

1 Introduction

This document is relevant to the targets:

- `bidomain`
- `bidomain_godunov_2`
- `bidomain_godunov`
- `bidomain_godunov_2`

The section "Toy Problem" is relevant to `bidomain` and `bidomain_godunov`, while "Toy Problem 2" is relevant to `bidomain_2` and `bidomain_godunov_2`. The section "Time Integration" is relevant to `bidomain` and `bidomain_2`. Naturally, `bidomain_godunov` and `bidomain_godunov_2` rely on the section "Godunov Splitting"; this section makes the same assumption and method as the previous section, "Linear Formulation". See the section "Version History" for more information about these different targets.

Let:

$$\begin{aligned}
 n &\in \mathbb{N} \\
 T &\in \mathbb{R} \\
 \Omega &\subseteq \mathbb{R}^n \\
 \chi, C_m &\in \mathbb{R} \\
 \boldsymbol{\sigma}_i, \boldsymbol{\sigma}_e &\in \mathbb{R}^{n \times n} \\
 I_{\text{ion}} &: \mathbb{R} \rightarrow \mathbb{R} \\
 f, f_e &: (0, T) \times \Omega \rightarrow \mathbb{R} \\
 v, u_e &: (0, T) \times \Omega \rightarrow \mathbb{R}
 \end{aligned}$$

Denote $\mathbf{w}(t, \mathbf{x}) = \begin{bmatrix} v \\ u_e \end{bmatrix}$. The Bidomain problem is given as:

$$\chi C_m \frac{\partial v}{\partial t} + \chi I_{\text{ion}}(v) = \nabla \cdot (\boldsymbol{\sigma}_i \nabla (v + u_e)) + f \quad (1)$$

$$0 = \nabla \cdot (\boldsymbol{\sigma}_i \nabla (v + u_e)) + \nabla \cdot (\boldsymbol{\sigma}_e \nabla u_e) + f_e \quad (2)$$

With initial conditions:

$$\mathbf{w}(0, \mathbf{x}) = \mathbf{w}_0(\mathbf{x}) \quad (3)$$

For all $\mathbf{x} \in \Omega$ and with boundary conditions:

$$\boldsymbol{\sigma}_i \nabla (v + u_e) \cdot \hat{\mathbf{n}} = \boldsymbol{\sigma}_e \nabla u_e \cdot \hat{\mathbf{n}} = 0 \quad (4)$$

For all $(t, \mathbf{x}) \in (0, T) \times \partial\Omega$.

Let $R_m \in \mathbb{R}$. For a passive cell model, we take $I_{\text{ion}}(v) = \frac{1}{R_m}v$. We also assume that $\boldsymbol{\sigma}_i = \sigma_i \mathbf{I}$ and $\boldsymbol{\sigma}_e = \sigma_e \mathbf{I}$ for some $\sigma_i, \sigma_e \in \mathbb{R}$, so that $\boldsymbol{\sigma}_i$ and $\boldsymbol{\sigma}_e$ can be replaced with σ_i and σ_e in the equations above.

2 Toy Problem

We consider a toy problem with $T = 1$, and $\Omega = [0, 1]^n$:

$$u_e(t, \mathbf{x}) = t^3 \prod_{i=1}^n \cos(\pi x_i) \quad (5)$$

$$v(t, \mathbf{x}) = -\frac{\sigma_i + \sigma_e}{\sigma_i} u_e(t, \mathbf{x}) \quad (6)$$

It is clear that $v(0, \mathbf{x}) = u_e(0, \mathbf{x}) = 0$, so we take $\mathbf{w}_0(\mathbf{x}) = \mathbf{0}$.

Next, we consider:

$$\begin{aligned} \sigma_i \nabla(v + u_e) &= \sigma_i (\nabla v + \nabla u_e) \\ &= \sigma_i \left(-\frac{\sigma_i + \sigma_e}{\sigma_i} \nabla u_e + \nabla u_e \right) \\ &= -\sigma_i \nabla u_e - \sigma_e \nabla u_e + \sigma_i \nabla u_e \\ &= -\sigma_e \nabla u_e \end{aligned}$$

Since $\frac{\partial u_e}{\partial x_k} = -\pi t^3 \sin(\pi x_k) \prod_{i \neq k} \cos(\pi x_i)$, we see that $\frac{\partial u_e}{\partial x_k} = 0$ when $x_k \in \mathbb{Z}$. Recall that $\Omega = [0, 1]^n$; then, on the x_k -faces of $\partial\Omega$, we have that $\frac{\partial u_e}{\partial x_k} = 0$ and $\hat{\mathbf{n}} = \pm \hat{\mathbf{e}}_k$. So, $\nabla u_e \cdot \hat{\mathbf{n}} = 0$ for all $\mathbf{x} \in \partial\Omega$, and both boundary conditions in Equation 4 are satisfied.

Then, we determine f and f_e . From above, we have that $\sigma_i \nabla(v + u_e) = -\sigma_e \nabla u_e$, and so, rewriting Equation 2:

$$\begin{aligned} f_e &= -\nabla \cdot (\sigma_i \nabla(v + u_e)) - \nabla \cdot (\sigma_e \nabla u_e) \\ &= \nabla \cdot (\sigma_e \nabla u_e) - \nabla \cdot (\sigma_e \nabla u_e) \\ &= 0 \end{aligned}$$

When we note that:

$$\begin{aligned} \frac{\partial^2 u_e}{\partial x_k^2} &= -\frac{\partial}{\partial x_k} \pi t^3 \sin(\pi x_k) \prod_{i \neq k} \cos(\pi x_i) \\ &= -\pi^2 t^3 \prod_{i=1}^n \cos(\pi x_i) \\ &= -\pi^2 u_e \\ \Delta u_e &= \sum_{k=1}^n \frac{\partial^2 u_e}{\partial x_k^2} \\ &= -\sum_{k=1}^n \pi^2 u_e \\ &= -n\pi^2 u_e \end{aligned}$$

We rewrite Equation 1:

$$\begin{aligned}
f &= \chi C_m \frac{\partial v}{\partial t} + \chi I_{\text{ion}}(v) - \nabla \cdot (\sigma_i \nabla (v + u_e)) \\
&= -\chi C_m \frac{\sigma_i + \sigma_e}{\sigma_i} \frac{\partial u_e}{\partial t} + \chi I_{\text{ion}}(v) - \nabla \cdot (-\sigma_e \nabla u_e) \\
&= -\chi C_m \frac{\sigma_i + \sigma_e}{\sigma_i} \frac{3}{t} u_e - \chi \frac{1}{R_m} \frac{\sigma_i + \sigma_e}{\sigma_i} u_e + \sigma_e \Delta u_e \\
&= -\chi \left(\frac{3C_m}{t} + \frac{1}{R_m} \right) \frac{\sigma_i + \sigma_e}{\sigma_i} u_e - n \sigma_e \pi^2 u_e \\
&= -\left(\chi \frac{\sigma_i + \sigma_e}{\sigma_i} \left(\frac{3C_m}{t} + \frac{1}{R_m} \right) + n \sigma_e \pi^2 \right) u_e
\end{aligned}$$

3 Toy Problem 2

We will consider another toy problem, to address the fact that $u_e = 0$ is a simple way to minimize the residual in the previous toy problem:

$$\begin{aligned}\omega(\mathbf{x}) &= \prod_{i=1}^n \cos(\pi x_i) \\ v(t, \mathbf{x}) &= t^2 \omega(\mathbf{x}) \\ u_e(t, \mathbf{x}) &= t \omega(\mathbf{x})\end{aligned}$$

We see that:

$$\begin{aligned}\Delta \omega &= -n\pi^2 \omega \\ \Delta v &= -n\pi^2 v \\ \Delta u_e &= -n\pi^2 u_e\end{aligned}$$

And:

$$\begin{aligned}\frac{\partial v}{\partial t} &= \frac{2}{t} v \\ &= 2u_e\end{aligned}$$

So, rewriting the Bidomain equations:

$$\begin{aligned}f &= \chi C_m \frac{\partial v}{\partial t} + \frac{\chi}{R_m} v - \nabla \cdot (\sigma_i \nabla (v + u_e)) \\ &= 2\chi C_m u_e + \frac{\chi t}{R_m} u_e - \sigma_i \Delta v - \sigma_i \Delta u_e \\ &= (2\chi C_m + \frac{\chi t}{R_m}) u_e + n\pi^2 \sigma_i v + n\pi^2 \sigma_i u_e \\ &= (2\chi C_m + \frac{\chi t}{R_m} + n\pi^2 \sigma_i) u_e + n\pi^2 \sigma_i t u_e \\ &= (2\chi C_m + \frac{\chi t}{R_m} + n\pi^2 \sigma_i (t + 1)) u_e \\ f_e &= -\nabla \cdot (\sigma_i \nabla (v + u_e)) - \nabla \cdot (\sigma_e \nabla u_e) \\ &= -\sigma_i \Delta v - (\sigma_i + \sigma_e) \Delta u_e \\ &= n\pi^2 \sigma_i v + n\pi^2 (\sigma_i + \sigma_e) u_e \\ &= n\pi^2 (\sigma_i t + \sigma_i + \sigma_e) u_e\end{aligned}$$

4 Spatial Discretization

We consider a set of basis functions $\Phi_i : \mathbb{R}^n \rightarrow \mathbb{R}^2$, where each $\Phi_i(\mathbf{x}) = \varphi_{b_i} \hat{\mathbf{e}}_{c_i}$ for some set of basis functions $\varphi_i : \mathbb{R} \rightarrow \mathbb{R}$ (b_i is the "base" of the index i , and c_i is the "component" of index i). For some fixed time t , we make the approximation that $\mathbf{w} = W_j \Phi_j$ (Einstein summation) where $\mathbf{W} \in \mathbb{R}^N$ (N is the number of DoFs). Furthermore, we write that $\frac{\partial \mathbf{w}}{\partial t} = X_j \Phi_j$ for some $\mathbf{X} \in \mathbb{R}^N$. Then, we may write:

$$\begin{aligned} v &= \mathbf{w} \cdot \hat{\mathbf{e}}_1 \\ &= W_j \varphi_{b_j} \delta_{1,c_j} \\ u_e &= \mathbf{w} \cdot \hat{\mathbf{e}}_2 \\ &= W_j \varphi_{b_j} \delta_{2,c_j} \\ \frac{\partial v}{\partial t} &= \frac{\partial \mathbf{w}}{\partial t} \cdot \hat{\mathbf{e}}_1 \\ &= X_j \varphi_{b_j} \delta_{1,c_j} \end{aligned}$$

Equation 1 becomes:

$$\chi C_m X_j \varphi_{b_j} \delta_{1,c_j} + \chi I_{\text{ion}}(W_j \varphi_{b_j} \delta_{1,c_j}) = \nabla \cdot (\sigma_i \nabla(W_j \varphi_{b_j} \delta_{1,c_j} + W_j \varphi_{b_j} \delta_{2,c_j})) + f$$

And, because $\delta_{1,c_j} + \delta_{2,c_j} = 1$:

$$\chi C_m X_j \varphi_{b_j} \delta_{1,c_j} + \chi I_{\text{ion}}(W_j \varphi_{b_j} \delta_{1,c_j}) = \nabla \cdot (\sigma_i \nabla(W_j \varphi_{b_j})) + f$$

Similarly, Equation 2 becomes:

$$\begin{aligned} 0 &= \nabla \cdot (\sigma_i \nabla(W_j \varphi_{b_j} \delta_{1,c_j} + W_j \varphi_{b_j} \delta_{2,c_j})) + \nabla \cdot (\sigma_e \nabla(W_j \varphi_{b_j} \delta_{2,c_j})) + f_e \\ &= \nabla \cdot (\sigma_i \nabla(W_j \varphi_{b_j})) + \nabla \cdot (\sigma_e \nabla(W_j \varphi_{b_j} \delta_{2,c_j})) + f_e \end{aligned}$$

Next, we multiply by a test function Φ_i and integrate:

$$\begin{aligned} &\left\langle \Phi_i \mid \chi C_m X_j \varphi_{b_j} \delta_{1,c_j} + \chi I_{\text{ion}}(W_j \varphi_{b_j} \delta_{1,c_j}) \right\rangle \\ &= \left\langle \Phi_i \mid \nabla \cdot (\sigma_i \nabla(W_j \varphi_{b_j})) \right\rangle + \left\langle \Phi_i \mid f \right\rangle \end{aligned}$$

Since $\Phi_i(\mathbf{x}) = \varphi_{b_i} \hat{\mathbf{e}}_{c_i}$:

$$\begin{aligned} &\chi C_m X_j \langle \varphi_{b_i} \mid \varphi_{b_j} \rangle \delta_{1,c_i} \delta_{1,c_j} + \frac{\chi}{R_m} W_j \langle \varphi_{b_i} \mid \varphi_{b_j} \rangle \delta_{1,c_i} \delta_{1,c_j} \\ &= \sigma_i W_j \langle \varphi_{b_i} \mid \nabla \cdot \nabla \varphi_{b_j} \rangle \delta_{1,c_i} \\ &+ \sigma_i W_j \langle \varphi_{b_i} \mid \nabla \cdot \nabla \varphi_{b_j} \rangle \delta_{2,c_i} + \sigma_e W_j \langle \varphi_{b_i} \mid \nabla \cdot \nabla \varphi_{b_j} \rangle \delta_{2,c_i} \delta_{2,c_j} \\ &+ \langle \varphi_{b_i} \mid f \rangle \delta_{1,c_i} + \langle \varphi_{b_i} \mid f_e \rangle \delta_{2,c_i} \end{aligned}$$

Then, since $\langle \varphi_{b_i} | \nabla \cdot \nabla \varphi_{b_j} \rangle = -\langle \nabla \varphi_{b_i} | \nabla \varphi_{b_j} \rangle$, and since $\delta_{1,c_i} + \delta_{2,c_i} = 1$:

$$\begin{aligned} \chi C_m X_j \langle \varphi_{b_i} | \varphi_{b_j} \rangle \delta_{1,c_i} \delta_{1,c_j} &= -\frac{\chi}{R_m} W_j \langle \varphi_{b_i} | \varphi_{b_j} \rangle \delta_{1,c_i} \delta_{1,c_j} \\ &\quad - \sigma_i W_j \langle \nabla \varphi_{b_i} | \nabla \varphi_{b_j} \rangle - \sigma_e W_j \langle \nabla \varphi_{b_i} | \nabla \varphi_{b_j} \rangle \delta_{2,c_i} \delta_{2,c_j} \\ &\quad + \langle \varphi_{b_i} | f \rangle \delta_{1,c_i} + \langle \varphi_{b_i} | f_e \rangle \delta_{2,c_i} \end{aligned}$$

So, denote:

$$\begin{aligned} A_{ij} &= -\frac{\chi}{R_m} \langle \varphi_{b_i} | \varphi_{b_j} \rangle \delta_{1,c_i} \delta_{1,c_j} - (\sigma_i + \sigma_e \delta_{2,c_i} \delta_{2,c_j}) \langle \nabla \varphi_{b_i} | \nabla \varphi_{b_j} \rangle \\ M_{ij} &= \chi C_m \langle \varphi_{b_i} | \varphi_{b_j} \rangle \delta_{1,c_i} \delta_{1,c_j} \\ f_i &= \langle \varphi_{b_i} | f \rangle \delta_{1,c_i} + \langle \varphi_{b_i} | f_e \rangle \delta_{2,c_i} \end{aligned}$$

Then:

$$\mathbf{M}\mathbf{X} = \mathbf{A}\mathbf{W} + \mathbf{f} \quad (7)$$

We note that: 1) Although we have formulated the problem as an affine equation, this is only the case because $I_{\text{ion}}(v) = \frac{1}{R_m} v$, and if we were not using a passive cell model, we may not be able to represent \mathbf{A} as a matrix; 2) \mathbf{M} and (in this case) \mathbf{A} are constant matrices, while \mathbf{X} , \mathbf{W} , and \mathbf{f} each depend on the current time step.

5 Time Integration

Let us discretize the time domain $(0, T)$ into k fixed-size time steps, so denote the step size $h = \frac{T}{k}$. We make the approximation:

$$\frac{\partial \mathbf{w}_{t+1}}{\partial t} = \frac{\mathbf{w}_{t+1} - \mathbf{w}_t}{h}$$

Then, recall that we have already made the approximation:

$$\begin{aligned} \mathbf{w} &= W_j \varphi_j \\ \frac{\partial \mathbf{w}}{\partial t} &= X_j \varphi_j \end{aligned}$$

This implies:

$$X_{t+1,j} \varphi_j = \frac{W_{t+1,j} - W_{t,j}}{h} \varphi_j$$

To keep these approximations consistent, we may simply define:

$$\mathbf{X}_{t+1} = \frac{1}{h} (\mathbf{W}_{t+1} - \mathbf{W}_t)$$

We wish to integrate using the Crank-Nicolson scheme, which we apply to Equation 7 as follows:

$$\begin{aligned} \frac{1}{h} M(\mathbf{W}_{t+1} - \mathbf{W}_t) &= \frac{1}{2} (\mathbf{A} \mathbf{W}_t + \mathbf{f}_t) + \frac{1}{2} (\mathbf{A} \mathbf{W}_{t+1} + \mathbf{f}_{t+1}) \\ M(\mathbf{W}_{t+1} - \mathbf{W}_t) &= \frac{h}{2} \mathbf{A}(\mathbf{W}_{t+1} + \mathbf{W}_t) + \frac{h}{2} (\mathbf{f}_{t+1} + \mathbf{f}_t) \end{aligned}$$

As mentioned in the previous section, although this equation is affine, there are cases in which the vector function \mathbf{A} may be nonlinear; so, we will use Newton's method to solve for \mathbf{W}_{t+1} . So, define:

$$\mathbf{R}_{t+1}(\mathbf{W}) = M(\mathbf{W} - \mathbf{W}_t) - \frac{h}{2} \mathbf{A}(\mathbf{W} + \mathbf{W}_t) - \frac{h}{2} (\mathbf{f}_{t+1} + \mathbf{f}_t)$$

Where \mathbf{W}_t above has been determined by the initial conditions or by the previous time step. The iteration proceeds as follows:

$$\begin{aligned} \mathbf{W}_{t+1}^{(0)} &= \mathbf{W}_t \\ \mathbf{J}[\mathbf{R}_{t+1}] \Delta \mathbf{W}_{t+1}^{(\ell+1)} &= -\mathbf{W}_{t+1}^{(\ell)} \\ \mathbf{W}_{t+1}^{(\ell+1)} &= \mathbf{W}_{t+1}^{(\ell)} + \alpha_{t+1} \Delta \mathbf{W}_{t+1}^{(\ell+1)} \end{aligned}$$

Where $\mathbf{J}[\mathbf{R}_{t+1}]$ is the Jacobian of \mathbf{R}_{t+1} and α_{t+1} is a step size to be chosen by the Newton solver. Once the residual $\|\mathbf{W}_{t+1}^{(\ell)}\|$ is sufficiently small, we take $\mathbf{W}_{t+1} = \mathbf{W}_{t+1}^{(\ell)}$ as the solution for the time step.

6 Linear Formulation

We can also solve the problem using the assumption that $I_{\text{ion}}(v) = \frac{1}{R_m}v$. In this case, the Bidomain equations become linear, allowing us to forego Newton iteration. If we solve using Backward Euler, we have that:

$$\begin{aligned}\frac{1}{h}\mathbf{M}(\mathbf{W}_{t+1} - \mathbf{W}_t) &= \mathbf{A}\mathbf{W}_{t+1} + \mathbf{f}_{t+1} \\ (\frac{1}{h}\mathbf{M} - \mathbf{A})\mathbf{W}_{t+1} &= \frac{1}{h}\mathbf{M}\mathbf{W}_t + \mathbf{f}_{t+1}\end{aligned}$$

Or, using Crank-Nicolson:

$$\begin{aligned}\frac{1}{h}\mathbf{M}(\mathbf{W}_{t+1} - \mathbf{W}_t) &= \frac{1}{2}(\mathbf{A}(\mathbf{W}_{t+1} + \mathbf{W}_t) + \mathbf{f}_{t+1} + \mathbf{f}_t) \\ (\frac{1}{h}\mathbf{M} - \frac{1}{2}\mathbf{A})\mathbf{W}_{t+1} &= (\frac{1}{h}\mathbf{M} + \frac{1}{2}\mathbf{A})\mathbf{W}_t + \frac{1}{2}(\mathbf{f}_{t+1} + \mathbf{f}_t)\end{aligned}$$

7 Godunov Splitting

Let us slightly rewrite the original system of equations:

$$\begin{aligned}\chi C_m \frac{\partial v}{\partial t} &= \nabla \cdot (\sigma_i \nabla (v + u_e)) - \chi I_{\text{ion}}(v) + f \\ 0 &= \nabla \cdot (\sigma_i \nabla (v + u_e)) + \nabla \cdot (\sigma \nabla u_e) + f_e\end{aligned}$$

Which, after spatial discretization, has been written as:

$$\mathbf{M}\mathbf{X} = \mathbf{A}\mathbf{W} + \mathbf{f}$$

Where \mathbf{M} is the mass matrix, \mathbf{A} is the stiffness matrix, \mathbf{f} is the stiffness right-hand side vector, \mathbf{W} is the solution state vector, and \mathbf{X} is the state vector for the solution's time derivative (which is typically written as $\mathbf{X}_{t+1} = \frac{1}{h}(\mathbf{W}_{t+1} - \mathbf{W}_t)$).

We split the right-hand side operator as follows:

$$\begin{aligned}\mathbf{M}\mathbf{X} &= \mathbf{f} \\ \mathbf{M}\mathbf{X} &= \mathbf{A}\mathbf{W}\end{aligned}$$

And, applying Backward Euler time stepping to each system, we get:

$$\begin{aligned}\mathbf{M}\mathbf{X}_t^{(*)} &= \mathbf{f}_{t+1} \\ \mathbf{M}\mathbf{X}_{t+1} &= \mathbf{A}\mathbf{W}_{t+1}\end{aligned}$$

Substituting $\mathbf{X}_t^{(*)} = \frac{1}{h}(\mathbf{W}_t^{(*)} - \mathbf{W}_t)$, we get:

$$\begin{aligned}\frac{1}{h}\mathbf{M}(\mathbf{W}_t^{(*)} - \mathbf{W}_t) &= \mathbf{f}_{t+1} \\ \frac{1}{h}\mathbf{M}\mathbf{W}_t^{(*)} &= \frac{1}{h}\mathbf{M}\mathbf{W}_t + \mathbf{f}_{t+1}\end{aligned}$$

And substituting $\mathbf{X}_{t+1} = \frac{1}{h}(\mathbf{W}_{t+1} - \mathbf{W}_t^{(*)})$, we get:

$$\begin{aligned}\frac{1}{h}\mathbf{M}(\mathbf{W}_{t+1} - \mathbf{W}_t^{(*)}) &= \mathbf{A}\mathbf{W}_{t+1} \\ (\frac{1}{h}\mathbf{M} - \mathbf{A})\mathbf{W}_{t+1} &= \frac{1}{h}\mathbf{M}\mathbf{W}_t^{(*)}\end{aligned}$$

So, our time stepping scheme is:

$$\begin{aligned}\frac{1}{h}\mathbf{M}\mathbf{W}_t^{(*)} &= \frac{1}{h}\mathbf{M}\mathbf{W}_t + \mathbf{f}_{t+1} \\ (\frac{1}{h}\mathbf{M} - \mathbf{A})\mathbf{W}_{t+1} &= \frac{1}{h}\mathbf{M}\mathbf{W}_t^{(*)}\end{aligned}$$

8 Version History

The `bidomain[_2]` targets are an attempt to solve the Bidomain problem assuming that the equations may be nonlinear. However, the version using the data described in "Toy Problem" performs poorly; in particular, the u_e component is forced to 0, and while the shape of the v component is correct, it appears scaled down. However, when solving the problem according to the method described in "Linear Formulation", these problems do not arise. `bidomain_godunov[_2]` does Godunov splitting using this linear method, which performs similarly well.

The `*_2` targets use a different toy problem in which v and u_e are linearly independent, and the `bidomain_2` performs better on this problem (although the scaling of both v and u_e are slightly off, the u_e component is not forced to 0). The `bidomain_godunov_2` code performs about as well as `bidomain_godunov`.