A Finite Element Method for Neural Fields on Surfaces: Simulations and Bifurcation Analysis

Aaron Hagstrom
Adviser: Dr. Evan Gawlik

A thesis presented for the degree of
Master of Arts

Mathematics
University of Hawaii at Manoa
USA

1 Abstract

We introduce a finite element method for simulating neural fields on surfaces and we make observations on how neural field dynamics and the underlying synchrony might depend on the nature of the surface. The system of partial differential equations that we study is a generalization to curved surfaces of a system that has been previously used to model neural field dynamics on planar domains. We also conduct a Turing instability analysis through numerical methods to determine how patterns created by a neural field are dependent on curvature. This is a worthwhile research direction because of the need to understand neuronal behavior on the irregular curvature of the brain.

2 Introduction

Neural fields aim to describe the spatiotemporal evolution of coarse-grained variables in populations of neurons [5].

By expressing brain activity in terms of neural fields, we acknowledge that the brain may be using *population coding* to function, that is, representing information not in the spike train of any single neuron but as collective neuronal dynamics. We would like to solve a neural field on a surface or manifold, in this case, a sphere, that simulates the brain cortex [27].

A key part of the study of neural fields is pattern formation to make sense of the patterns seen in neuroimaging and other experimentation. Travelling waves in in-vitro slices of a rat brain were first evoked in the 1980s. Scientists have also witnessed spiral waves in schizophrenic and epileptic patients. Stationary localised pulses of high activity (i.e. bumps) or periodic patterns are characteristics of neural field models.

Many pattern-forming systems can be modelled as stable in-homogeneous solutions of the equations describing the system. These systems usually are non-linear, symmetric, and include spatial coupling and competition between two processes. In the neural context the two competing processes are usually synaptic excitation and inhibition[27].

Visser [28] has sought to describe the generation of EEG signals in terms of a neural field (PIDE) solved on a sphere. He reports exotic pattern formation, including a large rotating wave reported in EEG studies of schizophrenic patients. Visser utilized an unorthodox combination of linear stability analysis, symmetric bifurcation theory, center manifold reduction, and numerical simulations.

Martin, in her PhD thesis [19], uses collocation methods to solve the simplest neural field model called the Amari equation on a periodic square domain, a torus, and on neuroimaging representations of a human and rat brain.

On a representation of the human brain, Martin made two observations on the connection between manifold shape and the neural field: traveling bump solutions get trapped in regions of large negative curvature when moving along non-geodesic paths (similar in case of rat brain), and when following winding paths, bump solutions often break up into two bumps and traverse the sulcus (i.e. grooves in brain) in opposite directions.

Jirsa [15] has modeled a neural field as a nonlinear PDE and then mapped the field onto the folded cortex to help explain EEG and MEG patterns.

While all these researchers mentioned above, except Jirsa, studied very simple PIDEs on a surface, in this thesis, we are more interested in studying the PIDE reformulated as a PDE.

Neural fields can explain a phenomenon seen in EEG/MEG frequency bands called beta-rebound. This occurs when there is a sudden decrease in neural oscillatory power. This phenomenon may be created by local synchrony or *within-population synchrony*, which can witnessed in simulation as patterns within bumps.

We can construct network models, often called Firing Rate Models (FREs), at the microscopic level by connecting individual neurons or at the macroscopic level by coupling neuron-like units with outputs consisting of firing rates rather than action potentials. However, the main challenge is that FREs usually are not exact models of the underlying microscopic dynamics and assume the spiking activity of the neurons is asynchronous and so uncorrelated. Thus, they can't characterize the underlying system very well by themselves.

So, an overarching problem in mathematical and computational neuroscience is then the means by which to link the microscopic and macroscopic network descriptions of neurons in order to create a model that incorporates properties of both and that can be analyzed fairly easily and without great computational expense. For this reason, *mean-field reduction methods* have been instrumental in understanding the underlying dynamics of neuronal systems including the level of synchrony within a population of neurons.

With these methods, we can sometimes replace a large network of interacting neurons (i.e. oscillators) by a mean-field that exactly describes the properties of these underlying oscillators. These are called *next-generation models*. After the reduction process is complete, we obtain a low-dimensional dynamical system, a *neural mass model*. A neural field model can be thought of as a spatially-extended neural mass model. In this thesis, we study a next-generation neural field model of Byrne used to study EEG/MEG patterns [4].

3 The Finite Element Method

The finite element method is a numerical method used for solving differential equations, primarily partial differential equations (PDEs). The three main types of PDEs are elliptic, hyperbolic, and parabolic.

There are two main approaches to solving elliptic PDEs, namely the finite difference methods (FDM) and variational methods, under which the finite element method falls. The finite element method was largely developed by engineers beginning in the 1950s.

The main idea of the method is to develop the weak form, also known as the *integral form*, which involves multiplying the given equation by a trial function and then integrating. The next step is to *discretize* or break up the weak form into pieces. We do this by dividing the domain into *elements*. We then define interpolations or shape functions for each element in the form of summations, which enable us to write the PDE as a matrix equation, which can then be solved [29]. We motivate the problem by beginning with solving the Poisson Equation, the partial differential equation

$$\begin{cases} -\Delta u = f \quad \text{in} \quad \Omega \\ u = 0 \quad \text{on} \quad \partial\Omega \end{cases} \tag{1}$$

where $\Omega \subset \mathbf{R}^n$ and f is a given function.

We first define a classical or strong solution of this system as a function $u \in C^2(\Omega) \cap C^0(\overline{\Omega})$ that satisfies (1). We define

$$C^k(\Omega) = \{u : \Omega \to \mathbf{R}| \\ \text{u and all of its partial derivatives of order} \leq k \text{ are continuous on } \Omega\}$$

The next task is to derive the *weak formulation*. A weak solution of the Poisson equation is a function $u \in V$ satisfying

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v dx \quad \forall v \in V$$

For the 2D case,

$$V = \left\{ u \in L^2(\Omega) \middle| \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \in L^2(\Omega), \quad u = 0 \quad \text{on} \quad \partial\Omega \right\} \tag{2}$$

where V is called a Sobolev space (which is a type of Hilbert space) and the derivatives are called `weak derivatives', which are generalizations of the derivative of a function.

The space L^2 is a Lebesgue space. That is,

$$L^2(\Omega) = \left\{ u : \Omega \to \mathbf{R} \middle| \int_\Omega u^2 dxdy < \infty \right\}$$

We derive the weak formulation by means of the divergence theorem. For every smooth function v that vanishes on $\partial\Omega$, we have

$$\begin{aligned} \int_\Omega (fv)dxdy &= \int_\Omega (-\Delta uv)dxdy \\ &= \int_\Omega -div(\nabla u)vdxdy \\ &= \int_\Omega -div((\nabla u)v)dxdy + \int_\Omega (\nabla u \cdot \nabla v)dxdy \\ &= \int_{\partial\Omega} -((\nabla u)v) \cdot nds + \int_\Omega (\nabla u \cdot \nabla v)dxdy \\ &= \int_{\partial\Omega} -v\frac{\partial u}{\partial n}ds + \int_\Omega (\nabla u \cdot \nabla v)dxdy \\ &= \int_\Omega (\nabla u \cdot \nabla v)dxdy \end{aligned}$$

The third step follows from the product rule applied to $div((\nabla u)v)$ The third to last step follow from Green's Theorem. The last step follows from $v = 0$ on the boundary.

In abstract terms, we want to find a $u \in V$ such that

$$a(u,v) = \ell(v) \quad \forall v \in V$$

where $a : V \times V \to \mathbf{R}$ is a bilinear form that is symmetric and positive and ℓ is a linear functional. Here,

$$\begin{aligned} a(u,v) &= \int_\Omega (\nabla u \cdot \nabla v) dx dy \\ \ell(v) &= \int_\Omega (fv) dx dy \end{aligned}$$

To solve this problem, we can define a finite dimensional subspace V_h of V, consisting of continuous, piecewise linear functions and thereby we can discretize the weak formulation in the following terms: Find a $u_h \in V_h$ such that

$$a(u_h, v_h) = \ell(v_h) \quad \forall v_h \in V_h$$

where

$$V_h = \{u_h \in C^0(\overline{\Omega}) \mid u_h|_T \quad \text{is linear} \quad \forall T \in T_h \quad \text{and} \quad u_h = 0 \quad \text{on} \quad \partial\Omega\}$$

where T_h is a triangulation of Ω or a set of triangles $T_1, ...T_m$ such that

$$\begin{gathered} T_h = \{T_1, T_2, ..., T_m \mid \overline{\Omega} = \cup_{i=1}^m T_i \quad \forall i \neq j, T_i \cap T_j \\ \text{is either empty, a common vertex, or a common edge}\} \end{gathered}$$

The next task is to solve for u_h.

To do this, we first define a basis for the space V_h. Define $N = \text{Dim}(V_h)$. If $N_{\overline{\Omega}}$ is the total number of vertices and $N_{\partial\Omega}$ is the total number of vertices on $\partial\Omega$, then

$$N = N_{\overline{\Omega}} - N_{\partial\Omega}$$

Then, define the basis of V_h as $\{\phi_j(x,y)\}_{j=1}^N$, where

$$\phi_i(x_j, y_j) = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}$$

where (x_j, y_j) is the position of the jth vertex. Our solution will be a weighted sum of these basis functions. That is:

$$\begin{aligned} u_h(x,y) &= \sum_{i=1}^N c_i \phi_i(x,y) \\ v_h(x,y) &= \sum_{j=1}^N d_j \phi_j(x,y) \end{aligned}$$

Substituting these functions into

$$a(u_h, v_h) = \ell(v_h)$$

and defining the "stiffness" matrix as

$$\begin{aligned} A_{ij} &= a(\phi_i, \phi_j) \\ &= \sum_{T \in T_h} \int_T (\nabla \phi_i \cdot \nabla \phi_j) dxdy \end{aligned}$$

and defining

$$\begin{aligned} b_j &= \ell(\phi_j) \\ &= \sum_{T \in T_h} \int_T f \phi_j dxdy \end{aligned}$$

we get

$$\begin{aligned} a(u_h, v_h) &= \sum_{i,j} a(c_i \phi_i, d_j \phi_j) \\ &= \sum_{i.j} c_i a(\phi_i, \phi_j) d_j \\ &= \sum_{i,j} c_i A_{ij} d_j \\ &= \mathbf{c^t A d} \end{aligned}$$

and

$$\begin{aligned} \ell(v_h) &= \sum_j \ell(\phi_j) d_j \\ &= \mathbf{b^t d} \end{aligned}$$

and so,

$$\begin{aligned} \mathbf{c^t A d} = \mathbf{b^t d} \quad \forall \mathbf{d} \in \mathbf{R}^N \\ \iff \mathbf{c^t A} = \mathbf{b^t} \\ \iff \mathbf{A^t c} = \mathbf{b} \\ \iff \mathbf{A c} = \mathbf{b} \\ \iff \mathbf{c} = \mathbf{A}^{-1} \mathbf{b} \end{aligned}$$

3.1 Natural Coordinates

We use what are called *natural coordinates* so that we can use Gaussian quadrature to compute the integrals appearing in the equations for a and ℓ.

We would like to map the vertices of an element $T \in T_h$ in the triangulation in the xy plane to the vertices of a reference triangle $\hat{T}$ in the $\hat{x}\hat{y}$ plane, that is, to a right triangle in the first quadrant of the Real plane having one vertex at the origin and with two sides of length 1. Define the mapping

$$\begin{aligned} F : \hat{T} &\to T \\ (\hat{x}, \hat{y}) &\mapsto (x, y) \end{aligned}$$

Denote the vertex coordinates of T as

$$\begin{gathered}(x_{i_1}, y_{i_1})\\(x_{i_2}, y_{i_2})\\(x_{i_3}, y_{i_3})\end{gathered}$$

Denote the vertex coordinates of $\hat{T}$ as

$$\begin{gathered}(0,0)\\(1,0)\\(0,1)\end{gathered}$$

Define the map between $\hat{T}$ and T as

$$\begin{bmatrix}x\\y\end{bmatrix} = F(\hat{x},\hat{y}) = \begin{bmatrix}x_{i_1}\\y_{i_1}\end{bmatrix} + \begin{bmatrix}x_{i2}-x_{i1} & x_{i3}-x_{i1}\\y_{i2}-y_{i1} & y_{i3}-y_{i1}\end{bmatrix}\begin{bmatrix}\hat{x}\\\hat{y}\end{bmatrix}$$

Define $\hat{\phi}_a(\hat{x},\hat{y}) = \phi_{i_a}(x,y) = \phi_{i_a}(F(\hat{x},\hat{y}))$ for $a = 1,2,3$.
$\hat{\phi}_1(\hat{x},\hat{y})$ is the unique linear function on $\hat{T}$ satisfying

$$\begin{cases}\hat{\phi}_1(0,0) = 1\\\hat{\phi}_1(1,0) = 0\\\hat{\phi}_1(0,1) = 0\end{cases}$$

$\hat{\phi}_2(\hat{x},\hat{y})$ is the unique linear function on $\hat{T}$ satisfying

$$\begin{cases}\hat{\phi}_2(0,0) = 0\\\hat{\phi}_2(1,0) = 1\\\hat{\phi}_2(0,1) = 0\end{cases}$$

$\hat{\phi}_3(\hat{x},\hat{y})$ is the unique linear function on $\hat{T}$ satisfying

$$\begin{cases}\hat{\phi}_3(0,0) = 0\\\hat{\phi}_3(1,0) = 0\\\hat{\phi}_3(0,1) = 1\end{cases}$$

Solving these systems of equations, we get that

$$\begin{aligned}\hat{\phi}_1(\hat{x},\hat{y}) &= \phi_{i_1}(x,y) = \phi_{i_1}(F(\hat{x},\hat{y})) = 1-\hat{x}-\hat{y}\\\hat{\phi}_2(\hat{x},\hat{y}) &= \phi_{i_2}(x,y) = \phi_{i_2}(F(\hat{x},\hat{y})) = \hat{x}\\\hat{\phi}_3(\hat{x},\hat{y}) &= \phi_{i_3}(x,y) = \phi_{i_3}(F(\hat{x},\hat{y})) = \hat{y}\end{aligned}$$

And so,

$$\begin{aligned}\hat{\nabla}\hat{\phi}_1 &= \begin{bmatrix}-1\\-1\end{bmatrix}\\\hat{\nabla}\hat{\phi}_2 &= \begin{bmatrix}1\\0\end{bmatrix}\\\hat{\nabla}\hat{\phi}_3 &= \begin{bmatrix}0\\1\end{bmatrix}\end{aligned}$$

Observe that

$$\hat{\nabla}\hat{\phi}_a = \begin{bmatrix} \frac{\partial\hat{\phi}_a}{\partial\hat{x}} \\ \frac{\partial\hat{\phi}_a}{\partial\hat{y}} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial\hat{x}} & \frac{\partial y}{\partial\hat{x}} \\ \frac{\partial x}{\partial\hat{y}} & \frac{\partial y}{\partial\hat{y}} \end{bmatrix} \begin{bmatrix} \frac{\partial\phi_{i_a}}{\partial x} \\ \frac{\partial\phi_{i_a}}{\partial y} \end{bmatrix} = J^T\nabla\phi_{i_a}$$

So,

$$\nabla\phi_{i_a} = J^{-T}\hat{\nabla}\hat{\phi}_a$$

Also,

$$dydy = |J|d\hat{x}d\hat{y}$$

So,

$$\int_T \nabla\phi_{i_a}\cdot\nabla\phi_{i_b}dxdy = \int_{\hat{T}}(J^{-T}\hat{\nabla}\hat{\phi}_a)\cdot(J^{-T}\hat{\nabla}\hat{\phi}_b)|J|d\hat{x}d\hat{y}$$

In the same way,

$$\int_T f\phi_{i_a}dxdy = \int_{\hat{T}}\hat{f}\hat{\phi}_a|J|d\hat{x}d\hat{y}$$

where $\hat{f}(\hat{x},\hat{y}) = f(x,y) = f(F(\hat{x},\hat{y}))$.

3.2 Time Dependent Heat Equation

To motivate time-dependent neural field problems, we will examine the time-dependent heat equation, which we can formulate as the following:

$$\begin{cases} -\Delta u & = f - \frac{\partial u}{\partial t} \quad \text{in} \quad \Omega \\ u & = 0 \quad \text{on} \quad \partial\Omega \end{cases} \tag{3}$$

where t is time.

We want to find $u \in H_0^1(\Omega)$, which is the Sobolev space of equation (2), such that

$$\int_\Omega \frac{\partial u}{\partial t}v + \int_\Omega \nabla u\cdot\nabla v = \int_\Omega fv$$

$\forall v \in H_0^1(\Omega)$. We can reframe this as finding a $u_h \in V_h$ as in the case of the Poisson problem such that

$$\int_\Omega \frac{\partial u_h}{\partial t}v_h + \int_\Omega \nabla u_h\cdot\nabla v_h = \int_\Omega fv_h$$

$\forall v_h \in V_h$.

Next, choosing $\{\phi_i\}_{i=1}^N$ as a basis for V_h, we can write that

$$u_h(t,x,y) = \sum_{i=1}^N c_i(t)\phi_i(x,y)$$

$$v_h(t,x,y) = \sum_{j=1}^N d_j(t)\phi_j(x,y)$$

This leads to

$$\sum_{i,j}\int_\Omega \frac{dc_i}{dt}\phi_i(x,y)\cdot d_j\phi_j(x,y) + \int_\Omega c_i(t)\nabla\phi_i(x,y)\cdot d_j\nabla\phi_j(x,y) = \sum_j\int_\Omega f d_j\phi_j(x,y)$$

$$\sum_{i,j}\frac{dc_i}{dt}\left(\int_\Omega \phi_i\cdot\phi_j\right)d_j + c_i\left(\int_\Omega \nabla\phi_i\cdot\nabla\phi_j\right)d_j = \sum_j d_j\int_\Omega f\phi_j$$

Replacing $\int_\Omega \phi_i\cdot\phi_j$ with M_{ij} and replacing $\int_\Omega \nabla\phi_i\cdot\nabla\phi_j$ with A_{ij} and $\int_\Omega f\phi_j$ with b_j, we get

$$\sum_{i,j}\frac{dc_i}{dt}M_{ij}d_j + c_iA_{ij}d_j = \sum_j d_jb_j$$

Or as matrices, we can write as

$$\mathbf{\dot{c}^T M d} + \mathbf{c^T A d} = \mathbf{b^T d} \quad \forall\mathbf{d}$$

where **M** is the mass matrix and **A** is the stiffness matrix.
So,

$$\begin{aligned}\mathbf{\dot{c}^T M} + \mathbf{c^T A} &= \mathbf{b^T}\\ \mathbf{M^T\dot{c}} + \mathbf{A^T c} &= \mathbf{b}\\ \mathbf{M\dot{c}} + \mathbf{Ac} &= \mathbf{b}\end{aligned}$$

This is called the *semi-discrete Galerkin* formulation because we have only discretized the spatial variable but not the time derivative. One of the more common methods to discretize the time derivative is the *θ-method*, which involves replacing the time derivative with the difference

$$\frac{\partial\mathbf{c}}{\partial t} \approx \frac{\mathbf{c^{n+1}} - \mathbf{c^n}}{\Delta t} \tag{4}$$

where Δt is the time increment, and $t_{n+1} = t_n + \Delta t$. We introduce a *relaxation parameter θ* and write **c** as

$$\mathbf{c} = \theta\mathbf{c^{n+1}} + (1-\theta)\mathbf{c^n} \tag{5}$$

for $t_n \le t \le t_{n+1}$ where θ is usually specified as being in the range $0 \le \theta \le 1$ and the n exponent denotes a point in time. This parameter controls the accuracy and stability of the algorithm. Setting $\theta = 0$, we get $\mathbf{c} = \mathbf{c^n}$

$$\begin{aligned}\mathbf{M}\left(\frac{\mathbf{c^{n+1}} - \mathbf{c^n}}{\Delta t}\right) + \mathbf{Ac^n} &= \mathbf{b}\\ \mathbf{Mc^{n+1}} &= \mathbf{Mc^n} + \Delta t(\mathbf{b} - \mathbf{Ac^n})\\ \mathbf{c^{n+1}} &= \mathbf{M^{-1}}(\mathbf{Mc^n} + \Delta t(\mathbf{b} - \mathbf{Ac^n}))\end{aligned}$$

which is the solution.

3.3 Implementation

We take six general steps when implementing the Finite Element Method for the time-dependent heat equation in MATLAB.

1. Input data for boundary conditions, forces, and geometry of domain.

2. Define the triangulation T_h

3. Assemble the global stiffness matrix $\mathbf{A}$, the mass matrix $\mathbf{M}$, and force vector $\mathbf{b}$.

4. Discretize the time derivative $\dot{\mathbf{c}}$

5. Solve the system $\mathbf{M}\dot{\mathbf{c}} + \mathbf{A}\mathbf{c} = \mathbf{b}$

6. Plot the resulting solution.

4 Error Analysis

4.1 Energy Norm (i.e. a-norm)

We would like to understand how much error results from an FEM formulation. Ultimately, we are interested in analyzing finite element methods for neural fields on surfaces, but for now, we will be content with a 1D formulation that gives the general idea of what we are trying to accomplish.

Let $V_h \subset V$ be a finite-dimensional subspace of V, where V is a Hilbert Space, and let $a : V \times V \to \mathbf{R}$ be a symmetric bilinear form satisfying three conditions. The first and last are continuity properties and the second is called coercivity.

$$\begin{cases} |a(u,v)| & \leq M\|u\|_V\|v\|_V \quad \forall u,v \in V \\ a(u,u) & \geq \alpha\|u\|_V^2 \quad \forall u \in V \\ \|\ell\|_{V'} & = \underset{v\in V}{sup}\frac{|\ell(v)|}{\|v\|_V} < \infty \end{cases}$$

where M and α are positive constants.

We can first conduct an error analysis of the a-norm of the error, where

$$\|u\|_a = a(u,u)^{\frac{1}{2}}$$

We can prove that if $u \in V$ satisfies

$$\begin{aligned} a(u,v) &= \ell(v) \\ &\forall v \in V \end{aligned}$$

and $u_h \in V_h$ satisfies

$$\begin{aligned} a(u_h,v_h) &= \ell(v_h) \\ &\forall v_h \in V_h \end{aligned}$$

then,

$$\|u - u_h\|_a = \inf_{v_h \in V_h} \|u - v_h\|_a$$
$$\|u\|_a = a(u,u)^{1/2}$$

The weak formulation implies that

$$a(u, v_h) = \ell(v_h)$$
$$\forall v_h \in V_h$$

and the Galerkin formulation implies that

$$a(u_h, v_h) = \ell(v_h)$$
$$\forall v_h \in V_h$$

Subtracting the second expression from the first, we obtain

$$a(u - u_h, v_h) = 0$$
$$\forall v_h \in V_h$$

which can be interpreted as "$u - u_h$ is a-orthogonal to V_h."

This fact becomes useful in evaluating the expression

$$\begin{aligned} \|u - u_h\|_a^2 &= a(u - u_h, u - u_h) \\ &= a(u - u_h, u - v_h) + a(u - u_h, v_h - u_h) \\ &= a(u - u_h, u - v_h) \\ &\leq \|u - u_h\|_a \|u - v_h\|_a \\ \forall v_h \in V_h & \end{aligned}$$

So, we have

$$\|u - u_h\|_a^2 \leq \|u - u_h\|_a \|u - v_h\|_a$$
$$\forall v_h \in V_h$$

So,

$$\|u - u_h\|_a = \inf_{v_h \in V_h} \|u - v_h\|_a \tag{6}$$

4.2 Discretization Error

Now we will analyze the error in the V-norm.

We can prove that

$$\|u - u_h\|_V \leq \sqrt{\frac{M}{\alpha}} \inf_{v_h \in V_h} \|u - v_h\|_V$$

by the following argument

$$
\begin{aligned}
\alpha \|u - u_h\|_V^2 &\leq a(u - u_h, u - u_h) \\
&= \|u - u_h\|_a^2 \\
&\leq \|u - v_h\|_a^2 \\
&= a(u - v_h, u - v_h) \\
&\leq M \|u - v_h\|_V^2 \\
\forall v_h \in V_h
\end{aligned}
$$

So,

$$
\|u - u_h\|_V \leq \sqrt{\frac{M}{\alpha}} \|u - v_h\|_V \qquad \forall v_h \in V_h
$$

4.3 Interpolation Error

We wish to consider a one-dimensional problem $-u'' = f$ on $\Omega = (0,1)$ with boundary conditions $u(0) = u(1) = 0$, and we're partitioning the interval $(0,1)$ into subintervals of length h and assume $u \in H^2(\Omega)$. Let

$$
\begin{cases}
V & = H_0^1(\Omega) \\
V_h & = \{v_h \in C^0(\Omega) | v_h|_{[jh,(j+1)h]} \text{ is linear } \forall j \text{ and } v_h(0) = v_h(1) = 0\} \\
\|u\|_a & = (\int_\Omega u'(x)^2 dx)^{\frac{1}{2}} = |u|_{H^1(\Omega)} \\
\|u\|_V & = \|u\|_{H^1(\Omega)} = (\int_0^1 u(x)^2 dx + \int_0^1 u'(x)^2 dx)^{\frac{1}{2}} = (\|u\|_{L^2(\Omega)}^2 + |u|_{H^1(\Omega)}^2)^{\frac{1}{2}} \\
a(u,v) & = \int_\Omega u'(x) v'(x) dx \\
\ell(v) & = \int_\Omega f(x) v(x) dx
\end{cases}
$$

By equation (6), we can write

$$
|u - u_h|_{H^1(\Omega)} = \underset{v_h \in V_h}{inf} |u - v_h|_{H^1(\Omega)} \tag{7}
$$

We want to show that the right-hand side of (7) is small.

We need to find some $v_h \in V_h$, (where v_h is an arbitrary piecewise linear function) such that $|u - v_h|_{H^1(\Omega)}$ is small. We can choose v_h to be a piecewise linear interpolant of u. In other words

$$
v_h = I_h u \tag{8}
$$

This mean that u is approximated with an interpolant. We can express v_h as a summation over basis functions defined at gridpoints x_i.

As noted in (8), we are defining v_h as interpolant of u and so

$$
\begin{aligned}
v_h(x) &= I_h u(x) \\
&= \sum_{i=1}^{N} u(x_i) \phi_i(x)
\end{aligned}
$$

We know that

$$\begin{aligned} v_h(x_j) &= \sum_{i=1}^{N} u(x_i)\phi_i(x_j) \\ &= u(x_j) \end{aligned}$$

We can prove that in one-dimension

$$|u - v_h|_{H^1(\Omega)} \leq h|u|_{H^2(\Omega)}$$

Denote

$$e(x) = u(x) - v_h(x)$$

such that

$$\begin{aligned} |u - v_h|^2_{H^1(\Omega)} &= \int_0^1 e'(x)^2 dx \\ &= \sum_{j=0}^{N} \int_{jh}^{(j+1)h} e'(x)^2 dx \end{aligned}$$

By the Mean Value Theorem, there exists $\xi \in [jh, (j+1)h]$ such that

$$e'(\xi) = \frac{e((j+1)h) - e(jh)}{h} = 0$$

And so, $\forall x \in [jh, (j+1)h]$,

$$\begin{aligned} e'(x) &= \int_\xi^x e''(y)dy \\ |e'(x)| &= \left| \int_\xi^x e''(y)dy) \right| \\ &\leq |x - \xi|^{\frac{1}{2}} \left(\int_\xi^x e''(y)^2 dy \right)^{\frac{1}{2}} \\ &\leq h^{\frac{1}{2}} \left(\int_{jh}^{(j+1)h} e''(y)^2 dy \right)^{\frac{1}{2}} \\ \int_{jh}^{(j+1)h} |e'(x)|^2 dx &\leq \int_{jh}^{(j+1)h} h \int_{jh}^{(j+1)h} e''(y)^2 dydx \\ &= h^2 \int_{jh}^{(j+1)h} e''(y)^2 dy \\ &= h^2 \int_{jh}^{(j+h)j} u''(y)^2 dy \end{aligned}$$

Since,

$$e'' = u'' - v_h'' = u'' \text{ on } (jh, (j+1)h).$$

So,

$$\sum_{j=0}^{N} \int_{jh}^{(j+1)h} |e'(x)|^2 dx \leq h^2 \sum_{j=0}^{N} \int_{jh}^{(j+1)h} u''(y)^2 dy$$
$$|e|^2_{H^1(\Omega)} \leq h^2 |u|^2_{H^2(\Omega)}$$
$$|u - v_h|_{H^1(\Omega)} \leq h|u|_{H^2(\Omega)}$$

This means that we can reduce the error between u and v_h in the H^1 seminorm significantly just by making h arbitrarily small.

To summarize, if $u \in V$ is the solution of $a(u,v) = \ell(v) \quad \forall v \in V$ and $u_h \in V_h$ is the solution of $a(u_h, v_h) = \ell(v_h) \quad \forall v_h \in V_h$, then

$$|u - u_h|_{H^1(\Omega)} \leq h|u|_{H^2(\Omega)} \tag{9}$$

if $u \in H^2(\Omega)$. Equation (9) means that the derivatives of u_h are converging to the derivatives of u at a linear rate.

5 Surface PDEs

5.1 The Basics

We are interested in simulating neural fields, reformulated as partial differential equations, on surfaces. These types of equations currently have applications in image analysis [17, 20], materials sciences[11], among other areas.

Begin by defining a smooth, connected, oriented hypersurface in $\mathbf{R}^{n+1}$, $n = 1, 2$ [10].

Let $d = d(x)$ with $x \in \mathbf{R}^{n+1}$ be a smooth level set function. Let $\mathcal{N}$ be an open subset of $\mathbf{R}^{n+1}$, where $\nabla d \neq 0$.

Define a surface Γ as

$$\Gamma = \{x \in \mathcal{N} | d(x) = 0\}$$

Assume d is twice differentiable,

$$d \in C^2(\mathcal{N})$$

Fix the orientation of Γ such that the normal $\hat{n}$ to Γ is in the direction of increasing d.

The normal vector field can be defined as

$$\hat{n}(x) = \frac{\nabla d(x)}{|\nabla d(x)|}$$

where $x \in \mathbf{R}^{n+1}$.

Denote the projection at x onto the tangent space of Γ as having the i,j-element

$$P(x)_{ij} = \delta_{ij} - \hat{n}(x)_i \hat{n}(x)_j$$

We can choose $\mathcal{N}$ and a $d(x)$ (i.e. one possible choice is a signed distance function) such that for every $x \in N$ there exists a unique $a(x) \in \Gamma$ such that

$$x = a(x) + d(x)\hat{n}(a(x))$$

Define the tangential gradient on Γ by

$$\nabla_\Gamma \eta = \nabla \eta - \nabla \eta \cdot \hat{n}\hat{n} = P\nabla \eta$$

for a function η defined on $\mathcal{N}$, where $\nabla \eta$ denotes the gradient on $\mathbf{R}^{n+1}$.

We know that

$$\nabla_\Gamma \eta \cdot \hat{n} = 0.$$

Let $\underline{D}$ be the tangential gradient. The components of the tangential gradient are defined as

$$\nabla_\Gamma \eta = (\underline{D}_1 \eta, ..., \underline{D}_{n+1}\eta)$$

Define the Laplace-Beltrami operator on Γ as the tangential divergence of the tangential gradient:

$$\Delta_\Gamma \eta = \nabla_\Gamma \cdot \nabla_\Gamma \eta = \sum_{i=1}^{n+1} \underline{D}_i \underline{D}_i \eta$$

If we denote the outer unit normal vector of $\partial\Gamma$ by μ we can see that

$$\int_\Gamma \underline{D}_i \eta = -\int_\Gamma \eta H \hat{n}_i + \int_{\partial\Gamma} \eta \mu_i$$

where $H = -\nabla \cdot \hat{n}$ is the mean curvature (the sum of the principal curvatures) with respect to $\hat{n}$.

This equation yields the divergence theorem for

$$\xi = (\xi_1, \xi_2, ..., \xi_{n+1}) :$$

$$\int_{\partial\Gamma} \xi \cdot \mu = \int_\Gamma \nabla_\Gamma \cdot \xi + \int_\Gamma \xi \cdot \hat{n} H$$

Green's formula on the surface Γ is

$$\int_\Gamma \nabla_\Gamma \xi \cdot \nabla_\Gamma \eta = \int_{\partial\Gamma} \xi \nabla_\Gamma \eta \cdot \mu - \int_\Gamma \xi \Delta_\Gamma \eta$$

Much of our results are computed on the sphere. In this case, we let

$$\Gamma = \{x \in \mathbf{R}^{n+1} || x - x_0| = R\}$$

and

$$d(x) = R - |x - x_0|$$

where R is the radius and x_0 is the center of the sphere. The normal vectors are pointing inward and mean curvature is

$$H = \frac{n}{R}$$

For Hilbert spaces on surfaces, define the following.

$$H^1(\Gamma) = \{\eta \in L^2(\Gamma) | \nabla_\Gamma \eta \in L^2(\Gamma)^{n+1}\}$$

$$H_0^1(\Gamma) = \{\eta \in H^1(\Gamma) | \eta = 0 \quad \text{on} \quad \partial\Gamma\}$$

5.2 The Jacobian and its Pseudo-determinant

In our finite element formulation on a sphere, we will need to use a different Jacobian and determinant. The Jacobian for a two-dimensional manifold immersed in $\mathbf{R}^3$ is given by

$$\begin{aligned}\mathbf{J} &= [\mathbf{J_1}, \mathbf{J_2}] \\ &= \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \\ \frac{\partial z}{\partial \hat{x}} & \frac{\partial z}{\partial \hat{y}} \end{bmatrix}\end{aligned}$$

The Jacobian pseudo-determinant is the transformation of the volume of the differential integral measure.
The determinant is the volume of the parallelogram spanned by the two columns of $\mathbf{J}$.

$$|\mathbf{J}| = |\mathbf{J_1} \times \mathbf{J_2}|_2$$

where $|\cdot|_2$ is the Euclidean norm.

6 Mathematical Models of Neurons

6.1 Properties of Neurons

A neuron is an electrically excitable cell comprising three distinct parts: the soma, the axon, and the dendrites. The central processing unit is the soma. The output device is the axon, where the signals go to other neurons. Where the axon meets the dendrites of another neuron is called a chemical synapse. Signals from the neuron are short electrical pulses called action potentials or spikes and a chain of these spikes is called a spike train. Gap junctions are a type of direct electrical junction between neurons and occur between dendrites and dendrites or axons and axons [12]. Neurons mainly communicate with one another by sending signals along their axons. There are about 10^{11} neurons in the human brain. Per cubic millimeter, there are 10^4 neural cells.

A neuron receives inputs from about $10,000$ other neurons through synapses on the dendritic tree. This produces changes in the membrane potential of the neuron called post-synaptic potentials (PSPs) and leads to an action potential or spike when the PSPs are greater than the *firing threshold*.

7 Neural Mass Models

We use the term *neural mass model* to refer to low-dimensional models that attempt to describe the coarse-grained activity of large populations of neurons and synapses. They are typically systems of ordinary differential equations (ODEs) and track the activity of an excitatory population of neurons coupled to an inhibitory population. Neural mass models can also be thought of as the neural field models without the spatial component. We only consider one population of neurons in this thesis.

Byrne and co-authors [4] modify a reduction model of Montbrio [21] by adding gap junctions which are known to promote synchrony and expanding it to a PDE model on a planar surface. They derive FREs that correspond exactly to an all-to-all coupled population of what are called all-to-all coupled Quadratic Integrate-and-Fire (QIF) neurons. This system is the neural mass model

$$\begin{cases} \tau\dot{R} = -\kappa_V R + 2RV + \frac{\gamma}{\pi\tau} \\ \tau\dot{V} = \eta_0 + V^2 - \pi^2\tau^2 R^2 + \kappa_s U \\ QU = R \end{cases} \tag{10}$$

where R is the instantaneous mean firing rate or fraction of neurons firing at a time t, V is the average membrane potential and U is the synaptic activity being driven by the mean firing rate. Here, Q is a linear differential operator that we discuss further below. As for parameters, κ_s is the synaptic coupling constant, κ_V is the gap-junction coupling constant, η_0 is a constant that describes the excitability of neurons in the system, τ is some constant, γ is the degree of heterogeneity in the network.

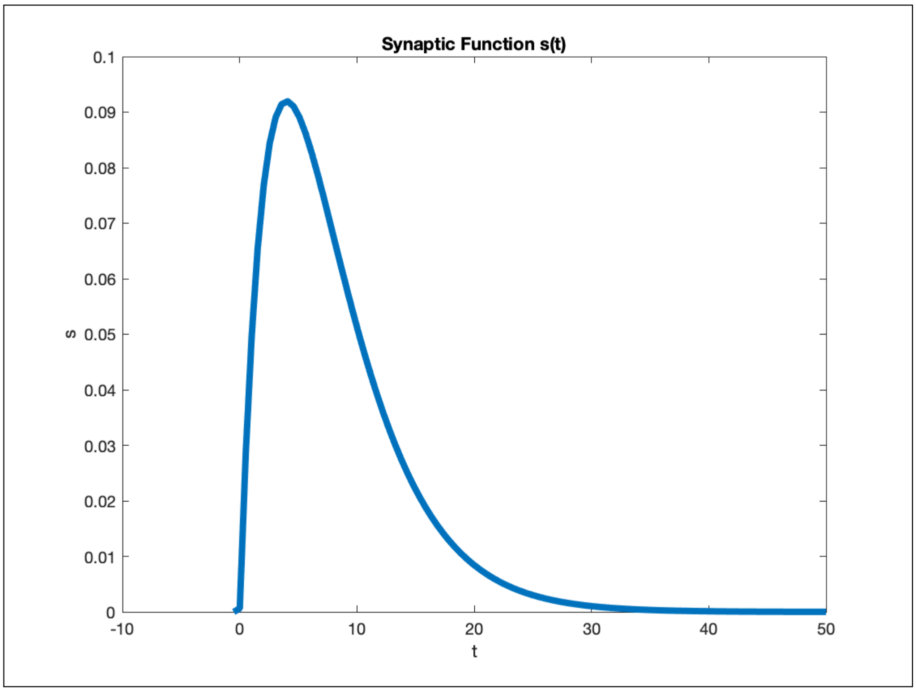
To describe synaptic activity U, we denote the $m'th$ firing time of a neuron j by T_j^m, and we say that the current received by neuron i from neuron j is proportional to

$$\sum_{m\in\mathbf{Z}} s(t - T_j^m)$$

where s describes rise and fall of post-synaptic response. We often denote s to be the Green's function of a linear differential operator Q. In other words,

$$Qs = \delta$$

where δ is a Dirac delta function. If $s(t) = \alpha^2 t e^{-\alpha t} H(t)$ where H is a Heaviside step function and $\frac{1}{\alpha}$ is the time to peak, then $Q = \left(1 + \frac{1}{\alpha}\frac{d}{dt}\right)^2$.



If we take a mean-field approach, then we derive a neural mass model, which can be seen as a neural field without the spatial aspect. The synaptic activity s becomes a mean synaptic activity U and the Dirac delta becomes a mean firing rate, which gives us

$$QU = R,$$

which is the third equation in (10).

A gap junction is modeled as the difference in voltage between two neurons (i.e. $v_j - v_i$), where current is flowing from neuron i to neuron j.

8 Neural Fields

8.1 The Basic Model

Neural fields are continuum models for analyzing large scale population dynamics of neurons that we can derive from the neural mass models. In a continuum model on a 2D spatial domain, each point (x, y) corresponds to a density of neurons to which is associated an ordered pair (R, U) where R is the instantaneous mean firing rate or fraction of neurons firing at a time t and U describes the mean synaptic activity at a time t. In Byrne's model [4], we also include V, the average membrane potential, so we will have an ordered triple $(R(\boldsymbol{r}, t), V(\boldsymbol{r}, t), U(\boldsymbol{r}, t))$ at each point. We obtain partial differential equations for (R, V, U) by replacing temporal derivatives with partial derivatives in the neural mass model and also replacing the temporal dynamics $QU = R$ with the spatial dynamics $QU = \Psi$ where Ψ is the spatial drive. Here

$$\Psi = w \otimes R = \int_{\mathbf{R}} dt' \int_{\mathbf{R}^2} d\boldsymbol{r}' G(\boldsymbol{r} - \boldsymbol{r}', t - t') R(\boldsymbol{r}', t')$$

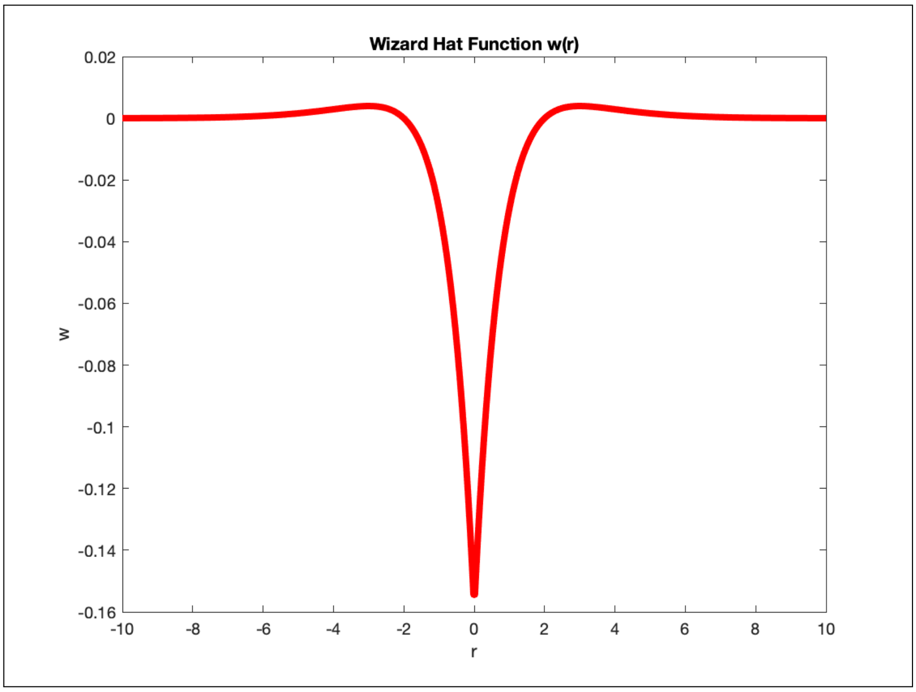
where where $\otimes$ denotes spatial interactions within the neural field model, $G(\boldsymbol{r}, t) = w(r)\delta(t - \frac{r}{v})$, $r = |\boldsymbol{r}|$, w is a connectivity function describing how neurons are spatially influencing each other, R is called a firing rate function, and $\boldsymbol{r} = (x, y)$ and $\boldsymbol{r}' = (x', y')$ are two points on the surface and t and t' are two different times.

8.2 The Connectivity Function

For w, we choose a *wizard-hat function*

$$w(r) = \frac{(\frac{r}{2} - 1)e^{-r}}{2\pi}$$

The function describes the phenomenon of *short-range inhibition long-range excitation*, which means neurons close to each other inhibit each other but neurons far from each other more excite each other. We could also have long-range inhibition and short-range excitation if we flip this function over the x-axis.



8.3 PDE Methods in 2D

There are two methods of translating neural fields, which are Partial Integro-Differential Equations (PIDEs), into PDEs. The first method uses the fact that the Fourier transform of the convolution of two functions is the product of their Fourier transforms[18].
A second method relies on fact that solutions to inhomogeneous linear differential equations can be written as the convolution of a Green's function and boundary conditions. For a certain choice of w, namely, the wizard-hat function above, we can obtain a PDE model referred to as a 2D *brain-wave equation*

$$\left[\left(1+\frac{1}{\nu}\frac{\partial}{\partial t}\right)^2-\frac{3}{2}\nabla^2\right]^2\Psi=-\left[\frac{1}{\nu}\frac{\partial}{\partial t}\left(1+\frac{1}{\nu}\frac{\partial}{\partial t}\right)-\frac{3}{2}\nabla^2\right]R \tag{11}$$

Coupling this brain-wave equation with our neural mass model yields the PDE system:

$$\begin{cases} \tau\frac{\partial R}{\partial t} & = -\kappa_V R + 2RV + \frac{\gamma}{\pi\tau} \\ \tau\frac{\partial V}{\partial t} & = \eta_0 + V^2 - \pi^2\tau^2R^2 + \kappa_s U \\ QU & = \Psi \\ \left[\left(1+\frac{1}{\nu}\frac{\partial}{\partial t}\right)^2-\frac{3}{2}\nabla^2\right]^2\Psi & = -\left[\frac{1}{\nu}\frac{\partial}{\partial t}\left(1+\frac{1}{\nu}\frac{\partial}{\partial t}\right)-\frac{3}{2}\nabla^2\right]R \end{cases} \tag{12}$$

8.4 FEM Discretization

We are considering the generalization of (12) to surfaces, which is obtained by replacing the Laplace operator Δ (or ∇^2) by the surface Laplacian Δ_Γ in (11).

We can discretize the 2D neural field model (12) with a mixed finite element method using substitutions of Byrne [4]. In the first step, we break up (11), which corresponds to the last equation of (12) into six PDEs with the substitutions A_1, A_2, A_3, A_4, p, g defined as:

$$\begin{aligned}
A_1 &= 1 + \frac{1}{\nu}\frac{\partial}{\partial t}\Psi \\
A_2 &= \left(1 + \frac{1}{\nu}\frac{\partial}{\partial t}\right)A_1 - \frac{3}{2}\Delta\Psi \\
A_3 &= \left(1 + \frac{1}{\nu}\frac{\partial}{\partial t}\right)A_2 \\
A_4 &= \frac{3}{2}\Delta A_2 - \left(\frac{1}{\nu}\frac{\partial R}{\partial t} + \frac{1}{\nu^2}\frac{\partial^2 R}{\partial t^2} - \frac{3}{2}\Delta R\right) \\
p &= \left(1 + \frac{1}{\alpha}\frac{\partial}{\partial t}\right)U \\
g &= U
\end{aligned}$$

These substitutions yield the following system of six partial differential equations.

$$\begin{cases}
\frac{\partial\Psi}{\partial t} & = \nu(-\Psi + A_1) \\
\frac{\partial A_1}{\partial t} & = \nu(-A_1 + A_2 + \frac{3}{2}\Delta\Psi) \\
\frac{\partial A_2}{\partial t} & = \nu(A_3 - A_2) \\
\frac{\partial A_3}{\partial t} & = \nu(A_4 - A_3) \\
\frac{\partial p}{\partial t} & = \alpha(\Psi - p) \\
\frac{\partial g}{\partial t} & = \alpha(p - g).
\end{cases}$$

Coupling this system with the two PDEs for $\dot{R}$ and $\dot{V}$, we get the eight-equation system

$$\begin{cases}
\frac{\partial R}{\partial t} & = \frac{1}{\tau}\big(- \kappa_V R + 2RV + \frac{\gamma}{\tau\pi}\big) \\
\frac{\partial V}{\partial t} & = \frac{1}{\tau}\big(\kappa_s g - (\pi\tau R)^2 + V^2 + \eta_0\big) \\
\frac{\partial\Psi}{\partial t} & = \nu(-\Psi + A_1) \\
\frac{\partial A_1}{\partial t} & = \nu(-A_1 + A_2 + \frac{3}{2}\Delta\Psi) \\
\frac{\partial A_2}{\partial t} & = \nu(A_3 - A_2) \\
\frac{\partial A_3}{\partial t} & = \nu(A_4 - A_3) \\
\frac{\partial p}{\partial t} & = \alpha(\Psi - p) \\
\frac{\partial g}{\partial t} & = \alpha(p - g).
\end{cases} \tag{13}$$

As in the case of the time-dependent heat equation, we would like to find

$$R_h, V_h, \Psi_h, (A_1)_h, (A_2)_h, (A_3)_h, p_h, g_h \in V_h$$

where V_h is the space of continuous piecewise linear functions, such that our weak formulation holds for all test functions $D_h \in V_h$. That is,

$$\begin{cases} (\dot{R}_h, D_h) & = \frac{1}{\tau}(-\kappa_V(R_h, D_h) + 2(R_h V_h, D_h) + \frac{\gamma}{\pi\tau}(1, D_h)) \\ (\dot{V}_h, D_h) & = \frac{1}{\tau}(\kappa_S(g_h, D_h) - (\pi^2\tau^2(R_h^2, D_h)) + (V_h^2, D_h) + \eta_0(1, D_h)) \\ (\dot{\Psi}_h, D_h) & = \nu(-(\psi_h, D_h) + ((A_1)_h, D_h)) \\ ((\dot{A}_1)_h, D_h) & = \nu(-((A_1)_h, D_h) + ((A_2)_h, D_h) - \frac{3}{2}(\nabla\Psi_h, \nabla D_h)) \\ ((\dot{A}_2)_h, D_h) & = \nu(-((A_2)_h, D_h) + (A_3, D_h)) \\ ((\dot{A}_3)_h, D_h) & = \nu(-((A_3)_h, D_h) + ((A_4)_h, D_h)) \\ (\dot{p}_h, D_h) & = \alpha(-(p_h, D_h) + (\Psi_h, D_h)) \\ (\dot{g}_h, D_h) & = \alpha(-(g_h, D_h) + (p_h, D_h)) \end{cases}$$

for every $D_h \in V_h$, where

$$((A_4)_h, D_h) = -\frac{3}{2}(\nabla(A_2)_h, \nabla D_h) - \frac{3}{2}(\nabla R_h, \nabla D_h) - \frac{1}{\nu^2}(\ddot{R}_h, D_h) - \frac{1}{\nu}(\dot{R}_h, D_h)$$

and where

$$(\ddot{R}_h, D_h) = \frac{1}{\tau}(-\kappa_V(\dot{R}_h, D_h) + 2(V_h\dot{R}_h, D_h) + 2(R_h\dot{V}_h, D_h))$$

Choosing $\{\phi_r(x,y,z)\}_{r=1}^N$ as a basis for V_h, we can write

$$\begin{cases} R_h(x,y,z,t) & = \sum_i R_i(t)\phi_i(x,y,z) \\ V_h(x,y,z,t) & = \sum_j V_j(t)\phi_j(x,y,z) \\ \Psi_h(x,y,z,t) & = \sum_k \Psi_k(t)\phi_k(x,y,z) \\ U_h(x,y,z,t) & = \sum_l U_l(t)\phi_l(x,y,z) \\ (A_1)_h(x,y,z,t) & = \sum_m (A_1)_m(t)\phi_m(x,y,z) \\ (A_2)_h(x,y,z,t) & = \sum_n (A_2)_n(t)\phi_n(x,y,z) \\ (A_3)_h(x,y,z,t) & = \sum_o (A_3)_o(t)\phi_o(x,y,z) \\ p_h(x,y,z,t) & = \sum_s p_s(t)\phi_s(x,y,z) \\ g_h(x,y,z,t) & = \sum_q g_q(t)\phi_q(x,y,z) \end{cases} \tag{14}$$

Now, define the summations below

$$\begin{cases}
(\dot{R}_h, D_h) = \sum_{i,r}(\int \dot{R}_i(\phi_i \cdot \phi_r))D_r & = \dot{\mathbf{R}}^{\mathbf{T}}\mathbf{MD} \\
(\dot{V}_h, D_h) = \sum_{j,r}(\int \dot{V}_j(\ \phi_j \cdot \phi_r))D_r & = \dot{\mathbf{V}}^{\mathbf{T}}\mathbf{MD} \\
(\dot{\Psi}_h, D_h) = \sum_{k,r}(\int \dot{\Psi}_k(\ \phi_k \cdot \phi_r))D_r & = \dot{\mathbf{\Psi}}^{\mathbf{T}}\mathbf{MD} \\
((\dot{A}_1)_h, D_h) = \sum_{m,r}(\int (\dot{A}_1)_m \phi_m \cdot \phi_r))D_r & = (\dot{\mathbf{A}}_{\mathbf{1}})^{\mathbf{T}}\mathbf{MD} \\
((\dot{A}_2)_h, D_h) = \sum_{n,r}(\int (\dot{A}_2)_n(\ \phi_n \cdot \phi_r))D_r & = (\dot{\mathbf{A}}_{\mathbf{2}})^{\mathbf{T}}\mathbf{MD} \\
((\dot{A}_3)_h, D_h) = \sum_{o,r}(\int (\dot{A}_3)_o(\ \phi_o \cdot \phi_r))D_r & = (\dot{\mathbf{A}}_{\mathbf{3}})^{\mathbf{T}}\mathbf{MD} \\
(\dot{p}_h, D_h) = \sum_{s,r}(\int \dot{p}_s(\ \phi_s \cdot \phi_r))D_r & = \dot{\mathbf{p}}^{\mathbf{T}}\mathbf{MD} \\
(\dot{g}_h, D_h) = \sum_{q,r}(\int \dot{g}_q(\ \phi_q \cdot \phi_r))D_r & = \dot{\mathbf{g}}^{\mathbf{T}}\mathbf{MD} \\
(R_h, D_h) = \sum_{i,r}(\int R_i(\phi_i \cdot \phi_r))D_r & = \mathbf{R}^{\mathbf{T}}\mathbf{MD} \\
(V_h, D_h) = \sum_{j,r}(\int V_j(\phi_j \cdot \phi_r))D_r & = \mathbf{V}^{\mathbf{T}}\mathbf{MD} \\
(\Psi_h, D_h) = \sum_{k,r}(\int \Psi_k(\phi_k \cdot \phi_r))D_r & = \mathbf{\Psi}^{\mathbf{T}}\mathbf{MD} \\
((A_1)_h, D_h) = \sum_{m,r}(\int (A_1)_m(\phi_m \cdot \phi_r))D_r & = \mathbf{A}_{\mathbf{1}}^{\mathbf{T}}\mathbf{MD} \\
((A_2)_h, D_h) = \sum_{n,r}(\int (A_2)_n(\phi_n \cdot \phi_r))D_r & = \mathbf{A}_{\mathbf{2}}^{\mathbf{T}}\mathbf{MD} \\
((A_3)_h, D_h) = \sum_{o,r}(\int (A_3)_o(\phi_o \cdot \phi_r))D_r & = \mathbf{A}_{\mathbf{3}}^{\mathbf{T}}\mathbf{MD} \\
(p_h, D_h) = \sum_{s,r}(\int p_s(\phi_s \cdot \phi_r))D_r & = \mathbf{p}^{\mathbf{T}}\mathbf{MD} \\
(g_h, D_h) = \sum_{q,r}(\int g_q(\phi_q \cdot \phi_r))D_r & = \mathbf{g}^{\mathbf{T}}\mathbf{MD} \\
(R_h V_h, D_h) = \sum_{i,j,r}(\int R_i V_j(\phi_i \cdot \phi_j \cdot \phi_r))D_r & \approx (\mathbf{R} \odot \mathbf{V})^{\mathbf{T}}\mathbf{MD} \\
((R^2)_h, D_h) = \sum_{i,r}(\int R_i^2(\phi_i \cdot \phi_i \cdot \phi_r))D_r & \approx (\mathbf{R} \odot \mathbf{R})^{\mathbf{T}}\mathbf{MD} \\
((V^2)_h, D_h) = \sum_{j,r}(\int V_j^2(\phi_j \cdot \phi_j \cdot \phi_r))D_r & \approx (\mathbf{V} \odot \mathbf{V})^{\mathbf{T}}\mathbf{MD} \\
(\nabla\Psi_h, \nabla D_h) = \sum_{k,r}(\int \Psi_k(\nabla\phi_k \cdot \nabla\phi_r))D_r & = \mathbf{\Psi}^{\mathbf{T}}\mathbf{AD} \\
(1, D_h) = \sum_{t,r}(\int 1(\phi_t \cdot \phi_r))D_r & = \mathbf{1}^{\mathbf{T}}\mathbf{MD}
\end{cases} \tag{15}$$

Where $\mathbf{A}$ is the stiffness matrix and $\mathbf{M}$ is the mass matrix and

$$\begin{aligned}
(\dot{R}_h, D_h) &= \frac{1}{\tau}\Big(- \kappa_V(R_h, D_h) + 2(R_h V_h, D_h) + \frac{\gamma}{\pi\tau}(1_h, D_h)\Big) \\
&\approx \frac{1}{\tau}\Big(- \kappa_V(\mathbf{R}^{\mathbf{T}}\mathbf{MD}) + 2(\mathbf{R} \odot \mathbf{V})^{\mathbf{T}}\mathbf{MD} + \frac{\gamma}{\pi\tau}(\mathbf{1}^{\mathbf{T}}\mathbf{MD})\Big) \\
&= \frac{1}{\tau}\Big(- \kappa_V(\mathbf{R}^{\mathbf{T}}\mathbf{M}) + 2((\mathbf{R} \odot \mathbf{V}^{\mathbf{T}})\mathbf{M}) + \frac{\gamma}{\pi\tau}(\mathbf{1}^{\mathbf{T}}\mathbf{M})\Big)\mathbf{D} \\
&= \mathbf{B}_{\mathbf{1}}^{\mathbf{T}}\mathbf{D}
\end{aligned}$$

where $\mathbf{B}_{\mathbf{1}}^{\mathbf{T}} = \frac{1}{\tau}\Big(- \kappa_V(\mathbf{R}^{\mathbf{T}}\mathbf{M}) + 2(\mathbf{R} \odot \mathbf{V})^T M + \frac{\gamma}{\pi\tau}(\mathbf{1}^{\mathbf{T}}\mathbf{M})\Big)$. Similarly,

$$\begin{aligned}
(\dot{V}_h, D_h) &= \frac{1}{\tau}\Big(\kappa_S(g_h, D_h) - (\pi^2\tau^2(R_h^2, D_h)) + (V_h^2, D_h) + \eta_0(1_h, D_h)\Big) \\
&\approx \frac{1}{\tau}\Big(k_S(\mathbf{g}^{\mathbf{T}}\mathbf{MD}) - \pi^2\tau^2(\mathbf{R} \odot \mathbf{R})^{\mathbf{T}}\mathbf{MD} + (\mathbf{V} \odot \mathbf{V})^{\mathbf{T}}\mathbf{MD} + \eta_0(\mathbf{1}^{\mathbf{T}}\mathbf{MD})\Big) \\
&= \frac{1}{\tau}\Big(k_S(\mathbf{g}^{\mathbf{T}}\mathbf{M}) - \pi^2\tau^2(\mathbf{R} \odot \mathbf{R})^{\mathbf{T}}\mathbf{M} + (\mathbf{V} \odot \mathbf{V})^{\mathbf{T}}\mathbf{M} + \eta_0(\mathbf{1}^{\mathbf{T}}\mathbf{M})\Big)\mathbf{D} \\
&= \mathbf{B}_{\mathbf{2}}^{\mathbf{T}}\mathbf{D}
\end{aligned}$$

So,

$$\begin{aligned}
(\ddot{R}_h, D_h) &= \frac{1}{\tau}\Big(-\kappa_V(\dot{R}_h, D_h) + 2(V_h\dot{R}_h, D_h) + 2(R_h\dot{V}_h, D_h)\Big) \\
&\approx \frac{1}{\tau}\Big(-\kappa_V \mathbf{B_1^T M} + 2(\mathbf{V}\odot\mathbf{B_1})^{\mathbf{T}}\mathbf{M} + 2(\mathbf{R}\odot\mathbf{B_2})^{\mathbf{T}}\mathbf{M}\Big)\mathbf{D} \\
&= \mathbf{B_3^T D}
\end{aligned}$$

and

$$\begin{aligned}
(A_4, D_h) &= -\frac{3}{2}(\nabla(A_2)_h, \nabla D_h) - \frac{3}{2}(\nabla R_h, \nabla D_h) - \frac{1}{\nu^2}(\ddot{R}_h, D_h) - \frac{1}{\nu}(\dot{R}_h, D_h) \\
&\approx \Big(-\frac{3}{2}(\mathbf{A_2^T AD}) - \frac{3}{2}(\mathbf{R^T AD}) - \frac{1}{\nu^2}(\mathbf{B_3^T D}) - \frac{1}{\nu}(\mathbf{B_1^T D})\Big)
\end{aligned}$$

Substituting (15) and (14) into (13), we get the system of ODEs

$$\begin{cases}
\mathbf{\dot{R}^T MD} &= \frac{1}{\tau}\Big(-\kappa_V \mathbf{R^T MD} + 2((\mathbf{R}\odot\mathbf{V})^{\mathbf{T}}\mathbf{MD}) + \frac{\gamma}{\tau\pi}(\mathbf{1^T M})\mathbf{D}\Big) \\
\mathbf{\dot{V}^T MD} &= \frac{1}{\tau}\Big(\kappa_s(\mathbf{g^T M})\mathbf{D} - (\pi^2\tau^2(\mathbf{R}\odot\mathbf{R})^{\mathbf{T}}\mathbf{MD}) + ((\mathbf{V}\odot\mathbf{V})^{\mathbf{T}}\mathbf{MD}) + \eta_0(\mathbf{1^T M})\mathbf{D}\Big) \\
\mathbf{\dot{\Psi}^T MD} &= \nu(-\mathbf{\Psi^T MD} + \mathbf{A_1^T MD}) \\
\mathbf{\dot{A}_1^T MD} &= \nu\Big(-\mathbf{A_1^T MD} + \mathbf{A_2^T MD} + \frac{3}{2}(\mathbf{\Psi^T AD})\Big) \\
\mathbf{\dot{A}_2^T MD} &= \nu(\mathbf{A_3^T MD} - \mathbf{A_2^T MD}) \\
\mathbf{\dot{A}_3^T MD} &= \nu\Big(\frac{3}{2}(\mathbf{A_2^T AD}) - \frac{3}{2}(\mathbf{R^T AD}) + \frac{1}{\nu^2}(\mathbf{B_3^T D}) - \frac{1}{\nu}(\mathbf{B_1^T D}) - \mathbf{A_3^T MD}\Big) \\
\mathbf{\dot{p}^T MD} &= \alpha(\mathbf{\Psi^T MD} - \mathbf{p^T MD}) \\
\mathbf{\dot{g}^T MD} &= \alpha(\mathbf{p^T MD} - \mathbf{g^T MD})
\end{cases}$$

which must hold for every $\mathbf{D}$.

This ODE system is of the form

$$\mathbf{N\dot{y}} = \mathbf{f}(\mathbf{y})$$

where

$$\mathbf{N} = \begin{bmatrix} \mathbf{M} & & \\ & \ddots & \\ & & \mathbf{M} \end{bmatrix}$$

and

$$\mathbf{y} = \begin{bmatrix} \mathbf{R} \\ \mathbf{V} \\ \mathbf{\Psi} \\ \mathbf{A_1} \\ \mathbf{A_2} \\ \mathbf{A_3} \\ \mathbf{p} \\ \mathbf{g} \end{bmatrix}$$

and

$$\mathbf{f}(\mathbf{y}) = \begin{bmatrix} \frac{1}{\tau}\big(-\kappa_V \mathbf{MR} + 2(\mathbf{M}(\mathbf{R}\odot\mathbf{V})) + \frac{\gamma}{\tau\pi}(\mathbf{M1})\big) \\ \frac{1}{\tau}\big(\kappa_s(\mathbf{Mg}) - (\pi^2\tau^2\mathbf{M}(\mathbf{R}\odot\mathbf{R})) + (\mathbf{M}(\mathbf{V}\odot\mathbf{V})) + \eta_0(\mathbf{M1})\big) \\ \nu(-\mathbf{M\Psi} + \mathbf{MA_1}) \\ \nu\Big(-\mathbf{MA_1} + \mathbf{MA_2} + \frac{3}{2}(\mathbf{A\Psi})\Big) \\ \nu(\mathbf{MA_3} - \mathbf{MA_2}) \\ \nu\Big(\frac{3}{2}\mathbf{AA_2} - \frac{1}{v}\mathbf{B_1} + \frac{1}{v^2}\mathbf{B_3} - \frac{3}{2}\mathbf{AR} - \mathbf{MA_3}\Big) \\ \alpha(\mathbf{M\Psi} - \mathbf{Mp}) \\ \alpha(\mathbf{Mp} - \mathbf{Mg}) \end{bmatrix}$$

If we use a Eulerian discretization of time as in the case of the time-dependent heat equation, we can find solutions of the form

$$\mathbf{y_{k+1}} = \mathbf{y_k} + \Delta t \mathbf{N}^{-1}\mathbf{f}(\mathbf{y_k})$$

We solved this system with MATLAB's ODE45 function, which implements the Runge-Kutta method with a variable time-step.

8.5 Synchrony

Synchrony, when nodes oscillate in unison, is one of the most studied behaviour of coupled oscillators and can yield exotic patterns. In neuroscience, nodes can be single neurons or densities of neurons. Synchrony can either be global or local and is thought to be integral to certain brain functions and neurological disorders such as epilepsy. The phenomenon has also been associated with memory, cognition, control of gain and motion, and breathing [1]. Local synchrony, also called *within-population synchrony* [4] which is what we are most interested in, can be seen as oscillations within bumps, even as the surrounding activity stays constant with a low firing rate. This type of synchrony can explain a phenomenon of temporal variation in EEG/MEG frequency bands called *beta-decrease* and *beta-rebound*. Multiple electrodes, which count as the nodes of the neural network, can map spatiotemporal dynamics of the brain. [1]. While an EEG scalp electrode can only detect an electric field if all individual cells act coherently – meaning they are in near global synchrony – in practise, there are fluctuations in the power of frequency bands, seeming to imply that neurons' level of synchronization is changing locally. An oscillation within a bump usually occurs when there is a collision between a Turing bifurcation and a Hopf bifurcation. Data on local synchrony can be merged with connectome data to give a new perspective on whole brain dynamics [4]. We can determine the *within-population* dynamics by a conformal map derived by Montbrio.

$$\begin{cases} Z = \frac{1-W^*}{1+W^*} \\ W = \pi\tau R + iV \end{cases}$$

We can follow a point in mesh through a simulation and plot the R,V and corresponding $|Z|$-values which fall between 0 and 1, 0 being a low synchrony level and 1 being a high level [4]. Z is called a Kuramoto order parameter.

9 Bifurcation Analysis

The mathematician Alan Turing developed the concept of Turing instability analysis in 1952 to study the emergence of patterns in spatio-temporal systems [26]. Turing instabilities occur when a ``homogeneous steady state becomes unstable to spatial perturbations with certain wave-number $k$ as a system parameter is varied" and a ``globally periodic stable solution emerges with an intrinsic wavelength [27]." Physiologically, we can interpret the Turing instability as a critical point beyond which macroscopic dynamics subsume the microscopic dynamics. We can derive the conditions for growth of in-homogeneous solutions through a linear stability analysis. Thereby, we can find for which parameter values solutions become unstable to ``globally periodic spatial perturbations," $e^{i\boldsymbol{k}\cdot\boldsymbol{r}}$, and also find the intrinsic wavelength $\frac{2\pi}{k}$ of resulting patterns. From this, we obtain an algebraic equation known as the dispersion relation, $\lambda = \lambda(k)$, which relates the wavelengths of perturbations to their growth rate. A Turing instability occurs when the real part for the first time becomes positive for some critical wave number k_c, that is, ``when the dispersion curve lies in the negative half-plane with only its maximum touching the imaginary axis."

9.1 Planar Surface

We can do a stability analysis of the system in the manner of Byrne [4] by first finding the homogeneous steady state and perturbing from this value. More specifically, the steady state will be given by solving the equation:

$$\begin{aligned} -\kappa_V R_0 + 2R_0V_0 + \frac{\gamma}{\pi\tau} &= 0 \\ \eta_0 + V_0^2 - \pi^2\tau^2R_0^2 &= 0 \end{aligned}$$

for steady state

$$(0, 0, R_0, V_0)$$

and then applying perturbations of the form

$$(U(\boldsymbol{r},t), \Psi(\boldsymbol{r},t), V(\boldsymbol{r},t)) = (0,0,R_0,V_0) + \epsilon(\overline{U}, \overline{\Psi}, \overline{R}, \overline{V})e^{\lambda t}e^{i\boldsymbol{k}\cdot\boldsymbol{r}} \tag{16}$$

where $k = |\boldsymbol{k}|$ is an arbitrary scalar and λ is also a scalar.

If this equation is substituted into the brain wave equation, then, we get

$$\left[\left(1+\frac{\lambda}{\nu}\right)^2 + \frac{3}{2}k^2\right]^2 \overline{\Psi} = -\left[\frac{\lambda}{\nu}\left(1+\frac{\lambda}{\nu}\right) + \frac{3}{2}k^2\right]\overline{R}$$

Linearizing the dynamics of (R, V), we get

$$\begin{aligned} A(\lambda)\begin{bmatrix}\overline{R}\\ \overline{V}\end{bmatrix} &= \begin{bmatrix}0\\ \kappa_s\overline{U}\end{bmatrix} \\ A(\lambda) &= \tau\lambda I_2 - J \end{aligned}$$

$$J = \begin{bmatrix} -\kappa_V + 2V_0 & 2R_0 \\ -2\pi^2\tau^2 R_0 & 2V_0 \end{bmatrix}$$

Now, use Cramer's Rule to obtain expression for

$$\overline{R} = \frac{1}{|A(\lambda)|}\begin{vmatrix} 0 & -2R_0 \\ \kappa_s\overline{U} & \tau\lambda - 2V_0 \end{vmatrix} = \frac{2\kappa_s R_0}{|A(\lambda)|(1+\frac{\lambda}{\alpha})^2}\overline{\Psi}$$

We used the fact that

$$\left(1+\frac{\lambda}{\alpha}\right)^2 \overline{U} = \overline{\Psi} \tag{17}$$

Substitute back into original equation and ask for a nontrivial solution for $\overline{\Psi}$. This means that

$$\mathcal{E}(\lambda, k) = 0$$

where

$$\mathcal{E}(\lambda,k) = |A(\lambda)|\left(1+\frac{\lambda}{\alpha}\right)^2\left[\left(1+\frac{\lambda}{\nu}\right)^2 + \frac{3}{2}k^2\right]^2 + 2\kappa_s R_0\left[\frac{\lambda}{\nu}\left(1+\frac{\lambda}{\nu}\right) + \frac{3}{2}k^2\right]$$

To do bifurcation analysis on a manifold, we would perform a similar analysis, replacing the quantity $e^{i\mathbf{k}\cdot\mathbf{r}}$ by an eigenfunction of the surface Laplacian in our ansatz (16). Many of the same tools used for analysis of pattern formation of reaction diffusion equations on surfaces, for which there is a large body of literature already extant, can be applied to studying pattern formation of neural fields as well. That is our aim.

The roots dictate the stability of the steady state. When branch of solutions $\lambda(k)$ to $\mathcal{E}(\lambda, k) = 0$ touches imaginary axis, the system undergoes a bifurcation. There are three main bifurcations to look at:

1. Hopf: $\operatorname{Re}\lambda(0) = 0$
2. Turing-Hopf: $\operatorname{Re}\lambda(k_c) = \operatorname{Re}\lambda'(k_c) = 0$ for some $k_c \neq 0$
3. Intersection of Hopf and Turing-Hopf

As we adjust κ_v, the gap-junction coupling constant, the steady state changes. At some point, the steady state changes from a stable spiral to an unstable spiral with a limit cycle, meaning the system approaches an oscillatory state of constant amplitude and frequency. We used Wolfram Mathematica and suggested parameters from [4] to find bifurcation points.

9.2 Sphere

We are interested in investigating the effect of surface geometry on pattern formation on the human cortex. To study the dynamics, we applied techniques used in the study of nonlinear reaction diffusion systems on arbitrary surfaces [9]. These techniques involve conducting numerical analysis of the Laplace-Beltrami operator on surface meshes. Our final goal is to understand the possible patterns in the system and which solutions are stable and unstable.

Because the Laplace-Beltrami operator acting on an arbitrary surface is Hermitian, we can compute the spectral basis for an emerging pattern.

We can substitute a potential pattern into a linearized system to obtain an explicit form of the pattern. Thereby, we can define a bifurcation point in terms of spectral eigenvalues and system parameters.

We first perform bifurcation analysis near homogeneity through spectral analysis of the Laplace-Beltrami operator. From this operator, we can obtain eigenmodes and eigenvalues for the operator.

To compute the *Laplacian basis eigenfunctions*, we solve the eigenvalue problem

$$\nabla^2 \phi_k = -\lambda_k \phi_k,$$

In numerical terms, we discretize the eigenvalue problem as

$$\boldsymbol{A}\boldsymbol{b}_k = \lambda_k \boldsymbol{M}\boldsymbol{b}_k$$

where $\boldsymbol{b}_k$ is a discrete vector representation of ϕ_k, $\boldsymbol{A}$ is the stiffness matrix and $\boldsymbol{M}$ is the mass matrix.

Martin [19] performed a numerical bifurcation analysis on the torus using a *pseudo-arclength continuation scheme*. One of Martin's observations was that, on the torus, as the major curvature radius decreases to a certain point, a stable bump solution evolves into a stable ring solution.

10 Numerical Experiments

We ran MATLAB simulations on two surfaces: a planar region with periodic boundary conditions and a sphere. We plot the mean firing rate R, mean membrane potential V, and the within-population synchrony measure $|Z|$ for a particular point (in red) in the mesh.

For the Hopf bifurcations, we see that oscillations gradually decay toward a steady state before the bifurcation point and approach a constant amplitude and frequency after the bifurcation point (i.e. a limit cycle) – both for the neural mass model and the 2D model on a sphere.

For the Turing-Hopf bifurcation, we observe how several patterns eventually converge to one specific pattern, as seen in a standing wave eventually forming. For the planar surface, we ran a simulation at the intersect of Hopf and Turing-Hopf bifurcation and observe high within-population synchrony as a pattern emerges within a bump of activity.

10.1 Hopf

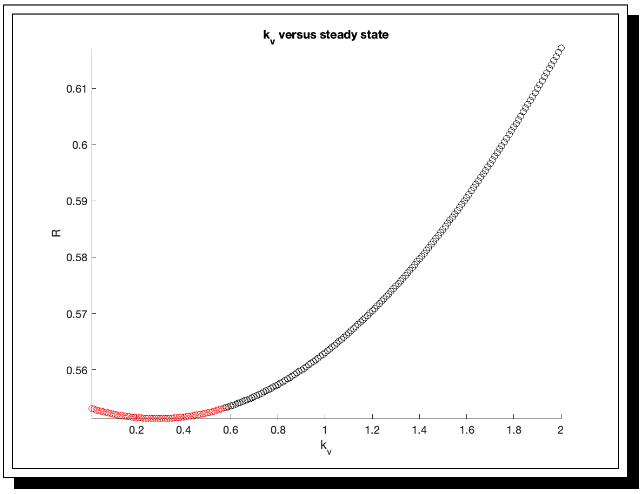
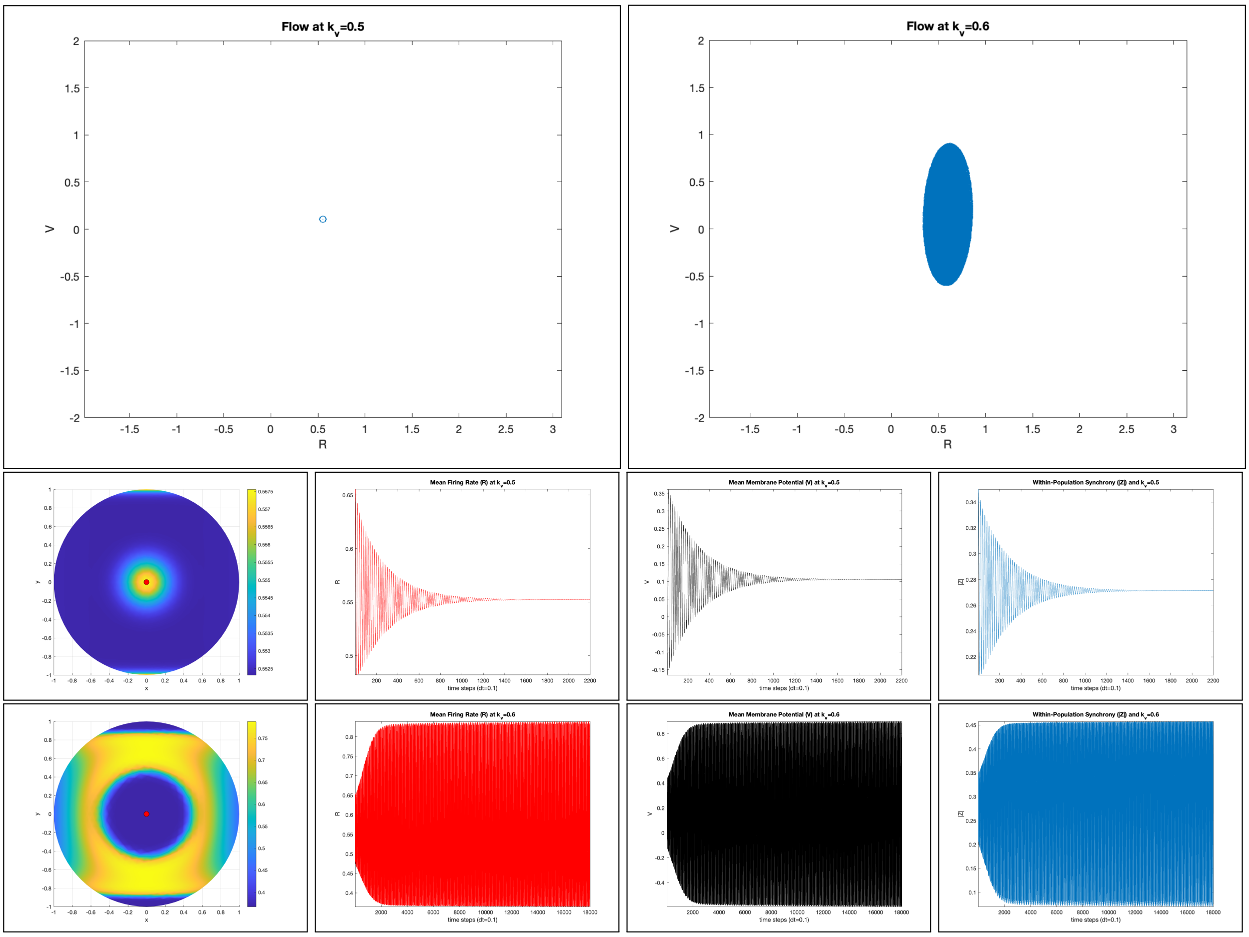


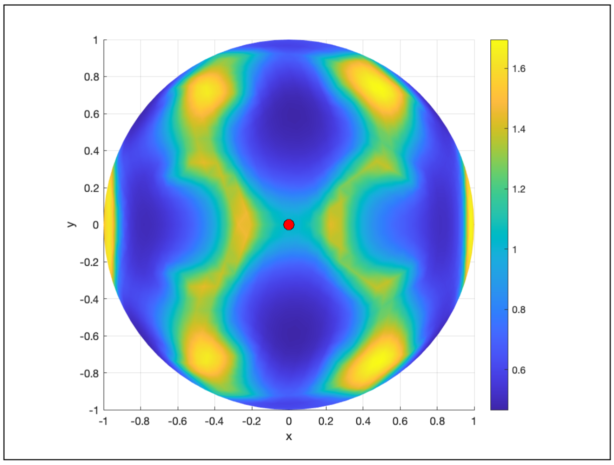
Figure 1: κ_v versus the R coordinate of the steady state. The red indicates the stable regime and black the unstable regime.

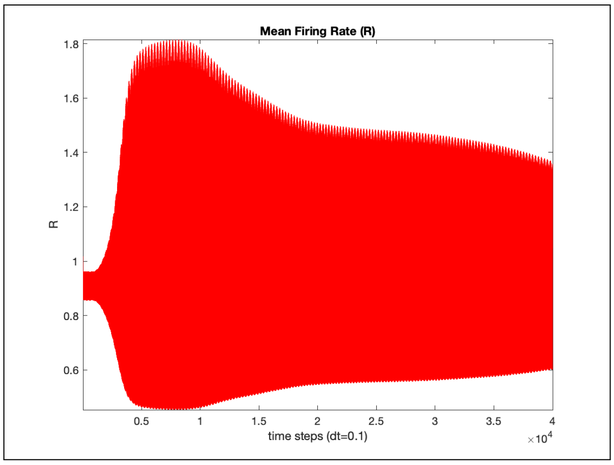
The figures below show the flow of the neural mass model (top row) and R,V, and $|Z|$ at a point on a sphere: middle row (stable), bottom row (unstable). Parameters are $v = 1$, $\eta_0 = 3$, $\gamma = 1$, $\alpha = 1$, $k_c = 1$,$\kappa_v = 0.5/0.6$, $\kappa_s = 1$, $\tau = 1$.

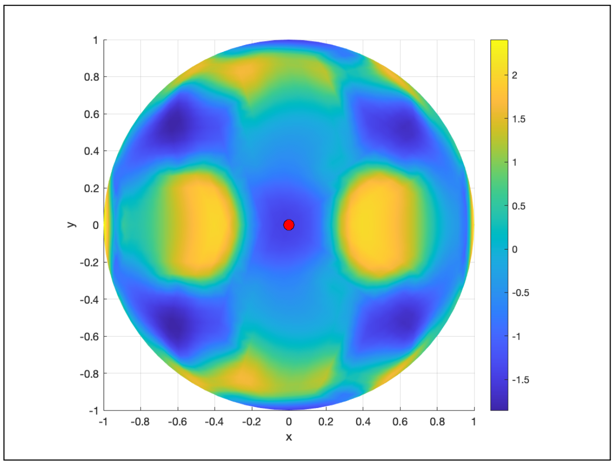


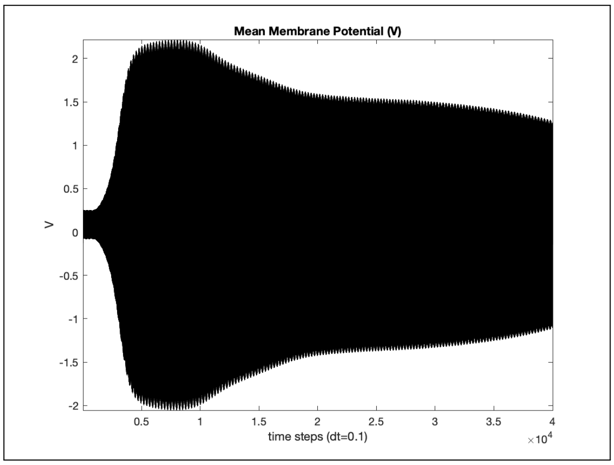
10.2 Turing-Hopf

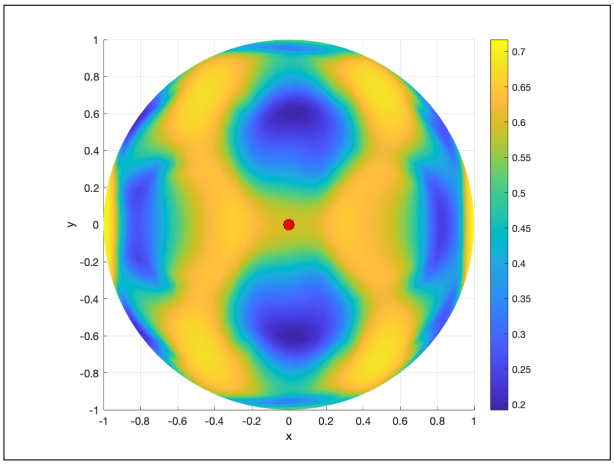
10.2.1 Sphere

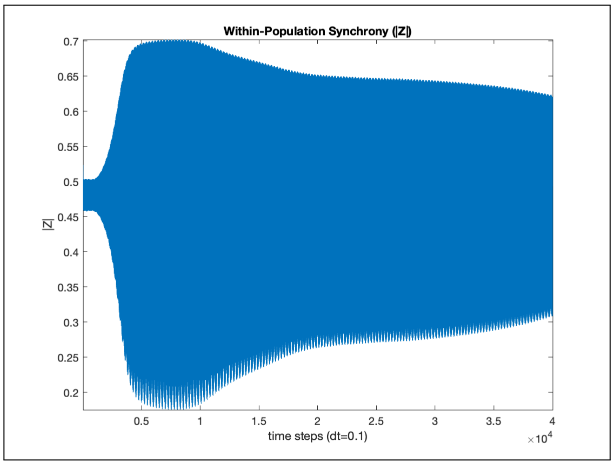










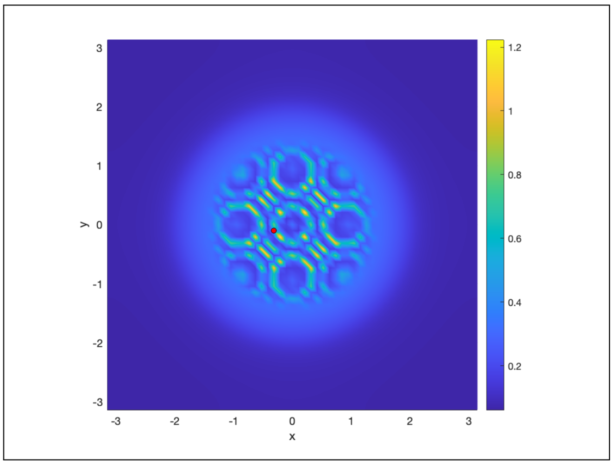


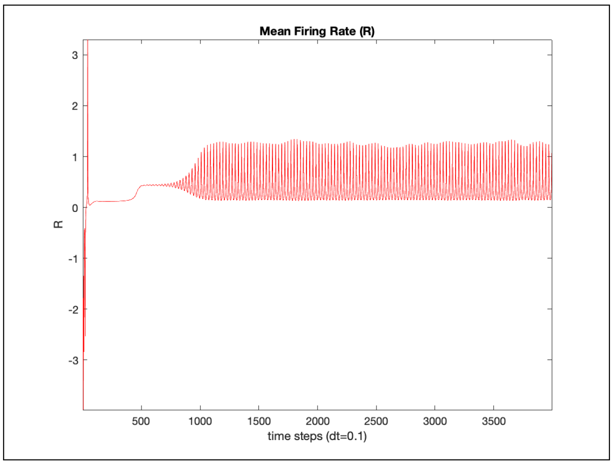
The figures in left column above show, R, V, and $|Z|$ near a Turing-Hopf bifurcation plotted on a sphere.

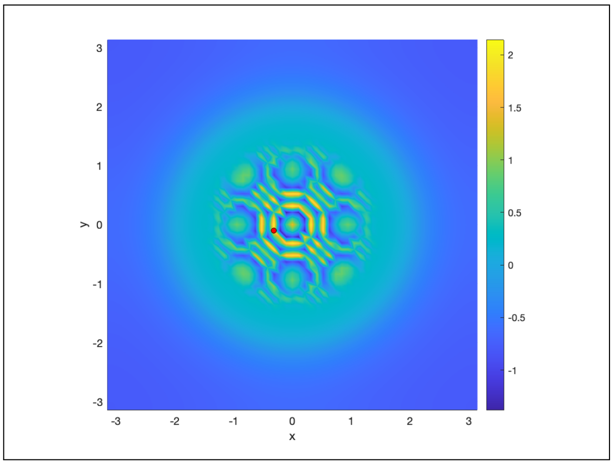
The figures in the right column above show R, V, and $|Z|$ plotted at a pole (red dot) of the sphere.

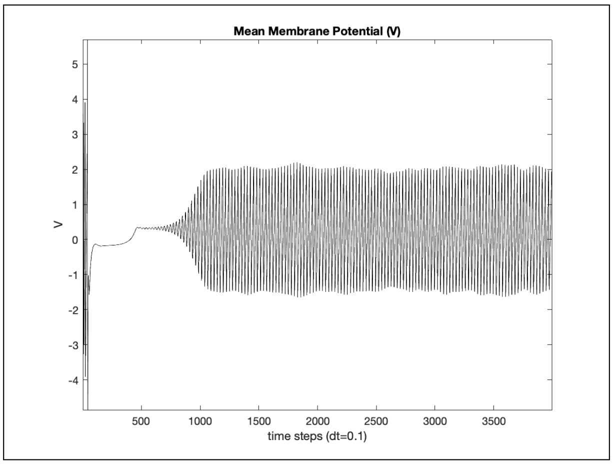
Here, the parameters are $v = 1.04683$, $\eta_0 = 8.12$, $\gamma = 0.5$, $\alpha = 5$, $k_c = 4.49022271163$, $\kappa_v = 0.346252$, $\kappa_S = 12$, $\tau = 1$.

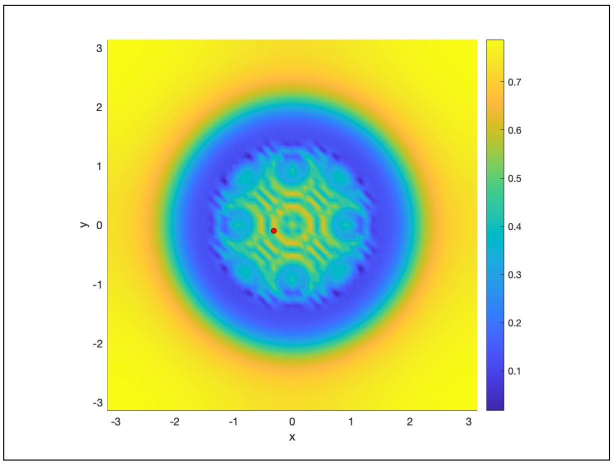
10.2.2 Plane

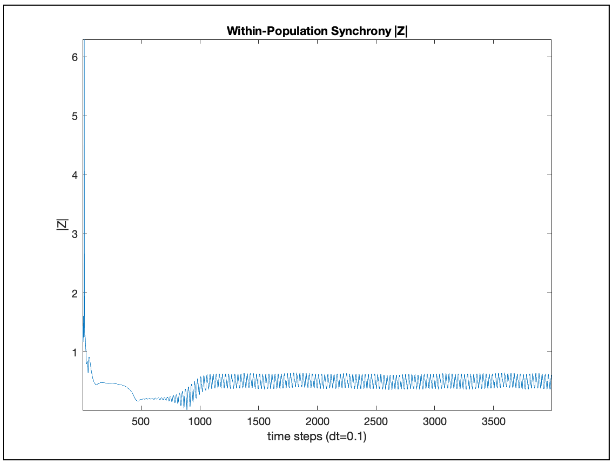












The figures in the left column above show R, V, and $|Z|$ at the intersection of a Turing and Hopf bifurcation plotted on a periodic planar domain. We see that there is a high level of within-population synchrony (close to 1).

The figures in the right column above show R, V, and $|Z|$ plotted at a point (red dot) near the center of the domain.

Here, the parameters are $v = 10.0$, $\eta_0 = 0.1$,$\gamma = 0.5$, $\alpha = 5.0$, $k_c = 1$, $\kappa_v = 1.0$, $\kappa_s = -25.0$, $\tau = 1$.

11 Future Research

It could be beneficial to study in greater detail the dynamics of the bump patterns (localized states) we obtain from the PDE model on the sphere, for which there is a growing literature. For example, Rozada, Ruuth, and Ward investigate the existence and stability of multi-spot formation on the unit sphere for the Brusselator reaction-diffusion [24]. Additionally, we would like to gauge the discretization error in the model by comparing with an analytical solution on the sphere, if possible.

12 Conclusion

On the sphere and plane, we noticed similar complex patterns forming: bumps, bumps splitting into multiple bumps, circular waves, bands/stripes. Both are also capable of fairly high levels of within-population synchrony. We were however not able to duplicate, as seen in the plane, patterns within patterns, which signal particularly high levels of within-population synchrony. By finding an intersection point between a Hopf and Turing-Hopf bifurcation however, this might be possible.

References

[1] Bick, C., Goodfellow, M., Laing, C. R., amp; Martens, E. A. (2020). Understanding the dynamics of biological and neural oscillator networks through exact mean-field reductions: a review. The Journal of Mathematical Neuroscience, 10(1). https://doi.org/10.1186/s13408-020-00086-9

[2] Bressloff, P. C. (2014). Waves in neural media: from single neurons to neural fields. Springer.

[3] Brain for Blender. Brainder. (2018, March 29). https://brainder.org/research/brain-for-blender/.

[4] Byrne, Á., Ross, J., Nicks, R., &amp Coombes, S. (2020). Mean-field models for EEG/MEG: from oscillations to waves. https://doi.org/10.1101/2020.08.12.246256

[5] Coombes, S. (n.d.). Neural fields. Scholarpedia. http://www.scholarpedia.org/article/Neuralfields.

[6] Coombes, S., amp; Byrne, Á. (2018). Next Generation Neural Mass Models. Nonlinear Dynamics in Computational Neuroscience, 1–16.

[7] Dayan, P., amp; Abbott, L. (2014). Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. MIT Press.

[8] Dabrowski, K. M., Castano, D. J., amp; Tartar, J. L. (2013). Basic Neuron Model Electrical Equivalent Circuit: An Undergraduate Laboratory Exercise. Journal of Undergraduate Neuroscience Education, 12(1).

[9] Dhillon, D. S., Milinkovitch, M. C., amp; Zwicker, M. (2017). Bifurcation Analysis of Reaction Diffusion Systems on Arbitrary Surfaces. Bulletin of Mathematical Biology, 79(4), 788–827. https://doi.org/10.1007/s11538-017-0255-8

[10] Dziuk, G., amp; Elliott, C. M. (2013). Finite element methods for surface PDEs. Acta Numerica, 22, 289–396. https://doi.org/10.1017/s0962492913000056

[11] Fife, P., Cahn, J., amp; Elliott, C. (2001). A free-boundary model for diffusion-induced grain boundary motion. Interfaces and Free Boundaries, 291–336. https://doi.org/10.4171/ifb/42

[12] Gerstner, W. (n.d.). Neuronal Dynamics. Neuronal Dynamics - a neuroscience textbook by Wulfram Gerstner, Werner M. Kistler, Richard Naud and Liam Paninski. https://neuronaldynamics.epfl.ch/.

[13] Golomb, D. (n.d.). Neuronal synchrony measures. Scholarpedia.

[14] Izhikevich, E. M. (2014). Dynamical Systems in Neuroscience: the Geometry of Excitability and Bursting. MIT Press.

[15] Jirsa, V. K., Jantzen, K. J., Fuchs, A., amp; Kelso, J. A. S. (2001). Neural Field Dynamics on the Folded Three-Dimensional Cortical Sheet and Its Forward EEG and MEG. Lecture Notes in Computer Science, 286–299.

[16] Johnson, C. (2009). Numerical solution of partial differential equations by the finite element method. Dover Publications.

[17] Kimmel, R. (1997). Intrinsic scale space for images on surfaces: The geodesic curvature flow. Scale-Space Theory in Computer Vision, 212–223.

[18] Laing, C. R., amp; Troy, W. C. (2003). PDE Methods for Nonlocal Models. SIAM Journal on Applied Dynamical Systems, 2(3), 487–516. https://doi.org/10.1137/030600040

[19] Martin, R. (n.d.). Collocation techniques for solving neural field models on complex cortical geometries (dissertation).

[20] Mémoli, F., Sapiro, G., amp; Thompson, P. (2004). Implicit brain imaging. NeuroImage, 23. https://doi.org/10.1016/j.neuroimage.2004.07.072

[21] Montbrió, E., Pazó, D., amp; Roxin, A. (2015). Macroscopic Description for Networks of Spiking Neurons. Physical Review X, 5(2). https://doi.org/10.1103/physrevx.5.021028

[22] Ott, E., amp; Antonsen, T. M. (2008). Low dimensional behavior of large systems of globally coupled oscillators. Chaos: An Interdisciplinary Journal of Nonlinear Science, 18(3), 037113. https://doi.org/10.1063/1.2930766

[23] Rognes, M. E., Ham, D. A., Cotter, C. J., amp; McRae, A. T. (2013). Automating the solution of PDEs on the sphere and other manifolds in FEniCS 1.2. Geoscientific Model Development, 6(6), 2099–2119. https://doi.org/10.5194/gmd-6-2099-2013

[24] Rozada, I., Ruuth, S. J., amp; Ward, M. J. (2014). The Stability of Localized Spot Patterns for the Brusselator on the Sphere. SIAM Journal on Applied Dynamical Systems, 13(1), 564–627. https://doi.org/10.1137/130934696

[25] Trinh, P. H., amp; Ward, M. J. (2016). The dynamics of localized spot patterns for reaction-diffusion systems on the sphere. Nonlinearity, 29(3), 766–806. https://doi.org/10.1088/0951-7715/29/3/766

[26] Turing, A. M. (1990). The chemical basis of morphogenesis. *Bulletin of mathematical biology*, 52(1), 153-197.

[27] Venkov, N. A. (2008). Dynamics of Neural Field Models (thesis).

[28] Visser, S., Nicks, R., Faugeras, O., amp; Coombes, S. (2017). Standing and travelling waves in a spherical brain model: The Nunez model revisited. Physica D: Nonlinear Phenomena, 349, 27–45. https://doi.org/10.1016/j.physd.2017.02.017

[29] What Is FEM amp; FEA Explained: Finite Element Method. SimScale. (2020, December 20). https://www.simscale.com/blog/2016/10/what-is-finite-element-method/.