both α_y and β_y being functions of V . Complete expressions for the α_y and β_y can be found in [7]. There are a total of 15 gating variables in the model. For the fast sodium current there are 3 gates, m , h , and j . Five potassium currents are included in the model, four of which are time dependent: the transient outward potassium current, with gates o_a and o_i , the ultrarapid delayed rectifier potassium current, with gates u_a and u_i , the rapid delayed outward rectifier potassium current, with gate x_r , and the slow delayed outward rectifier potassium current, with gate x_s . The L-type calcium current is controlled by three gates, d , f , and f_{Ca} , and the calcium release from the junctional sarcoplasmic reticulum (JSR) is also controlled by three gates, u , v , and w . All 15 gates satisfy equation (1), with $y = m, h, j$, etc.

In addition to the ODEs for the gating parameters, the Courtemanche et al. model includes ODEs for the intracellular sodium, potassium, and calcium concentrations ($[Na^+]_i$, $[K^+]_i$, $[Ca^{2+}]_i$), including the calcium uptake and release by the SR ($[Ca^{2+}]_{up}$, $[Ca^{2+}]_{rel}$),

$$\frac{d[Na^+]_i}{dt} = \frac{-3I_{Na,K} - 3I_{NaCa} - I_{b,Na} - I_{Na}}{FV_i},$$

$$\frac{d[K^+]_i}{dt} = \frac{2I_{Na,K} - I_{K1} - I_{to} - I_{Kur} - I_{Kr} - I_{Ks} - I_{b,K}}{FV_i},$$

$$\frac{d[Ca^{2+}]_i}{dt} = \frac{B1}{B2},$$

$$B1 = \frac{2I_{NaCa} - I_{p,Ca} - I_{Ca,L} - I_{b,Ca}}{2FV_i} + \frac{V_{up}(I_{up,leak} - I_{up}) + I_{rel}V_{rel}}{V_i},$$

$$B2 = 1 + \frac{[Trpn]_{max}K_{m,Trpn}}{([Ca_i^{2+}] + K_{m,Trpn})^2} + \frac{[Cmdn]_{max}K_{m,Cmdn}}{([Ca^{2+}]_i + K_{m,Cmdn})^2},$$

$$\frac{d[Ca^{2+}]_{up}}{dt} = I_{up} - I_{up,leak} - I_{tr} \frac{V_{rel}}{V_{up}},$$

$$\frac{d[Ca^{2+}]_{rel}}{dt} = (I_{tr} - I_{rel}) \left\{ 1 + \frac{[Csqn]_{max}K_{m,Csqn}}{([Ca^{2+}]_{rel} + K_{m,Csqn})^2} \right\}^{-1}.$$

In total, there are 21 ODEs in the Courtemanche et al. atrial cell model. A more complete description can be found in [7].

2.2 The ventricular cell model

We here list the ODEs of the 1999 canine model by Winslow et al. We only present the structure of the equations; for a full specification of ionic currents and coefficient functions we refer to the original presentation in [5].

The transmembrane potential is governed by a sum of 13 ionic currents;

$$\begin{aligned} \frac{dV}{dt} = & -(I_{Na} + I_{Ca} + I_{Ca,K} + I_{Kr} + I_{Ks} + I_{to} + I_{K1} + I_{Kp} \\ & + I_{NaCa} + I_{NaK} + I_{p(Ca)} + I_{Ca,b} + I_{Na,b}). \end{aligned}$$

The ionic currents, specified in [5] are in general nonlinear functions of V .

Membrane currents of sodium and potassium are governed by 7 gating variables, which are described by

$$\begin{aligned} \frac{dm}{dt} &= \alpha_m(1 - m) - \beta_m m, \\ \frac{dh}{dt} &= \alpha_h(1 - h) - \beta_h h, \\ \frac{dj}{dt} &= \alpha_j(1 - j) - \beta_j j, \\ \frac{dX_{Kr}}{dt} &= \frac{(X_{Kr}^\infty - X_{Kr})}{\tau_{X_{Kr}}}, \\ \frac{dX_{Ks}}{dt} &= \frac{(X_{Ks}^\infty - X_{Ks})}{\tau_{X_{Ks}}}, \\ \frac{dX_{to}}{dt} &= \alpha_{X_{to}}(1 - X_{to}) - \beta_{X_{to}} X_{to}, \\ \frac{dY_{to}}{dt} &= \alpha_{Y_{to}}(1 - Y_{to}) - \beta_{Y_{to}} Y_{to}. \end{aligned}$$

All coefficients are functions of the transmembrane potential V only.

A characteristic of this model is its advanced handling of calcium dynamics. This

includes the following system to describe release of calcium from the SR.

$$\begin{aligned}\frac{dP_{C_1}}{dt} &= -k_a^+ [Ca^{2+}]_{ss}^4 P_{C_1} + k_a^- P_{O_1}, \\ \frac{dP_{O_1}}{dt} &= k_a^+ [Ca^{2+}]_{ss}^4 P_{C_1} - k_a^- P_{O_1}, -k_b^+ [Ca^{2+}]_{ss}^3 P_{O_1} \\ &\quad + k_b^- P_{O_2} - k_c^+ P_{O_1} + k_c^- P_{C_2}, \\ \frac{dP_{O_2}}{dt} &= k_b^+ [Ca^{2+}]_{ss}^3 P_{O_1} - k_b^- P_{O_2}, \\ \frac{dP_{C_2}}{dt} &= k_c^+ P_{O_1} - k_c^- P_{C_2}.\end{aligned}$$

All the k -coefficients are given constants.

The following system describes the membrane current of calcium through the so-called L-type channels.

$$\begin{aligned}\frac{dC_0}{dt} &= \beta C_1 + \omega C_{Ca0} - (4\alpha + \gamma)C_0, \\ \frac{dC_1}{dt} &= 4\alpha C_0 + 2\beta C_2 + \frac{\omega}{b} C_{Ca1} - (\beta + 3\alpha + \gamma a)C_1, \\ \frac{dC_2}{dt} &= 3\alpha C_1 + 3\beta C_3 + \frac{\omega}{b^2} C_{Ca2} - (2\beta + 2\alpha + \gamma a^2)C_2, \\ \frac{dC_3}{dt} &= 2\alpha C_2 + 4\beta C_4 + \frac{\omega}{b^3} C_{Ca3} - (3\beta + \alpha + \gamma a^3)C_3, \\ \frac{dC_4}{dt} &= \alpha C_3 + gO + \frac{\omega}{b^4} C_{Ca4} - (4\beta + f + \gamma a^4)C_4, \\ \frac{dO}{dt} &= fC_4 - gO, \\ \frac{dC_{Ca0}}{dt} &= \beta' C_{Ca1} + \gamma C_0 - (4\alpha' + \omega)C_{Ca0}, \\ \frac{dC_{Ca1}}{dt} &= a\alpha' C_{Ca0} + 2\beta' C_{Ca2} + \gamma a C_1 - \left(\beta' + 3\alpha' + \frac{\omega}{b}\right) C_{Ca1}, \\ \frac{dC_{Ca2}}{dt} &= 3\alpha' C_{Ca1} + 3\beta' C_{Ca3} + \gamma a^2 C_2 - \left(2\beta' + 2\alpha' + \frac{\omega}{b^2}\right) C_{Ca2}, \\ \frac{dC_{Ca3}}{dt} &= 2\alpha' C_{Ca2} + 4\beta' C_{Ca4} + \gamma a^3 C_3 - \left(3\beta' + \alpha' + \frac{\omega}{b^3}\right) C_{Ca3}, \\ \frac{dC_{Ca4}}{dt} &= \alpha' C_{Ca3} + \gamma a^4 C_4 - \left(4\beta' + f' + \frac{\omega}{b^4}\right) C_{Ca4},\end{aligned}$$

$$\frac{dy}{dt} = \frac{y_{\infty} - y}{\tau_y}.$$

The coefficients α , β , α' , and β' are functions of the transmembrane potential V . The parameter γ , describing the transition from state normal to state Ca, is a function of the subspace Ca^{2+} concentration $[\text{Ca}^{2+}]_{\text{ss}}$. The gating variables of the closing variable y are functions of V , and the rest of the coefficients are constants. A detailed description of the channel is given in [21]. Intracellular calcium buffering is described by

$$\begin{aligned} \frac{d[\text{HTRPNCa}]}{dt} &= k_{\text{htrpn}}^+ [\text{Ca}^{2+}]_i ([\text{HTRPN}]_{\text{tot}} - [\text{HTRPNCa}]) \\ &\quad - k_{\text{htrpn}}^- [\text{HTRPNCa}], \\ \frac{d[\text{LTRPNCa}]}{dt} &= k_{\text{ltrpn}}^+ [\text{Ca}^{2+}]_i ([\text{LTRPN}]_{\text{tot}} - [\text{LTRPNCa}]) \\ &\quad - k_{\text{ltrpn}}^- [\text{LTRPNCa}], \end{aligned}$$

where the k -coefficients are constants.

Finally, the model describes the dynamics of five different intracellular ionic concentrations by the equations

$$\begin{aligned} \frac{d[\text{K}^+]_i}{dt} &= -[I_{\text{Kr}} + I_{\text{Ks}} + I_{\text{to}} + I_{\text{K1}}, \\ &\quad + I_{\text{Kp}} + I_{\text{Ca,K}} - 2I_{\text{NaK}}] \frac{A_{\text{cap}} C_{\text{sc}}}{V_{\text{myo}} F}, \\ \frac{d[\text{Ca}^{2+}]_i}{dt} &= \beta_i \left[J_{\text{xfer}} - J_{\text{up}} - J_{\text{trpn}}, \right. \\ &\quad \left. - (I_{\text{Ca,b}} - 2I_{\text{NaCa}} + I_{\text{p(Ca)}}) \frac{A_{\text{cap}} C_{\text{sc}}}{2V_{\text{myo}} F} \right], \\ \frac{d[\text{Ca}^{2+}]_{\text{ss}}}{dt} &= \beta_{\text{ss}} \left(J_{\text{rel}} \frac{V_{\text{JSR}}}{V_{\text{ss}}} - J_{\text{xfer}} \frac{V_{\text{myo}}}{V_{\text{ss}}} - I_{\text{Ca}} \frac{A_{\text{cap}} C_{\text{sc}}}{2V_{\text{myo}} F} \right), \\ \frac{d[\text{Ca}^{2+}]_{\text{JSR}}}{dt} &= \beta_{\text{JSR}} (J_{\text{tr}} - J_{\text{rel}}), \\ \frac{d[\text{Ca}^{2+}]_{\text{NSR}}}{dt} &= J_{\text{up}} \frac{V_{\text{myo}}}{V_{\text{NSR}}} - J_{\text{tr}} \frac{V_{\text{JSR}}}{V_{\text{NSR}}}. \end{aligned}$$

The ionic currents and the coefficient functions β_i , β_{ss} , and β_{NSR} are nonlinear functions of their respective variables, making all these equations nonlinear.

As noted above, the reader is referred to [5] for a full specification of the system.

3 Numerical methods

The models described in the previous section can be written as general initial-value problems of the form

$$\frac{dy}{dt} = f(y, t), \quad y(0) = y^0. \quad (2)$$

Many standard solvers for such initial-value problems can be derived from the formula

$$y(t^{n+1}) = y(t^n) + \int_{t^n}^{t^{n+1}} f(\tau, y(\tau)) d\tau, \quad (3)$$

where $y(t^n)$ is assumed to be known. The right-hand side function f is generally nonlinear, and y is unknown, so it is normally not possible to compute the integral exactly. Different approximations of the integral give rise to different numerical methods. The forward and backward Euler methods are derived by viewing the integrand $f(t, y)$ as constant, evaluated either at the left ($t = t^n$) or right ($t = t^{n+1}$) endpoint of the integral, respectively. Both of these methods are formulated in vector form, and in principle all components of the vector function f are integrated simultaneously and in the same manner.

The solvers considered in this paper differ from this normal structure in that the components of f are integrated separately, i.e., by

$$y_i(t^{n+1}) = y_i(t^n) + \int_{t^n}^{t^{n+1}} f_i(\tau, y(\tau)) d\tau, \quad (4)$$

and different approximation rules may be applied to the different components. In component form, all the equations in the models considered can be written in the form

$$\frac{dy_i}{dt} = f_i(y), \quad (5)$$

where $f_i(y)$ is component i of the nonlinear vector function $f(y)$. However, a significant portion of the equations can be written in the form

$$\frac{dy_i}{dt} = A(y_*)y_i + B(y_*), \quad (6)$$

where A and B are nonlinear functions, and y_* is a vector of components of y that does not include component i . For a system of m equations, y_* is a vector of at most length $m - 1$, containing various components y_j for $j \neq i$. An example of equations that can be put on this form is the gating variable equations (1), for which A and B depend only on the transmembrane potential V ; i.e., in this case $y_* = V$. We see that in a sequential solution algorithm, where each equation is stepped forward holding all other variables constant, (6) reduces to a linear ODE with constant coefficients.

The NSFD solvers are based on special treatment of the equations of the form (6). More precisely, the integral in (4) is approximated differently for these equations. The ODEs associated with the two models considered here are therefore divided into two parts, one part consisting of equations that can be written as (6), and one part consisting of the equations that must be kept in the fully nonlinear form (5). With a slight abuse of nomenclature, we refer to the group of equations of the form (6) as the linear part of the system, and the group of the form (5) as the nonlinear part. For the model by Courtemanche et al., the linear part consists of the variables $m, h, j, o_a, o_i, u_a, u_i, x_r, x_s, d, f, f_{Ca}, u, v$, and w , while the nonlinear part contains $V, [Na^+]_i, [K^+]_i, [Ca^{2+}]_i, [Ca^{2+}]_{rel},$ and $[Ca^{2+}]_{up}$. Similarly, for the model by Winslow et al. the variables $m, h, j, X_{Kr}, X_{Ks}, X_{to}, Y_{to}, P_{C_1}, P_{O_1}, P_{O_2}, P_{C_2}, C_0, C_1, C_2, C_3, C_4, O, C_{Ca0}, C_{Ca1}, C_{Ca2}, C_{Ca3}, C_{Ca4}, [HTRPNCa],$ and $[LTRPNCa]$ form the linear part, and $V, [K^+]_i, [Ca^{2+}]_i, [Ca^{2+}]_{ss}, [Ca^{2+}]_{JSR},$ and $[Ca^{2+}]_{NSR}$ form the nonlinear part.

It will be useful to compare the performance of the NSFD solvers to that of the forward Euler method. Forward Euler is a standard solver in the sense that all components of the integral are approximated in the same manner, and the method can be derived from (3). However, because of its simple, explicit structure, the method can also be derived from the component-based version (4). As described above, the integral in (4) is approximated by inserting the values t^n, y^n in f and treating the integrand as constant. The linear components are stepped forward by

$$y_i^{n+1} = y_i^n + \Delta t A(y^n) y_i^n + B(y^n), \quad (7)$$

and the nonlinear components are updated from

$$y_i^{n+1} = y_i^n + \Delta t f(y^n). \quad (8)$$

The NSFD solver derived by Rush and Larsen [14] is based on a special treatment of the linear part of the equations. Inserting the values t_n, y_n in the expressions for A and B and using (4) and (6), we get

$$y_i(t^{n+1}) = y_i(t^n) + \int_{t^n}^{t^{n+1}} (Ay_i + B) d\tau. \quad (9)$$

Since A and B are treated as constant, the integral can be evaluated analytically, and we obtain

$$y_i^{n+1} = \left(y_i^n + \frac{B}{A} \right) e^{A\Delta t} - \frac{B}{A}. \quad (10)$$

In the original formulation of the Rush–Larsen method, this update formula is used for all components of the linear part of the system, and the forward Euler method is used for the nonlinear components.

One can suggest a variety of modifications of the original Rush–Larsen method. A possible weakness of the method is that the forward Euler method is used for the nonlinear part of the system. In the case that this part of the system is stiff, this will lead to severe restrictions on the time step. An improvement of the method may therefore be to solve some or all of these equations implicitly with for instance the backward Euler method. Although this discretization leads to nonlinear algebraic equations that must be solved with an iterative technique such as Newton’s method, this process is fairly simple because only scalar versions of the Newton iteration must be applied. Using the backward Euler method, (4) becomes

$$y_i^{n+1} = y_i^n + \Delta t f_i(y_*, y_i^{n+1}), \quad (11)$$

where again y_* is a vector containing components of y , but not y_i itself. This equation is solved by Newton’s method. Combining this formula for the nonlinear equations with (10) for the linear parts, we get a first-order algorithm that is slightly more computationally expensive than the original Rush–Larsen method, but it may have improved stability properties and hence be ultimately more efficient. We also test a component-wise version of backward Euler, where (11) is applied to both the linear and nonlinear parts of the system.

Another weakness of the Rush–Larsen method is that although the accuracy is improved compared to the forward Euler method, the overall accuracy is still only first order. Although this is hard to avoid for a sequential treatment of the equations like the one we consider here, second-order accuracy may be obtained by performing each step as a series of two steps. First, we use the Rush–Larsen method with implicit treatment of the nonlinear part (given in (11)) to compute the value of all variables at time $t = t_n + \Delta t/2$; i.e., all components of $y_i^{n+1/2}$. These midpoint values may then be used to compute the constants in the update (10). If we also use an implicit midpoint discretization of the equations in the nonlinear part, we obtain an algorithm that is second-order accurate. The two steps of the algorithm can be summarized as follows.

(i) Compute a midpoint-approximation of all variables using

$$y_i^{n+1/2} = \left(y_i^n + \frac{B}{A} \right) e^{A\Delta t/2} - \frac{B}{A}$$

for the linear equations and

$$y_i^{n+1/2} = y_i^n + \frac{\Delta t}{2} \left(f_i(y_*^n, \frac{1}{2}(y_i^n + y_i^{n+1/2})) \right).$$

for the nonlinear ones.

(ii) Use the midpoint values to compute new values of A and B , and integrate the linear part of the system for a full time step with (10). The nonlinear equations are then updated with the implicit midpoint formula,

$$y_i^{n+1} = y_i^n + \Delta t \left(f_i(y_*^{n+1/2}, \frac{1}{2}(y_i^n + y_i^{n+1})) \right).$$

Because it consists of two normal steps, this step is almost exactly twice as expensive as the NSFD method based on a single step of (10) and the backward Euler method. Numerical experiments must be performed to determine whether this extra cost is outweighed by the increased accuracy or stability of the method.

As an alternative second-order NSFD method, we discretize the equations in the nonlinear part using the following singly diagonally implicit Runge–Kutta (SDIRK) method,

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ \hline 1 & 1-\gamma & \gamma \\ \hline & 1-\gamma & \gamma \end{array},$$

where $\gamma = \frac{2-\sqrt{2}}{2}$. As with the backward Euler method, this SDIRK method is L -stable and has the property of stiff decay [18].

4 Numerical experiments

The cell model ODE systems are known to be stiff. In general it is difficult to give a precise, quantitative definition of stiffness; see e.g. [8, 18, 22] for discussions. Stiffness is often better defined in qualitative terms, relating the performance of explicit ODE solvers to that of implicit ODE solvers. We will say that a problem is stiff if the step size required for stability of a given numerical method is much smaller than the step size dictated by accuracy requirements. This definition illustrates the important

Cell model	λ_{\min}
Beeler-Reuter	-82.15
Luo-Rudy phase 1	-174
Luo-Rudy phase 2	-365
Winslow et al.	-19167
Courtemanche et al.	-106

Table 1. Largest negative eigenvalues occurring in some popular electrophysiology models for cardiac cells.

fact that stiffness is not only a property of the ODE system, but also depends on the requested accuracy.

Stiffness is often related to the existence of a wide range of time scales in a problem. For a linear homogeneous problem of the form

$$\frac{dy}{dt} = Ay,$$

the time scales of the solution are characterized by the eigenvalues of A . A more quantitative measure of stiffness can sometimes be obtained from the distribution of the eigenvalues of A . In order to have a stable solution all eigenvalues of A must have negative real part, and a stiff problem can sometimes be characterized by the eigenvalues being distributed over a large interval of the negative real axis.

For nonlinear problems it is even harder to give a meaningful quantitative definition of stiffness. However, (local) stiffness of these problems will often be related to the eigenvalues of the Jacobian of the right-hand side function f . Similar to the linear case, a wide distribution of eigenvalues corresponds to the existence of multiple time scales, and thereby usually a stiff system. By this definition, because the Jacobian is not constant, a nonlinear problem may be stiff in some intervals and non-stiff in others. In an attempt to characterize the stiffness of various cell model ODE systems, we have evaluated the largest negative Jacobian eigenvalue occurring in a typical simulation. We have only considered the real part of the eigenvalues, and the results are given in Table 1. For the Courtemanche and Winslow cell models, the maximum (largest positive) and minimum (largest negative) eigenvalues are plotted versus time (for the course of one action potential) in Figure 1.

We see that among these models the canine model by Winslow et al. stands out as having extremely stiff regions. Based on the eigenvalues it is possible to estimate the step size required for stability of various methods. The step size Δt must be chosen so that the complex number $\lambda\Delta t$ is inside the absolute stability domain of the method for all eigenvalues λ . For a definition of the absolute stability domain we refer to e.g., [8, 18]. Using the forward Euler method as an example, the stability domain of this method is a circle in the complex plane with center in $(-1, 0)$ and radius one. From the eigenvalues in Table 1 we see that for $\lambda\Delta t$ to be inside this region, we must

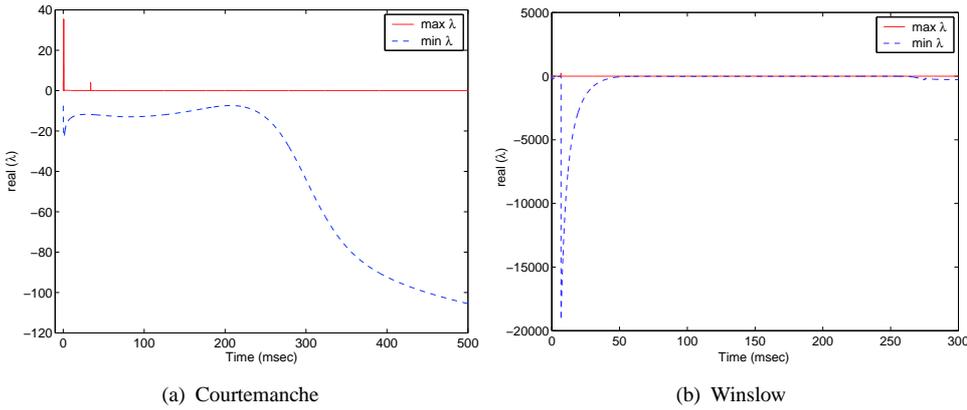


Figure 1. Maximum and minimum eigenvalues for the Courtemanche and Winslow cell models.

have $\Delta t \approx 0.0189$ msec for the Courtemanche et al. model and $\Delta t \approx 0.000104$ msec for the model by Winslow et al. Therefore, although these models are comparable in terms of physiological accuracy, they are markedly different in terms of stiffness, resulting in much stricter step size requirements for a constant step-size solution of the model by Winslow et al.

In our experiments, we determine the largest allowable time step based on accuracy. To determine the error in a numerical solution, we compute a reference solution, V , generated using the Matlab solver `ode15s`, to serve as the exact solution. Matlab's `ode15s` is a variable step size stiff solver based on a family of linear multistep methods known as *numerical differentiation formulas* [23]. The absolute and relative tolerances were set to $1e-10$ ($atol=rtol=1e-10$) in generating the reference solution. To check convergence of the reference solution, it was compared to solutions generated by `ode15s` with $atol=rtol=1e-11$, and `ode45` (a Matlab solver based on the Dormand-Prince pair) with $atol=rtol=1e-10$ and $atol=rtol=1e-11$; we found the solutions to be identical up to 7 digits. The approximate solutions for the transmembrane potential \hat{V} obtained with the numerical methods under investigation were compared to the reference solution using the following measures: a relative root-mean-square (RRMS) error [24] given by

$$RRMS = \sqrt{\frac{\sum_{i=1}^N (\hat{V}_i - V_i)^2}{\sum_{i=1}^N V_i^2}}, \quad (12)$$

and a measure of the global error

$$e_{global} = \max(|\hat{V} - V|). \quad (13)$$

The allowable error was determined in terms of the RRMS error. Specifically, for all

methods the step size was increased until the RRMS error first exceeded 5%. This level of accuracy is chosen to reflect typical accuracies required in biomedical engineering applications. We note that allowable step size does not completely determine the efficiency of a given method because each method has a different computational cost per step.

Table 2 gives the results for the forward Euler method, the NSFD method using forward Euler on the nonlinear components (Rush–Larsen method), and the component-wise version of backward Euler. Table 3 gives the results for three NSFD methods using backward Euler, implicit midpoint, and SDIRK, respectively, on the nonlinear components. In Tables 4 and 5 we verify that the NSFD methods with implicit midpoint and SDIRK solvers are indeed second order; i.e., as Δt is halved, the (asymptotic) error decreases by a factor of 4. (The *rate* of convergence is calculated as \log_2 of this factor.) The CPU times for all six methods are given in Table 6. Figures 2 and 3 compare the solutions for the transmembrane potential using the various numerical methods discussed here with the step sizes Δt given in Tables 2 and 3.

		Forward Euler	NSFD w/ FE	Component-wise BE
Courtemanche	Δt	0.0194	0.345	0.408
	RMS	0.0023	0.0497	0.0498
	e_{global}	1.9396	37.4717	46.3757
Winslow	Δt	0.000107	0.00028	0.000125
	RMS	0.000776	0.0486	0.0405
	e_{global}	0.0991	6.0760	5.0574

Table 2. Maximum step sizes (in msec) for various methods.

		NSFD w/ BE	NSFD w/ Impl Midpt	NSFD w/ SDIRK
Courtemanche	Δt	0.541	0.80	0.89
	RMS	0.0499	0.0490	0.0493
	e_{global}	55.5594	26.1142	26.9269
Winslow	Δt	0.00014	0.00028	0.00123
	RMS	0.0474	0.0478	0.0497
	e_{global}	5.9119	5.9667	6.2596

Table 3. Maximum step sizes (in msec) for various methods.

Step size	RRMS error	Rate	Global error	Rate
0.0025	0.2453		29.3822	
0.00125	0.0494	2.3120	6.2208	2.2398
0.000625	0.0123	2.0059	1.5483	2.0064
0.0003125	0.0032	1.9425	0.4050	1.9347
0.00015625	0.0008414	1.9272	0.1060	1.9339

Table 4. Error values for the Winslow model illustrating that the NSFD with the Implicit Midpoint method is second order.

Step size	RRMS error	Rate	Global error	Rate
0.001	0.0321		4.0449	
0.0005	0.0081	1.9866	1.0149	1.9948
0.00025	0.0021	1.9475	0.2645	1.9400
0.000125	0.00054621	1.9429	0.0688	1.9428

Table 5. Error values for the Winslow model illustrating that the NSFD with the SDIRK method is second order.

Method	Courtemanche	Winslow
FE	3.11	48.37
NSFD w/ FE	0.20	27.37
Component-wise BE	0.16	100.05
NSFD w/ BE	0.14	58.61
NSFD w/ Implicit Midpoint	0.14	25.77
NSFD w/ SDIRK	0.15	38.98

Table 6. CPU times (in seconds) for the various methods

5 Discussion

Cellular reactions in the heart are commonly modelled by systems of ODEs. Although these models share many common goals and features, their stiffness properties may be very different. This dramatically affects the choice of the most efficient numerical method for their solution.

We have investigated the performance of 4 NSFD methods as well as standard forward and backward Euler methods. Performance was measured in terms of CPU time required to achieve a given accuracy. In this study an RRMS error (12) of no greater than 5% was deemed acceptable.

Using this error criterion, all the NSFD methods perform much better than the standard forward Euler method for the Courtemanche et al. atrial cell model. All the

NSFD methods require about 15–20 times less CPU time than the forward Euler method. In general we find that the errors produced by the forward Euler method are *too small*; i.e., larger step sizes with the method are impossible without inducing instability. This is one telltale sign of stiffness. The performances of the NSFD methods and the component-wise backward Euler method are comparable. In this case, the NSFD method with forward Euler (Rush–Larsen method) may be preferable because this method can be implemented in an explicit fashion.

For the Winslow et al. ventricular cell model, the conclusions are less straightforward. The forward Euler method is outperformed by all the NSFD methods except for NSFD with backward Euler. The NSFD method using forward Euler (Rush–Larsen) and the NSFD method using implicit midpoint show the best performance, requiring about 2 times less CPU time than the forward Euler method. However, the actual RRMS and global errors for forward Euler are significantly smaller than all other methods. Hence in some sense much better results can be obtained with not much more cost; this may be an acceptable tradeoff in some situations. It also appears clear from this model that the properties of L -stability and stiff decay (shared by the backward Euler and SDIRK method) do not appear to enhance performance. The A -stability of the implicit midpoint method seems to suffice.

Although not shown explicitly, the Winslow et al. model has more dynamic (or transient) behaviour than the Courtemanche et al. model; i.e., the solutions of the ODEs have relatively more regions of rapid variation. This dynamic behaviour necessitates smaller step sizes for its resolution independent of other considerations. Hence the Winslow et al. model generally requires smaller time steps than the Courtemanche et al. In such cases, explicit methods generally outperform implicit methods because they incur less cost per step. We can see the effects of this in Tables 2, 3, and 6, where the smaller step sizes lead to smaller global errors and less discrepancy in CPU time between forward Euler and the rest of the methods. The importance of higher order for the NSFD methods is also less clear in this case, at least when comparing orders 1 and 2.

It is important to note that these conclusions are based on a specific (although generally accepted) definition of what constitutes “acceptable error” for these types of simulations. However, changes to this definition may result in different conclusions. In particular, the error in the different simulations may be too large by other measures of error, such as the global error (13), despite the efficiency gains.

We also note that we have not contemplated integration with variable step sizes. Such a possibility could again change the conclusions. Variable step sizes are not generally used for such models because in a realistic simulation of the heart, the cell model ODEs must be coupled to PDEs that describe the macroscopic electrical properties of the tissue. The models are often integrated using fixed step sizes in an operator splitting formulation; hence results from a study with variable step sizes would have limited generalizability to such situations.

References

- [1] D. Noble. A modification of the hodgkin-huxley equations applicable to purkinje fibre action and pace-maker potentials. *Journal of Physiology*, 160:317–352, 1962.
- [2] D. Di Francesco and D. Noble. A model of the cardiac electrical activity incorporating ionic pumps and concentration changes - simulations of ionic currents and concentration changes. *Phil. Trans. R. Soc. Lond.*, B307:353–398, 1985.
- [3] C.-H. Luo and Y. Rudy. A model of the ventricular cardiac action potential - depolarisation, repolarisation and their interaction. *Circulation Research*, 68:1501–1526, 1991.
- [4] C.-H. Luo and Y. Rudy. A dynamic model of the cardiac ventricular action potential - simulations of ionic currents and concentration changes. *Circulation Research*, 74:1071–1097, 1994.
- [5] R. L. Winslow, J. Rice, S. Jafri, E. Marban, and B. O'Rourke. Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, II. model studies. *Circulation Research*, 1999.
- [6] A. Nygren, C. Fiset, L. Firek, J.W. Clark, D.S. Lindblad, R.B. Clark, and W.R. Giles. Mathematical model of an adult human atrial cell: The role of K^+ currents in repolarization. *Circulation Research*, 82:63–81, 1998.
- [7] M. Courtemanche, R. J. Ramirez, and S. Nattel. Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. *American Journal of Physiology*, 275:H301–H321, 1998.
- [8] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II, Stiff and Differential Algebraic Problems*. Springer-Verlag, 1991.
- [9] W. T. Miller and D. B. Geselowitz. Simulation studies of the electrocardiogram: I. the normal heart. *Circ. Res.*, 43:301–315, 1978.
- [10] L. Tung. *A Bi-domain model for describing ischemic myocardial D-C potentials*. PhD thesis, MIT, Cambridge, 1978.
- [11] J. Sundnes, G. T. Lines, and A. Tveito. Efficient solution of ordinary differential equations modeling electrical activity in cardiac cells. *Mathematical Biosciences*, 172:55–72, 2001.
- [12] B. J. Roth. A mathematical model of make and break electrical stimulation of cardiac tissue by a unipolar anode or cathode. *IEEE Transactions on Biomedical Engineering*, 42(12):1174–1184, 1995.
- [13] H. I. Saleheen and K. T. Ng. A new three-dimensional finite-difference bidomain formulation for inhomogeneous anisotropic cardiac tissues. *IEEE Transactions on Biomedical Engineering*, 45(1):15–25, 1998.
- [14] S. Rush and H. Larsen. Practical algorithm for solving dynamic membrane equations. *IEEE Transactions on Biomedical Engineering*, 25(4):389–392, 1978.
- [15] Z. L. Qu, J. N. Weiss, and A. Garfinkel. From local to global spatiotemporal chaos in a cardiac tissue model. *Physical Review E*, 61(1):727–732, 2000.
- [16] N. Virag, V. Jacquemet, and C. S. Henriquez. Study of atrial arrhythmias in a computer model based on magnetic resonance image of human atria. *Chaos*, 12(3):754–763, 2002.
- [17] K. H. W. J. ten Tusscher, D. Noble, P. J. Noble, and A. V. Panfilov. A model for human ventricular tissue. *American Journal of Physiology - Heart and Circulatory Physiology*, 286(4):H1573–89, 2004.
- [18] U. M. Ascher and L. R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM, 1998.
- [19] R. J. Spiteri and M. C. MacLachlan. An efficient non-standard finite difference scheme for an ionic model of cardiac action potentials. *Journal of Difference Equations and Applications*, 9(12):1069–1082, 2003.
- [20] J. Sundnes, G. T. Lines, and A. Tveito. An operator splitting method for solving the bidomain equations coupled to a volume conductor model for the torso. *Mathematical Biosciences*, 194(2):233–248, 2005.
- [21] M. Saleet Jafri, J. Jeremy Rice, and Raimond L. Winslow. Cardiac calcium dynamics: The roles of ryanodine receptor adaptation and sarcoplasmic reticulum load. *Biophysical Journal*, 74:1149–1168, 1998.
- [22] J. D. Lambert. *Numerical methods for ordinary differential systems*. John Wiley & Sons Ltd., Chichester, 1991. The initial value problem.
- [23] L. F. Shampine and M. W. Reichelt. The Matlab ODE Suite. *SIAM Journal on Scientific Computing*, 18:1–22, 1997.
- [24] B. M. Horacek, J. W. Warren, P. Stovicek, and C. L. Feldman. Diagnostic accuracy of derived versus standard 12-lead electrocardiograms. *Journal of Electrocardiology*, 33:Suppl:155–60, 2000.

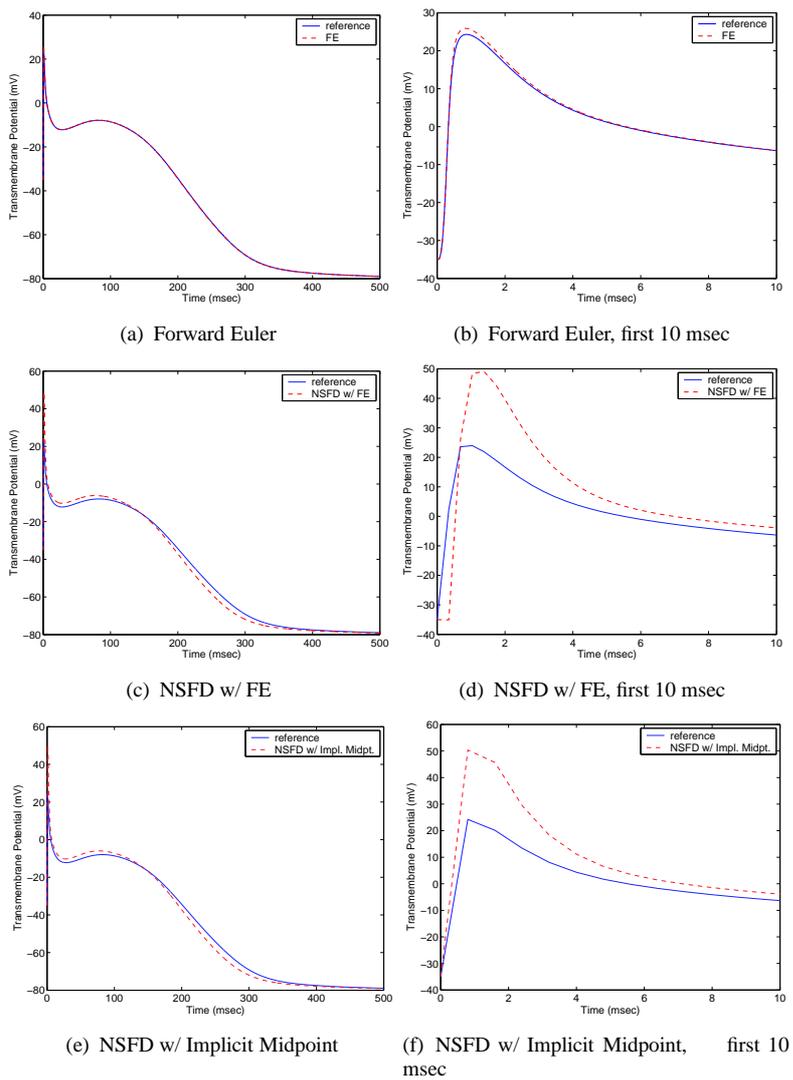


Figure 2. Comparison of computed transmembrane potential for various numerical methods; model of Courtemanche et al.

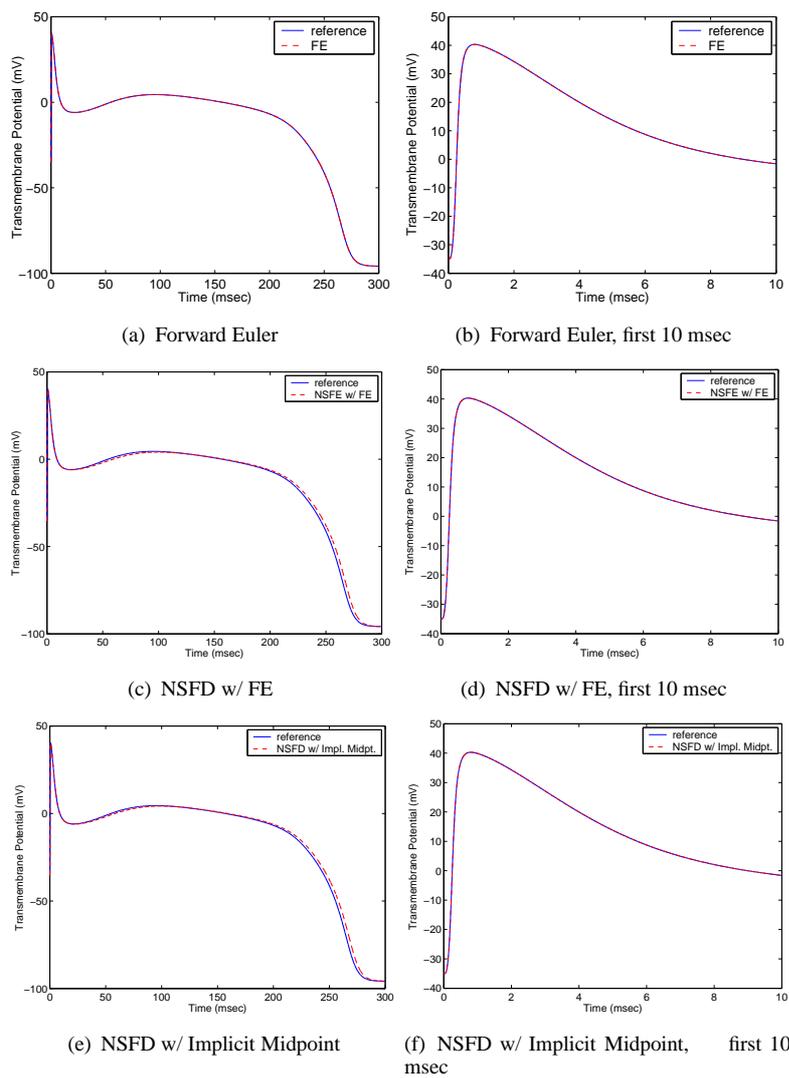


Figure 3. Comparison of computed transmembrane potential for various numerical methods; model of Winslow et al.