Finite element implementation

First, we briefly summarize the strong form of the governing equations in Section 4.4. The strong form of the coupled PDEs are given by

\[
\begin{cases}
\dot{c}_i + c_i \frac{j}{J} - \text{div} q_i - f^c_i = 0 & \text{in } B_t, \\
c_i = \bar{c}_i & \text{on } S_c, \\
\text{div} T = 0 & \text{in } B_t, \\
u = \bar{u} & \text{on } S_u, \\
Tn = \bar{t} & \text{on } S_t.
\end{cases}
\] (S.1)

Next, we let \( w_1, w_2 \) denote two test fields which vanish on \( S_c \) and \( S_u \), respectively. After multiplying the PDEs with two test fields and performing integration by parts, the weak form could be expressed as

\[
\begin{cases}
\int_{B_t} w_1 \left( \dot{c}_i + c_i \frac{j}{J} \right) dv + \int_{B_t} (\text{grad} w_1 \cdot q_i) dv - \int_{B_t} (w_1 f^c_i) dv = 0, \\
\int_{S_t} (T : \text{grad} w_2) dv = \int_{S_t} (w_2 \cdot \bar{t}) da.
\end{cases}
\] (S.2)

As is routine, the body is discretized into finite elements such that \( B_t = \cup B^e_t \), and the nodal variables are taken to be the cell densities and displacement, which are interpolated inside each element by

\[
c_i = \sum c_i^A N^A \quad \text{and} \quad u = \sum u^A N^A.
\] (S.3)

with the index \( A = 1, 2, \ldots \) denoting the nodes of the element, \( c_i^A \) and \( u^A \) are the nodal cell densities and displacements, respectively, and \( N^A \) the shape functions. Next, the test fields \( w_1 \) and \( w_2 \) are interpolated by the same shape functions in the Galerkin approach, viz.

\[
w_1 = \sum w_1^A N^A, \quad \text{and} \quad w_2 = \sum w_2^A N^A.
\] (S.4)
After substitution of Eq (S.3) and Eq (S.4) into the weak forms in Eq (S.2), we arrive at element-level of residuals. Since the expressions are similar for different neuron types, we only demonstrate the residual for \( c_1(x, t) \),

\[
R_{c_1}^A = \int_{\Omega^e} N^A \left( \dot{c}_1 + c_1 \dot{J} / J \right) + (\text{grad} N^A \cdot \mathbf{q}_1) dv - \left( N^A f_1^c \right) dv.
\]  

(S.5)

The element level residual for the displacement field \( \mathbf{u}(x, t) \) is given by

\[
R_u^A = -\int_{\Omega^e} (\mathbf{T} \cdot \text{grad} N^A) dv + \int_{\Gamma^e} (N^A \dot{\mathbf{u}}) da.
\]  

(S.6)

Since we consider cohorts of electroporated neurons at three different times in this work, four element-level residuals in total are assembled into a global residual, which, when set to zero, represents a system of non-linear equations for the nodal degrees of freedom.

The following tangents

\[
(K_{uu})^{AB} = -\frac{\partial R_u^A}{\partial u^B}, \quad (K_{uc_1})^{AB} = -\frac{\partial R_u^A}{\partial c_1^B}, \quad (K_{uc_2})^{AB} = -\frac{\partial R_u^A}{\partial c_2^B}, \quad (K_{uc_3})^{AB} = -\frac{\partial R_u^A}{\partial c_3^B},
\]

\[
(K_{c_1u})^{AB} = -\frac{\partial R_{c_1}^A}{\partial u^B}, \quad (K_{c_1c_1})^{AB} = -\frac{\partial R_{c_1}^A}{\partial c_1^B}, \quad (K_{c_1c_2})^{AB} = -\frac{\partial R_{c_1}^A}{\partial c_2^B}, \quad (K_{c_1c_3})^{AB} = -\frac{\partial R_{c_1}^A}{\partial c_3^B},
\]

\[
(K_{c_2u})^{AB} = -\frac{\partial R_{c_2}^A}{\partial u^B}, \quad (K_{c_2c_1})^{AB} = -\frac{\partial R_{c_2}^A}{\partial c_1^B}, \quad (K_{c_2c_2})^{AB} = -\frac{\partial R_{c_2}^A}{\partial c_2^B}, \quad (K_{c_2c_3})^{AB} = -\frac{\partial R_{c_2}^A}{\partial c_3^B},
\]

\[
(K_{c_3u})^{AB} = -\frac{\partial R_{c_3}^A}{\partial u^B}, \quad (K_{c_3c_1})^{AB} = -\frac{\partial R_{c_3}^A}{\partial c_1^B}, \quad (K_{c_3c_2})^{AB} = -\frac{\partial R_{c_3}^A}{\partial c_2^B}, \quad (K_{c_3c_3})^{AB} = -\frac{\partial R_{c_3}^A}{\partial c_3^B},
\]

are required by the iterative Newton-Raphson procedure for convergence.

More specifically, the tangent \( K_{uu}^{AB} \) is given by

\[
K_{uu}^{AB} = -\int_{\Omega^e} \frac{\partial N^A}{\partial x_j} \mathbb{A}_{ijkl} \frac{\partial N^B}{\partial x_l} dv - \int_{\Gamma^e} N^A N^B \frac{\partial \mathbf{F}}{\partial u_k} da,
\]  

(S.7)

where the spatial tangent modulus \( \mathbb{A} \) is related to referential tangent modulus \( \mathbb{A}_R \) through

\[
\mathbb{A}_{ijkl} = J^{-1} F_{jm} F_{ln} (\mathbb{A}_R)_{imkn},
\]  

(S.8)

and the referential tangent modulus is given by \( \mathbb{A}_R \overset{\text{def}}{=} \frac{\partial \mathbf{T}_R}{\partial \mathbf{F}} \) with \( \mathbf{T}_R = J \mathbf{T} \mathbf{F}^{-\top} \) denoted as the Piola stress [1]. After using Eq (15), the Piola stress is now given by

\[
\mathbf{T}_R = J^6 (\mu F_{jm} + \mu \mathbf{F}^{-\top} \mathbf{F}^g^{-1} \mathbf{F}^g^{-1}),
\]  

(S.9)

and hence

\[
(\mathbb{A}_R)_{ijkl} = J^6 (\mu F_{jm} + \mu \mathbf{F}^{-\top} \mathbf{F}^g^{-1} \mathbf{F}^g^{-1})
\]

\[
+ J^6 (\mu \mathbf{F}^{-\top} \mathbf{F}^{g-1} \mathbf{F}^{g-1} + \mu \delta_{ij} F_{jk} F_{ik}^{-1} F_{lk}^{-1} F_{ab}^{-1} F_{jb}^{-1} F_{ab}^{-1} F_{jb}^{-1}).
\]  

(S.10)
The three diagonal tangents for neuron density are similar, and here we only demonstrate $K_{c1c1}^{AB}$, which is given by

$$K_{c1c1}^{AB} = -\int_{B_e^c} N^A N^B \left( \frac{1}{\Delta t} + \frac{j}{J} \right) dv - \int_{B_e^c} \text{grad} N^A \cdot \left( \frac{\partial q_i}{\partial c_1} \right) N^B dv$$

$$= -\int_{B_e^c} N^A N^B \left( \frac{1}{\Delta t} + \frac{j}{J} \right) dv - \int_{B_e^c} \text{grad} N^A \cdot \text{D} \cdot \text{grad} N^B dv$$

$$- \int_{B_e^c} \text{grad} N^A \left( \tilde{H} + c_1 \frac{\partial \tilde{H}}{\partial c_1} \right) \cdot \text{v} N^B dv,$$

where the derivative $\frac{\partial \tilde{H}}{\partial c_1}$ is given by

$$\frac{\partial \tilde{H}}{\partial c_1} = \frac{\alpha_c e^{\alpha_c (c_1 - c_0)}}{e^{\alpha_c (c_1 - c_0)} + 1} - \frac{\alpha_c e^{2 \alpha_c (c_1 - c_0)}}{(e^{\alpha_c (c_1 - c_0)} + 1)^2}.$$

We note the current work does not address coupling between cohorts of neurons, thus we set $K_{c1c2}, K_{c1c3}, K_{c2c1}, K_{c2c3}, K_{c3c1},$ and $K_{c3c2}$ equal to zero.

The displacement–cell density tangents are similar among the three types of neurons; we only demonstrate $K_{u_i c1}^{AB}$, which is given by

$$K_{u_i c1}^{AB} = -\int_{B_e^c} \frac{\partial N^A}{\partial x_j} \left( \frac{\partial T_{ij}}{\partial c_1} \right) N^B dv,$$

where the derivative $\left( \frac{\partial T}{\partial c_1} \right)_{ij}$ could be further expressed as

$$\left( \frac{\partial T}{\partial c_1} \right)_{ij} = \left( \frac{\partial T}{\partial F^g} \right)_{ijkl} \left( \frac{\partial F^g}{\partial c_1} \right)_{kl}.$$

The derivative $\left( \frac{\partial T}{\partial F^g} \right)_{ijkl}$ in Eq (S.14) is given by

$$\left( \frac{\partial T}{\partial F^g} \right)_{ijkl} = J^{e-1} \left( \lambda \ln J^e - \mu - \lambda \right) \delta_{ij} F_k^{g-1} + \mu B_{ij}^e F_k^{g-1}$$

$$- \mu F_{ik}^e F_{ja}^{g-1} - \mu F_{ia}^e F_{jk}^{g-1}.$$

The derivative $\left( \frac{\partial F^g}{\partial c_1} \right)_{kl}$ in Eq (S.14) is given by

$$\left( \frac{\partial F^g}{\partial c_1} \right)_{kl} = k^\parallel \delta_{kl} + (k^\parallel - k^\perp) N_k N_l.$$

The cell density–displacement tangents among the three neuron types are similar, and we only
demonstrate $K_{c_1 u_k}^{AB}$, which is given by

$$K_{c_1 u_k}^{AB} = -\int_{B_t} N^A (\dot{c}_1 + \frac{c_1}{\Delta t} + f_1^c) dv$$
$$+ \int_{B_t} \frac{\partial N^A}{\partial x_k} q_i \frac{\partial N^B}{\partial x_i} dv$$
$$- \int_{B_t} \frac{\partial N^A}{\partial x_i} q_i \frac{\partial N^B}{\partial x_k} dv$$
$$- \int_{B_t} \frac{\partial N^A}{\partial x_i} \left( \frac{\partial q_i}{\partial F_{kn}} F_{nm} \right)_{ikm} \frac{\partial N^B}{\partial x_m} dv.$$  \hfill (S.17)

After implementing a two-noded linear element in Matlab for verification of our numerical implementation (see below), we then implement four-noded quadrilateral and eight-noded brick elements, (UPE4 and U3D8, respectively) in [2] by writing user-defined element (UEL) subroutines, following our previous work [1, 3, 4, 5, 6].

**Verification of our numerical implementation**

In order to verify our finite element implementations, we decouple the problem and consider each phenomenon (spatiotemporal cell density and mechanical deformation) separately. For the former, we compare our independent numerical solutions in Matlab and Abaqus to each other; in the latter case, we compare our numerical solutions in Abaqus with an analytically tractable solution.

First, we consider cell migration in a rigid slender body with a length of $l = 1$ m. Here, we only restrict our attention on one cohort of cells, $c_1$, and allow them to migrate along $x$ direction. Under these assumptions, the PDE in Eq (5) is now reduced to 1-D,

$$\dot{c}_1 - \frac{dq_1}{dx} - f_1^c = 0 \quad \text{in} \quad 0 < x \leq l.$$  \hfill (S.18)

The cell density flux $q_1$ in Eq (S.18) is given by

$$q_1 = -c_1 \hat{H}(c_1 - c_0; \alpha_c) \hat{v}_1(x) + D \frac{dc_1}{dx},$$  \hfill (S.19)

where the velocity field $\hat{v}_1(x)$ is expressed as

$$\hat{v}_1(x) = v_1 \hat{H}(\delta_1^v - x; \alpha_v) - 0.5v_1.$$  \hfill (S.20)

The source term $f_1^c$ in Eq (S.18) is given by the following function,

$$f_1^c = G^c \left[ \frac{\epsilon^2}{(t - \delta_1^t)^2 + \epsilon^2} \right] \hat{H}(x - \delta^x; \alpha_G).$$  \hfill (S.21)

For initial and boundary conditions, cell density is absent throughout the domain at $t = 0$, and we assign zero cell flux at the boundary. We consider a 1-D linear element in Matlab and a 3-D eight-noded brick element (U3D8) in Abaqus. The material parameters used for the verification are: $c_0 = 100 \text{ m}^{-3}$, $v_1 = 0.01 \text{ m s}^{-1}$, $\alpha_c = 0.1$, $\alpha_G = 100$, $\alpha_v = 100$, $D = 0.005 \text{ m}^2 \text{ s}^{-1}$, $G_c = 25 \text{ m}^{-3} \text{ s}^{-1}$, $\epsilon = 20 \text{ s}$, $\delta_1^v = 0.8 \text{ m}$, $\delta_1^t = 200 \text{ s}$, and $\delta^x = 0.2 \text{ m}$. As expected, both programs produce identical results at $t = 500 \text{ s}$, and thus we consider them verified (SI Fig 1a).
SI Fig 1: Verification of numerical implementations. a) Comparison of numerical solutions of rigid cell migration problem between Matlab and Abaqus; note that the normalized cell density $c_1/c_0$ is plotted against the normalized length $x/l$. b) Schematic of a unit cubic undergoing simple shear motion with an angle of $\theta$. c) The normalized shear stress $T_{12}/\mu$ and normalized normal stress difference $(T_{11} - T_{33})/\mu$ are plotted against the amount of shear $\gamma = \tan \theta$. 
For the mechanical problem, we prescribe a simple shear motion with an angle of $\theta$ to a unit cube (SI Fig 1b), such that the associated deformation gradient is given by [7]

$$
\mathbf{F} = \begin{bmatrix}
1 & \gamma & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}, \\
\mathbf{B} = \begin{bmatrix}
1 + \gamma^2 & \gamma & 0 \\
\gamma & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}, \\
\mathbf{C} = \begin{bmatrix}
1 & \gamma & 0 \\
\gamma & \gamma^2 + 1 & 0 \\
0 & 0 & 1
\end{bmatrix},
$$  \tag{S.22}

where $\gamma = \tan \theta$ denotes the amount of shear. Next, two assumptions are made to make the analytical solution tractable: 1) perfect incompressibility (i.e., $J_e = 1$), and 2) no growth (i.e., $\mathbf{F}^g = 1$). Under these assumptions, the Cauchy stress in Eq (15) is now given by

$$
\mathbf{T} = \mu \mathbf{B} + P \mathbf{1},
$$  \tag{S.23}

where $P$ is the constitutively indeterminate pressure that is introduced to satisfy the incompressibility constraint. In our simulation, we prescribe the same deformation to a single U3D8 element in Abaqus. Note that in our numerical treatment, we take $L = 10^3 \mu$ to approximate a nearly incompressible material in the simulation. We compare the analytical solutions for shear stress and normal stress difference, given by

$$
T_{12} = \mu \gamma \quad \text{and} \quad T_{11} - T_{33} = \mu \gamma^2,
$$  \tag{S.24}

respectively, against the numerical solutions (SI Fig 1c). Once again, the excellent agreement between analytical and numerical results indicate the mechanical portion of our finite element implementation is verified.

**Robustness of genetic algorithm**

To investigate the robustness of the genetic algorithm used in our model calibration, we focus on exploring the evolution of some of the key material parameters. More specifically, we keep track of four key material parameters ($G^c$, $v_i$, $D$, and $k_s$) and plot them as a function of genome up to 40 generations (SI Fig 2). We note that, under different ranges of initial guesses used in the calibration, most parameters can converge to steady values after roughly about 10 generations/100 genomes. Parameter $G^c$ appears to fall into a local minimum around 100 genomes, before converging to a lower value after approximately 300 genomes. This study reveals the robustness of the genetic algorithm.
SI Fig 2: Evolution of material parameters of a) baseline division rate $G_c$, b) migration velocity $v_i$, c) diffusion coefficient $D$, and d) subcortical growth parameter $k_s$ as a function of the genome for different ranges of initial guesses.
References

[1] Shawn A. Chester, Claudio V. Di Leo, and Lallit Anand. A finite element implementation of a
coupled diffusion-deformation theory for elastomeric gels. *International Journal of Solids and
Structures*, 52:1 – 18, 2015.

[2] Abaqus/Standard. *Abaqus Reference Manuals*. Dassault Systemes Simulia, Providence, RI,
2020.

[3] Shuolun Wang, Martina Decker, David L Henann, and Shawn A Chester. Modeling of dielectric
viscoelastomers with application to electromechanical instabilities. *Journal of the Mechanics
and Physics of Solids*, 95:213–229, 2016.

[4] Craig M Hamel, Fangda Cui, and Shawn A Chester. A finite element method for light activated
shape-memory polymers. *International Journal for Numerical Methods in Engineering*, 111(5):
447–473, 2017.

[5] Nikola Bosnjak, Shuolun Wang, Daehoon Han, Howon Lee, and Shawn A Chester. Modeling of
fiber-reinforced polymeric gels. *Mechanics Research Communications*, 96:7–18, 2019.

[6] Shuolun Wang and Shawn A Chester. Multi-physics modeling and finite element formulation
of corneal uv cross-linking. *Biomechanics and Modeling in Mechanobiology*, pages 1–18, 2021.

[7] Morton E Gurtin, Eliot Fried, and Lallit Anand. *The mechanics and thermodynamics of con-
inua*. Cambridge University Press, 2010.