Chaste: Finite Element Implementations

Pras Pathmanathan*

July 18, 2012

This document lists the equations and finite element implementations used in various solvers in the Chaste codebase. Section 1 can be read as an introduction to the stages required in converting Poisson's equation and the heat equation into finite element linear systems, while the remaining sections can be used as references for the equations and the solvers.

Contents

*Please send comments or corrections to pras@cs.ox.ac.uk

1 Finite element solution of simple equations

1.1 Poisson's equation

Let $\Omega \subset \mathbb{R}^n$, and let $\partial\Omega^{\text{dir}}$ and $\partial\Omega^{\text{neu}}$ be two non-intersecting subsets of the boundary of Ω whose union is the entire boundary. Consider Poisson's equation with mixed Dirichlet-Neumann boundary conditions:

$$\begin{aligned} \nabla^2 u &= f(x) \\ u &= u^*(x) \quad \text{on } \partial\Omega^{\text{dir}} \\ \boldsymbol{\nabla} u \cdot \mathbf{n} &= g(x) \quad \text{on } \partial\Omega^{\text{neu}} \end{aligned} \tag{1}$$

where $\mathbf{n}$ is the unit *outward-facing* normal.

The weak form of this equation is found by multiplying by a test function[1] satisfying $v(\partial\Omega^{\text{dir}}) = 0$ (i.e. v is zero on the Dirichlet part of the boundary), and integrating using the divergence theorem:

$$\begin{aligned} 0 &= -\int_\Omega \left(\nabla^2 u\right) v \,\mathrm{d}V + \int_\Omega f v \,\mathrm{d}V \qquad \forall v \in V_0 \\ &= \int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \,\mathrm{d}V - \int_{\partial\Omega} v \boldsymbol{\nabla} u \cdot \mathbf{n} \,\mathrm{d}S + \int_\Omega f v \,\mathrm{d}V \qquad \forall v \in V_0 \\ &= \int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \,\mathrm{d}V - \int_{\partial\Omega^{\text{dir}}} v \boldsymbol{\nabla} u \cdot \mathbf{n} \,\mathrm{d}S - \int_{\partial\Omega^{\text{neu}}} v \boldsymbol{\nabla} u \cdot \mathbf{n} \,\mathrm{d}S + \int_\Omega f v \,\mathrm{d}V \qquad \forall v \in V_0 \end{aligned}$$

Using the facts that v is zero on $\partial\Omega^{\text{dir}}$ and $\boldsymbol{\nabla} u \cdot \mathbf{n} = g$ on $\partial\Omega^{\text{neu}}$, we have the weak form: find u such that $u = u^*$ on $\partial\Omega^{\text{dir}}$ satisfying:

$$\int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \,\mathrm{d}V - \int_{\partial\Omega^{\text{neu}}} g v \,\mathrm{d}S + \int_\Omega f v \,\mathrm{d}V = 0 \qquad \forall v \in V_0$$

For simplicity we now just consider the case $u^* = 0$. The finite element discretisation is obtained by choosing a set of piecewise polynomial basis functions (such as linear basis functions), one for each node in $\Omega \backslash \partial\Omega^{\text{dir}}$, i.e. one for each node at which u is unknown: $\{\phi_1, \phi_2, \ldots, \phi_N\}$, where N is the number of nodes; and restricting the test functions v to just these basis functions, to obtain N equations:

$$\int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} \phi_i \,\mathrm{d}V - \int_{\partial\Omega^{\text{neu}}} g \phi_i \,\mathrm{d}S + \int_\Omega f \phi_i \,\mathrm{d}V = 0 \qquad i = 1, \ldots, N,$$

and then approximating u by $u = \sum_{j=1}^N U_j \phi_j$, where U_j is the approximation of $u(x_j)$. Since $\boldsymbol{\nabla} u$ is then equal to $\sum_{j=1}^N U_j \boldsymbol{\nabla} \phi_j$, this gives

$$\int_\Omega \sum_{j=1}^N U_j \boldsymbol{\nabla} \phi_j \cdot \boldsymbol{\nabla} \phi_i \,\mathrm{d}V - \int_{\partial\Omega^{\text{neu}}} g \phi_i \,\mathrm{d}S + \int_\Omega f \phi_i \,\mathrm{d}V = 0 \qquad i = 1, \ldots, N,$$

or

$$\sum K_{ij} U_j = b_i, \quad i = 1, \ldots, N,$$

where K is the *stiffness matrix*:

$$K_{ij} = \int_\Omega \boldsymbol{\nabla} \phi_i \cdot \boldsymbol{\nabla} \phi_j \,\mathrm{d}V, \tag{2}$$

and

$$b_i = \int_{\partial\Omega^{\text{neu}}} g \phi_i \,\mathrm{d}S - \int_\Omega f \phi_i \,\mathrm{d}V,$$

or in other words, the linear system

$$K\mathbf{U} = \mathbf{b}.$$

[1] From some suitable space V (actually the Sobolev space H^1). We write V_0 for $\{v \in V : v(\partial\Omega^{\text{dir}}) = 0\}$.

1.1.1 Applying Dirichlet boundary conditions

In practice, we allow the test functions to vary over *all* the nodes, and then apply the Dirichlet boundary conditions by altering the rows of the matrix K: zeroing the row and setting the diagonal to be one; and the vector $\mathbf{b}$: setting its value to be the boundary condition value. If k corresponds to a node with a Dirichlet boundary condition, the new row of the linear system looks like:

$$\begin{pmatrix} 0 \dots 0 & 1 & 0 \dots 0 \end{pmatrix} \begin{bmatrix} \vdots \\ U_k \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ u^*(x_k) \\ \vdots \end{bmatrix}$$

(i.e. the equation $U_k = u^*(x_k)$ in matrix form). And actually, in the code, we then alter the matrix and right-hand-side vector again so that the k-th *column* of the matrix becomes zeroed with a one on the diagonal, in order to maintain symmetry.

1.2 The heat equation

Now consider the heat equation with mixed Dirichlet-Neumann boundary conditions and an initial condition: find $u(t, x)$ satisfying

$$\begin{aligned} \frac{\partial u}{\partial t} &= \nabla^2 u \\ u &= u^*(x) \quad \text{on } \partial\Omega^{\text{dir}} \\ \boldsymbol{\nabla} u \cdot \mathbf{n} &= g(x) \quad \text{on } \partial\Omega^{\text{neu}} \\ u(0, x) &= u_0(x) \end{aligned} \tag{3}$$

There are a number of possible time-discretisations. The explicit approach is:

$$\frac{u^{m+1} - u^m}{\Delta t} = \nabla^2 u^m,$$

and the implicit approach is:

$$\frac{u^{m+1} - u^m}{\Delta t} = \nabla^2 u^{m+1}.$$

We choose the latter as it is unconditionally stable, and as with the finite element method (in contrast to the finite difference method), a linear system would still have to be solved if the explicit approach were taken. For the heat equation with a nonlinear source term (such as the monodomain equation—see later):

$$\frac{\partial u}{\partial t} = \nabla^2 u + f(u),$$

we choose a *semi-implicit* discretisation

$$\frac{u^{m+1} - u^m}{\Delta t} = \nabla^2 u^{m+1} + f(u^m), \tag{4}$$

since a fully-implicit discretisation would require the solution of a nonlinear system.

The weak form corresponding to (3) with an implicit time-discretisation is: given u^m, find u^{m+1} such that $u^{m+1} = u^*$ on $\partial\Omega^{\text{dir}}$ satisfying:

$$\frac{1}{\Delta t} \int_\Omega u^{m+1} v \, \mathrm{d}V + \int_\Omega \boldsymbol{\nabla} u^{m+1} \cdot \boldsymbol{\nabla} v \, \mathrm{d}V = \frac{1}{\Delta t} \int_\Omega u^m v \, \mathrm{d}V + \int_{\partial\Omega^{\text{neu}}} g v \, \mathrm{d}S \qquad \forall v \in V_0$$

Letting v be $\phi_1, \ldots, \phi_N$ as before, and with $u = \sum U_j \phi_j$, we get the finite element approximation

$$\frac{1}{\Delta t} M \mathbf{U}^{m+1} + K \mathbf{U}^{m+1} = \frac{1}{\Delta t} M \mathbf{U}^{m} + \mathbf{b},$$

where M is the *mass matrix*

$$M_{ij} = \int_{\Omega} \phi_i \phi_j \, \mathrm{d}V,$$

where K is the stiffness matrix defined in (2), and here (since we have no source term), $\mathbf{b}$ is just

$$b_i = \int_{\partial \Omega^{\mathrm{neu}}} g \phi_i \, \mathrm{d}S.$$

Dirichlet boundary conditions are then applied as described in Section 1.1.1.

2 Chaste PDE solvers

2.1 `SimpleLinearEllipticSolver`

This takes in an `AbstractLinearEllipticPde` and boundary conditions, which overall are of the form

$$\begin{aligned} \nabla \cdot (D(x)\boldsymbol{\nabla} u) + \alpha(x)u + c(x) &= 0 \\ u &= u^*(x) \qquad \text{on } \partial\Omega^{\text{dir}} \\ (D(x)\boldsymbol{\nabla} u)\cdot \mathbf{n} &= g(x) \qquad \text{on } \partial\Omega^{\text{neu}} \end{aligned}$$

where $D(x)$ is a matrix-valued function, and α and c are scalar-valued functions, all provided by the user. *Note that the Neumann boundary condition the user provides corresponds to the value* $(D(x)\boldsymbol{\nabla} u)\cdot\mathbf{n}$ *on the bounday, not* $\boldsymbol{\nabla} u \cdot \mathbf{n}$. When D is the identity matrix and $\alpha \equiv 0$ this is just (1).

The weak form is: find u such that $u = u^*$ on $\partial\Omega^{\text{dir}}$ satisfying:

$$\int_\Omega (D\boldsymbol{\nabla} u)\cdot\boldsymbol{\nabla} v \,\mathrm{d}V - \int_{\partial\Omega^{\text{neu}}} gv \,\mathrm{d}S - \int_\Omega \alpha u v \,\mathrm{d}V - \int_\Omega cv \,\mathrm{d}V = 0 \qquad \forall v \in V_0$$

Letting v be $\phi_1, \ldots, \phi_N$ and with $u = \sum U_j \phi_j$ as before, the finite element approximation is

$$K\mathbf{U} - M\mathbf{U} = \mathbf{b},$$

where here the stiffness matrix is now dependent on the diffusion tensor $D(x)$

$$K_{ij} = \int_\Omega \boldsymbol{\nabla}\phi_i \cdot (D\boldsymbol{\nabla}\phi_j)\,\mathrm{d}V, \tag{5}$$

and the mass-matrix is dependent on $\alpha(x)$

$$M_{ij} = \int_\Omega \alpha\phi_i\phi_j \,\mathrm{d}V,$$

and the right-hand-side vector is

$$b_i = \int_{\partial\Omega^{\text{neu}}} g\phi_i \,\mathrm{d}S + \int_\Omega c\phi_i \,\mathrm{d}V.$$

Dirichlet boundary conditions are then applied as described in Section 1.1.1. (Note also that in the code we do not (currently) distinguish between the two matrices: the 'full' matrix to be assembled is $A = K - M$, so the linear system is $A\mathbf{U} = \mathbf{b}$).

2.2 `SimpleLinearParabolicSolver`

This solver is for parabolic problems defined by an `AbstractLinearParabolicPde` and boundary conditions, which overall are of the form

$$\begin{aligned} k(x)\frac{\partial u}{\partial t} &= \nabla\cdot(D(x)\boldsymbol{\nabla} u) + f(x,u) \\ u &= u^*(x) \qquad \text{on } \partial\Omega^{\text{dir}} \\ \boldsymbol{\nabla} u\cdot\mathbf{n} &= g(x) \qquad \text{on } \partial\Omega^{\text{neu}} \\ u(0,x) &= u_0(x) \end{aligned} \tag{6}$$

where $D(x)$ is a matrix-valued function, and k and f are scalar-valued functions, all provided by the user. Again, the Neumann boundary condition the user provides is $(D(x)\boldsymbol{\nabla} u)\cdot\mathbf{n}$, not $\boldsymbol{\nabla} u\cdot\mathbf{n}$.

Using a semi-implicit discretisation as in (4), the weak form is find u^{m+1} such that $u^{m+1} = u^*$ on $\partial\Omega^{\text{dir}}$ satisfying:

$$\begin{aligned} & \frac{1}{\Delta t}\int_\Omega k u^{m+1} v \,\mathrm{d}V + \int_\Omega \left(D\boldsymbol{\nabla} u^{m+1}\right)\cdot\boldsymbol{\nabla} v \,\mathrm{d}V \\ = \; & \frac{1}{\Delta t}\int_\Omega k u^{m} v \,\mathrm{d}V + \int_\Omega f(x,u^m) v \,\mathrm{d}V + \int_{\partial\Omega^{\text{neu}}} g v \,\mathrm{d}S \qquad \forall v \in V_0 \end{aligned}$$

The finite element approximation is therefore given by

$$\frac{1}{\Delta t} M\mathbf{U}^{m+1} + K\mathbf{U}^{m+1} = \frac{1}{\Delta t} M\mathbf{U}^{m} + \mathbf{c},$$

where K is the diffusion-tensor dependent stiffness matrix given by (5), the mass matrix here depends on $k(x)$

$$M_{ij} = \int_\Omega k\phi_i\phi_j \,\mathrm{d}V,$$

and $\mathbf{c}$ depends on the solution at the previous time

$$c_i = \int_\Omega f(x,u^m)\phi_i \,\mathrm{d}V + \int_{\partial\Omega^{\text{neu}}} g\phi_i \,\mathrm{d}S.$$

Dirichlet boundary conditions are then applied as described in Section 1.1.1. Again, in the code we do not (currently) distinguish between the two matrices: the 'full' matrix to be assembled is $A = K + \frac{1}{\Delta t}M$, and the 'full' right-hand-side vector is $\mathbf{b} = \frac{1}{\Delta t}M\mathbf{U}^m + \mathbf{c}$; and we then solve $A\mathbf{U}^{m+1} = \mathbf{b}$.

2.3 `SimpleNonlinearEllipticSolver`

This takes in an `AbstractNonlinearEllipticPde` and boundary conditions, which overall are of the form

$$\begin{aligned} \nabla\cdot(D(x,u)\boldsymbol{\nabla}u) + f(x,u) &= 0 \\ u &= u^*(x) \qquad \text{on } \partial\Omega^{\text{dir}} \\ (D(x,u)\boldsymbol{\nabla}u)\cdot\mathbf{n} &= g(x) \qquad \text{on } \partial\Omega^{\text{neu}} \end{aligned}$$

The weak form is: find u such that $u = u^*$ on $\partial\Omega^{\text{dir}}$ satisfying:

$$\int_\Omega (D(x,u)\boldsymbol{\nabla}u)\cdot\boldsymbol{\nabla}v \,\mathrm{d}V - \int_{\partial\Omega^{\text{neu}}} g v \,\mathrm{d}S - \int_\Omega f(x,u) v \,\mathrm{d}V = 0 \qquad \forall v \in V_0$$

The *nonlinear* finite element problem is then: find the solution $\mathbf{U}$ of the nonlinear set of equations: $\mathbf{f}(\mathbf{U}) = 0$, where

$$f_i(\mathbf{U}) = \int_\Omega \left(D\left(x, \sum U_j\phi_j\right)\boldsymbol{\nabla}u\right)\cdot\boldsymbol{\nabla}\phi_i \,\mathrm{d}V - \int_{\partial\Omega^{\text{neu}}} g\phi_i \,\mathrm{d}S - \int_\Omega f\left(x, \sum U_j\phi_j\right)\phi_i \,\mathrm{d}V = 0$$

(using $u = \sum U_j\phi_j$), together with the Dirichlet boundary conditions. This can be solved using the Newton (or a Newton-like) method. $\mathbf{f}$ is the residual vector, and the matrix that is computed is the Jacobian, $\frac{\partial f_i}{\partial U_j}$.

3 Cardiac electrophysiology

3.1 The monodomain equations

The monodomain equations are

$$\chi\left(\mathcal{C}\frac{\partial V}{\partial t}+I_{\text{ion}}(\mathbf{u},V)\right)-\nabla\cdot(\sigma\boldsymbol{\nabla}V)+I^{(\text{vol})} = 0, \tag{7}$$

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}(\mathbf{u},V), \tag{8}$$

where $\sigma(x)$ is an effective conductivity[2], χ is the surface-area-to-volume ratio, $\mathcal{C}$ is the capacitance across the membrane, $I^{(\text{vol})}(x)$ the stimulus current (per unit volume) and $I_{\text{ion}}(\mathbf{u},V)$ the ionic current (per unit area) provided by the cell model.

This is exactly the form given by (6), with $k=\chi\mathcal{C}$ (constant), $D=\sigma$ and $f(x,u)=-I^{(\text{vol})}-\chi I_{\text{ion}}(\mathbf{u},V)$. We usually take the Neumann boundary condition to be zero for monodomain problems, although we can also use an surface stimulus, in which case $g=I^{(\text{surf})}(x)$. The weak form and finite element discretisation are therefore those given in Section 2.2, but for easy comparison with the source code we write this out fully. The weak form is: find V^{m+1} satisfying

$$\begin{aligned}&\frac{\chi\mathcal{C}}{\Delta t}\int_\Omega V^{m+1}v\,\mathrm{d}^3\mathbf{x}+\int_\Omega\left(\sigma\boldsymbol{\nabla}V^{m+1}\right)\cdot\boldsymbol{\nabla}v\,\mathrm{d}^3\mathbf{x}\\ =\;&\frac{\chi\mathcal{C}}{\Delta t}\int_\Omega V^{m}v\,\mathrm{d}^3\mathbf{x}-\int_\Omega\left(I^{(\text{vol})}+\chi I_{\text{ion}}(\mathbf{u},V^m)\right)v\,\mathrm{d}^3\mathbf{x}\\ &+\int_{\partial\Omega^{\text{neu}}}I^{(\text{surf})}v\,\mathrm{d}S\qquad\forall v\in V_0\end{aligned}$$

(having used $\mathrm{d}^3\mathbf{x}$ rather than $\mathrm{d}V$ as V denotes voltage), giving the FE problem

$$\left(\frac{\chi\mathcal{C}}{\Delta t}M+K\right)\mathbf{V}^{m+1}=\frac{\chi\mathcal{C}}{\Delta t}M\mathbf{V}^m+\mathbf{c},$$

where $M_{ij}=\int\phi_i\phi_j\,\mathrm{d}^3\mathbf{x}$, $K_{ij}=\int\boldsymbol{\nabla}\phi_i\cdot(\sigma\boldsymbol{\nabla}\phi_j)\,\mathrm{d}^3\mathbf{x}$ and

$$c_i=-\int_\Omega\left(I^{(\text{vol})}+\chi I_{\text{ion}}(\mathbf{u},V^m)\right)\phi_i\,\mathrm{d}^3\mathbf{x}+\int_{\partial\Omega^{\text{neu}}}I^{(\text{surf})}\phi_i\,\mathrm{d}S.$$

The only complication is that the cell model ODEs are given at the nodes (not quadrature points), so $I_{\text{ion}}(\mathbf{u},V)$ is only initially known at the nodes, and would have to be interpolated onto the quadrature points. This leads to the so-called 'matrix-based-assembly' of the RHS vector: interpolating $I_{\text{ion}}(\mathbf{x})$ as $\sum_{j=1}^N(I_{\text{ion}})_j\phi_j$, where $(I_{\text{ion}})_j$ is the ionic current at node j, we get that $\int I_{\text{ion}}\phi_i\,\mathrm{d}^3\mathbf{x}$ is just the product of the mass matrix with the vector of ionic currents. By assuming the stimulus is also defined node-wise and also interpolated linearly (this is the default behaviour in the code), we overall end up with

$$\left(\frac{\chi\mathcal{C}}{\Delta t}M+K\right)\mathbf{V}^{m+1}=\frac{\chi\mathcal{C}}{\Delta t}M\mathbf{V}^m-M\mathbf{F}^m+\mathbf{c}^{(\text{surf})}$$

where $M_{ij}=\int\phi_i\phi_j\,\mathrm{d}^3\mathbf{x}$, $K_{ij}=\int\boldsymbol{\nabla}\phi_i\cdot(\sigma\boldsymbol{\nabla}\phi_j)\,\mathrm{d}^3\mathbf{x}$, and $\mathbf{F}^m$ is the vector of nodal currents and stimuli:

$$F_i^m=(I^{(\text{vol})})_i+\chi(I_{\text{ion}})_i$$

and

$$c_i^{(\text{surf})}=\int_{\partial\Omega^{\text{neu}}}I^{(\text{surf})}\phi_i\,\mathrm{d}S.$$

[2]The monodomain equations apply if the extracellular conductivity σ_e is a multiple of the intracellular conductivity σ_i: $\sigma_e=\lambda\sigma_i$ say. The effective conductivity for the monodomain equation is then $\sigma=\frac{\lambda}{1+\lambda}\sigma_i$.

3.2 The bidomain equations

The bidomain problem in full generality is: find $V(t,x)$ and $\phi_e(t,x)$ satisfying:

$$\begin{aligned}
\chi\left(\mathcal{C}\frac{\partial V}{\partial t} + I_{\text{ion}}(\mathbf{u},V)\right) - \nabla\cdot\left(\sigma_i\boldsymbol{\nabla}\left(V+\phi_e\right)\right) &= -I_i^{(\text{vol})}, &(9)\\
\nabla\cdot\left(\sigma_i\boldsymbol{\nabla}V + \left(\sigma_i+\sigma_e\right)\boldsymbol{\nabla}\phi_e\right) &= I_{\text{total}}^{(\text{vol})}, &(10)\\
\frac{\partial\mathbf{u}}{\partial t} &= \mathbf{f}\left(\mathbf{u},V\right),
\end{aligned}$$

where $I_{\text{total}}^{(\text{vol})} = I_i^{(\text{vol})} + I_e^{(\text{vol})}$, with boundary conditions

$$\begin{aligned}
\mathbf{n}\cdot\left(\sigma_i\boldsymbol{\nabla}\left(V+\phi_e\right)\right) &= I_i^{(\text{surf})}, &(11)\\
\mathbf{n}\cdot\left(\sigma_e\boldsymbol{\nabla}\phi_e\right) &= I_e^{(\text{surf})}. &(12)
\end{aligned}$$

There are now two PDEs, so there are two equations in the weak form: find V, ϕ_e satisfying

$$\begin{aligned}
&\frac{\chi\mathcal{C}}{\Delta t}\int_\Omega V^{m+1}v\,\mathrm{d}^3\mathbf{x} + \int_\Omega\left(\sigma_i\boldsymbol{\nabla}(V^{m+1}+\phi_e^{m+1}\right)\cdot\boldsymbol{\nabla}v\,\mathrm{d}^3\mathbf{x}\\
= &\frac{\chi\mathcal{C}}{\Delta t}\int_\Omega V^m v\,\mathrm{d}^3\mathbf{x} - \int_\Omega\left(I_i^{(\text{vol})} + \chi I_{\text{ion}}(\mathbf{u},V^m)\right)v\,\mathrm{d}^3\mathbf{x}\\
&+\int_{\partial\Omega^{\text{neu}}} I_i^{(\text{surf})}v\,\mathrm{d}S \qquad \forall v\in V_0 \qquad (13)
\end{aligned}$$

and

$$\begin{aligned}
&\int_\Omega\left(\sigma_i\boldsymbol{\nabla}V + \left(\sigma_i+\sigma_e\right)\boldsymbol{\nabla}\phi_e\right)\cdot\boldsymbol{\nabla}v\,\mathrm{d}^3\mathbf{x}\\
= &-\int_\Omega I_{\text{total}}^{(\text{vol})}v\,\mathrm{d}^3\mathbf{x} + \int_{\partial\Omega^{\text{neu}}} I_{\text{total}}^{(\text{surf})}v\,\mathrm{d}S \qquad \forall v\in V_0 \qquad (14)
\end{aligned}$$

(where $I_{\text{total}}^{(\text{surf})}$ is obviously $I_i^{(\text{surf})} + I_e^{(\text{surf})}$).

For the finite element discretisation, we choose a set of basis functions $\psi_1,\psi_2,\ldots,\psi_N$ (now using ψ instead of ϕ for basis functions as the latter denotes electrical potential), set $V=\sum V_k\psi_k$ and $\phi_e=\sum\Phi_k\psi_k$ (now avoiding the use of i as a subscript, since it denotes 'intracellular'), and set $v=\psi_j$ in turn in (13) and (14) to obtain $2N$ equations. For any particular conductivity σ, let us define the stiffness matrix $K[\sigma]$ by

$$\left(K[\sigma]\right)_{jk} = \int_\Omega\boldsymbol{\nabla}\psi_j\cdot\left(\sigma\boldsymbol{\nabla}\psi_k\right)\,\mathrm{d}^3\mathbf{x}.$$

The first N equations are:

$$\frac{\chi\mathcal{C}}{\Delta t}M\mathbf{V}^{m+1} + K[\sigma_i]\mathbf{V}^{m+1} + K[\sigma_i]\boldsymbol{\Phi}_e^{m+1} = \frac{\chi\mathcal{C}}{\Delta t}M\mathbf{V}^m + \mathbf{c}^{(1)},$$

(here $\boldsymbol{\Phi}_e=(\Phi_1,\ldots,\Phi_N)$), where

$$c_j^{(1)} = \int_\Omega -\left(I_i^{(\text{vol})} + \chi I_{\text{ion}}(\mathbf{u},V^m)\right)\psi_j\,\mathrm{d}^3\mathbf{x} + \int_{\partial\Omega^{\text{neu}}} I_i^{(\text{surf})}\psi_j\,\mathrm{d}S.$$

However, exactly as in the monodomain section, we have that

$$\mathbf{c}^{(1)} = -M\mathbf{F}^m + \mathbf{c}^{(\text{surf})}.$$

The second N equations are:

$$K[\sigma_i]\mathbf{V}^{m+1} + K[\sigma_i + \sigma_e]\mathbf{\Phi}_e^{m+1} = \mathbf{d},$$

where

$$d_j = -\int_\Omega I_{\text{total}}^{(\text{vol})}\psi_j \,\mathrm{d}^3\mathbf{x} + \int_{\partial\Omega^{\text{neu}}} I_e^{(\text{surf})}\psi_j \,\mathrm{d}S.$$

Note: in the code, we do not allow the direct specification of $I_e^{(\text{vol})}$, and in order for compatibility conditions to be satisfied[3], force $I_{\text{total}}^{(\text{vol})}$ to be zero (i.e. implicitly choose $I_e^{(\text{vol})} = I_i^{(\text{vol})}$), so that actually

$$d_j = \int_{\partial\Omega^{\text{neu}}} I_{\text{total}}^{(\text{surf})}\psi_j \,\mathrm{d}S$$

only[4].

Overall, we have the $2N$ equations

$$\begin{pmatrix} \frac{\chi\mathcal{C}}{\Delta t}M + K[\sigma_i] & K[\sigma_i] \\ K[\sigma_i] & K[\sigma_i + \sigma_e] \end{pmatrix} \begin{pmatrix} \mathbf{V}^{m+1} \\ \mathbf{\Phi}_e^{m+1} \end{pmatrix} = \begin{pmatrix} \frac{\chi\mathcal{C}}{\Delta t}M\mathbf{V}^m - M\mathbf{F}^m + \mathbf{c}^{(\text{surf})} \\ \mathbf{d} \end{pmatrix}$$

3.3 The bidomain equations with a perfusing bath

For the bidomain problem with a perfusing bath, the implementation quickly becomes more difficult to write, so for clarity let us temporarily take all the stimuli to be zero, and also take $\chi = \mathcal{C} = 1$. We will also occasionally write ϕ_i instead of $V + \phi_e$, again for clarity.

We suppose there are two disjoint domains, Ω (tissue) and Ω_b (the bath), with interface $\partial\Omega$ (the boundary of the tissue). In this problem ϕ_i (and therefore V) is only defined in Ω, but ϕ_e is defined throughout $\Omega \cup \Omega_b$.

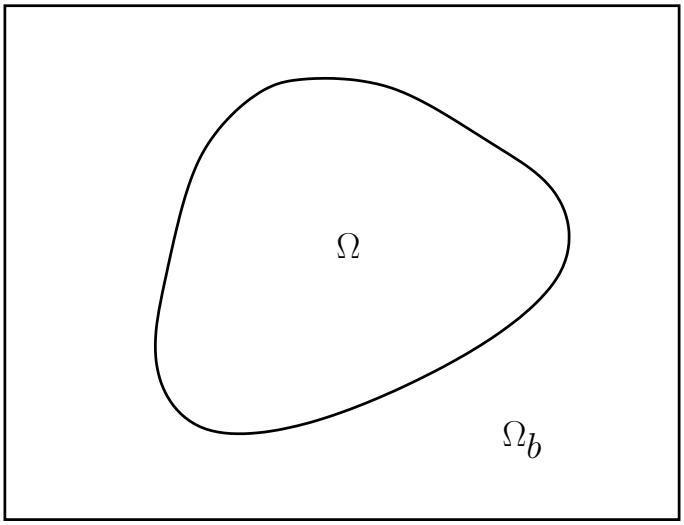

Figure 1: Domains in a model of cardiac tissue contained in a conductive bath.

[3] See Pathmanathan et al., "A numerical guide to the solution of the bidomain equations of cardiac electrophysiology", 2010

[4] $\int_{\partial\Omega^{\text{neu}}} I_{\text{total}}^{(\text{surf})}\,\mathrm{d}S$ also needs to be zero (if no Dirichlet boundary conditions), so the electrodes class, in bidomain with bath problems, makes sure $I_e^{(\text{surf})}$ takes equal and opposite values on the opposite faces of the bath.

The (simplified) problem to be solved is (see also Section 3.3.1): find $V \in H^1(\Omega)$ and $\phi_e \in H^1(\Omega \cup \Omega_b)$ satisfying

$$\begin{aligned} \frac{\partial V}{\partial t} - \nabla \cdot (\sigma_i \boldsymbol{\nabla} \phi_i) + I_{\text{ion}} &= 0, \qquad \text{in } \Omega && (15) \\ \nabla \cdot (\sigma_i \boldsymbol{\nabla} \phi_i + \sigma_e \boldsymbol{\nabla} \phi_e) &= 0, \qquad \text{in } \Omega && (16) \\ \nabla \cdot (\sigma_b \boldsymbol{\nabla} \phi_e) &= 0, \qquad \text{in } \Omega_b && (17) \\ \frac{\partial \mathbf{u}}{\partial t} &= \mathbf{f}(\mathbf{u}, V), \end{aligned}$$

with boundary conditions:

$$\begin{aligned} \mathbf{n} \cdot (\sigma_i \boldsymbol{\nabla} \phi_i) &= 0, \qquad \text{on } \partial\Omega \text{ (i.e. tissue boundary)} && (18) \\ \mathbf{n} \cdot (\sigma_b \boldsymbol{\nabla} \phi_e) &= 0, \qquad \text{on } \partial\Omega_b \backslash \partial\Omega \text{ (i.e. bath boundary)} && (19) \end{aligned}$$

and suitable initial conditions. An interface boundary condition is also required: it is

$$\mathbf{n} \cdot (\sigma_e \boldsymbol{\nabla} \phi_e) \big|_{\partial\Omega^{\text{tiss}}} + \mathbf{n} \cdot (\sigma_b \boldsymbol{\nabla} \phi_e) \big|_{\partial\Omega^{\text{bath}}} = 0 \qquad \text{on } \partial\Omega \tag{20}$$

(where $\big|_{\partial\Omega^{\text{tiss}}}$ denotes the evaluation on $\partial\Omega$ as a limit of from the tissue side and $\big|_{\partial\Omega^{\text{bath}}}$ the bath side). This is the condition which will arise naturally in the weak form (see below).

The first equation of weak form is found by multiplying (15) by $v \in H^1(\Omega)$ and integrating using the divergence theorem:

$$\begin{aligned} 0 &= \int_\Omega \frac{\partial V}{\partial t} v \, \mathrm{d}^3\mathrm{x} + \int_\Omega (\sigma_i \boldsymbol{\nabla} \phi_i) \cdot \boldsymbol{\nabla} v \, \mathrm{d}^3\mathrm{x} - \int_{\partial\Omega} v \, (\sigma_i \boldsymbol{\nabla} \phi_i) \cdot \mathbf{n} \, \mathrm{d}S + \int_\Omega I_{\text{ion}} v \, \mathrm{d}^3\mathrm{x} \qquad \forall v \in H^1(\Omega) \\ &= \int_\Omega \frac{\partial V}{\partial t} v \, \mathrm{d}^3\mathrm{x} + \int_\Omega (\sigma_i \boldsymbol{\nabla} \phi_i) \cdot \boldsymbol{\nabla} v \, \mathrm{d}^3\mathrm{x} + \int_\Omega I_{\text{ion}} v \, \mathrm{d}^3\mathrm{x} \qquad \forall v \in H^1(\Omega) \end{aligned}$$

the boundary integral vanishing due to (18).

The second equation in the weak form is found by multiplying (16) and (17) (essentially one equation over the whole domain $\Omega \cup \Omega_b$) by $w \in H^1(\Omega \cup \Omega_b)$ (note the larger domain) and integrating using the divergence theorem:

$$\begin{aligned} 0 = \quad & \int_\Omega (\sigma_i \boldsymbol{\nabla} \phi_i + \sigma_e \boldsymbol{\nabla} \phi_e) \cdot \boldsymbol{\nabla} w \, \mathrm{d}^3\mathrm{x} - \int_{\partial\Omega} w \, (\sigma_i \boldsymbol{\nabla} \phi_i + \sigma_e \boldsymbol{\nabla} \phi_e) \cdot \mathbf{n} \big|_{\partial\Omega^{\text{tiss}}} \, \mathrm{d}S \\ & + \int_{\Omega_b} (\sigma_b \boldsymbol{\nabla} \phi_e) \cdot \boldsymbol{\nabla} w \, \mathrm{d}^3\mathrm{x} - \int_{\partial\Omega} w \, (\sigma_b \boldsymbol{\nabla} \phi_e) \cdot \mathbf{n} \big|_{\partial\Omega^{\text{bath}}} \, \mathrm{d}S - \int_{\partial\Omega_b} w \, (\sigma_b \boldsymbol{\nabla} \phi_e) \cdot \mathbf{n} \, \mathrm{d}S \quad \forall w \in H^1(\Omega \cup \Omega_b) \end{aligned}$$

Here, the last boundary integral vanishes due to the boundary condition (19), as does the first part of the first boundary integral, $\int_{\partial\Omega} w \, (\sigma_i \boldsymbol{\nabla} \phi_i) \cdot \mathbf{n} \big|_{\partial\Omega^{\text{tiss}}} \, \mathrm{d}S$, due to (18). The remaining boundary term is

$$- \int_{\partial\Omega} w \, (\sigma_e \boldsymbol{\nabla} \phi_e) \cdot \mathbf{n} \big|_{\partial\Omega^{\text{tiss}}} \, \mathrm{d}S - \int_{\partial\Omega} w \, (\sigma_b \boldsymbol{\nabla} \phi_e) \cdot \mathbf{n} \big|_{\partial\Omega^{\text{bath}}} \, \mathrm{d}S.$$

$w \in H^1(\Omega \cup \Omega_b)$, so w continuous across the interface, i.e. $w\big|_{\partial\Omega^{\text{tiss}}} = w\big|_{\partial\Omega^{\text{bath}}}$, so the above is equal to

$$- \int_{\partial\Omega} w \left((\sigma_i \boldsymbol{\nabla} \phi_i) \cdot \mathbf{n} \big|_{\partial\Omega^{\text{tiss}}} + (\sigma_b \boldsymbol{\nabla} \phi_e) \cdot \mathbf{n} \big|_{\partial\Omega^{\text{bath}}} \right) \mathrm{d}S$$

which is zero due to the interface condition (20). Hence all the surface integrals in the second equation of the weak form also vanish.

The weak problem is therefore (see also Section 3.3.1): find $V \in H^1(\Omega)$ and $\phi_e \in H^1(\Omega \cup \Omega_b)$ satisfying (initial conditions and):

$$\int_\Omega \frac{\partial V}{\partial t} v \, \mathrm{d}^3\mathrm{x} + \int_\Omega (\sigma_i \boldsymbol{\nabla} \phi_i) \cdot \boldsymbol{\nabla} v \, \mathrm{d}^3\mathrm{x} + \int_\Omega I_{\text{ion}} v \, \mathrm{d}^3\mathrm{x} = 0 \quad \forall v \in H^1(\Omega) \tag{21}$$

and

$$\int_\Omega (\sigma_i \boldsymbol{\nabla}\phi_i + \sigma_e \boldsymbol{\nabla}\phi_e)\cdot\boldsymbol{\nabla} w \,\mathrm{d}^3\mathrm{x} + \int_{\Omega_b} (\sigma_b \boldsymbol{\nabla}\phi_e)\cdot\boldsymbol{\nabla} w \,\mathrm{d}^3\mathrm{x} = 0 \quad \forall w \in H^1(\Omega\cup\Omega_b) \tag{22}$$

For the finite element discretisation, assume for convenience that Ω and Ω_b are open (i.e. $\partial\Omega$ is not contained in either Ω or Ω_b. Let $K < N < N+M$, and suppose there are K nodes in the interior of Ω, $N-K$ nodes on the boundary $\partial\Omega$, and M nodes in Ω_b:

$$\begin{aligned} \mathbf{x}_1,\ldots,\mathbf{x}_K &\in \Omega \qquad \text{and therefore } \notin \partial\Omega \\ \mathbf{x}_{K+1},\ldots,\mathbf{x}_N &\in \partial\Omega \\ \mathbf{x}_{N+1},\ldots,\mathbf{x}_{N+M} &\in \Omega_b \qquad \text{and therefore } \notin \partial\Omega \end{aligned}$$

The basis functions are then

$$\underbrace{\psi_1,\ldots,\psi_K}_{=0 \text{ in } \Omega_b}, \underbrace{\psi_{K+1},\ldots,\psi_N}_{\neq 0 \text{ in } \Omega \text{ or } \Omega_b}, \underbrace{\psi_{N+1},\ldots,\psi_{N+M}}_{=0 \text{ in } \Omega}$$

We can write $V = \sum_{j=1}^N V_j\psi_j$. This technically would give V non-zero in a small band outside Ω, where V isn't defined, so this has to be understood to apply only for $\mathbf{x}\in\Omega\cup\partial\Omega$ $(=\overline{\Omega})$. Also, $\phi_e = \sum_{j=1}^{N+M}\Phi_j\psi_j$. Let $\mathbf{V} = (V_1,\ldots,V_N)$, $\boldsymbol{\Phi} = (\Phi_1,\ldots,\Phi_{N+M})$ and define $\boldsymbol{\Phi}_{(1)} = (\Phi_1,\ldots,\Phi_N)$, i.e. the first N components of $\boldsymbol{\Phi}$.

The final finite element linear system will of size $2N+M$. The first N equations are given by setting $v=\psi_j$, $j=1,\ldots,N$ in (21). Since $\psi_{N+1},\ldots,\psi_{N+M}$ are zero in $\Omega\cup\partial\Omega$, this equation is only dependent on $\Phi_1,\ldots,\Phi_N$, i.e. on $\boldsymbol{\Phi}_{(1)}$ rather than the full $\boldsymbol{\Phi}$. We get, as in Section 3.2,

$$\frac{1}{\Delta t}M\mathbf{V}^{m+1} + K[\sigma_i]\left(\mathbf{V}^{m+1} + \boldsymbol{\Phi}_{(1)}^{m+1}\right) = \frac{1}{\Delta t}M\mathbf{V}^m + \mathbf{c}^{(1)} \tag{23}$$

where M in the $N\times N$ mass stiffness, and $K[\sigma_i]$ is the $N\times N$ stiffness matrix using conductivity σ_i.

The remaining $N+M$ equations are obtained by setting $w=\psi_1,\ldots,\psi_{N+M}$ in (22). This gives

$$\begin{aligned} 0 \;=\; & \sum_{k=1}^{N} V_k \int_\Omega (\sigma_i\boldsymbol{\nabla}\psi_k)\cdot\boldsymbol{\nabla}\psi_j \,\mathrm{d}^3\mathbf{x} \\ & + \sum_{k=1}^{N+M}\Phi_k\left(\int_\Omega ((\sigma_i+\sigma_e)\boldsymbol{\nabla}\psi_k)\cdot\boldsymbol{\nabla}\psi_j\,\mathrm{d}^3\mathbf{x} + \int_{\Omega_b}(\sigma_b\boldsymbol{\nabla}\psi_k)\cdot\boldsymbol{\nabla}\psi_j\,\mathrm{d}^3\mathbf{x}\right) \qquad j=1,\ldots,N+M \end{aligned} \tag{24}$$

The first term here is: $K[\sigma_i]\mathbf{V}$ for equations $j=1,\ldots,N$; and zero for equations $j=N+1,\ldots,N+M$ (as then $\psi_j=0$ in Ω), i.e.

$$\begin{pmatrix} K[\sigma_i]\mathbf{V} \\ \mathbf{0} \end{pmatrix} \begin{matrix} \} \text{ size } N \\ \} \text{ size } M \end{matrix}$$

The second term can be written as the product $\mathcal{K}\boldsymbol{\Phi}$, where $\mathcal{K}$ is the $(N+M)\times(N+M)$ matrix:

$$\mathcal{K}_{jk} = \int_\Omega ((\sigma_i+\sigma_e)\boldsymbol{\nabla}\psi_k)\cdot\boldsymbol{\nabla}\psi_j\,\mathrm{d}^3\mathbf{x} + \int_{\Omega_b}(\sigma_b\boldsymbol{\nabla}\psi_k)\cdot\boldsymbol{\nabla}\psi_j\,\mathrm{d}^3\mathbf{x} \tag{25}$$

which we write in 2 by 2 block matrix form

$$\mathcal{K} = \begin{pmatrix} \mathcal{K}_{(1,1)} & \mathcal{K}_{(1,2)} \\ \mathcal{K}_{(2,1)} & \mathcal{K}_{(2,2)} \end{pmatrix} \begin{matrix} \} \text{ size } N \\ \} \text{ size } M \end{matrix}$$

So overall we have that (24) in matrix form is:

$$\begin{bmatrix} K[\sigma_i] & \mathcal{K}_{(1,1)} & \mathcal{K}_{(1,2)} \\ 0 & \mathcal{K}_{(2,1)} & \mathcal{K}_{(2,2)} \end{bmatrix} \begin{bmatrix} \mathbf{V}^{m+1} \\ \boldsymbol{\Phi}_{(1)}^{m+1} \\ \boldsymbol{\Phi}_{(2)}^{m+1} \end{bmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \begin{matrix} \} \text{ first } N \text{ equations of (24)} \\ \} \text{ next } M \text{ equations of (24)} \end{matrix} \tag{26}$$

Finally, we put together (23) and (26) to get the full finite element system (see also Section 3.3.1)

$$\begin{pmatrix} \frac{1}{\Delta t}M + K[\sigma_i] & K[\sigma_i] & 0 \\ K[\sigma_i] & \mathcal{K}_{(1,1)} & \mathcal{K}_{(1,2)} \\ 0 & \mathcal{K}_{(2,1)} & \mathcal{K}_{(2,2)} \end{pmatrix} \begin{bmatrix} \mathbf{V}^{m+1} \\ \mathbf{\Phi}^{m+1}_{(1)} \\ \mathbf{\Phi}^{m+1}_{(2)} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}M\mathbf{V}^m + \mathbf{c}^{(1)} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \} \text{ size } N \\ \} \text{ size } N \\ \} \text{ size } M \end{matrix}$$

Note that $\mathcal{K}_{(1,2)} = \mathcal{K}_{(2,1)}$ and is nearly all zero—it is only non-zero at values corresponding to nodes on or near $\partial\Omega$.

In practice, for implementation/parallelisation reasons, we introduce a set of dummy voltages values at the bath nodes, $V_{N+1}, \ldots, V_{N+M}$, and introduce the extra equations

$$V_j = 0, \qquad j = N+1, \ldots, N+M.$$

Letting $\mathbf{V}_{(1)} = \mathbf{V}$, and $\mathbf{V}_{(2)} = (V_{N+1}, \ldots, V_{N+M})$ (i.e. the vector of dummy values), and letting I_M denote the M by M identity matrix, we have

$$\begin{bmatrix} \frac{1}{\Delta t}M + K[\sigma_i] & 0 & K[\sigma_i] & 0 \\ 0 & I_M & 0 & 0 \\ K[\sigma_i] & 0 & \mathcal{K}_{(1,1)} & \mathcal{K}_{(1,2)} \\ 0 & 0 & \mathcal{K}_{(2,1)} & \mathcal{K}_{(2,2)} \end{bmatrix} \begin{bmatrix} \mathbf{V}^{m+1}_{(1)} \\ \mathbf{V}^{m+1}_{(2)} \\ \mathbf{\Phi}^{m+1}_{(1)} \\ \mathbf{\Phi}^{m+1}_{(2)} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}M\mathbf{V}^m + \mathbf{c}^{(1)} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \} \text{ size } N \\ \} \text{ size } M \\ \} \text{ size } N \\ \} \text{ size } M \end{matrix}$$

3.3.1 The bidomain problem with a bath, including stimuli and parameters

We know re-state the above without the simplications. The problem to be solved is: find $V \in H^1(\Omega)$ and $\phi_e \in H^1(\Omega \cup \Omega_b)$ satisfying

$$\begin{aligned} \chi\mathcal{C}\frac{\partial V}{\partial t} - \nabla\cdot(\sigma_i\boldsymbol{\nabla}\phi_i) + \chi I_{\text{ion}} + I_i^{(\text{vol})} &= 0, && \text{in } \Omega \\ \nabla\cdot(\sigma_i\boldsymbol{\nabla}\phi_i + \sigma_e\boldsymbol{\nabla}\phi_e) &= 0, && \text{in } \Omega \\ \nabla\cdot(\sigma_b\boldsymbol{\nabla}\phi_e) &= 0, && \text{in } \Omega_b \\ \frac{\partial \mathbf{u}}{\partial t} &= \mathbf{f}(\mathbf{u}, V), \end{aligned}$$

with boundary conditions:

$$\begin{aligned} \mathbf{n}\cdot(\sigma_i\boldsymbol{\nabla}\phi_i) &= I_i^{(\text{surf})}, && \text{on } \partial\Omega \text{ (i.e. tissue boundary)} \\ \mathbf{n}\cdot(\sigma_b\boldsymbol{\nabla}\phi_e) &= I_e^{(\text{surf})}, && \text{on } \partial\Omega_b\backslash\partial\Omega \text{ (i.e. bath boundary)} \end{aligned}$$

and interface boundary condition

$$\mathbf{n}\cdot(\sigma_e\boldsymbol{\nabla}\phi_e)\big|_{\partial\Omega^{\text{tiss}}} + \mathbf{n}\cdot(\sigma_b\boldsymbol{\nabla}\phi_e)\big|_{\partial\Omega^{\text{bath}}} = 0 \qquad \text{on } \partial\Omega$$

Note: we are assuming $I_{\text{total}}^{(\text{vol})} = 0$ in the 2nd PDE, as described in the end of Section 3.2, i.e. implicitly choosing $I_e^{(\text{vol})} = -I_i^{(\text{vol})}$; we have not allowed a volume bath stimulus; and $I_e^{(\text{surf})}$, acting on the bath boundary only, corresponds to electrodes.

The finite element problem is

$$\begin{bmatrix} \frac{\chi\mathcal{C}}{\Delta t}M + K[\sigma_i] & 0 & K[\sigma_i] & 0 \\ 0 & I_M & 0 & 0 \\ K[\sigma_i] & 0 & \mathcal{K}_{(1,1)} & \mathcal{K}_{(1,2)} \\ 0 & 0 & \mathcal{K}_{(2,1)} & \mathcal{K}_{(2,2)} \end{bmatrix} \begin{bmatrix} \mathbf{V}^{m+1}_{(1)} \\ \mathbf{V}^{m+1}_{(2)} \\ \mathbf{\Phi}^{m+1}_{(1)} \\ \mathbf{\Phi}^{m+1}_{(2)} \end{bmatrix} = \begin{bmatrix} \frac{\chi\mathcal{C}}{\Delta t}M\mathbf{V}^m + \mathbf{c}^{(1)} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{c}^{(2)} \end{bmatrix} \begin{matrix} \} \text{ size } N \\ \} \text{ size } M \\ \} \text{ size } N \\ \} \text{ size } M \end{matrix}$$

where $M_{jk} = \int \psi_j \psi_k \, \mathrm{d}^3\mathbf{x}$ is the standard mass matrix, $K[\sigma]$ is defined by (3.2), $\mathcal{K}$ is defined by (25), and

$$\begin{aligned} c_j^{(1)} &= \int_\Omega - \left(I_i^{(\mathrm{vol})} + \chi I_{\mathrm{ion}}(\mathbf{u}, V^m) \right) \psi_j \, \mathrm{d}^3\mathbf{x} + \int_{\partial\Omega} I_i^{(\mathrm{surf})} \psi_j \, \mathrm{d}S, \\ c_j^{(2)} &= \int_{\partial\Omega_b} I_e^{(\mathrm{surf})} \psi_{j+N} \, \mathrm{d}S. \end{aligned}$$

4 Solid mechanics

We now describe how to solve solid mechanics problems using the finite element method. We assume the material in question undergoes large deformations, which means that nonlinear elasticity (also known as finite elasticity) has to be used, and is hyperelastic, which means there exists a strain-energy function (to be defined shortly). We will consider both the incompressible and compressible cases.

4.1 Formulation

Let $\Omega_0 \subset \mathbb{R}^3$ denote a body in its undeformed, stress-free configuration, and let $\mathbf{X}$ be a point in Ω_0. Let the deformed configuration, under some given loads, be given by Ω, and let $\mathbf{x}(\mathbf{X}) \in \Omega$ be the corresponding point in a deformed configuration.

In compressible elasticity, the unknowns to be computed are the deformed positions $\mathbf{x}$ (or equivalently the displacements $\mathbf{u} = \mathbf{x} - \mathbf{X}$).

In incompressible elasticity, there are two unknowns, the deformation $\mathbf{x}$ and the pressure p. The pressure is essentially a Lagrangian multiplier arising from the constraint of incompressibility (see below).

4.1.1 Kinematics

The important deformation-based quantities are the deformation gradient and deformation tensor. The deformation gradient is defined to be

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}, \qquad \text{i.e.} \qquad F_{iM} = \frac{\partial x_i}{\partial X_M},$$

from which the Green deformation tensor is defined to be

$$C = F^{\mathrm{T}} F \qquad \text{i.e.} \qquad C_{MN} = F_{iM} F_{iN},$$

The Lagrangian strain tensor is $E = \frac{1}{2}(C - I)$ but it is easier to work with C rather than E.

Now, $\det(F)$ must be positive everywhere; this is the constraint of non-interpenetrability. In incompressible problems, the constraint of incompressibility can be stated as $\det(F) = 1$ everywhere in the body, or, equivalently, $\det(C) = 1$.

The following variables and notation are often used: $J = \det(F)$, $I_1 = \mathrm{tr}(C)$, $I_2 = \frac{1}{2}((\mathrm{tr}(C))^2 - \mathrm{tr}(C^2))$, and $I_3 = \det C$ (so $J^2 = I_3$). The latter three variables are the *principal invariants* of C.

4.1.2 Stress

There are three important stress tensors: the *Cauchy stress*, σ or σ_{ij}, which is a measure of the true stress in the body, the force acting on a surface in the deformed body, per unit deformed area; the first Piola-Kirchhoff tensor, S or S_{Mi}, which measures the force acting on a surface in the deformed body per unit *undeformed* area; and the *second Piola-Kirchhoff stress*, T, or T_{MN}, which is a transformed (non-physical) stress, the force 'acting' on a surface in the *undeformed* body, per unit *undeformed* area. σ and T are symmetric. The relationship between the stresses are

$$\sigma = \frac{1}{\det F} F S, \qquad T = S F^{-\mathrm{T}}, \qquad \sigma = \frac{1}{\det F} F T F^{\mathrm{T}}$$

To link stress and strain (or stress and deformation), a material law is required, which is a nonlinear functional relationship linking stress and deformation. The material law is material-dependent and can only be determined by experiment. For hyper-elastic materials, the law is defined via a strain-energy function (see Section 4.2).

For compressible materials, the material law is a relationship between stress and deformation gradient: $T \equiv T(C)$ (or alternatively, $T \equiv T(E)$, $S \equiv S(F)$, $\sigma = \sigma(F)$, etc). For incompressible materials, the stress is also dependent on the pressure: $T \equiv T(C, p)$.

Example material relationships are given in Section 4.2.

4.1.3 Equilibrium equations

Let $\mathbf{b}$ be the body force per unit mass (generally equal to $(0, 0, -g)$ if the effect of gravity is not neglected, or zero otherwise), and ρ_0 the density. The equation of *static* equilibrium, which determines the new configuration $\mathbf{x}(\mathbf{X})$, given a material law, is $\nabla \cdot S + \rho_0 \mathbf{b} = 0$—"divergence of stress plus body force equals zero." Written in terms of the 2nd Piola-Kirchoff stress, this becomes

Compressible case: find $\mathbf{x} \equiv \mathbf{x}(\mathbf{X})$, given a material law $T \equiv T(C(\mathbf{x}))$, satisfying

$$\frac{\partial}{\partial X_M}\left(T_{MN}(\mathbf{x})F_{iN}(\mathbf{x})\right) + \rho_0 b_i = 0 \qquad \text{in } \Omega_0, \tag{27}$$

Suitable boundary conditions are the specification of the deformation on part of the boundary of Ω_0 and surface traction on the remainder of $\partial\Omega_0$:

$$\begin{aligned} \mathbf{x} &= \mathbf{x}^* && \text{on } \partial\Omega_0^{\text{disp}} \\ TF^{\mathrm{T}}\mathbf{N} &= \mathbf{s} && \text{on } \partial\Omega_0^{\text{trac}} \end{aligned}$$

where $\mathbf{x}^*$ is the specified deformation, $\mathbf{N}$ is the undeformed unit normal, $\mathbf{s}$ is a specified surface traction (force per unit area), and $\partial\Omega_0^{\text{disp}}$ and $\partial\Omega_0^{\text{trac}}$ are disjoint subsets of $\partial\Omega_0$ whose union makes up $\partial\Omega_0$.

In incompressible elasticity, we have in addition the constraint of incompressibility, which is $\det F = 1$, and the fact that the stress T is dependent on the pressure.

Incompressible case: find $\mathbf{x} \equiv \mathbf{x}(\mathbf{X})$ and $p \equiv p(\mathbf{X})$, given a material law $T \equiv T(C(\mathbf{x}), p)$, satisfying

$$\frac{\partial}{\partial X_M}\left(T_{MN}(\mathbf{x}, p)F_{iN}(\mathbf{x})\right) + \rho_0 b_i = 0, \tag{28}$$

$$\det F = 1, \tag{29}$$

where $\mathbf{b}$ is the body force per unit mass, ρ_0 is the density, and with boundary conditions as above:

$$\begin{aligned} \mathbf{x} &= \mathbf{x}^* && \text{on } \partial\Omega_0^{\text{disp}} \\ TF^{\mathrm{T}}\mathbf{N} &= \mathbf{s} && \text{on } \partial\Omega_0^{\text{trac}} \end{aligned}$$

4.2 Hyper-elasticity and material laws

To close the above formulation we need to specify the relationship between T and C.

Consider the compressible case first. The definition of hyper-elasticity is that there exists a strain energy function $W \equiv W(E)$, from which the 2nd Piola-Kirchhoff stress is determined by $T = \frac{\partial W}{\partial E}$. Equivalently, $W \equiv W(C)$ and $T = 2\frac{\partial W}{\partial C}$.

In the *isotropic* compressible case, the strain energy becomes dependent on just the three principal invariants of C, i.e. $W \equiv W(I_1, I_2, I_3)$. Defining the notation $w_k = \frac{\partial W}{\partial I_k}(I_1, I_2, I_3)$, the stress is then given by

$$T = 2w_1\frac{\partial I_1}{\partial C} + 2w_2\frac{\partial I_2}{\partial C} + 2w_3\frac{\partial I_3}{\partial C},$$

which, using some useful formulae[5], becomes

$$T = 2\frac{\partial W}{\partial C} = 2w_1 I + 2w_2(I_1 I - C) + 2w_3 I_3 C^{-1}. \tag{30}$$

[5]For a (general, not-necessarily symmetric) matrix A: for $I_1(A)$, $\frac{\partial I_1}{\partial A_{MN}} = \delta_{MN}$ and $\frac{\mathrm{d}^2 I_1}{\mathrm{d}A_{MN}\partial A_{PQ}} = 0$; for $I_2(A)$, $\frac{\partial I_2}{\partial A_{MN}} = I_1\delta_{MN} - A_{MN}$ and $\frac{\partial^2 I_2}{\partial A_{MN}\partial A_{PQ}} = \delta_{MN}\delta_{PQ} - \delta_{MP}\delta_{NQ}$; and for $I_3 = \det(A)$, $\frac{\partial(\det(A))}{\partial A_{pq}} = \det(A)(A^{-1})_{qp}$ and $\frac{\partial A^{-1}_{MN}}{\partial A_{PQ}} = -A^{-1}_{MP}A^{-1}_{QN}$ (differentiate $A^{-1}A = I$).

Any compressible material law should satisfy $T(I) = 0$, i.e. zero stress when no deformation, which provides a constraint on material parameters.

To implement isotropic compressible material laws, w_1, w_2 and w_3, as well as each second derivative w_{ij}, needs to be calculated and coded—see `AbstractIsotropicCompressibleMaterialLaw` and child classes. Compressible material laws are often written in terms of the *deviatoric invariants*, $\bar{I}_1 = I_1 I_3^{-\frac{1}{3}}$, and $\bar{I}_2 = I_2 I_3^{-\frac{2}{3}}$ (these are the strain invariants of C after being scaled to have unit determinant—see for example [Horgan and Saccomandi, Journal of Elasticity, 2004] for more details). For example, the Neo-Hookean material law is

$$\begin{aligned} W(I_1, I_2, I_3) &= c_1(\bar{I}_1 - 3) + c_3(J-1)^2, \\ &= c_1(I_1 I_3^{-\frac{1}{3}} - 3) + c_3(I_3^{\frac{1}{2}} - 1)^2, \end{aligned}$$

(note the $-1/3$ here is 3D-specific).

Incompressible laws are similar except they depend on the pressure and I_3 in a specific way:

$$W(C) = W^{\text{mat}}(C) - \frac{p}{2}(I_3 - 1),$$

where $W^{\text{mat}}(C)$ is material-dependent and to be measured experimentally. This gives the stress

$$T = 2\frac{\partial W^{\text{mat}}}{\partial C} - pC^{-1}.$$

Incompresible laws have a particular pressure p_0, generally non-zero, which is present in the undeformed body, i.e. satisfying $T(I, p_0) = 0$.

In the isotropic incompressible case, we have $W = W^{\text{mat}}(I_1, I_2) - \frac{p}{2}(I_3 - 1)$, and $T = 2w_1^{\text{mat}} I + 2w_2^{\text{mat}}(I_1 I - C) - pC^{-1}$.

4.3 Weak form

For the weak form, we use the notation $\delta\mathbf{x}$ for the test functions (one vector-valued function rather than, say, three function v_1, v_2, v_3). Write $\mathcal{V}$ for the space of deformations[6]. Let $\mathcal{V}_0$ be the subspace of deformations which are zero on $\partial\Omega_0^{\text{disp}}$, i.e. $\mathcal{V}_0 = \{\mathbf{y} \in \mathcal{V} : \mathbf{y}(\mathbf{X}) = 0 \text{ if } \mathbf{X} \in \partial\Omega_0^{\text{disp}}\}$.

The weak form for the compressible equilibrium equation (27) is obtained by taking the inner product of (27) with $\delta\mathbf{x}$ and integrating using the divergence theorem, from which we obtain:

Compressible case: find $\mathbf{x} \in \mathcal{V}$ such $\mathbf{x} = \mathbf{x}^*$ on $\partial\Omega_0^{\text{disp}}$ and

$$\int_{\Omega_0} T_{MN} \frac{\partial x_i}{\partial X_N} \frac{\partial(\delta x_i)}{\partial X_M} \,\mathrm{d}V_0 = \int_{\Omega_0} \rho_0 b_i \delta x_i \,\mathrm{d}V_0 + \int_{\partial\Omega_0^{\text{trac}}} s_i \delta x_i \,\mathrm{d}S_0 \qquad \forall\, \delta\mathbf{x} \in \mathcal{V}_0 \tag{31}$$

For the incompressible case, we also have the constraint equation, which we have to multiply with a test function δp, from a suitable space[7] $\mathcal{W}$, and integrate.

Incompressible case: find $\mathbf{x} \in \mathcal{V}$ and $p \in \mathcal{W}$ such $\mathbf{x} = \mathbf{x}^*$ on $\partial\Omega_0^{\text{disp}}$ and

$$\int_{\Omega_0} T_{MN} \frac{\partial x_i}{\partial X_N} \frac{\partial(\delta x_i)}{\partial X_M} \,\mathrm{d}V_0 - \int_{\Omega_0} \rho_0 b_i \delta x_i \,\mathrm{d}V_0 - \int_{\partial\Omega_0^{\text{trac}}} s_i \delta x_i \,\mathrm{d}S_0 = 0 \qquad \forall\, \delta\mathbf{x} \in \mathcal{V}_0 \tag{32}$$

$$\int_{\Omega_0} \delta p \,(\det F - 1) \,\mathrm{d}V_0 = 0 \qquad \forall\, \delta p \in \mathcal{W} \tag{33}$$

(This can also be written by summing these two equations and saying $\forall\, \delta\mathbf{x} \in \mathcal{V}_0, \delta p \in \mathcal{W}$).

[6] This is the space $H^1(\Omega)^3$.
[7] $L^2(\Omega)$.

4.4 Normal pressure on deformed surface boundary conditions

The traction boundary conditions stated in Section 4.1.3 are the *natural boundary conditions* for this problem, i.e. those that arise naturally when writing down the weak form. They represent however a traction specified on surfaces on the *undeformed* body. More physically realistic boundary conditions are that there is an external pressure applied to surfaces on the *deformed* body, and in particular acting in the normal direction. Specifically, the boundary condition is

$$\sigma_{ij} n_j = P n_i \quad \text{on } \partial\Omega^{\text{trac}}$$

where $\mathbf{n}$ is the normal on the deformed body. This corresponds to a *deformation-dependent* boundary condition (i.e. $\mathbf{s} \equiv \mathbf{s}(\mathbf{x})$) on the undeformed state of

$$\mathbf{s} = JPF^{-\mathrm{T}}\mathbf{N} \tag{34}$$

Hence in this situation, the boundary integral term in the weak form is also nonlinear, like the volume integrals.

4.5 Finite element discretisation

For the FEM implementation we just consider the incompressible case—the compressible case just involves ignoring the pressure equations or blocks.

Suppose we have a mesh in which $\mathbf{x}$ will be solved for at $\mathcal{N}$ nodes and p will be solved for at $\mathcal{M}$ nodes. For example, when using a quadratic mesh, with quadratic interpolation for displacement (i.e. for $\mathbf{x}$), and linear interpolation for pressure[8], then $\mathcal{N}$ is the total number of nodes, and $\mathcal{M}$ is the number of vertices. Now, the total number of unknowns is $3\mathcal{N} + \mathcal{M}$ (assuming a 3D problem). Let $\phi_1, \phi_2, \ldots, \phi_{\mathcal{N}}$ by the bases used for displacement (in this example, quadratic bases), and $\psi_1, \psi_2, \ldots, \psi_{\mathcal{M}}$ those for pressure (in this example, linear bases).

Let the unknown $\mathbf{x}$-values at the nodes be denoted $\boldsymbol{\mathcal{X}}_I = ((\mathcal{X}_I)_1, (\mathcal{X}_I)_2, (\mathcal{X}_I)_3) = (\mathcal{X}_I, \mathcal{Y}_I, \mathcal{Z}_I)$, so that $\mathbf{x} = \sum_{I=1}^{\mathcal{N}} \boldsymbol{\mathcal{X}}_I \phi_I$. Similarly, let the unknown pressures be $\mathcal{P}_1, \ldots, \mathcal{P}_{\mathcal{M}}$, so $p = \sum_{I=1}^{\mathcal{M}} \mathcal{P}_I \psi_I$. Let us write the vector of all the unknowns as

$$\boldsymbol{\mathcal{A}} = (\mathcal{X}_1, \ldots, \mathcal{X}_{\mathcal{N}}, \mathcal{Y}_1, \ldots, \mathcal{Y}_{\mathcal{N}}, \mathcal{Z}_1, \ldots, \mathcal{Z}_{\mathcal{N}}, \mathcal{P}_1, \ldots, \mathcal{P}_{\mathcal{M}}) .$$

This is not the ordering used in the code–as with bidomain-with-bath we write the equations down using one ordering which gives a block format of vectors/matrices, but in the code we use a 'striped' ordering for parallelisation reasons—see Section 4.6.

Now, suppose $\mathcal{I}$ is an index into $\boldsymbol{\mathcal{A}}$, in others words, that $1 \le \mathcal{I} \le 3\mathcal{N} + \mathcal{M}$. Clearly $\mathcal{A}_{\mathcal{I}}$ is either a spatial variable, $(\mathcal{X}_I)_d$ for some $d = 1, 2$ or 3 and some $I \le \mathcal{N}$; or a pressure variable, $\mathcal{P}_I$ for some $I \le \mathcal{M}$. Let us introduce the notation

$$\mathcal{I} = \text{disp}(I, d)$$

if $\mathcal{I}$ corresponds to a spatial unknown and $\mathcal{A}_{\mathcal{I}} = (\mathcal{X}_I)_d$, and

$$\mathcal{I} = \text{pressure}(I)$$

if $\mathcal{I}$ corresponds to a pressure unknown and $\mathcal{A}_{\mathcal{I}} = \mathcal{P}_I$. For example, for small I, if $\mathcal{I} = I$ then $\mathcal{I} = \text{disp}(I, 1)$; if $\mathcal{I} = 2\mathcal{N} + I$ then $\mathcal{I} = \text{disp}(I, 3)$, if $\mathcal{I} = 3\mathcal{N} + I$ then $\mathcal{I} = \text{pressure}(I)$.

$$\boldsymbol{\mathcal{A}} = \left(\underbrace{\mathcal{X}_1, \ldots, \mathcal{X}_{\mathcal{N}}}_{\mathcal{I}=\text{disp}(I,1)}, \underbrace{\mathcal{Y}_1, \ldots, \mathcal{Y}_{\mathcal{N}}}_{\mathcal{I}=\text{disp}(I,2)}, \underbrace{\mathcal{Z}_1, \ldots, \mathcal{Z}_{\mathcal{N}}}_{\mathcal{I}=\text{disp}(I,3)}, \underbrace{\mathcal{P}_1, \ldots, \mathcal{P}_{\mathcal{M}}}_{\mathcal{I}=\text{pressure}(I)} \right) .$$

[8]The order of polynomial interpolation for pressure must be lower than that for displacement.

There will be $3\mathcal{N}+\mathcal{M}$ nonlinear equations in the finite element problem. The first $3\mathcal{N}$ equations are obtained by setting $\delta\mathbf{x}=(\phi_I,0,0)$ in (32) for $I=1,\dots,\mathcal{N}$, then $\delta\mathbf{x}=(0,\phi_I,0)$, then $\delta\mathbf{x}=(0,0,\phi_I)$; and the next $\mathcal{M}$ equations obtained by setting $\delta p=\psi_I$ in (33), $I=1,\dots,\mathcal{M}$. Overall, we have: solve $\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{A}})=0$, where

$$\mathcal{F}_{\mathcal{I}}(\boldsymbol{\mathcal{A}})=\begin{cases}\int_{\Omega_0} T_{MN}F_{dN}\frac{\partial\phi_I}{\partial X_M}\,\mathrm{d}V_0-\int_{\Omega_0}\rho_0 b_d\phi_I\,\mathrm{d}V_0-\int_{\partial\Omega_0^{\mathrm{trac}}}s_d\phi_I\,\mathrm{d}S_0 & \text{if } \mathcal{I}=\mathrm{disp}(I,d)\\ \int_{\Omega_0}\psi_I\,(\det F-1)\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{pressure}(I)\end{cases}$$

In this equation T and F should be considered as functions of $\boldsymbol{\mathcal{A}}$, through $T\equiv T(C(\mathbf{x}),p)$ with $\mathbf{x}=\sum_{I=1}^{\mathcal{N}}\boldsymbol{\mathcal{X}}_I\phi_I$ and $p=\sum_{I=1}^{\mathcal{M}}\mathcal{P}_I\psi_I$. In the case of pressure-on-deformed-surface boundary conditions, as described in Section 4.4, $\mathbf{s}$ will also depend on $\boldsymbol{\mathcal{A}}$—for this case see also Section 4.7.

This equation has to be solved using a nonlinear solver such as Newton's method. We use Newton's method with damping, for which we need to compute the Jacobian, $\frac{\partial\mathcal{F}_{\mathcal{I}}}{\partial\mathcal{A}_{\mathcal{J}}}$. Let us ignore the pressure-on-deformed-surface case for the time being. After some calculation, the Jacobian can be shown to be

$$\frac{\partial\mathcal{F}_{\mathcal{I}}}{\partial\mathcal{A}_{\mathcal{J}}}=\begin{cases}\int_{\Omega_0}\frac{\partial T_{MN}}{\partial C_{PQ}}\left(F_{eQ}\frac{\partial\phi_J}{\partial X_P}+F_{eP}\frac{\partial\phi_J}{\partial X_Q}\right)F_{dN}\frac{\partial\phi_I}{\partial X_M}+T_{MN}\delta_{de}\frac{\partial\phi_J}{\partial X_N}\frac{\partial\phi_I}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{disp}(I,d),\ \mathcal{J}=\mathrm{disp}(J,e)\\ \int_{\Omega_0}-\psi_J C^{-1}_{MN}F_{dN}\frac{\partial\phi_I}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{disp}(I,d),\ \mathcal{J}=\mathrm{pressure}(J)\\ \int_{\Omega_0}\psi_I(\det F)F^{-1}_{Me}\frac{\partial\phi_J}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{pressure}(I),\ \mathcal{J}=\mathrm{disp}(J,e)\\ 0 & \text{if } \mathcal{I}=\mathrm{pressure}(I),\ \mathcal{J}=\mathrm{pressure}(J)\end{cases}$$

which can be simplified further to (here, we write the first term using the symmetrisation of $\frac{\partial T}{\partial C}$ so that a smaller number of contractions (i.e. tensor-matrix multiplications) is required, which significantly reduces the computational cost)

$$\mathrm{Jac}_{\mathcal{I}\mathcal{J}}=\begin{cases}\int_{\Omega_0}\left(\frac{\partial T_{MN}}{\partial C_{PQ}}+\frac{\partial T_{MN}}{\partial C_{QP}}\right)F_{dN}F_{eQ}\frac{\partial\phi_J}{\partial X_P}\frac{\partial\phi_I}{\partial X_M}+T_{MN}\delta_{de}\frac{\partial\phi_J}{\partial X_N}\frac{\partial\phi_I}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{disp}(I,d),\ \mathcal{J}=\mathrm{disp}(J,e)\\ \int_{\Omega_0}-\psi_J F^{-1}_{Md}\frac{\partial\phi_I}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{disp}(I,d),\ \mathcal{J}=\mathrm{pressure}(J)\\ \int_{\Omega_0}\psi_I(\det F)F^{-1}_{Me}\frac{\partial\phi_J}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{pressure}(I),\ \mathcal{J}=\mathrm{disp}(J,e)\\ 0 & \text{if } \mathcal{I}=\mathrm{pressure}(I),\ \mathcal{J}=\mathrm{pressure}(J)\end{cases}\tag{35}$$

$\frac{\partial T}{\partial C}$ has to be provided by the user (through the material law).

Rewriting slightly, we have

$$\mathrm{Jac}_{\mathcal{I}\mathcal{J}}=\begin{cases}\int_{\Omega_0}\left(\left(\frac{\partial T_{MP}}{\partial C_{NQ}}+\frac{\partial T_{MP}}{\partial C_{QN}}\right)F_{dP}F_{eQ}+T_{MN}\delta_{de}\right)\frac{\partial\phi_J}{\partial X_N}\frac{\partial\phi_I}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{disp}(I,d),\ \mathcal{J}=\mathrm{disp}(J,e)\\ \int_{\Omega_0}-\psi_J F^{-1}_{Md}\frac{\partial\phi_I}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{disp}(I,d),\ \mathcal{J}=\mathrm{pressure}(J)\\ \int_{\Omega_0}\psi_I(\det F)F^{-1}_{Me}\frac{\partial\phi_J}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{pressure}(I),\ \mathcal{J}=\mathrm{disp}(J,e)\\ 0 & \text{if } \mathcal{I}=\mathrm{pressure}(I),\ \mathcal{J}=\mathrm{pressure}(J)\end{cases}\tag{36}$$

Note that the term multiplying the basis gradients is just $\frac{\partial S}{\partial F}$ written in terms of $\frac{\partial T}{\partial E}$ and T

$$\frac{\partial S_{Mi}}{\partial F_{jN}}=\left(\frac{\partial T_{MP}}{\partial C_{NQ}}+\frac{\partial T_{MP}}{\partial C_{QN}}\right)F_{iP}F_{jQ}+T_{MN}\delta_{ij}$$

If we had used S in the weak form rather than TF^{T} than this $\frac{\partial S}{\partial F}$ term would have dropped out immediately. Overall, we have

$$\mathrm{Jac}_{\mathcal{I}\mathcal{J}}=\begin{cases}\int_{\Omega_0}\frac{\partial S_{Md}}{\partial F_{eN}}\frac{\partial\phi_J}{\partial X_N}\frac{\partial\phi_I}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{disp}(I,d),\ \mathcal{J}=\mathrm{disp}(J,e)\\ \int_{\Omega_0}-\psi_J F^{-1}_{Md}\frac{\partial\phi_I}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{disp}(I,d),\ \mathcal{J}=\mathrm{pressure}(J)\\ \int_{\Omega_0}\psi_I(\det F)F^{-1}_{Me}\frac{\partial\phi_J}{\partial X_M}\,\mathrm{d}V_0 & \text{if } \mathcal{I}=\mathrm{pressure}(I),\ \mathcal{J}=\mathrm{disp}(J,e)\\ 0 & \text{if } \mathcal{I}=\mathrm{pressure}(I),\ \mathcal{J}=\mathrm{pressure}(J)\end{cases}\tag{37}$$

This only applies when $\mathbf{s}$ is not a function of deformation, i.e. not the case described in Section 4.4. When there are pressure-on-deformed-surface boundary conditions, there is an extra term which needs to be added to the Jacobian, which is described in Section 4.7.

Note that the Jacobian has a natural 2×2 block structure

$$\text{Jac} = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & 0 \end{pmatrix} \begin{matrix} \} \text{ displacement} \\ \} \text{ pressure} \end{matrix}$$

J_{11} is not in general symmetric, and J_{12} is not in general equal to $-J_{21}$. (It would be if we set $\det F = 1$ in the third equation in (37), but although $\det F$ must be 1 for the solution, it may not be of the current Newton guess).

4.6 Actual ordering of unknowns

The following ordering was used in the above

$$\boldsymbol{\mathcal{A}} = (\mathcal{X}_1, \ldots, \mathcal{X}_\mathcal{N}, \mathcal{Y}_1, \ldots, \mathcal{Y}_\mathcal{N}, \mathcal{Z}_1, \ldots, \mathcal{Z}_\mathcal{N}, \mathcal{P}_1, \ldots, \mathcal{P}_\mathcal{M})$$

as it allows the residual vector and Jacobian matrix to be written nicely in block form (and also because the code used to have this ordering). This assumes that there are N nodes in total, and the first M of those are vertices.

The implementation in the source code however does not assume the first M nodes are vertices. Instead, we introduce *dummy pressure variables* at nodes which are not vertices (i.e. internal nodes), in the same way that dummy variables were introduced in the bidomain-with-bath solver. We use the ordering:

$$\boldsymbol{\mathcal{A}} = (\mathcal{X}_1, \mathcal{Y}_1, \mathcal{Z}_1, \mathcal{P}_1, \ldots, \mathcal{X}_N, \mathcal{Y}_N, \mathcal{Z}_N, \mathcal{P}_N)\,.$$

This striped ordering is used, as always, for parallelisation reasons: keeping all unknowns at a node together in the ordering means they can be kept on the same processor.

As we have introduced extra unknowns, we have to add extra constraints; these are:

$$\mathcal{P}_i = 0 \qquad \text{if node } i \text{ is an internal node.}$$

This constraint is used during the nonlinear solve. *After* the solve is complete, the pressure at each internal nodes is linearly interpolated from the solution pressure at the neighbouring vertex nodes, which is the natural definition of the pressure at internal nodes.

Overall, the Jacobian is of the form

$$\text{Jac} = P^{\mathrm{T}} \begin{pmatrix} J_{11} & J_{12} & 0 \\ J_{21} & 0 & 0 \\ 0 & 0 & I \end{pmatrix} P \begin{matrix} \} \text{ displacement} \\ \} \text{ pressure} \\ \} \text{ dummy pressure} \end{matrix}$$

where P is some permutation matrix, and I is the identity matrix with equal to the number of non-vertices[9].

4.7 FE residual/Jacobian for the case of normal pressure on deformed surface BCs

With this type of boundary condition, $\mathbf{s}$ is a function of the deformation, as given in (34). The finite element residual is unchanged from that stated in Section 4.5. However, it should be noted that when looping over surface elements in order to compute the surface integral term, knowing the deformation at the surface element nodes is not enough to be able to compute F in the surface element, hence not enough to compute

[9] A final note: to complicate matters further, the element-level residual and jacobian in contrast to the the full residual/jacobian currently do not use dummy pressure variables or striped ordering, largely for historical reasons (but also because they do not need to).

s. In this part of the code, the volume element containing this surface element is first found, and then F can be be computed, using the deformation at the nodes of the volume element.

For the Jacobian, the derivative of **s** with respect to the spatial unknowns is not zero, so this introduces additions terms into the matrix. After a calculation, the term that has to be added to (37) can be shown to be

$$\text{Jac}^{\text{extra}}_{\mathcal{I}\mathcal{J}} = \begin{cases} -\int_{\partial\Omega_0^{\text{trac}}} JPN_M \left(F^{-1}_{Ne}F^{-1}_{Md} - F^{-1}_{Me}F^{-1}_{Nd}\right) \frac{\partial \phi_J}{\partial X_N}\phi_I \, \mathrm{d}S_0 & \text{if } \mathcal{I} = \text{disp}(I,d),\ \mathcal{J} = \text{disp}(J,e) \\ 0 & \text{if } \mathcal{I} = \text{disp}(I,d),\ \mathcal{J} = \text{pressure}(J) \\ 0 & \text{if } \mathcal{I} = \text{pressure}(I),\ \mathcal{J} = \text{disp}(J,e) \\ 0 & \text{if } \mathcal{I} = \text{pressure}(I),\ \mathcal{J} = \text{pressure}(J) \end{cases}$$

4.8 Test problems

We can use test problems with exact solutions to test the solvers. It is almost impossible to prescribe normal body forces and/or boundary conditions on a regular domain and solve the equations analytically; however we can instead choose a deformation which we want to be the solution, and then choose the appropriate body force and surface tractions which will give that solution.

The incompressible test problems are either based on the simple deformation $\mathbf{x} = (\alpha X, \beta Y)$, and then documented in the code, or a nonlinear deformation described in [Pathmanathan et al., Journal of Strain Analysis for Engineering Design, 2009]. See tests in `continuum_mechanics/test/TestIncompressibleNonlinearElasticitySolver.hpp`.

One of the compressible test problems also uses $\mathbf{x} = (\alpha X, \beta Y)$ and is relatively simple. The other test problem uses a nonlinear deformation and is described fully here, as it requires a large calculation to determine the appropriate body force and tractions. The compressible tests are defined in the test pack `continuum_mechanics/test/TestCompressibleNonlinearElasticitySolver.hpp`.

Nonlinear compressible test problem

The compressible test problem is 2D and uses the Neo-Hookean material law

$$W(I_1, I_3) = c(\bar{I}_1 - 3) + d(J-1)^2,$$

where (defining $m = -1/\texttt{DIM}$), $\bar{I}_1 = I_1 I_3^m$. Then

$$w_1 = cI_3^m \qquad w_3 = mcI_1 I_3^{m-1} + d\left(1 - I_3^{-\frac{1}{2}}\right).$$

The stress is $T = 2w_1 I + 2w_3 I_3 C^{-1}$, which, written in terms of the 1st Piola-Kirchoff tensor, is

$$S = 2w_1 F^{\mathrm{T}} + 2w_3 I_3 F^{-1}.$$

Suppose the body is the unit square, and the $X = 0$ side is given displacement boundary conditions, and the remaining surfaces given traction boundary conditions. We choose the following deformation

$$\mathbf{x} = \begin{pmatrix} q\left(X + \frac{a}{2}X^2\right) \\ \frac{Y}{1+aX} \end{pmatrix},$$

where q and a are parameters. Note that the deformation is compressible if $q \neq 1$. Writing $\lambda = 1 + aX$, this gives a deformation gradient

$$F = \begin{pmatrix} q\lambda & 0 \\ -Ya\lambda^{-2} & \lambda^{-1} \end{pmatrix} \qquad \Rightarrow \qquad F^{-1} = \frac{1}{q}\begin{pmatrix} \lambda^{-1} & 0 \\ Ya\lambda^{-2} & q\lambda \end{pmatrix}$$

and therefore

$$\begin{aligned} I_1 &= q^2\lambda^2 + a^2Y^2\lambda^{-4} + \lambda^{-2}, \\ I_3 &= q^2. \end{aligned}$$

Then

$$\begin{aligned} w_1 &= cq^{2m}, \\ w_3 &= mcI_1q^{2m-2} + d\left(1 - q^{-1}\right), \end{aligned}$$

and

$$S = 2\begin{pmatrix} w_1q\lambda + qw_3\lambda^{-1} & -w_1Ya\lambda^{-2} \\ qw_3Ya\lambda^{-2} & w_1\lambda^{-1} + w_3q^2\lambda \end{pmatrix}$$

We can now read off the tractions: the traction on the top surface is (S_{21}, S_{22}), the traction of the right-hand surface is (S_{11}, S_{12}), and the traction on the bottom surface is $(-S_{21}, -S_{22})$.

Finally, we can compute

$$\begin{aligned} \frac{\partial w_3}{\partial X} &= mcq^{2m-2}a\left(2q^2\lambda - 4a^2Y^2\lambda^{-5} - 2\lambda^{-3}\right) \\ \frac{\partial w_3}{\partial Y} &= 2mcq^{2m-2}aY\lambda^{-4} \end{aligned} \tag{38}$$

and using these

$$\begin{aligned} \frac{\partial S_{11}}{\partial X} &= 2\left(w_1qa - w_3aq\lambda^{-2} + q\lambda^{-1}\frac{\partial w_3}{\partial X}\right) \\ \frac{\partial S_{12}}{\partial X} &= 4w_1a^2Y\lambda^{-3} \\ \frac{\partial S_{21}}{\partial Y} &= 2\left(qw_3a\lambda^{-2} + qYa\lambda^{-2}\frac{\partial w_3}{\partial Y}\right) \\ \frac{\partial S_{22}}{\partial Y} &= 2\lambda q^2\frac{\partial w_3}{\partial Y} \end{aligned} \tag{39}$$

These are used in prescribing the appropriate body force: $\mathbf{b} = -\frac{1}{\rho_0}\left(\frac{\partial S_{11}}{\partial X} + \frac{\partial S_{21}}{\partial Y}, \frac{\partial S_{12}}{\partial X} + \frac{\partial S_{22}}{\partial Y}\right)$.

4.9 Other implementation calculations

The material law classes need to provide a function which returns T (a matrix, i.e. a 2nd-order tensor) and $\frac{\partial T}{\partial E}$ (a fourth order tensor, $\frac{\partial T_{MN}}{\partial E_{PQ}}$) given C.

In the isotropic cases, they make use of the following.

Compressible case

The stress is given by

$$T = 2w_1I + 2w_2(I_1I - C) + 2w_3I_3C^{-1}$$

which means that

$$\begin{aligned} \frac{1}{4}\frac{\partial T}{\partial E} = \quad & w_{11}I \otimes I + w_{12}I \otimes (I_1I - C) + w_{13}I_3I \otimes C^{-1} \\ + \quad & w_{21}(I_1I - C) \otimes I + w_{22}(I_1I - C) \otimes (I_1I - C) \\ + \quad & w_{23}I_3(I_1I - C) \otimes C^{-1} + w_2\left(I \otimes I - \frac{\partial C}{\partial C}\right) \\ + \quad & w_{31}I_3C^{-1} \otimes I + w_{32}I_3C^{-1} \otimes (I_1I - C) \\ + \quad & (w_3 + I_3w_{33})I_3C^{-1} \otimes C^{-1} + w_3I_3\frac{\partial C^{-1}}{\partial C} \end{aligned}$$

Here $A \otimes B$ represents the fourth-order tensor Z satisfying $Z_{MNPQ} = A_{MN} B_{PQ}$. Two fourth-order tensors in the above, $\frac{\partial C}{\partial C}$ and $\frac{\partial C^{-1}}{\partial C}$, cannot be written in this way: they are $\frac{\partial C_{MN}}{\partial C_{PQ}} = \delta_{MP}\delta_{NQ}$ and $\frac{\partial C^{-1}_{MN}}{\partial C_{PQ}} = -C^{-1}_{MP} C^{-1}_{QN}$.

This is implemented in `AbstractIsotropicCompressibleMaterialLaw`.

Incompressible case

This case is less complicated given the form of the strain energy function. The stress is given by

$$T = 2w_1 I + 2w_2(I_1 I - C) - pC^{-1}$$

which means that

$$\begin{aligned}
\frac{1}{4}\frac{\partial T}{\partial E} = \quad & w_{11} I \otimes I + w_{12} I \otimes (I_1 I - C) \\
+ \quad & w_{21}(I_1 I - C) \otimes I + w_{22}(I_1 I - C) \otimes (I_1 I - C) \\
+ \quad & w_2 \left(I \otimes I - \frac{\partial C}{\partial C} \right) \\
- \quad & \frac{p}{2} \frac{\partial C^{-1}}{\partial C}
\end{aligned}$$

This is implemented in `AbstractIsotropicIncompressibleMaterialLaw`.

5 Cardiac electro-mechanics

5.1 Formulation

In cardiac electro-mechanical problems, the monodomain/bidomain equations are amended slightly to take into account the deformation, an extra set of ODE systems—*the contraction model*, which model active force generation on the cell-level—are introduced, and this active force is added to the nonlinear elasticity equations.

Physiologically, the mechanical response in the cell is dependent on the electrical activity largely through the *intracellular calcium concentration*, $[\mathrm{Ca}^{2+}]$, and therefore most contraction models take this as input[10]. Now, let $\mathbf{f}$ denote the undeformed unit fibre direction. The *fibre-stretch* (stretch in the fibre-direction) is then given by $\lambda = \sqrt{\mathbf{f}^{\mathrm{T}} C \mathbf{f}}$ (C is the standard deformation tensor defined in Section 4). The contraction model can also be dependent on the fibre-stretch, as well as possibly on the fibre-stretch-rate, $\dot{\lambda}$. Let $\mathbf{w}$ be a vector of internal state variables for the contraction model. The contraction model is a set of ODEs determining how the state variables evolve and an equation which provides the *active tension*, σ_a. Despite the name, this is actually a stress not a force; a scalar stress generated at the cellular level in the fibre direction in response to excitation.

$$\begin{aligned} \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} &= \mathbf{g}(\mathbf{w}; [\mathrm{Ca}^{2+}], \lambda, \dot{\lambda}), && (40) \\ \sigma_a &\equiv \sigma_a(\mathbf{w}; [\mathrm{Ca}^{2+}], \lambda, \dot{\lambda}). && (41) \end{aligned}$$

Note that we have denoted active tension as σ_a rather than the more common T_a. This is to emphasise the fact that the active tension is likely to be a *Cauchy* (true) stress, rather than a Piola-Kirchhoff stress. This will be the case if the deformed cross-sectional area (rather than undeformed cross-sectional area) was used in the experiments used to fit contraction model parameters.

The equations of nonlinear elasticity are amended to take into account the active response of the tissue by introducing a third term to the stress which depends on the active tension [11]

$$T = 2\frac{\partial W}{\partial C} - pC^{-1} + \frac{J\sigma_a}{\mathbf{f}^{\mathrm{T}} C \mathbf{f}} \mathbf{f}\mathbf{f}^{\mathrm{T}}, \tag{42}$$

(where $J = \det(F)$), or

$$T = T^{\mathrm{passive}} + T^{\mathrm{active}}, \tag{43}$$

where

$$\begin{aligned} T^{\mathrm{passive}} &= 2\frac{\partial W^{\mathrm{mat}}}{\partial C} - pC^{-1}, \\ T^{\mathrm{active}} &= \frac{J\sigma_a}{\lambda^2} \mathbf{f}\mathbf{f}^{\mathrm{T}}. \end{aligned} \tag{44}$$

T^{active} is the active (tensor) stress corresponding to the cellular active tension that is induced in the fibre direction.

We assume the tissue is always instantaneously in equilibrium and that that inertial effects can be neglected (i.e. quasi-steady), which means the static equilibrium equations, (28) and (29), are used. We also take zero body force (i.e. neglect gravity), so the equations of equilibrium are:

$$\begin{aligned} \frac{\partial}{\partial X_M}(T_{MN} F_{iN}) &= 0, && (45) \\ \det(F) &= 1. && (46) \end{aligned}$$

[10] Some simpler models are instead dependent on the voltage and some also possibly explicitly on time.

[11] The derivation of the active stress term is as follows: we suppose the σ_a returned by the contraction model is a *Cauchy* stress, so that it has to be transformed to a 2nd Piola-Kirchoff stress. The active stress is assumed to act only in the fibre direction, so the tensor active Cauchy stress is $\sigma^{\mathrm{active}} = \sigma_a \tilde{\mathbf{f}}\tilde{\mathbf{f}}^{\mathrm{T}}$, where $\tilde{\mathbf{f}}$ is the *deformed normalised* fibre direction. Now, $\tilde{\mathbf{f}} = F\mathbf{f}/\|F\mathbf{f}\|$, so $\sigma^{\mathrm{active}} = \sigma_a \frac{F\mathbf{f}\mathbf{f}^{\mathrm{T}}F^{\mathrm{T}}}{\mathbf{f}^{\mathrm{T}}C\mathbf{f}}$. Now use the relationship between σ and T, $T = \det F F^{-1}\sigma F^{-\mathrm{T}}$

where T is now the total stress.

The deformation affects the electrical activity in two ways. First, the deformation of the tissue alters the geometry over which the voltage propagates, altering the spatial derivatives in (7); and secondly, the cell-model can be dependent on fibre-stretch through the so-called 'stretch-activated channels'. Equations (7) and (8) become (using D for the monodomain conductivity tensor as σ now represents stress):

$$\chi C_m \frac{\partial V}{\partial t} = \nabla \cdot (F^{-1} D F^{-\mathrm{T}} \nabla V) - \chi I_{\mathrm{ion}}(\mathbf{u}, V, \lambda), \tag{47}$$

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(\mathbf{u}, V, \lambda). \tag{48}$$

(Note: $F^{-1} D F^{-\mathrm{T}} = C^{-1} D$ if D is isotropic, i.e. if fibre directions are not being used in the electrical simulation). The code by default uses D instead of $F^{-1} D F^{-\mathrm{T}}$ by default as in the case of simple propagation it has been shown that this approximation gives very little error (essentially because, when there is diffusion of the voltage occurring, there is little deformation) whereas including the deformation-dependence massively increases computational cost. The user can switch on the deformation-dependence if they require it.

5.2 Implicit or explicit schemes

Consider evaluating a stress $T \equiv T(C)$. T is dependent on σ_a, which is dependent on C through possibly being dependent on λ and $\dot{\lambda}$. One possibility is to use the previous value of λ and $\dot{\lambda}$ for computing σ_a (so using both the current and previous value of C in evaluating T). This is the commonly-used *explicit method*, and has the advantage that the contraction models ODEs need only be solved once per timestep (per node/quadrature point). However, this can have serious stability/accuracy issues (see cardiac electromechanics paper). The alternative is to evaluate σ_a using the current value of C, which is the *implicit method*. This has no stability issues, but requires the contraction models ODEs to be re-integrated every time a stress is evaluated (which is several times during the nonlinear solve). Letting C^n be the value of C at the previous timestep, etc, the two schemes evaluate T using:

$$T = 2\frac{\partial W}{\partial C}(C^{n+1}) - p(C^{n+1})^{-1} + \frac{J^{n+1}\sigma_a(\lambda^n, \dot{\lambda}^n)}{(\lambda^{n+1})^2}\mathbf{f}\mathbf{f}^{\mathrm{T}}, \tag{49}$$

in the explicit method, and

$$T = 2\frac{\partial W}{\partial C}(C^{n+1}) - p(C^{n+1})^{-1} + \frac{J^{n+1}\sigma_a(\lambda^{n+1}, \dot{\lambda}^{n+1})}{(\lambda^{n+1})^2}\mathbf{f}\mathbf{f}^{\mathrm{T}}, \tag{50}$$

in the implicit method.

The weak form, finite element residual $\boldsymbol{\mathcal{F}}$ and finite element Jacobian are all unchanged, they just need to be computed using the *total* stress, using both passive and active parts. This introduces an extra term in the $\frac{\partial T}{\partial C}$ term used in (36), which can be computed to be:

$$\frac{\partial T^{\mathrm{active}}_{MN}}{\partial C_{PQ}} = J\left(-\frac{\sigma_a}{\lambda^4} + \frac{\partial\sigma_a/\partial\lambda + \frac{1}{\Delta t}\partial\sigma_a/\partial\dot{\lambda}}{2\lambda^3}\right) m_M m_N m_P m_Q + \frac{J\sigma_a}{2\lambda^2} m_M m_N (C^{-1})_{PQ}. \tag{51}$$

The derivatives of σ_a need to be approximated numerically and are zero if the explicit method is used (because then σ_a is a function of λ at the *previous* timestep).

5.3 Anisotropic passive material laws

Finally, a note on the implementation of anisotropic passive material laws, in particular cardiac material laws which depend on the fibre, sheet and normal directions[12]. An example is the pole-zero material law,

[12] More precisely, these are three orthonormal directions at each point, the first being the fibre direction, the second being a vector in the sheet orthogonal to the fibre, the third being normal to the sheet.

which has terms of the form $k_{ff}E_{ff}$, where k_{ff} is a parameter and $E_{ff} = \mathbf{f}^{\mathrm{T}} E \mathbf{f}$. Let us write $\mathbf{f}$, $\mathbf{s}$ and $\mathbf{n}$ for the fibre, sheet and normal directions. The material law implementations take in C and the change of basis matrix P, defined by

$$P = [\mathbf{f}\ \mathbf{s}\ \mathbf{n}].$$

The stress is computed by assuming the fibres are parallel to the X-axis, and the sheet in the XY plane, by transforming C to the fibre-sheet basis prior to the calculating the stress ($C \to C^*$) and transforming the resultant T^* after it has been computed. Also, the material law will return $\frac{\mathrm{d}T^*}{\mathrm{d}C^*}$, which has to be transformed as well. The appropriate computations are

$$C^* = P^{\mathrm{T}} C P \qquad T^* \equiv T^*(C^*) \qquad T = P T^* P^{\mathrm{T}}$$

and

$$\frac{\mathrm{d}T_{MN}}{\mathrm{d}C_{PQ}} = P_{Mm} P_{Nn} P_{Pp} P_{Qq} \frac{\mathrm{d}T^*_{mn}}{\mathrm{d}C^*_{pq}}.$$

6 Fluid dynamics

6.1 Stokes' flow

Let $\Omega \subset \mathbb{R}^3$, and let $\mathbf{x}$ denote position (the independent variable). The Stokes' flow equations can be used to determine the fluid flow when inertial forces are small compared to viscous forces (the low Reynolds number regime). The Stokes' flow equations are:

$$\nabla \cdot \sigma + \mathbf{f} = 0 \tag{52}$$
$$\nabla \cdot \mathbf{u} = 0 \tag{53}$$

where σ is the (total) fluid stress defined by $\sigma_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - p\delta_{ij}$, where $\mathbf{u}$ is the unknown flow velocity field, p is the unknown pressure field, and μ is the viscosity. $\mathbf{f}$ is an applied body force. The first equation represents conservation of momentum, the second conservation of mass. Note that $\nabla \cdot \sigma$ has i-th component given by $\frac{\partial \sigma_{ij}}{\partial x_j}$. Substituting σ in (52) and making use of (53), we obtain the alternative statement of Stokes' flow:

$$\mu\nabla^2\mathbf{u} - \boldsymbol{\nabla} p + \mathbf{f} = 0 \tag{54}$$
$$\nabla \cdot \mathbf{u} = 0 \tag{55}$$

Appropriate boundary conditions are

$$\mathbf{u} = \mathbf{u}^* \quad \text{on } \Gamma_1 \tag{56}$$
$$\sigma\mathbf{n} = \mathbf{s} \quad \text{on } \Gamma_2 \tag{57}$$

where Γ_1 and Γ_2 partition $\partial\Omega$, $\mathbf{u}^*$ is a presribed boundary flow and $\mathbf{s}$ the prescribed boundary stress (usually 0). Often Γ_2 is empty, and $\mathbf{u}^*$ takes one value on an inflow boundary, another value on an outflow boundary, and is zero on the remainder of the boundary (no slip boundary conditions). Note that if Γ_2 is empty p will only be defined up to a constant.

The weak form is obtained by multiplying by test functions and integrating using the divergence theorem, as normal. We use the first formulation (52) rather than the second formulation to make sure the boundary integral involves the Neumann boundary condition (57). The weak problem is[13]: *find* $\mathbf{u} \in V$ *and* $p \in W$ *such that* $\mathbf{u} = \mathbf{u}^*$ *on* Γ_1 *and*

$$\int_\Omega \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\frac{\partial v_i}{\partial x_j} - p\frac{\partial v_i}{\partial x_i}\,\mathrm{d}V = \int_\Omega f_i v_i\,\mathrm{d}V + \int_{\Gamma_2} s_i v_i\,\mathrm{d}S \qquad \forall \mathbf{v} \in V$$
$$\int_\Omega \frac{\partial u_i}{\partial x_i} q\,\mathrm{d}V = 0 \qquad \forall \mathbf{q} \in W$$

Now, let $\epsilon_{ij}(\mathbf{u}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ and note that $\epsilon_{ij}(\mathbf{u})\frac{\partial v_i}{\partial x_j} = \epsilon_{ij}(\mathbf{u})\epsilon_{ij}(\mathbf{v})$. We obtain the weak problem: *find* $\mathbf{u} \in V$ *and* $p \in W$ *such that* $\mathbf{u} = \mathbf{u}^*$ *on* Γ_1 *and*

$$\int_\Omega 2\mu\epsilon_{ij}(\mathbf{u})\epsilon_{ij}(\mathbf{v}) - p\frac{\partial v_i}{\partial x_i}\,\mathrm{d}V = \int_\Omega f_i v_i\,\mathrm{d}V + \int_{\Gamma_2} s_i v_i\,\mathrm{d}S \qquad \forall \mathbf{v} \in V$$
$$\int_\Omega \frac{\partial u_i}{\partial x_i} q\,\mathrm{d}V = 0 \qquad \forall \mathbf{q} \in W$$

For the finite element problem, we use quadratic basis functions to interpolate the flow, and linear basis functions for the pressure, as in the solid mechanics solvers. Assuming for the time being the vector of nodal unknowns is ordered

$$\boldsymbol{\mathcal{X}} = (\mathcal{U}_1, \ldots, \mathcal{U}_\mathcal{N}, \mathcal{V}_1, \ldots, \mathcal{V}_\mathcal{N}, \mathcal{W}_1, \ldots, \mathcal{W}_\mathcal{N}, \mathcal{P}_1, \ldots, \mathcal{P}_\mathcal{M}) \tag{58}$$

[13] Here $V = H^1(\Omega)^3$ and $W = L^2(\Omega)$.

where $(\mathcal{U}_k, \mathcal{V}_k, \mathcal{W}_k)$ is the unknown flow at node k, $\mathcal{P}_k$ the unknown pressure at node k, and there are assumed N nodes and M vertices. Let ϕ_k represent the (quadratic) basis function for node k, and ψ_k represent the (linear) basis function for vertex k. By defining $u = \sum_{k=1,\ldots,N} \mathcal{U}_k \phi_k$, $v = \sum_{k=1,\ldots,N} \mathcal{V}_k \phi_k$, etc, and $p = \sum_{k=1,\ldots,M} \mathcal{P}_k \psi_k$, and letting $\mathbf{v}$ and q range over the basis functions as usual, the finite element problem can be shown to be

$$\mathcal{A}\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{B}}$$

where $\mathcal{A}$ is a $(3M+N) \times (3M+N)$ matrix with entries (using the same notation as in Section 4.5)

$$\mathcal{A}_{\mathcal{I}\mathcal{J}} = \begin{cases} \mu \int_\Omega \frac{\partial \phi_I}{\partial x_k} \frac{\partial \phi_I}{\partial x_k} \delta_{de} + \frac{\partial \phi_I}{\partial x_e} \frac{\partial \phi_J}{\partial x_d} \, \mathrm{d}V & \text{if } \mathcal{I} = \mathrm{disp}(I,d),\ \mathcal{J} = \mathrm{disp}(J,e) \\ \int_\Omega \frac{\partial \phi_I}{\partial x_d} \psi_J \, \mathrm{d}V & \text{if } \mathcal{I} = \mathrm{disp}(I,d),\ \mathcal{J} = \mathrm{pressure}(J) \\ \int_\Omega \frac{\partial \phi_J}{\partial x_e} \psi_I \, \mathrm{d}V & \text{if } \mathcal{I} = \mathrm{pressure}(I),\ \mathcal{J} = \mathrm{disp}(J,e) \\ 0 & \text{if } \mathcal{I} = \mathrm{pressure}(I),\ \mathcal{J} = \mathrm{pressure}(J) \end{cases} \tag{59}$$

and $\boldsymbol{\mathcal{B}}$ a vector of size $3M+N$ with entries

$$\boldsymbol{\mathcal{B}}_{\mathcal{I}} = \begin{cases} \int_\Omega f_d \phi_I \, \mathrm{d}V + \int_{\Gamma_2} s_d \phi_I \, \mathrm{d}S & \text{if } \mathcal{I} = \mathrm{disp}(I,d) \\ 0 & \text{if } \mathcal{I} = \mathrm{pressure}(I) \end{cases}$$

Note that for the case $\Gamma_2 = \{\}$, an alternative (and somewhat simpler) weak form and finite element problem can be obtained by integrating (54) by parts. The problem is equivalent to the above when there are no Neumann boundary; however it would not enforce the Neumann boundary condition (56).

In block form the linear system has the following structure

$$\begin{pmatrix} A & B \\ B^{\mathrm{T}} & 0 \end{pmatrix} \boldsymbol{\mathcal{X}} = \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix} \begin{matrix} \} \text{ size } 3M \text{ (flow)} \\ \} \text{ size } N \text{ (pressure)} \end{matrix}$$

Ordering: finally, note that the implementation in the code does not actually use the ordering (58). See Section 4.6 for the true ordering. Everything in Section 4.6 applies to the Stokes' flow implementation, including dummy pressure variables, the identity block in the matrix, and linear interpolation of pressure values after the solve.