Centrum voor Wiskunde en Informatica

REPORTRAPPORT

MAS

Modelling, Analysis and Simulation

Modelling, Analysis and Simulation

Numerical solution of mixed gradient-diffusion equations modelling axon growth

B. Lastdrager

REPORT MAS-R0203 JANUARY 31, 2002

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

Numerical Solution of Mixed Gradient-Diffusion Equations Modelling Axon Growth

Boris Lastdrager
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT
In the current paper a numerical approach is presented for solving a system of coupled gradient-diffusion equations which acts as a first model for the growth of axons in brain tissue. The presented approach can be applied to a much wider range of problems, but we focus on the axon growth problem. In our approach time stepping is performed with a Rosenbrock solver with approximate matrix factorization. For the Jacobian an approximation is used that simplifies the solution of the coupled parabolic and gradient equations. A possible complication in the implementation of source terms is noted and a criterion that helps to avoid it is presented.

2000 Mathematics Subject Classification: 65M20, 65L06, 92C20.
Keywords and Phrases: diffusion equations, gradient equations, Rosenbrock methods, axon growth, neurobiology.
Note: This work was performed under a research contract with The Netherlands Organization for Scientific Research (NWO) and was carried out under CWI-projects MAS1.1 "Applied Analysis and Scientific Computing" and MAS2.1 "Computational Fluid Dynamics".

1. Introduction

In biological experiments it is often observed that in the initial growth phase axons approach each other to form a bundle. Then, in the intermediate phase the axons grow jointly towards a remotely located concentration of so-called targets. Once the axons have sufficiently approached the targets they debundle and attach to different individual targets. The ultimate goal of our research is to develop a numerical modelling tool that can establish which physical processes are necessary and which are not to predict above described behavior of bundling and debundling.

It is known that one of the mechanisms by which axons are guided to their targets relies on the diffusion of chemo-attractant molecules from the target through the tissue. The concentration of target derived chemo-attractant is largest near the targets and decays away from the targets. The growth cones at the tip of the axons can sense and follow the gradient in concentration to reach the targets [3]. Furthermore, it is also known that axons are repelled by diffusible molecules secreted by tissues the axons need to grow away from.

Several mechanisms have been suggested to explain that axons approach each other in the initial growth phase. Random movement might bring the axons together, repulsive signals from surrounding cells could do the same or diffusible molecules that the axons secrete may guide them towards each other. Following [5], we focus on the latter mechanism; we assume that the axons growth cones emit a diffusible chemo-attractant to which the other axons growth cones are sensitive.

Furthermore we assume that the growth cones can secrete diffusible substances that act as chemorepellant to the other axons. In particular we allow the rate of secretion of chemorepellant to be dependent on the concentration of target derived chemo-attractant. This enables the axons to repel one another progressively stronger as they approach the target area, ultimately leading to debundling.

In the current paper we restrict the interaction between axons to growth-cone to growth-cone interaction. I.e., axons only secrete chemicals from their growth cones and can only sense with their growth cones.

2. Biological model

In this paper we consider essentially the same model as in [5]. We consider a fixed spatial domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$, measured in millimeters. The model assumes that both the targets and the growth cones secrete diffusible chemical compounds to which the growth cones are sensitive. In particular, the targets release an attractant and the growth cones release both an attractant and a repellant. The rate of release of repellant is dependent on the concentration of target derived attractant. This allows the axons to repel each other once they have reached the target area, where there is a high concentration of target derived attractant.

The time evolution of the concentration fields is governed by diffusion equations of the form

$$\frac{\partial u}{\partial t} = d\Delta u - \kappa u + s,$$

where u is the concentration of a diffusible chemical compound. The diffusion constant d measures how quickly the compound diffuses through the medium, the loss constant κ measures the rate of absorption of the medium and the source term s contains the release of mass by the axons and targets. The growth of the axons is governed by gradient equations of the form

$$\frac{d\mathbf{r}}{dt} = \lambda \nabla u(\mathbf{r}(t), t), \tag{2.1}$$

where $\mathbf{r}$ is the position of an axon. The parameter λ measures the sensitivity of the axon to gradients in the concentration field u.

A property of (2.1) is that in a steady solution field u, limit points for $\mathbf{r}$ coincide with a maximum, minimum or saddle point in u. Depending on the sign of λ, maxima or minima can either be stable or unstable limit points. For $\lambda > 0$, $\mathbf{r}$ will move in the direction of the gradient of u, i.e. it will move towards a maximum of u. Once at a maximum, a small displacement in $\mathbf{r}$ will cause $\mathbf{r}$ to move back towards the maximum. Hence, for $\lambda > 0$ maxima in u are stable limit points for $\mathbf{r}$. Likewise, for $\lambda < 0$ minima in u are stable limit points for $\mathbf{r}$. For $\lambda < 0$ maxima are unstable limit points for $\mathbf{r}$ since a small displacement from a maximum will cause $\mathbf{r}$ to move away from the maximum. Likewise, for $\lambda > 0$ minima in u are unstable limit points for $\mathbf{r}$. Saddle points are always unstable since a small displacement from the saddle point can place $\mathbf{r}$ at a new position where the gradient in u can either point towards or away from the saddle point.

For the current model of cone derived attractant and repellant and target derived repellant the growth of the axons is governed by the following set of coupled gradient-diffusion equations:

$$\begin{aligned}
\frac{\partial u(\mathbf{x},t)}{\partial t} &= d_u \Delta u(\mathbf{x},t) - \kappa_u u(\mathbf{x},t) + \sigma_u \sum_{\beta} s(\mathbf{x} - \mathbf{T}_\beta), \\
\frac{\partial v(\mathbf{x},t)}{\partial t} &= d_v \Delta v(\mathbf{x},t) - \kappa_v v(\mathbf{x},t) + \sigma_v \sum_{\alpha} s(\mathbf{x} - \mathbf{r}_\alpha(t)), \\
\frac{\partial w(\mathbf{x},t)}{\partial t} &= d_w \Delta w(\mathbf{x},t) - \kappa_w w(\mathbf{x},t) + \sum_{\alpha} \sigma_w \left(u(\mathbf{x},t)\right) s(\mathbf{x} - \mathbf{r}_\alpha(t)), \\
\frac{d\mathbf{r}_\alpha(t)}{dt} &= \lambda_u \nabla u(\mathbf{r}_\alpha(t),t) + \lambda_v \nabla v(\mathbf{r}_\alpha(t),t) - \lambda_w \nabla w(\mathbf{r}_\alpha(t),t),
\end{aligned} \tag{2.2}$$

together with homogenous Dirichlet boundary conditions on u, v, and w.

Here t denotes time, $\mathbf{x} = (x, y)$ denotes a position in two dimensional space, u, v and w denote the concentrations of respectively, target derived attractant, cone derived attractant and cone derived repellant. Furthermore, d_u, d_v and d_w are the corresponding diffusion coefficients and κ_u, κ_v and κ_w are loss coefficients due to absorption in the tissue. The positions of the targets and growth cones are denoted by $\mathbf{T}_\beta$ and $\mathbf{r}_\alpha$, respectively, where β ranges from 1 to the number of targets and α ranges from 1 to the number of cones. The coefficients σ_u, σ_v and σ_w denote the rate of release of the different chemical compounds. The term $s(\mathbf{x})$ denotes a localised symmetric source term function with maximum at $\mathbf{x} = \mathbf{0}$. Finally, the positive coefficients λ_u, λ_v and λ_w measure the sensitivity of the cones to the gradients in the corresponding chemical concentration fields.

Note that in the gradient equation λ_u and λ_v both enter with a positive sign while λ_w enters with a negative sign. This represents the fact that both u and v act as attractants to the cones while w acts as a repellant. A positive λ causes $\mathbf{r}_\alpha$ to grow towards a maximum in the corresponding concentration field while a negative λ causes $\mathbf{r}_\alpha$ to grow away from a maximum.

Example
In Figure 1 the stationary solution for u is shown for a configuration with 5 targets located at $y = 0.25$ and $x = -0.25, -0.125, 0, 0.125, 0.25$. In Figures 2a-f the cones growth trajectories are displayed corresponding to the stationary target derived attractant field shown in Figure 1 with initial cone positions given by $y = -0.25$ and $x = -0.25, -0.125, 0, 0.125, 0.25$. The trajectories are shown at different time levels between $t = 0$ and $t = 2000$, measured in seconds, together with contour lines for the net gradient field $\lambda_u u + \lambda_v v - \lambda_w w$.

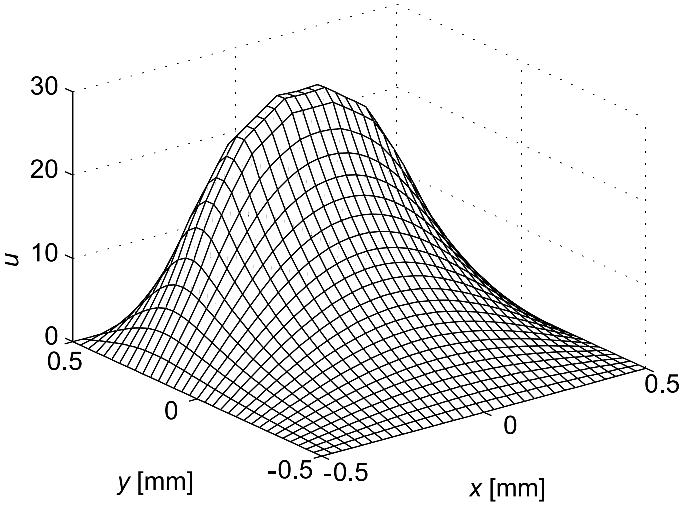


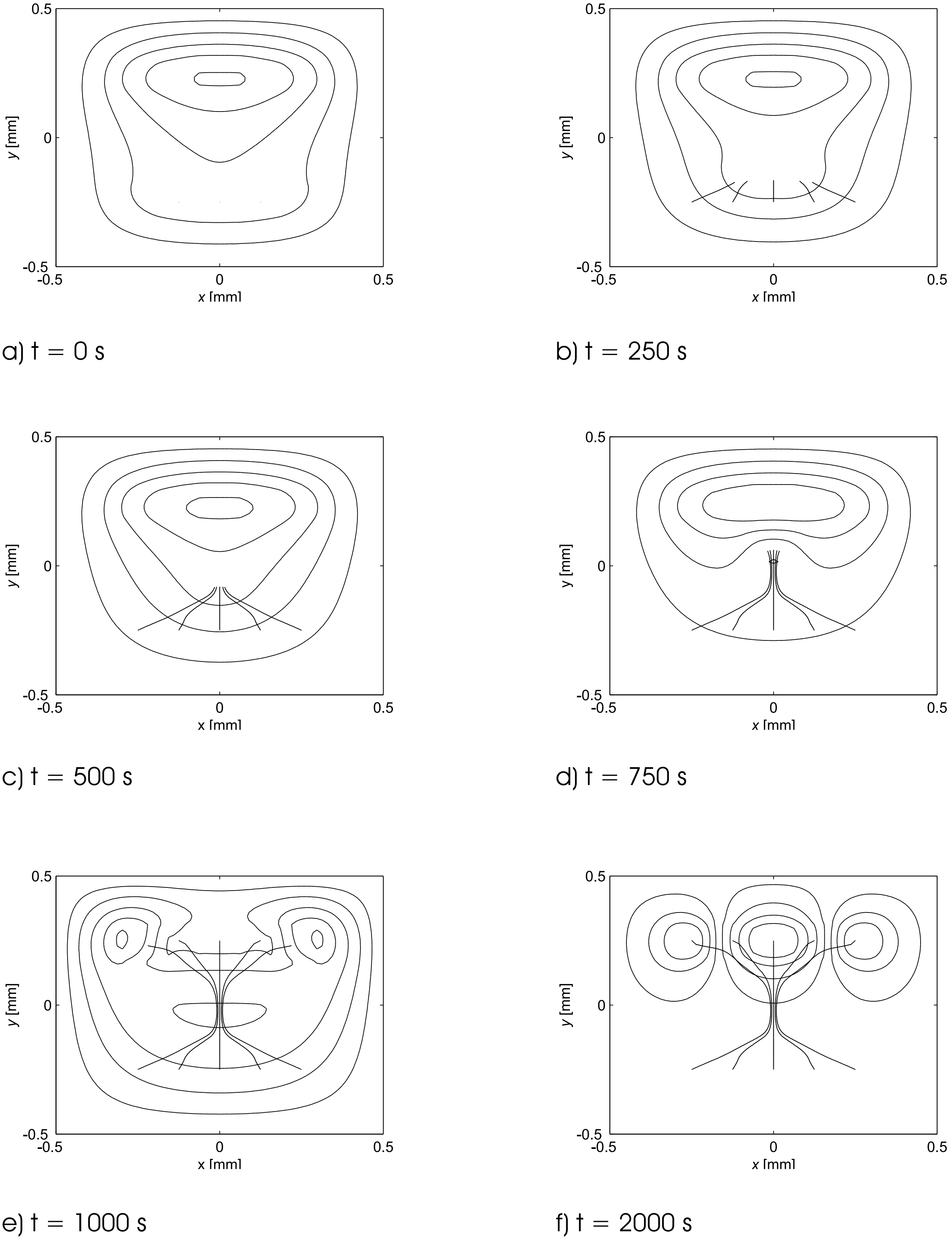
Figure 1: Stationary solution for u

Since the axons grow in the direction of the gradient in $\lambda_u u + \lambda_v v - \lambda_w w$, the direction of growth of a cone is always perpendicular to the contour lines. This does not imply that the growth paths should be perpendicular to the contour lines. The paths represent the growth of the axons at earlier times and need not be perpendicular to the contour lines at the current time.

Figures 1 and 2 are based on the following parameters:

$$\begin{aligned} d_u &= d_v = d_w = 10^{-4}, \\ \kappa_u &= \kappa_v = \kappa_w = 10^{-4}, \\ \lambda_u &= 10^{-5}, \\ \lambda_v &= 5 \cdot 10^{-6}, \\ \lambda_w &= 3.75 \cdot 10^{-5}, \\ \sigma_u &= \sigma_v = 3 \cdot 10^{-3}. \end{aligned}$$

These agree with the biological order estimates given in [8], but the used values were tuned to get a clear pattern of bundling and debundling. Following [5], the dependence of σ_w on $u(\mathbf{x}, t)$ is modelled



a) t = 0 s

b) t = 250 s

c) t = 500 s

d) t = 750 s

e) t = 1000 s

f) t = 2000 s

Figure 2: Axon growth trajectories together with contour lines for the net gradient field $\lambda_u u + \lambda_v v - \lambda_w w$

by the following relationship,

$$\sigma_w = \frac{3 \cdot 10^{-3} u^2}{u^2 + \rho^2}, \quad \rho = 25.$$

For the source function $s(\mathbf{x})$ we have used

$$\begin{aligned} s(\mathbf{x}) &= s_1(x) s_1(y), \\ s_1(\varphi) &= \begin{cases} 0, & |\varphi| > l. \\ 1 - \frac{|\varphi|}{l}, & |\varphi| \le l. \end{cases} \end{aligned}$$

Here l represents the radius of the axons' growth cone which we have taken $l = 0.15$ mm, which is quite large. A realistic estimate would be $l = 0.02$ mm, but we have taken the larger value to get convergence on relatively coarse grids; the model is not significantly altered by this choice.

3. Spatial Discretization

To evolve the governing system (2.2) in time, we employ a method of lines scheme (MOL). This implies that we first discretize the spatial operators on a spatial grid. This transforms the set of PDEs for the field quantities u, v, w into a large set of ODEs in time, i.e., we get an ODE for every spatial point. In the current paper the resulting set of ODEs together with the ODEs for the growth cone positions $\mathbf{r}_\alpha$ are solved with a Rosenbrock time stepping technique, as is explained in the next section.

The set of equations (2.2) presents a numerical challenge due to the presence of the equations for the growth cone positions $\mathbf{r}_\alpha$. Without these, the set could be solved in a straightforward manner with existing numerical techniques for PDEs. In its present form the model requires the solution of field equations coupled to particle equations. In principle it might be feasible to reformulate the field equations into a particle form or to rewrite the particle equations into field equations to obtain either a pure particle or a pure field problem, we however choose to apply a mixed field/particle setup. This approach has as an advantage in that both the field equations and the particle equations are tackled in a natural, efficient manner.

Consider a uniform spatial grid Ω_h, with mesh width $h = 1/(N+1)$, consisting of grid cells

$$\Omega_{i,j} = \{(x,y) | x_{i-1} \le x \le x_i, y_{j-1} \le y \le y_j\},$$

where $x_i = ih - 0.5$, $y_j = jh - 0.5$ and $i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, N$. We approximate the spatially dependent function $u(\mathbf{x}, t)$ with a grid function $u_h(t)$ such that at node (x_i, y_j) the grid function u_h has the value $u_{i,j}(t) \approx u(x_i, y_j, t)$. The spatial differential operators are approximated with second order central finite difference operators denoted by Δ_h and ∇_h. After discretization the ODEs for the axon positions read

$$\frac{d\mathbf{r}_\alpha(t)}{dt} = \lambda_u P_h(\mathbf{r}_\alpha(t)) \nabla_h u_h(t) + \lambda_v P_h(\mathbf{r}_\alpha(t)) \nabla_h v_h(t) - \lambda_w P_h(\mathbf{r}_\alpha(t)) \nabla_h w_h(t),$$

where $\mathbf{r}_\alpha(t)$ now represents the solution to the spatially discretized equation. To avoid excessive indices we do not replace $\mathbf{r}_\alpha(t)$ by $\mathbf{r}_{\alpha,h}(t)$. Note the appearance of $P_h(\mathbf{r}_\alpha(t))$. This is an interpolation operator which satisfies

$$P_h(\mathbf{r}_\alpha(t)) \nabla_h u_h(t) \approx \nabla u(\mathbf{r}_\alpha(t), t),$$

provided $\nabla_h u_h(t)|_{i,j} \approx \nabla u(x_i, y_j, t)$. The interpolation that we use is simple bilinear interpolation, combined with second order central differences for ∇_h, hence we have

$$P_h(\mathbf{r}_\alpha(t)) \nabla_h u_h(t) = \nabla u(\mathbf{r}_\alpha(t), t) + O(h^2).$$

Bilinear interpolation of the differences is sufficient to guarantee that $d\mathbf{r}_\alpha(t)/dt$ is continuous across cell interfaces and hence yields smooth trajectories for $\mathbf{r}_\alpha$. Note that with bilinear interpolation $d^2\mathbf{r}_\alpha(t)/dt^2$ does not exist across cell interfaces. This implies that bilinear interpolation is not to be used in conjunction with a higher order time integration method.

4. The Rosenbrock method

The full system of ODEs that results from spatial discretization, here denoted by $dc/dt = F(c)$, is stiff and therefore integrated with a second order Rosenbrock solver using an approximate Jacobian matrix. In [9] this solver has been successfully applied to an advection-diffusion-reaction system from air pollution modelling. The system is autonomous since the right hand side $F(c)$ contains no explicit time dependence; all time dependence in $F(c)$ enters through components of the solution being time dependent. The solution is advanced over a time step with

$$c^{n+1} = c^n + \frac{3}{2}\tau\kappa_1 + \frac{1}{2}\tau\kappa_2,$$

where τ is the step size $t_{n+1} - t_n$, c^n the approximation for $c(t_n)$, and

$$\begin{aligned}(I - \gamma\tau J)\,\kappa_1 &= F(c^n),\\ (I - \gamma\tau J)\,\kappa_2 &= F(c^n + \tau\kappa_1) - 2\kappa_1.\end{aligned}$$

Here J is an approximation of the Jacobian $\partial F/\partial c$ at $c = c^n$ and γ is a free parameter. With the exact Jacobian for J, the stability function reads

$$R(z) = \frac{1 + (1-2\gamma)z + (\frac{1}{2} - 2\gamma + \gamma^2)z^2}{(1-\gamma z)^2}, \tag{4.1}$$

from which it follows that the method is A-stable if and only if $\gamma \geq 1/4$ [4]. Furthermore the method is L-stable if $\gamma = 1 \pm \frac{1}{2}\sqrt{2}$ and the exact Jacobian matrix is used for J. The scheme is of second order in τ regardless of the choice for J.

In [4] one finds ample evidence that Rosenbrock methods are well suited to solve stiff ODEs in the low to moderate accuracy range. However, with the exact Jacobian $\partial F/\partial c$ the method cannot be efficiently applied since the linear system solutions are much too expensive. We therefore apply it with an approximation for $\partial F/\partial c$.

In the current work $c = (\rho_h, r_h)$, where $\rho_h = (u_h, v_h, w_h)$ and r_h is the vector of the r_α. To simplify the notation, in the following we suppress the index h. Furthermore, in an obvious notation, we write $F(c) = (F_\rho(c), F_r(c))$, then

$$\frac{\partial F}{\partial c} = \begin{pmatrix} \frac{\partial F_\rho}{\partial \rho} & \frac{\partial F_\rho}{\partial r} \\ \frac{\partial F_r}{\partial \rho} & \frac{\partial F_r}{\partial r} \end{pmatrix}.$$

We now exploit the fact that the Rosenbrock solver remains of second order for any choice of approximate matrix J and put

$$J = \begin{pmatrix} \frac{\partial F_\rho(c_n)}{\partial \rho} & 0 \\ 0 & 0 \end{pmatrix}.$$

That means we treat the $\partial F_\rho/\partial\rho$ part of the system linearly implicitly and the remainder explicitly. In other words, without the gradient equation we apply the A-stable Rosenbrock solver to the semi-discrete field equations with an exact Jacobian matrix, and without the field equations we integrate the gradient equations explicitly. The explicit method is obtained by substituting for J the zero matrix in the Rosenbrock method. This gives the explicit trapezoidal rule

$$c^{n+1} = c^n + \frac{1}{2}\tau F(c^n) + \frac{1}{2}\tau F(c^n + \tau F(c^n)). \tag{4.2}$$

We have also experimented with another implementation that does treat the whole system implicitly. In principle this allowed us to use the third order Rosenbrock method from [7], but we did not pursue this further because complications arose in the spatial discretization. The third order Rosenbrock method requires a smoother spatial discretization of higher order. So far we did not succeed in implementing such a discretization that worked better than the existing one.

4.1 Stability

While the order of the Rosenbrock method remains 2 with an approximate Jacobian, the stability properties can drastically change. Due to our choice for J, intuition says that the step size restriction will come from the explicit trapezoidal rule applied to the gradient equations and that the field equations do not give a restriction since these are solved linearly implicitly.

To get some insight into the stability of our method we borrow a model problem from [8] which the authors propose as a test model to investigate stability. The test model is a 2 × 2 linear ODE of the form

$$\frac{dc}{dt}=\begin{pmatrix} d_0 & 1 \\ 0 & d \end{pmatrix} c, \tag{4.3}$$

which in a number of steps with simplifying assumptions is obtained from (2.2) through 'linearization, freezing of coefficients, and Fourier-Von Neumann analysis'. Specifically, we have $-\infty < d_0 < 0$ with d_0 representing eigenvalues of the discrete Laplacian Δ_h. Hence d_0 depends on h^{-2} and can become large negative. Further, d represents an eigenvalue for the gradient equation and is determined by second order spatial derivatives of u, v, w. To see this, consider the 1D gradient equation

$$\begin{aligned} \frac{dr(t)}{dt} &= f(r(t)), \\ f(r(t)) &= \lambda u_x(r(t),t), \end{aligned}$$

then

$$\frac{\partial f(r(t))}{\partial r}=\lambda u_{xx}(r(t),t).$$

So, for d we can think of finite values. However, d can also take on positive values.

With the stability test model we have

$$J=\begin{pmatrix} d_0 & 0 \\ 0 & 0 \end{pmatrix}.$$

The Rosenbrock method then gives the recursion

$$c^{n+1}=\begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} c^n, \tag{4.4}$$

where

$$\begin{aligned} R_{11} &= 1+\tau\left(2(1-\gamma\tau d_0)^{-1}-(1-\gamma\tau d_0)^{-2}\right)d_0+\frac{1}{2}\tau^2(1-\gamma\tau d_0)^{-2}d_0^2, \\ R_{12} &= \tau\left(2(1-\gamma\tau d_0)^{-1}-(1-\gamma\tau d_0)^{-2}\right)+\frac{1}{2}\tau^2\left((1-\gamma\tau d_0)^{-1}d+(1-\gamma\tau d_0)^{-2}d_0\right), \\ R_{22} &= 1+\tau d+\frac{1}{2}\tau^2 d^2. \end{aligned}$$

Note that $R_{11}=R(\tau d_0)$ with R the stability function (4.1). Likewise $R_{22}(z)=1+z+z^2/2$ is the stability function of (4.2). Iterating the recursion (4.4) gives

$$\begin{aligned} c^n &= \begin{pmatrix} R_{11}^n & Q \\ 0 & R_{22}^n \end{pmatrix} c^0, \\ Q &= R_{12}\sum_{i=0}^{n-1} R_{22}^i R_{11}^{n-i-1}. \end{aligned}$$

For a stable scheme we must have power boundedness, i.e., $\|c^n\| \leq C \left\|c^0\right\|$, with C independent of n. We can guarantee power boundedness if we have

$$|R_{11}|^n \leq C_{11},\ |R_{22}|^n \leq C_{22},\ |Q| \leq C_{12},\ \text{with } C_{11}, C_{22}, C_{12} \text{ independent of } n.$$

In our implementation we limit ourselves to $\gamma \geq 1/4$, hence we have $|R_{11}| \leq 1$. Likewise, we consider only $\tau \leq 2/|d|$ and $d < 0$ ensuring $|R_{22}| \leq 1$. Now let $R_* = \max(|R_{11}|, |R_{22}|)$, then

$$\begin{aligned}
|Q| &= |R_{12}| \left| \sum_{i=0}^{n-1} R_{22}^i R_{11}^{n-i-1} \right| \\
&\leq |R_{12}| \sum_{i=0}^{n-1} |R_{22}|^i |R_{11}|^{n-i-1} \\
&\leq |R_{12}| \sum_{i=0}^{n-1} R_*^{n-1} \\
&= |R_{12}| (n-1) R_*^{n-1} \\
&\leq |R_{12}| e^{-1} \frac{1}{\ln\left(R_*^{-1}\right)}.
\end{aligned}$$

Hence $|Q|$ is bounded independent of n if $|R_{12}|$ and $1/\ln\left(R_*^{-1}\right)$ are bounded. For $1/\ln\left(R_*^{-1}\right)$ to be bounded it is necessary to have $|R_{11}|$ and $|R_{22}|$ strictly smaller than one, instead of $|R_{11}| \leq 1$ and $|R_{22}| \leq 1$. The entry R_{12} is even $O(\tau)$ and thus bounded. Hence we see that for a stable method we must have some damping for the test model (4.3), provided by the Rosenbrock method and the explicit trapezoidal rule, in addition to the step size restriction $\tau \leq 2/|d|$. If d is positive, some error growth is natural. Note that these conclusions are valid uniformly for $-\infty < d_0 < 0$.

4.2 Spatial factorization

Since we neglect the terms $\partial F_\rho/\partial r$, $\partial F_r/\partial \rho$ and $\partial F_r/\partial r$ in the true Jacobian, the systems that need to be solved are

$$\begin{aligned}
\left(I - \gamma\tau \frac{\partial F_\rho}{\partial \rho}\right) \kappa_{1\rho} &= F_\rho(c^n), && (4.5)\\
\kappa_{1r} &= F_r(c^n), && (4.6)\\
\left(I - \gamma\tau \frac{\partial F_\rho}{\partial \rho}\right) \kappa_{2\rho} &= F_\rho(c^n + \tau\kappa_1) - 2\kappa_{1\rho}, && (4.7)\\
\kappa_{2r} &= F_r(c^n + \tau\kappa_1) - 2\kappa_{1r}, && (4.8)
\end{aligned}$$

with the Jacobian $\partial F_\rho/\partial \rho$ evaluated at $c = c^n$. To further speed up the linear system solution, we approximate the matrix with the following spatial factorization,

$$\left(I - \gamma\tau \frac{\partial F_{\rho x}}{\partial \rho}\right) \left(I - \gamma\tau \frac{\partial F_{\rho y}}{\partial \rho}\right), \tag{4.9}$$

where $F_{\rho x}$ and $F_{\rho y}$ contain only difference operators in either the x- or the y-direction and $F_{\rho x} + F_{\rho y} = F_\rho$. Further, in our implementation we have distributed the loss terms and source terms equally (with factor 0.5) over $F_{\rho x}$ and $F_{\rho y}$. Solving the systems with factorized matrices requires the successive solution of systems of a much smaller dimension, that are only coupled in the x- or y-direction. For each grid line in Ω_h we encounter such a smaller sized system. These systems can be solved far more quickly than the original ones. This is an application of approximate matrix factorization. Examples of approximate matrix factorization can be found in [2], [1], [9], [6] and [7].

The use of (4.9) does not cause a loss of order in accuracy of the numerical solution since the order remains two for any approximation to $\partial F_\rho/\partial\rho$. Note that (4.9) implies that we approximate $\partial F_\rho/\partial\rho$ by

$$\partial F_{\rho x}/\partial\rho + \partial F_{\rho y}/\partial\rho + \gamma\tau\partial F_{\rho x}/\partial\rho\ \partial F_{\rho y}/\partial\rho.$$

The factorization does somewhat weaken the stability properties, e.g., L-stability is lost but otherwise the stability behaviour remains quite satisfactory.

5. Implementation of source terms and gradient

In our axon growth model it is assumed that all growth cones secrete identical chemical compounds through which they attract and repel each other. It therefore is possible that a certain growth cone responds to molecules secreted by that same growth cone. For instance, a growth cone can be slowed down in its growth due to the trail of attractant that builds up behind the growth cone. This self-interaction can be very troublesome in a numerical implementation. In the model the growth cones are described as small particles that secrete chemicals in a small spatial region and sense the gradients in chemical concentrations in the same region. To avoid self-interaction we consider only symmetrical source terms $s(x)$ which do not yield a direct gradient at the location of the emitting axon. Later on we will present a condition that the numerical implementation must satisfy to mimic this property.

To better understand how we can have self-interaction, and how this relates to the symmetry of $s(x)$, consider the following simplified version of the model which contains only one axon at position $r(t)$ and no targets:

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} - ku(x,t) + s(x - r(t)), \tag{5.1}$$

$$\frac{dr(t)}{dt} = \frac{\partial u(r(t),t)}{\partial x}. \tag{5.2}$$

By differentiating (5.2) with respect to t and substituting (5.1) we get

$$\frac{d^2r(t)}{dt^2} = \frac{\partial^3 u(r(t),t)}{\partial x^3} - k\frac{\partial u(r(t),t)}{\partial x} + \frac{\partial^2 u(r(t),t)}{\partial x^2}\frac{\partial u(r(t),t)}{\partial x} + \left.\frac{\partial s(x-r(t))}{\partial x}\right|_{x=r(t)}.$$

Hence when $\partial s(0)/\partial x \neq 0$, then the evolution of $r(t)$ will be influenced by the presence of $s(x)$. This is a self-interaction since $s(x - r(t))$ represents an emission by the axon located at $r(t)$. Higher temporal derivatives of $r(t)$ also only contain odd spatial derivatives of $s(x)$, i.e., $d^n r(t)/dt^n$ contains

$$\frac{\partial^{2n-1}s(0)}{\partial^{2n-1}x}, \frac{\partial^{2n-3}s(0)}{\partial^{2n-3}x}, \dots, \frac{\partial s(0)}{\partial x}.$$

To avoid self-interaction completely we must have that all odd derivatives of $s(x)$ vanish at $x = 0$, which holds if $s(x)$ is symmetric.

To see what kind of numerical complications we can have due to self-interaction we now consider some numerical implementations of (5.1) and (5.2). After spatial discretization at grid points x_i we get

$$\frac{du_i(t)}{dt} = D_{xx}u_i(t) - ku_i(t) + s(x_i - r(t)),$$

$$\frac{dr(t)}{dt} = P_h(r(t))D_x u_h(t).$$

Note that we use D_x and D_{xx} to denote spatial discretizations of $\partial/\partial x$ and $\partial^2/\partial x^2$, instead of ∇_h and Δ_h, to stress that we now model a one-dimensional case. Further, $P_h(r(t))$ interpolates the grid function $D_x u_h(t)$ at location $r(t)$. Note that as in Section 3 we keep the same notation for r after spatial discretization. When we now consider $d^2r(t)/dt^2$ we see the source term contribution appearing as

$$P_h(r(t))D_x s(x_i - r(t)). \tag{5.3}$$

In the remainder of this section we will examine this term for different spatial discretizations. It will turn out that depending on the discretization chosen, the contribution of this interpolated source term either vanishes or not. In our implementation a non-vanishing contribution is noticed as a growth cone that is blinded by its own attractant and repellant and randomly wanders about the domain. It will turn out that the choice for $P_h(r)D_x$ and $s(x - r(t))$ must be matched in some sense to avoid an excessive contribution of the interpolated source term.

Depending on the size of the mesh width, we treat the source terms in one of two ways. If the grid is coarse relative to the size of the sources, we treat the sources as point sources. When the grid is sufficiently fine we take into account the finite spatial extent of the sources.

5.1 Highly localized source term

First we consider the case of a relatively coarse grid. When the spatial extent of the source term is smaller than twice the mesh width, we distribute the source term over the nearest grid points. Note that in this section we attempt to model a point source and must therefore choose an appropriate distribution over the nearest grid points. In the next section we consider a source term with a spatial extent and the source term is in principle fixed by the biological model.

It is insightful to examine a few source term implementations together with choices for the gradient operator on a grid of points $x_i = ih$.

Example As a first implementation for the discrete gradient operator consider for a grid funtion f_h

$$P_h(r)D_x f_h = \frac{f_{i+1} - f_i}{h}, \quad x_i \le r < x_{i+1}. \tag{5.4}$$

As implementation of the source term consider

$$s(x_i - r) = s(x_{i+1} - r) = \frac{1}{2h}, \quad x_i \le r < x_{i+1}, \quad s_i(x_j - r) = 0, \quad j \ne i, i+1,$$

i.e., the source flux is divided evenly over the two nearest grid points. Taking these implementations together yields for the term (5.3)

$$P_h(r)D_x s(x_i - r) = \frac{s(x_{i+1} - r) - s(x_i - r)}{h} = \frac{1/2 - 1/2}{h^2} = 0, \quad x_i \le r < x_{i+1},$$

hence we have a vanishing contribution, which is satisfactory. Now consider the same implementation for the gradient but replace the source term interpolation with

$$s_i(x_i - r) = 1/h, \quad s(x_j - r) = 0, \quad r - h/2 \le x_i < r + h/2, \quad j \ne i,$$

i.e., all source flux is attributed to the nearest grid point. The term (5.3) then becomes

$$P_h(r)D_x s(x_i - r) = \frac{s(x_{i+1} - r) - s(x_i - r)}{h} = \frac{0 - 1}{h^2} = -\frac{1}{h^2}, \quad x_i \le r \le x_i + h/2.$$

Hence, we end up with a non-vanishing contribution that is inversely proportional to h^2. In the case that we would attempt to model a delta function type source term, this contribution would lead to a scheme that does not converge when we consider the limit $h \to 0$. In the case of a source term of finite spatial extent, the problem is less severe since, eventually, decreasing the mesh width h will resolve the source term on the spatial grid. Resolved source terms are discussed in the next section.

Condition on implementation of source term and gradient We have seen that for an acceptable implementation of source term and gradient we must demand that (5.3) vanishes. Now we introduce some notation to work out the consequences of this restriction. For the discretized approximation of the source term we write

$$s(x_i - r) = \frac{1}{h}\theta_i,$$

where θ_i are weight functions that smear out the point source behaviour over a limited number of points near r. To conserve mass the θ_i must satisfy

$$\sum_i \theta_i = 1. \tag{5.5}$$

For the discretized gradient operator, acting on a grid function f_h evaluated in point r, we write

$$P_h(r)D_x f_h = \sum_i \phi_i f_i,$$

where ϕ_i are weight functions that, under summation, construct an approximation to the gradient of f_h in point r out of surrounding values of f_h.

Using the above notation we have

$$P_h(r)D_x s(x_i - r) = \sum_j \frac{\phi_j \theta_j}{h}.$$

For a suitable combination of source term and gradient discretization $P_h(x)D_x s(x_i - r)$ must vanish, hence we must have

$$\sum_i \phi_i \theta_i = 0. \tag{5.6}$$

Using this condition we can choose matching discretizations for source term and gradient. Again this is clarified by looking at some examples.

Example 1 As before, consider

$$P_h(r)D_x f_h = \frac{f_{i+1} - f_i}{h}, \quad x_i \le r < x_{i+1},$$

as an implementation for the gradient. This corresponds to

$$\phi_{i+1} = \frac{1}{h}, \quad \phi_i = -\frac{1}{h}, \quad \phi_j = 0, \quad j \ne i, i+1.$$

Imposing restriction (5.6) then yields

$$\frac{1}{h}\theta_i - \frac{1}{h}\theta_{i+1} = 0,$$

while we also impose (5.5), which yields

$$\theta_i + \theta_{i+1} = 1.$$

These restrictions together yield

$$\theta_i = \theta_{i+1} = 1/2, \quad x_i \le r < x_{i+1}, \quad \theta_j = 0, \quad j \ne i, i+1.$$

Note that this corresponds to the source term implementation that led to $P_h(r)D_x s(x_i - r) = 0$ in the previous example.

Example 2 Now consider

$$P_h(r)D_x f_h = \frac{x_{i+1}-r}{h}\frac{f_{i+1}-f_{i-1}}{2h} + \frac{r-x_i}{h}\frac{f_{i+2}-f_i}{2h}, \quad x_i \le r < x_{i+1}, \tag{5.7}$$

which corresponds to

$$\phi_{i-1} = \frac{r-x_{i+1}}{2h^2}, \tag{5.8}$$

$$\phi_i = \frac{x_i - r}{2h^2}, \tag{5.9}$$

$$\phi_{i+1} = \frac{x_{i+1}-r}{2h^2}, \tag{5.10}$$

$$\phi_{i+2} = \frac{r-x_i}{2h^2}, \tag{5.11}$$

$$\phi_j = 0, \quad j \ne i-1, i, i+1, i+2.$$

This implementation for the gradient corresponds to second order central differences in x_i and x_{i+1} combined with linear interpolation. In choosing an implementation for the source term we now have four degrees of freedom, i.e., θ_{i-1}, θ_i, θ_{i+1}, θ_{i+2}. To fix these degrees of freedom we have only two restrictions, (5.5) and (5.6). To close the system we simply choose $\theta_{i-1} = \theta_{i+2} = 0$, which makes the discretized source term more local which is natural since the spatial extent of the source is smaller than twice the mesh width. We are left with

$$\frac{x_i - r}{2h^2}\theta_i + \frac{x_{i+1}-r}{2h^2}\theta_{i+1} = 0,$$

and

$$\theta_i + \theta_{i+1} = 1,$$

which leads to

$$\theta_i = 1 - \frac{r-x_i}{h}, \quad \theta_{i+1} = 1 + \frac{r-x_{i+1}}{h}, \tag{5.12}$$

i.e., the source flux is linearly distributed over the nearest grid points.

Implementations used for the current work Implementations (5.7) and (5.12) for the source term and gradient prove very useful in numerical practice. Hence these implementations were used to obtain the numerical results presented further on. Since we consider a spatially two-dimensional problem, the implementations were extended to two dimensions with a straightforward tensor product approach.

Higher order discretizations It seems straightforward to extend the above reasoning to obtain higher order discretizations for (highly localized) source terms and gradients. In particular, we have examined a source term implementation compatible with fourth order Hermite interpolation. However, this higher order implementation did not compete with (5.8) and (5.12) in numerical experiments.

5.2 Resolved source term

When the mesh width is smaller than the spatial extent of the source term we say that the source term is resolved. The source term function $s(x-r)$, which is still localized around $x = r$, then spans several mesh widths. Depending on the choice for the implementation of the gradient operator and interpolation, there exist leading order expressions (in the mesh width h) for the contribution of $P_h(r)D_x s(x_i - r)$. For the implementation (5.4) we obtain

$$P_h(r)D_x s(x_i - r) = \frac{1}{2}(1-2\alpha)hs^{(2)}(0) + R_1,$$

assuming smoothness, where $\alpha \in [0,1]$ measures the position r relative to the nearest grid points and R_1 is a remainder term of $O(h^2)$ that vanishes for $h \to 0$. For what follows R_1 is not negligible, but we focus on the formal leading order term anyway since that suffices to make our point. Likewise, for the implementation (5.7) we obtain

$$P_h(r)D_x s(x_i - r) = \frac{1}{6}(1-2\alpha)(\alpha-1)\alpha h^3 s^{(4)}(0) + R_2.$$

Note the absence of a term $\sim h^2$. This is due to $s^{(3)}(0) = 0$ since $s(x)$ is supposed to be symmetric around $x = 0$. Since these expressions vanish as $h \to 0$ they seem quite satisfactory. In practice h is finite of course and the contribution depends on the size of $s^{(2)}(0)$ and $s^{(4)}(0)$, respectively.

Smooth Gaussian type source term Like in [5] we now consider a source term of the type

$$s(x) = e^{-\gamma x^2},$$

which does not truly vanish away from $x = 0$ but can be made very small away from $x = 0$ by taking γ sufficiently large. In a practical implementation we typically choose the mesh width h such that the source term is resolved on several grid cells, i.e., we can put

$$s(h) = e^{-\beta},$$

where β is of order 1. For such a mesh width we thus have

$$\gamma = \beta / h^2.$$

For $s^{(2)}(0)$ and $s^{(4)}(0)$ this yields

$$s^{(2)}(0) = -\frac{2\beta}{h^2}, \quad s^{(4)}(0) = \frac{12\beta^2}{h^4}.$$

Hence for the contributions of the term $P_h(r)D_x s$ we get

$$P_h(r)D_x s(x_i - r) = -\beta(1-2\alpha)\frac{1}{h} + R_1,$$

for the implementation (5.4) and

$$P_h(r)D_x s(x_i - r) = 2\beta^2(1-2\alpha)(\alpha-1)\alpha\frac{1}{h} + R_2,$$

for the implementation (5.7). For both implementations we find

$$P_h(r)D_x s(x_i - r) \sim \frac{1}{h} + R.$$

In fact, for other gradient implementations we will still usually have

$$P_h(r)D_x s(x_i - r) = h^n s^{(n+1)}(0) + R \sim \frac{1}{h} + R.$$

Note that we are not stating this as an asymptotic result; when we consider the limit $h \to 0$ and keep γ fixed, the contribution $P_h(r)D_x s(x_i - r)$ will vanish. However, for a fixed sensible choice of the mesh width h relative to the spatial extent of the source term, the term $P_h(r)D_x s(x_i - r)$ yields a term proportional to $1/h$.

Implemented resolved source term We now consider a very simple type of source term for which $P_h(r)D_xs(x_i - r)$ vanishes in a natural manner. Denote the source term's spatial extent by l. When $h > l/2$ we treat the source term as a highly localized source term discussed earlier. When $h < l/2$ we treat the source term as a hat function that is zero outside its base of length l, has its maximum halfway that base and varies linearly between the maximum and the end points of the base. A straightforward calculation shows that for this source term the gradient implementation (5.7) always yields $P_h(r)D_xs(x_i - r) = 0$. The implementation (5.4) still gives $P_h(r)D_xs(x_i - r) \sim 1/h + R$ and should therefore not be used in conjunction with this source term.

6. Numerical results

6.1 Convergence

In Figure 3a-d axon growth paths are shown for 5 axons growing towards 5 targets. See section 2 for the values of the problem parameters that were used. The sub-figures are computed on grids of increasing resolution ranging from 65×65 grid cells to 513×513 grid cells. Before starting the actual integration the field equations were marched to steady state while keeping the axons fixed at their initial positions. During the integration the target concentration field was kept fixed at the steady state, which is natural since it is independent of the axons positions. Putting the field equations in steady state beforehand is justified if the time scale of the gradient equation is an order of magnitude larger then the time scale of the field equations. The source term was taken to be of the hat function type, the differential operators were discretized by central differences and the interpolation was done with bi-linear interpolation, as in (5.7) for the 1D case. The time stepsizes used for time integration were taken inversely proportional to the mesh width and are given in the figures' caption.

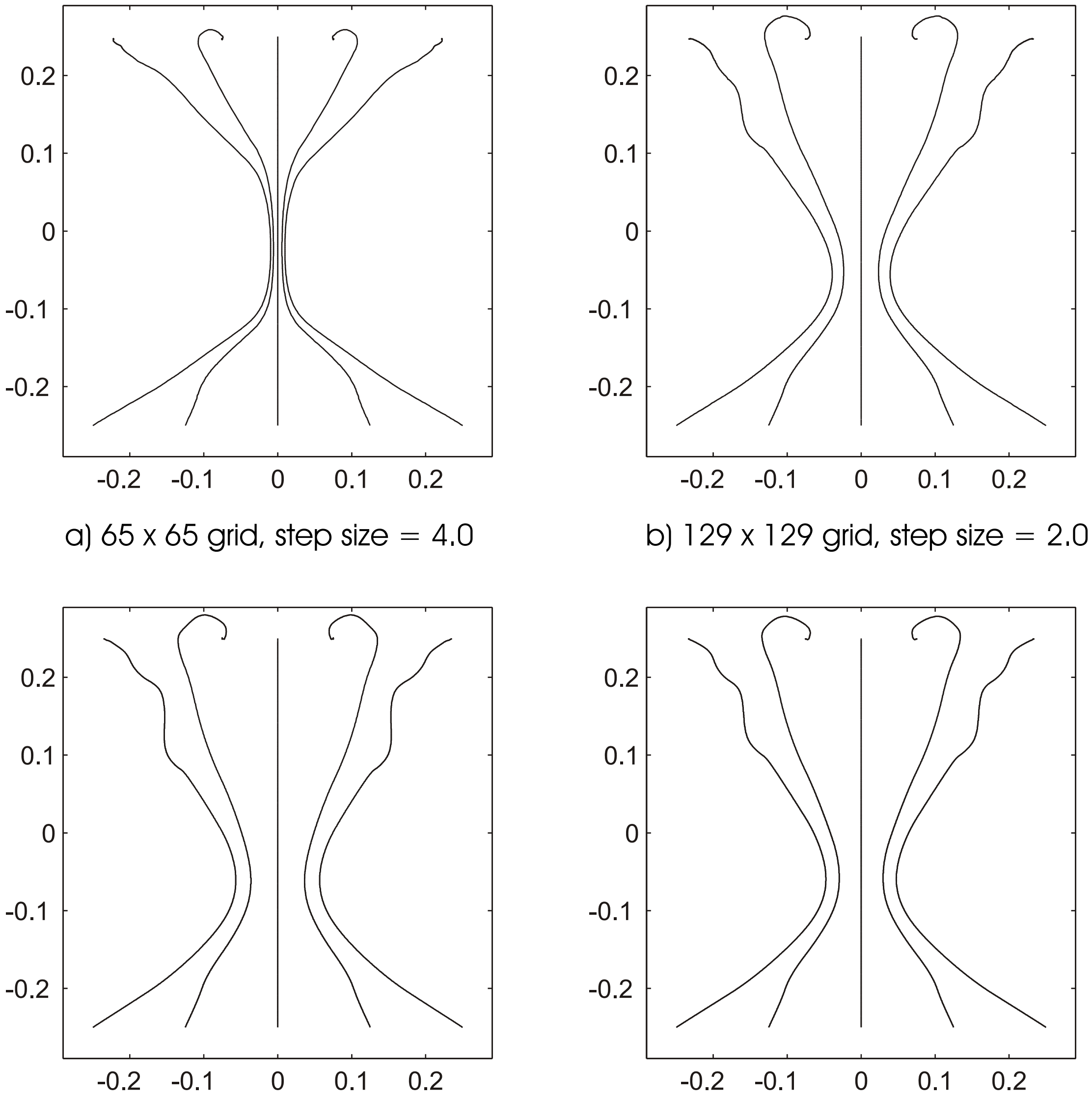
In Figure 4 the path of a single axon is plotted for different grid resolutions. From this plot we see that the observed path converges, however non-monotonically. Hence, provided the mesh width and step size are small enough, taking a smaller mesh width and step size does not yield a different path for the axons. Mesh width and step-size are reduced simultaneously to avoid stability problems. When only the mesh width is reduced, the method eventually becomes unstable. This is not in accordance with the stability analysis for the simple test model from Section 4. Convergence is only observed for a very fine mesh width due to the sensitivity of the problem. Small changes in mesh width can have a strong impact on the paths of the axons. This is not unexpected since the effects of mesh width related discretization errors on the axon path accumulate over the entire time integration interval. Therefore it seems attractive to use an effective higher order spatial discretization.

6.2 Stability

Since we do not have a textbook condition for the maximum allowable time step-size for our method applied to the axon growth problem, we have done numerical experiments to determine a maximum step size. We ran our program with different step-sizes and looked for signs of instability, e.g., amplified wiggles in the axon paths. When we saw these signs we deemed the step size too large. In Table 1 we have listed the thus determined maximum step sizes for several different mesh widths. From the table we see that a smaller step size is required on finer spatial grids. Halving the mesh widths requires halving the time step-size, approximately. It thus seems worthwhile to examine more stable methods.

Grid	maximum step size
65×65	12.5
129×129	6.1
257×257	3.0
513×513	1.4

Table 1: Experimentally observed maximum step sizes



a) 65 x 65 grid, step size = 4.0

b) 129 x 129 grid, step size = 2.0

c) 257 x 257 grid, step size = 1.0

d) 513 x 513 grid, step size = 0.5

Figure 3: Axon growth paths for different grid resolutions

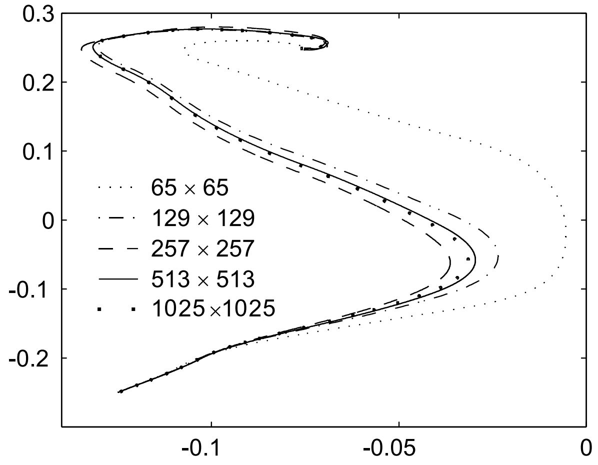


Figure 4: Single axon growth path for different grid resolutions

6.3 Efficiency

An important merit of our method is that it solves the problem at hand efficiently compared to a number of other methods that one could consider. This merit comes forth from the explicit gradient equation treatment and the spatial factorization which greatly reduces the complexity of the linear algebra problem that needs to be solved. Due to the factorization, small sized one-dimensional systems need to be solved instead of one large two-dimensional system.

To get some quantitative idea of the efficiency of our method, we compared it with another one, i.e., the Runge-Kutta-Chebyshev (RKC) method described in [8]. This method is fully explicit and is stabilized at the expense of additional function evaluations. We applied the RKC method to the same problem and obtained the same solutions with it. In Table 2 the wall-clock times for RKC and our Rosenbrock method (ROS2) are listed. These times refer to runs on a single processor on identical hardware with the same time step-sizes. The RKC program was written in Fortran while our ROS2 method was implemented in C, but this should not have a strong influence on the run times. As we can see from Table 2, ROS2 is faster than RKC, however the difference is only about 30%.

Grid	ROS2 runtime	RKC runtime
65 × 65	14	19
129 × 129	110	149
257 × 257	815	1207
513 × 513	7124	11431

Table 2: Runtimes for our ROS2 method versus the RKC method

6.4 Parameter sensitivity

For the numerical results to be in qualitative agreement with biological experiments, we ought to observe a bundling and debundling of the axons. For certain sets of problem parameters, this indeed does occur, see Figure 3. However, it is interesting to know whether this behaviour persists when the parameters are slightly perturbed. We have found that around a set of parameters for which bundling and debundling occurs, an interval of parameter values exists for which this still holds. Outside this interval, either bundling, debundling or both no longer occur. In Table 3 this parameter sensitivity is portrayed. The results in Table 3 refer to the same experiment as before, performed on a 257 × 257 grid. All parameters are kept fixed, except for λ_v, the parameter for growth cone sensitivity to cone derived attractant. For $\lambda_v \in [0, 3.88 \cdot 10^{-6})$ the axons do bundle but do not debundle, i.e., they grow jointly to a single target. For $\lambda_v \in [3.88 \cdot 10^{-6}, 6.04 \cdot 10^{-6})$ the axons bundle and partially debundle; they ultimately grow towards three targets. For $\lambda_v \in [6.04 \cdot 10^{-6}, 7.81 \cdot 10^{-6})$ the axons bundle and debundle completely in that they each grow to a different target. For $\lambda_v \geq 7.81 \cdot 10^{-6}$ the axons no longer bundle and hence do not debundle either. Instead they grow away from each other and eventually leave the computational domain. We see that around $\lambda_v = 5 \cdot 10^{-6}$ full bundling and debundling occurs but also that a change of λ_v by about 20% prohibits either bundling or debundling to occur. In other words, the system is rather sensitive to changes in parameters since a 20% change in a single parameter can change the qualitative behavior of the solution completely.

End points	λ_v
1	$[0, 3.88 \cdot 10^{-6})$
3	$[3.88 \cdot 10^{-6}, 6.04 \cdot 10^{-6})$
5	$[6.04 \cdot 10^{-6}, 7.81 \cdot 10^{-6})$
-	$[7.81 \cdot 10^{-6}, \infty)$

Table 3: Degree of debundling for different parameter ranges

6.5 Source term and gradient implementation

According to Section 5 it is imperative to have a source term and gradient implementation such that the discrete gradient of the discrete source vanishes at the origin of the source. To illustrate this claim numerically, we considered a single parameter set and ran tests for different implementations of source term and gradient. In Figure 5a the final axon paths are shown for the bilinear source term and the bilinear interpolation and gradient implementation. In Figure 5b the axon paths are shown for a piecewise constant source term and piecewise constant gradient implementation. We see that both figures show qualitatively the same set of growth paths. The paths shown in Figure 5c were obtained with piecewise constant source term and bilinear interpolation while the paths in Figure 5d were obtained with bilinear source term and piecewise constant interpolation. If we now consider condition (5.6) then we see that the implementations that yield Figures 5a and 5b satisfy this condition while the implementations that yield Figures 5c and 5d do not. This is clearly reflected in the axon paths; the paths seen in Figures 5c and 5d are qualitatively different from each other and from the paths in Figures 5a and 5b. Figures 5c and 5d clearly illustrate what can go wrong if the implementation of the source term and gradient are not chosen carefully.

7. Discussion

Rosenbrock time stepping with an appropriate Jacobian matrix proved well suited to the mixed parabolic-gradient problem that was considered. The freedom in choosing an approximation for the Jacobian allows the parabolic and gradient equations to be treated almost independently from each other. An extension to spatially three-dimensional problems seems certainly feasible. Spatial factorization would then be done for all three spatial dimensions leading to even larger gains in efficiency relative to a non-factorized approach.

It was shown that the implementation of the source term and gradient detection is somewhat delicate and should not be constructed independently of each other. In particular, a condition was derived that the combined implementation of source term and gradient should fulfil, see condition (5.6).

Since the axon paths are quite sensitive to refinements in the mesh width very fine meshes are needed to get spatially converged solutions. In a future method higher spatial order of discretization

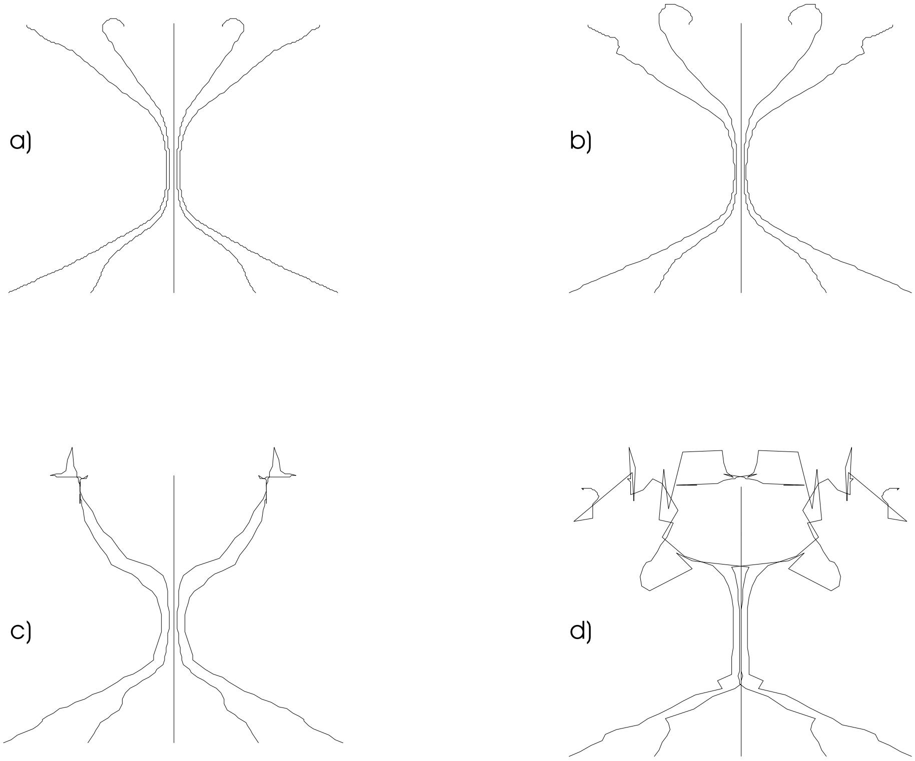


Figure 5: Axon growth paths for different source term and gradient implementations

might be attractive to get sufficiently accurate solutions on coarser grids so that very fine grids are no longer needed. Furthermore, it might be interesting to see wether the integration of the gradient equation can be stabilized since in the current Rosenbrock-Approximate Jacobian approach it seems to dictate the overall stability. The gradient equation could be handled implicitly or some form of RKC method could be used where the number of integration stages is increased to enhance stability, as in [8].

The model considered for axonal growth is a very simple one. To simulate more realistic axonal growth in any detail will require a more elaborate model. Furthermore, the current model is somewhat awkward for numerical simulation due to the absence of a physical dimension of growth cones and targets. As a result, in the simulations the growth cones are capable to approach each other to distances much smaller than could ever occur in biological experiments. The growth cones can even collapse onto each other, leaving them at the same location. When this happens, debundling is no longer possible since the cones can then no longer differentiate between themselves and the other cones and can no longer be pushed away from each other.

References

1. R. M. Beam, R. F. Warming, *An implicit finite-difference algorithm for hyperbolic systems in conservation-law form*, J. Comput. Phys. 22, pp. 87-110, 1976.
2. E. G. D'yakonov, *Difference systems of second order accuracy with a divided operator for parabolic equations without mixed derivatives*, USSR Comput. Math. Math. Phys., 4(5), pp. 206-216, 1964.
3. G. J. Goodhill, *Diffusion in axon guidance,* Eur. J. Neurosci. 9, pp. 1414-1421, 1997.
4. E. Hairer, G. Wanner, *Solving ordinary differential equations II. Stiff and differential algebraic problems,* 2nd ed., Springer-Verlag, Berlin, 1996.
5. H. G. E. Hentschel, A. van Ooyen, *Models of axon guidance during development,* Proc. R. Soc. Lond. B 266, pp. 2231-2238, 1999.
6. P. J. van der Houwen, B. P. Sommeijer, *Approximate factorization for time-dependent partial differential equations,* J. Comput. Appl. Math. 128, pp. 447-466, 2000.
7. B. Lastdrager, B. Koren, J. G. Verwer, *Solution of time-dependent advection-diffusion problems with the sparse-grid combination technique and a Rosenbrock solver,* Journal of Computational Methods in Applied Mathematics, Vol. 1, number 1, pp. 86-98, 2001.
8. J. G. Verwer, B. P. Sommeijer, *A numerical study of mixed parabolic-gradient systems,* J. Comput. Appl. Math. 132, pp. 191-210, 2001.
9. J. G. Verwer, E. J. Spee, J. G. Blom, W. Hundsdorfer, *A second-order Rosenbrock method applied to photochemical dispersion problems,* Siam J. Sci. Comput. 20, pp. 1456-1480, 1999.