# Tutorial on Finite-Element schemes for neural fields

Daniele Avitabile *

**Abstract.** In this tutorial we study the implementation of a basic Finite Element method for a neural field equation with one or more components. We will also compute and visualise patterns of cortical activity supported by these models, and observed in experiments. Before starting this tutorial download the latest version of the accompanying dataset. Solutions to the exercises can be found here: Q1, Q2, Q3, Q4 (Disk), and Q4 (Hexagon). Code can be downloaded from our repository.

**1. Introduction.** This tutorial deals with the implementation of a basic Finite Element Scheme for the standard Neural Field Equation

$$\begin{aligned} \partial_t u(r,t) &= -u(r,t) + \int_D w(r,r') f(u(r',t))\, dr', && (r,t) \in D \times [0,T] \\ u(r,0) &= \varphi(r) && r \in D. \end{aligned}$$

posed on a generic cortical surface $D \subset \mathbb{R}^3$. The exposition is non-technical, and downloadable exemplary files for relevant matrices are given to simplify the tutorial at a first pass. At the end of the tutorial you will have visualised patterns observed in experimental recordings of neuronal activity. A theory for numerical schemes for neural fields is given in [1] (see also references therein).

**1.1. Domain and triangulation.** We will assume that the cortex $D$ to be a piecewise-smooth compact surface $\mathbb{R}^3$. More specifically, $D$ is triangulated using $m$ triangles: we consider a triangulation of $\mathcal{T} = \{\tau_\alpha\}_{\alpha=1}^m$ of $D$ such that

$$\bigcup_{\alpha=1}^{m} \tau_\alpha = D, \qquad \tau_\alpha^\circ \cap \tau_\beta^\circ = \varnothing, \quad \alpha \neq \beta$$

where $\tau_\alpha^\circ$ denotes the interior of triangle $\tau_\alpha$. We will denote by $\mu_\alpha$ the area of the triangle $\tau_\alpha$. We will use $r = (x,y,z)$ for points in $\mathbb{R}^3$, and we number the nodes of the triangulation from 1 to $n$, with coordinates $\{r_i\}_{i=1}^n$.

It is convenient to introduce a family of mappings $\{B_\alpha\}_\alpha$ which send the reference triangle $\sigma$ (the standard simplex in $\mathbb{R}^2$) to

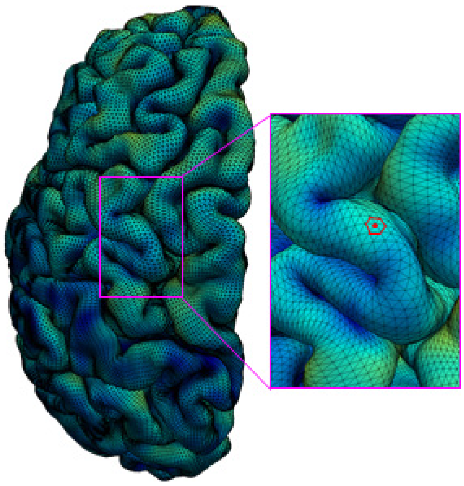

**Figure 1.1.** *Example of a cortical mesh (website)*

* Vrije Universiteit Amsterdam, Department of Mathematics, Faculteit der Exacte Wetenschappen, De Boelelaan 1082a, 1082 HV Amsterdam, The Netherlands. MathNeuro Team, Inria branch of the University of Montpellier, 861 rue Saint-Priest 34096 Montpellier Cedex 5 France. (d.avitabile@vu.nl, www.danieleavitabile.com).

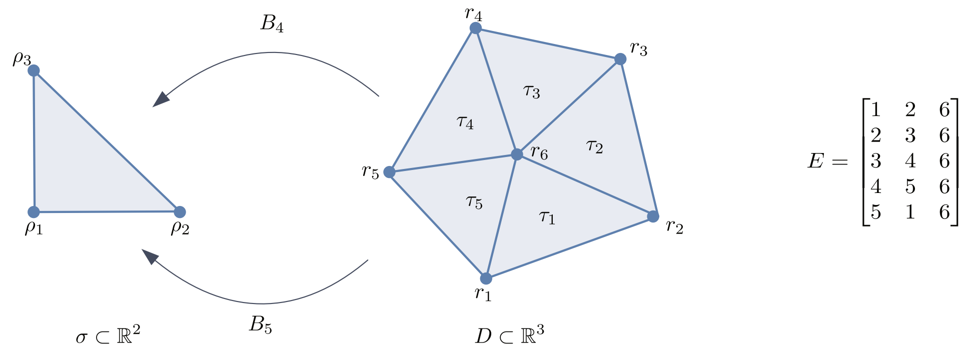


**Figure 1.2.** *A triangulation with 5 elements for a domain $D \subset \mathbb{R}^3$. Each triangle is mapped to the standard simplex $\sigma \subset \mathbb{R}^2$. To the right we show the corresponding element matrix $E$.*

the triangles $\{\tau_\alpha\}_\alpha$, that is, $B_\alpha \colon \sigma \mapsto \tau_\alpha$. The standard simplex in $\mathbb{R}^2$ is the triangle with nodes $\rho_1 = (0,0)$, $\rho_2 = (1,0)$, $\rho_3 = (0,1)$ (see schematic in Figure 1.2).

In this tutorial we use matrices to encode information about triangulation. Coordinates of the nodes are stored in a $n$-by-3 matrix $N$, so that the $i$th row of $N$ contains the coordinates $r_i$, while triangular elements are stored in an $m$-by-3 matrix $E$, so that the $\alpha$th row of $M$ indicates that triangle $\tau_\alpha$ has vertices at nodes $\{n_{\alpha,1}, n_{\alpha,2}, n_{\alpha,3}\}$,

$$N = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix}, \qquad E = \begin{bmatrix} n_{1,1} & n_{1,2} & n_{1,3} \\ n_{2,1} & n_{2,2} & n_{2,3} \\ \vdots & \vdots & \vdots \\ n_{3,1} & n_{3,2} & n_{3,3} \end{bmatrix}.$$

**1.2. Quadrature.** We address the problem of approximating the integral of a function $u \colon D \to \mathbb{R}$ over $D$. The starting point is defining a quadrature rule over the simplex $\sigma$, for a function $g \colon \sigma \to \mathbb{R}$. Rather than aiming for a general treatment, we work with a concrete quadrature choice to simplify the notation (more sophisticated quadrature rules can be obtained with minimal changes): its quadrature nodes coincide with the vertices $\{\rho_k\}_{k=1}^3$ of the simplex, and its quadrature weights are equal to $1/6$ for each node, hence

$$\int_\sigma g(\rho)\, d\rho \approx \sum_{k=1}^{3} g(\rho_k)/6,$$

Starting from this quadrature, we now write

$$\begin{aligned} \int_D u(r)\, dr' &= \sum_{\alpha=1}^{m} \int_{\tau_\alpha} u(r)\, dr = \sum_{\alpha=1}^{m} \int_{B_\alpha(\sigma)} u(r)\, dr = \sum_{\alpha=1}^{m} \mu_\alpha \int_\sigma u(B_\alpha(\rho))\, d\rho \\ &\approx \sum_{\alpha=1}^{m} \mu_\alpha/6 \sum_{k=1}^{3} u(B_\alpha(\rho_k)). \end{aligned}$$

With a further manipulation of the terms in the latest sum, we can write the quadrature formula above as a weighted sum of $u$ at the nodes of the triangulation

$$\begin{aligned}\int_D u(r)\,dr &\approx \sum_{\alpha=1}^{m} \mu_\alpha/6 \sum_{k=1}^{3} u(B_\alpha(\rho_k)) = \sum_{j=1}^{n} u(r_j) \sum_{\{\alpha\colon r_j\in\tau_\alpha\}} \mu_\alpha/6 \\ &=: \sum_{j=1}^{n} u(r_j)\delta_j\end{aligned}$$

Equipped with the quadrature rule of the previous section, we can now approximate the integral of functions in the form $r' \mapsto w(r,r')v(r')$

$$\int_D w(r,r')v(r')\,dr' \approx \sum_{j=1}^{n} w(r,r_j)v(r_j)\delta_j \tag{1.1}$$

**1.3. Finite Element Collocation Scheme for Neural Fields.** We are now ready to formulate a finite element collocation scheme for section 1. We collocate section 1 at the nodes $\{r_i\}_{i=1}^n$

$$\begin{aligned}\partial_t u(r_i,t) &= -u(r_i,t) + \int_D w(r_i,r')f(u(r',t))\,dr', \quad && i=1,\dots,n, \quad t\in[0,T] \\ u(r_i,0) &= \varphi(r_i) && i=1,\dots,n,\end{aligned}$$

and apply the quadrature rule (1.1) to arrive at the set of approximating ODEs

$$U'(t) = -U(t) + MF(U(t)), \qquad t\in[0,T], \qquad U(0)=\Phi, \tag{1.2}$$

where $U(t)\in\mathbb{R}^n$ for all $t\in[0,T]$ is a vector with components $U_i(t)\approx u(r_i,t)$, the vector $\Phi\in\mathbb{R}^n$ has components $\Phi_i=\varphi(r_i)$, and the nonlinear function $F\colon\mathbb{R}^n\to\mathbb{R}^n$, has components $(F(U))_i = f(u_i)$. Finally the matrix $M\in\mathbb{R}^{n\times n}$ has components $M_{ij}=w(r_i,r_j)\delta_j$.

## 2. Tutorial questions.

**Question 1.** This question is to make you familiar with triangulation, meshes and plotting with their data structure. We aim to plot the function

$$u(\rho,\theta) = ae^{-b\rho}(b\cos\rho + \sin\rho)\cos(\omega\theta), \qquad (r,\theta)\in[0,R]\times[-\pi,\pi)$$

with parameters $a=10$, $b=0.05$, $\omega=4$, $R=30$.

In principle this task involves creating a triangulation of the disk of radius $R$, which is a nontrivial task. I suggest to skip this step at a first pass, and load a precomputed dataset, providing nodes and elements in matrices $N$ and $E$, respectively, in the format specified in subsection 1.1.

1. Download the accompanying dataset. The relevant data is in the folder `Spots-Disk` folder in a binary file `mesh.mat`, which contains `nodes` and `elements` fields. This binary MATLAB format can be downloaded also in Python (see instructions here). Alternatively, the matrices $N$ and $E$ can be imported from the text files `nodes.dat` and `elements.dat`, respectively.

2. Create a vector `u` whose $i$th components evaluates $u$ at the $i$th node, and plot the function. In Matlab, this can be done with a command along these lines

```
trisurf(elements,x,y,z,u);
```

Plotting on triangulated meshes may not be straightforward, and you need to know how to do it for the remainder of the tutorial, so I have added a `minimal-plotter.m` and `minimal-plotter.py` to get you going.

**Question 2.** Next, we create the Finite Element (FE) matrix $M$ appearing in (1.2), for the following kernel

$$w(r,r') = W_{A,\varepsilon}(\|r-r'\|_2), \quad W_{A,\varepsilon}(x) = \begin{cases} A(x) & \text{if } |A(x)| \geq \varepsilon, \\ 0 & \text{otherwise,} \end{cases}, \quad A(x) = e^{-bx}(b\sin x + \cos x)$$

with parameters $b = 0.4$, $\varepsilon = 10^{-3}$. The synaptic kernel is thus distance dependent and truncated, meaning that only synaptic connections of sufficiently large strength are retained.

One way to form $M$ (there are faster and cheaper ones) is to first create a matrix $W$ with components $W_{ij} = w(r_i, r_j)$ and then assemble the FE matrix $M$ with components $M_{ij} = W_{ij}\delta_j$. The matrices $M$ and $W$ are sparse, and can be stored cheaply in sparse format, that is: one can store them cheaply specifying the matrix size, and three vector variables `rows`, `cols`, and `vals`, containing rows, column and values of the nonzero entries of the matrix. In this way the $i$th nonzero entry of the matrix $M$ is specified as

```
M(rows(i),cols(i)) = vals(i);
```

Working with sparse matrices also ensures that matrix-vector multiplications are carried out parsimoniously: if a matrix entry is null, we don't waste time using it in the multiplications. This results in large savings when one repeatedly evaluates the right-hand side of (1.2). To get further information on sparse matrices, you could look into Matlab's `sparse` command (here), or SciPy's `scipi.sparse` command (here).

As for the previous question, you could write your own code to generate $M$, but I suggest to load the matrices $W$ and $M$ at a first pass. You can load $W$ and $M$ using the binary files `synaptic-matrix.mat` and `fem-matrix.mat`, or their corresponding text (`.dat`) files.

Write code that loads (or computes) the matrices $W$, and $M$. Visualise the sparsity pattern of the matrix $M$ (commands `spy` in both Matlab and Python), and plot the function $r \mapsto W(r, r_{2000})$, the synaptic connections emanating from point $r_{2000}$ in the cortex.

**Question 3.** Cortical regions can support patterns that occupy the full space, (as we have seen in a previous tutorial on Turing-like bifurcations), as well as *localised patterns of activity*. Spots of activity in brain experiments can be found in connection with grid cells [2] (see also this video).

In a localised pattern, high activity is found only in a confined region of the cortex, that remains inactive elsewhere. We are going to explore these patterns as solutions to the following neural field equation

$$\begin{aligned} \partial_t u(r,t) &= -u(r,t) + \int_D w(r,r') f(u(r',t))\, dr', && (r,t) \in D \times [0,T] \\ u(r,0) &= \varphi(r) && r \in D. \end{aligned} \tag{2.1}$$

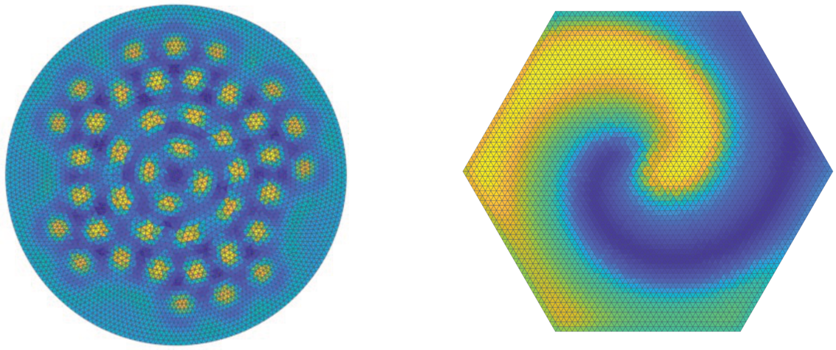

**Figure 2.1.** *Snapshot of the activity variable $u$ from time simulations of questions 3 and 4.*

The cortex $D$ is the disk of radius $R = 30$, and the time horizon is fixed to $T = 50$. The synaptic kernel is the one given in question 2, with parameters $b = 0.4$, $\varepsilon = 10^{-3}$. The firing rate is given by

$$f(u) = \frac{1}{1 + e^{-\mu u + \theta}} - \frac{1}{1 + e^{\theta}}, \qquad \mu = 5.5, \qquad \theta = 5.6,$$

and the initial condition by

$$\varphi(r) = \frac{\alpha}{\big[\cosh(\beta \|r\|_2)\big]^2}, \qquad \alpha = 20, \qquad \beta = \frac{1}{20}.$$

Run a simulation and produce numerical evidence that, when the system reaches an equilibrium, spots of activity form in a large portion of the cortex. This system is capable of sustaining many different configurations, in which spots are localised or not, and displaced in several parts of the cortex.

Perturb the system by setting $\mu = 6$ and taking

$$\varphi(r) = \begin{cases} u(r,T) & \text{if } \|r\|_2 \leq 15, \\ 0 & \text{otherwise,} \end{cases}$$

where $u(r,T)$ is the final state of the previous simulation. This perturbation changes the steepness of the sigmoid, and cuts the outer spots of the equilibrium found for $\mu = 5.5$. How does the system respond to this perturbation?

**Question 4.** Cortices support spatiotemporal patterns of activity, and one of the most striking one are rotating waves [3] (for an experimental recording, download this video).

We investigate the formation of spiral waves in neural fields with a *recovery variable* $v$ in addition to the activity variable $u$. Simulate the occurrence of spiral waves in the model

$$\begin{aligned} \partial_t u(r,t) &= -\alpha u(r,t) - \beta v(r,t) + \nu \int_D w(r,r') f(u(r',t))\, dr', && (r,t) \in D \times [0,T], \\ \tau \partial_t v(r,t) &= -\gamma u(r,t) - \delta v(r,t) && (r,t) \in D \times [0,T], \\ u(r,0) &= \varphi(r), && r \in D, \\ v(r,0) &= \psi(r), && r \in D. \end{aligned}$$

The cortex $D$ is either the disk of radius $R = 30$ (dataset `Spiral-Disk`) or a hexagon inscribed in the circle of radius $R = 30$ (dataset `Spiral-Hexagon`).

The synaptic kernel is distance dependent and truncated

$$w(r,r') = W_{A,\varepsilon}(\|r - r'\|_2), \quad W_{A,\varepsilon}(x) = \begin{cases} A(x) & \text{if } |A(x)| \geq \varepsilon, \\ 0 & \text{otherwise,} \end{cases},$$

and specified setting $\varepsilon = 10^{-3}$ and an integral form for the function $A$, involving the Bessel function of the first kind $J_0$

$$A(x) = \int_0^\infty J_0(xs) \frac{s}{s^4 + s^2 + 1}\, ds.$$

The datasets above provide synaptic and FEM matrices for this kernel on the respective grids.

The firing rate function is given by

$$f(u) = \frac{1}{1 + e^{-\mu(u-\theta)}}, \qquad \mu = 20, \qquad \theta = 0.6.$$

Note that, in this firing rate, the firing threshold is $\mu\theta$, not $\theta$ as in firing rates considered in other questions of the tutorial.

Observe the formation of spiral waves for $\tau = 5$, $\alpha = 1$, $\beta = 2$, $\gamma = -2.2$, $\delta = 1$, $\nu = 3.5$, $T = 50$, and initial conditions given by

$$\varphi(x,y,z) = \begin{cases} 1 & \text{if } y > 0, \\ 0 & \text{otherwise,} \end{cases} \qquad \psi(x,y,z) = \begin{cases} 4 & \text{if } x < 0, \\ 0 & \text{otherwise.} \end{cases}$$

Once a spiral wave has settled, perturb its core instantaneously, by starting a new simulation with initial conditions

$$\varphi(r) = \begin{cases} u(r,T) & \text{if } \|r\|_2 > 15, \\ 0 & \text{otherwise,} \end{cases} \qquad \psi(r) = \begin{cases} v(r,T) & \text{if } \|r\|_2 > 15, \\ 0 & \text{otherwise,} \end{cases}$$

and study whether the spiral survives to this large perturbation.

## REFERENCES

[1] D. Avitabile, *Projection Methods for Neural Field Equations*, SIAM Journal on Numerical Analysis, 61 (2023), pp. 562–591, https://doi.org/10.1137/21M1463768.

[2] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, *Microstructure of a spatial map in the entorhinal cortex*, Nature, 436 (2005), pp. 801–806, https://doi.org/10.1038/nature03721.

[3] X. Huang, W. C. Troy, Q. Yang, H. Ma, C. R. Laing, S. J. Schiff, and J.-Y. Wu, *Spiral Waves in Disinhibited Mammalian Neocortex*, The Journal of Neuroscience, 24 (2004), pp. 9897–9902, https://doi.org/10.1523/JNEUROSCI.2705-04.2004.