NeuroField: A Neural Field Theory simulation toolbox

P.K. Fung*, R.G. Abeysuriya, X. Zhao, P.A. Robinson

School of Physics, University of Sydney, New South Wales, Australia

Abstract

Neural field models are a powerful, computationally efficient approach to modeling large-scale brain activity. NeuroField is an extensible software package to simulate neural field equations in a wide range of models. The basic element of neural field theory (population activity, wave propagation, and synaptic effects) can be assembled into arbitrary networks and integrated numerically to predict brain activity. NeuroField also includes MATLAB and Python routines for higher-level analysis including the power spectrum. NeuroField is implemented in C++ and has been tested on a range of Linux distributions, Microsoft Windows, and Mac OS X. Extensive user documentation and examples are provided, and typical use of NeuroField does not require C++ experience. NeuroField is open-source and available (http://physics.usyd.edu.au/brain/neurofield) under the GNU license for non-commercial use.

Keywords: EEG, neurophysiology, methods, modeling

1. Introduction

Neural field modeling has proved to be a powerful technique for constructing relatively simple, physiologically based models of the brain that are capable of predicting EEG and correlate well with experimental data Deco et al. (2008), Pinotsis et al. (2012). We have developed a neural field corticothalamic model of the brain (Robinson et al., 2002, 2004, 2005, 2001, Rowe et al., 2004) that we

*Corresponding author. Tel. +61 9036 7274
Email address: `ffung@physics.usyd.edu.au` (P.K. Fung)

have previously used to investigate the alpha rhythm (O'Connor and Robinson, 2004, Robinson et al., 2003), age-related changes to the physiology of the brain (van Albada et al., 2010), evoked response potentials (Kerr et al., 2011, Rennie et al., 2002), seizures (Breakspear et al., 2006), and many other phenomena.

The key features of neural field models are captured by the three key equations governing general neural field theory

$$D_{ab} V_{ab}(\mathbf{r}, t) = \nu_{ab} \phi_{ab}(\mathbf{r}, t), \tag{1}$$

$$Q_a(\mathbf{r}, t) = S_a\big[\sum_b V_{ab}(\mathbf{r}, t)\big], \tag{2}$$

$$\mathcal{D}_{ab} \phi_{ab}(\mathbf{r}, t) = Q_b(\mathbf{r}, t - \tau_{ab}). \tag{3}$$

which represent synapto-dendritic smoothing, dendritic summation and firing response, and damped wave propagation, respectively. The differential operators are

$$D_\alpha(t) = \frac{1}{\alpha\beta}\frac{d^2}{dt^2} + \left(\frac{1}{\alpha} + \frac{1}{\beta}\right)\frac{d}{dt} + 1, \tag{4}$$

$$\mathcal{D}_a(\mathbf{r}, t) = \frac{1}{\gamma_a^2}\frac{\partial^2}{\partial t^2} + \frac{2}{\gamma_a}\frac{\partial}{\partial t} + 1 - r_a^2 \nabla^2, \tag{5}$$

and the sigmoid population response S_a is given by

$$Q_a = S(V_a) = \frac{Q_{\max}}{1 + \exp[-(V_a - \theta)/\sigma']}, \tag{6}$$

The relationship between these quantities is schematically illustrated in Fig. 1.

The most challenging part of applying neural field theory is the implementation of the numerical solver. Several factors contribute to making the numerical integration of neural field equations difficult. In particular, propagation delays between neural populations result in delay-differential equations that require special handling of temporal history. Further, propagation of neural fields according to a damped wave equation adds two dimensions to the system, and requires a relative sophisticated finite-differencing scheme that takes into account the geometry of the system. In addition, periodic boundary conditions must be correctly handled during the integration.

We have developed NeuroField to provide a software package that solves the neural field equations for arbitrary neural populations, and contains library code for analysis and visualization, thus removing the barriers to quickly testing and analyzing neural field models. The software is designed to be easily extensible with basic C++ programming skills, making it simple to expand upon the basic model to include new phenomena.

2. Method and Results

2.1. Key features/Basic functions

The essential role of NeuroField is to take as input a model and its initial conditions, and to output one or more time series corresponding to the result of integrating the neural field equations. A model is a specification of neural populations (amounting to defining their firing response to input from other populations including synapto-dendritic effects), and connections between the populations including how neural signals propagate through space. Sensory or other stimulus is implemented as a neural population that recieves no input from other populations, and has a pre-defined firing pattern. Integration of the neural field equations provides several quantities of interest. Most notably, the signals from populations can be associated with local field potentials (LFP)

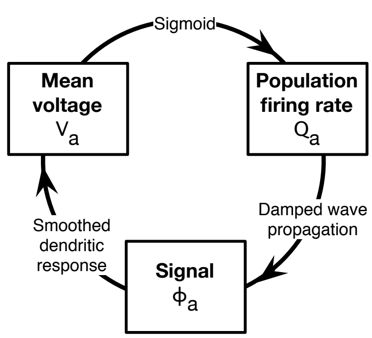

Figure 1: Schematic overview of the key dynamic quantities of neural field models, and the relationships between them.

or EEG depending, and these predictions can be directly compared against experimental data. The soma potential or firing rate of the neural populations can be compared to individual neuron data. Changes to synaptic strength can be monitored when simulating neural plasticity. When simulating spatially extended populations, spatial correlations and patterns of activity can also be analyzed.

The core of NeuroField is a C++ program that accept a human-readable plain text configuration file. The output from NeuroField is a plain text output file containing all of the requested simulation variables for each time step. The syntax of the output file has been designed to be simple to parse, and the NeuroField package includes reference parsers for Matlab and Python. These parsers may also help serve as a starting point for implementation of parsers other programming languages.

There are a number of common ways to analyze the output from neural field models. First, plots of the time series are useful for directly viewing neural oscillations, evoked responses, seizures, monitoring plasticiity, and verifying stability. Second, calculation of the power spectrum, which is often compared to experimental EEG. This can also involve detection of multiple spatial modes of activity and incorporation of volume conduction, to account for effects introduced by electrodes in real-world recordings. Finally, spatial patterns of activity and propagation of waves of activity can be visualized on a surface plot. All of these basic analyses are included as Matlab programs in the NeuroField toolbox.

2.2. Data structures

NeuroField is an object-oriented program where classes are used to encapsulate different components of the simulation. This structure makes it simple to write new components to customize parts of the simulation, that can be easily integrated into the rest of the simulation engine. An overview of the class structure is illustrated in Fig. 2.

The high-level classes `Solver` and `Array` serve as containers to drive the simulation, and to store collections of simulation elements, respectively. The

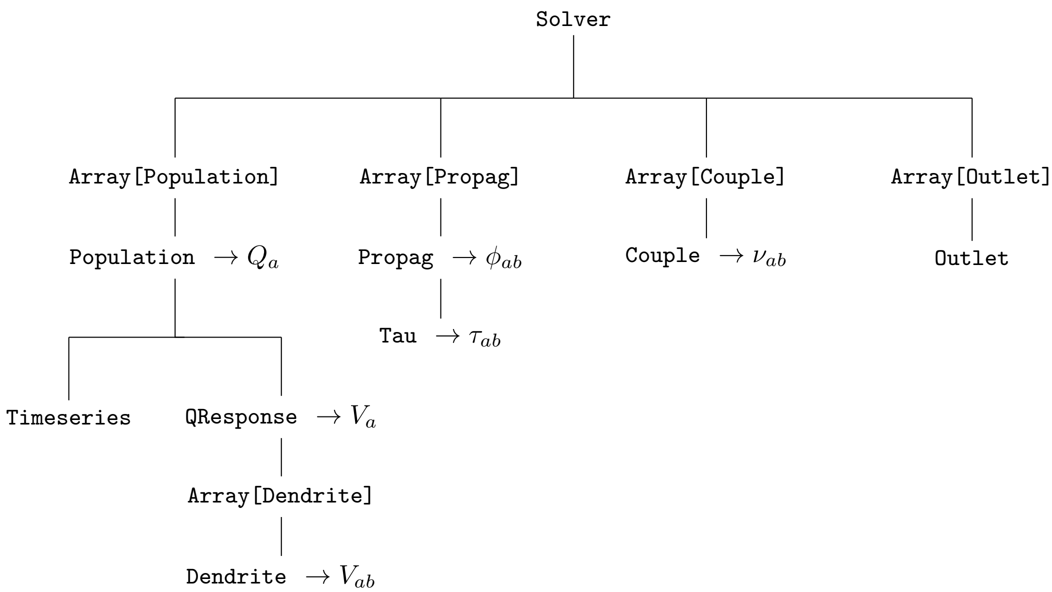

Figure 2: Schematic diagram showing key NeuroField objects, their hierarchical relationships, and their principal associated dynamic quantities.

`outlet` object serves as a modular container for writing variables into the output file. Creating an `outlet` object enables any variable (including new user-defined quantities) to be included in the output.

The main objects (`Couple`, `Dendrite`, `Qresponse`, `Population` and `Propag`) are each responsible for one part of the neural field model.

$$P = \nu_{ab}\phi_{ab}, \qquad \texttt{Couple} \tag{7}$$

$$D_{ab}V_{ab} = P, \qquad \texttt{Dendrite} \tag{8}$$

$$Q_a = S_a\Big[\sum_b V_{ab}\Big], \qquad \texttt{QResponse/Pop} \tag{9}$$

$$\mathcal{D}_{ab}\phi_{ab} = Q_b, \qquad \texttt{Propag} \tag{10}$$

2.2.1. Populations

A `Population` object represents a neural population, which is primarily characterized by a firing rate. A stimulus population is one that has no incoming connections, instead firing according to a pre-programmed selection (e.g., white

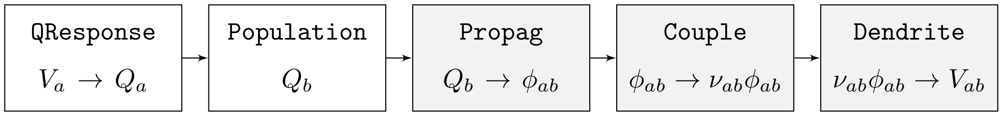

Figure 3: Schematic diagram showing the relationship between fundamental NeuroField objects. The white blocks conceptually relate to neural populations, and the shaded blocks relate to connections.

noise, or pulsed activity). Other populations recieve connections from of other populations, which are specified as a set of `Couple` objects. Each population contains two subsidiary objects, an array of `Dendrite` objects (one for each `Couple`), and a `QResponse`. The signal arriving through a `Couple` object is passed to a corresponding `Dendrite` which implements the synaptodendritic effects in Eq. (4). The contribution V_{ab} from each presynaptic population is then summed to provide the soma potential V_a. The population's `QResponse` object then provides the implementation of Eq. 6 which calculates the population's resulting firing rate. Another common alternative for the firing response is a linear function, which is suitable for small perturbations to a steady state. These behaviours are all specified within the `QResponse` object.

Finally, populations may be further customized to provide additional functionality. One notable example is the inclusion of bursting, which introduces two new dynamic properties of the population that are integrated at each time step. NeuroField includes a basic fourth-order Runge-Kutta integrator that is suitable for these types of additions. The modular nature of NeuroField enables this integrator to be easily substituted with a user-defined function.

2.2.2. Propagators

The neural field generated by a population propagates according to Eq. 5, which is encapsulated in a `Propag` object. There are as many `Propag` objects as there are connections in the model. There are three fundamental possibilities for the propagator. First, the propagator may simply be a direct mapping, with $\mathcal{D}_a(\mathbf{r}, t) = 1$. This is commonly used for short-range local connections. Second, for spatially localized activity we can include only the time derivaties in Eq. 5, which gives a *harmonic* propagator. Finally, we can consider the full expression in Eq. 5, which is the full wave propagator.

Much of the complexity of NeuroField lies in the solution to the wave equation. NeuroField uses an explicit finite difference (9 point) algorithm on a regular square grid with periodic boundary conditions to solve the wave equation. Implementation of the periodic boundary conditions requires that the 9 point stencil correctly wrap around the edges of the grid at every time step. Correct, efficient implementation of this step tends to be the biggest hurdle to implementing a neural field model.

The propagator also takes into account the spatial geometry of the problem. By default, NeuroField solves the wave equation on a flat grid. However, by considering a wave propagator of the form $\mathcal{D}_a(\mathbf{r}, \mathbf{r}', t)$, arbitrary metric tensors may be implemented. This type of propagator enables wave propagation on curved surfaces, which may be as simple as a sphere or as detailed as a surface based on structural MRI.

Finally, the propagator object also encapsulates any time delays, which typically arise due to spatial separation of neural populations (for example, between cortical and thalamic populations). By storing the time delay internally in the

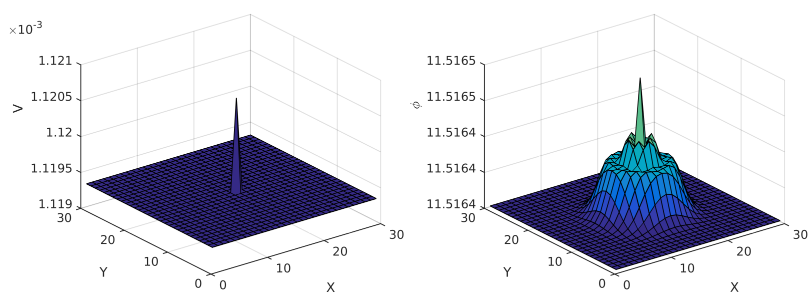

Figure 4: Effect of wave propagation in a single-population model. The stimulus is a single short pulse at the center node. The left panel shows population voltage V_a, which shows the spatial localization of the input signal. The right panel shows the signal ϕ_a after wave propagation.

`Propag` object, all other parts of the simulation are able to simply query the `Propag` to obtain ϕ_b, and the `Propag` will return the retarded value where applicable. Thus customizing other parts of the model requires no special handling of time delays.

2.2.3. Couples

The coupling strength of a connection is typically constant, but can be an arbitrary function of a range of different factors, allowing modeling of a wide variety of plasticity effects including STDP and CaDP

2.2.4. Input and output

- Mention structure of config and output files
- Largely human readable, but relatively simple to construct and parse programatically
- Examples of the output file

Example output, few times, maybe 2 nodes 2 traces

2.3. Visualization and analysis

2.3.1. Helper scripts

NeuroField is packaged with several helper scripts written in MATLAB to assist with running, analyzing and visualizing models.

2.3.2. Reading output files

nf_read() allows users to parse the output file from NeuroField into a MATLAB struct object. nf_grid reshapes the output for handling matrices.

2.3.3. Writing config files

nf_eirs() demonstrates writing a configuration file, running it with nf_run(), and then reading it with nf_read(). This demonstrates a complete MATLAB-based toolchain for using NeuroField.

2.3.4. Calculating power spectra

The power spectrum can be obtained by FFT, but correct normalization and calculation of the power spectrum including multiple spatial modes can be challenging to implement. We have implemented a 3D FFT algorithm that correctly normalizes the output and includes volume conduction effects that selective attenuate spatial modes depending on their wavenumber. The result can be directly compared to analytical predictions.

2.3.5. Visualizing output

The nf_extract() function makes it easy to select data for plotting from a NeuroField object. nf_movie can plot an animation of the output

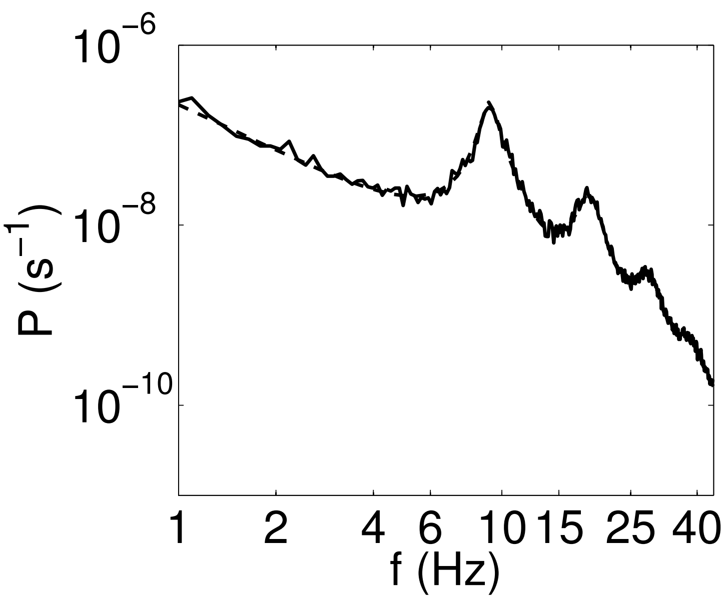

Figure 5: Comparision of linear analytic spectrum with the power spectrum computed using the NeuroField package analysis tools.

3. Discussion

We have developed NeuroField to provide an extensible, reliable framework for integrating nonlinear delay differential equations including spatial propagation. NeuroField is aimed for use by researchers who have constructed neural field models of the brain that require numerical integration. In this section, we review some usage and performance considerations.

3.1. White noise stimulus

- White noise requires stocastic DE integrator, effectively Euler
- Noise amplitude depends on grid resolution as this affects the possible bandwidth. Similar features depend on frequency domain power so noise needs to be normalized correctly

3.2. Performance

- Some numbers about the runtime and memory requirements of NeuroField
- Note that the memory requirements scale with the grid size, and the grid size depends on Lx and the propagator lengths (automatically enforced)
- Also that the delays in the system cause O(n) increases in memory usage

4. Acknowledgements

This work was supported by the Australian Research Council, National Health and Medical Research Council (through the Center for Integrated Research and Understanding of Sleep), and the Westmead Millennium Institute.

5. References

Breakspear M, Roberts JA, Terry JR, Rodrigues S, Mahant N, Robinson PA. A unifying explanation of primary generalized seizures through nonlinear brain modeling and bifurcation analysis. Cereb Cortex 2006; 16:1296–1313.

Deco G, Jirsa VK, Robinson PA, Breakspear M, Friston KJ. The dynamic brain: from spiking neurons to neural masses and cortical fields. PLoS Comput Biol 2008; 4:e1000092.

Kerr CC, Rennie CJ, Robinson PA. Model-based analysis and quantification of age trends in auditory evoked potentials. Clin Neurophysiol 2011; 122:134–147.

O'Connor SC, Robinson PA. Spatially uniform and nonuniform analyses of electroencephalographic dynamics, with application to the topography of the alpha rhythm. Phys Rev E 2004; 70:11911.

Pinotsis DA, Moran RJ, Friston KJ. Dynamic causal modeling with neural fields. NeuroImage 2012; 59:1261–1274.

Rennie CJ, Robinson PA, Wright JJ. Unified neurophysical model of EEG spectra and evoked potentials. Biol Cybern 2002; 86:457–471.

Robinson PA, Rennie CJ, Rowe DL. Dynamics of large-scale brain activity in normal arousal states and epileptic seizures. Phys Rev E 2002; 65:41924.

Robinson PA, Rennie CJ, Rowe DL, O'Connor SC. Estimation of multiscale neurophysiologic parameters by electroencephalographic means. Hum Brain Mapp 2004; 23:53–72.

Robinson PA, Rennie CJ, Rowe DL, O'Connor SC, Gordon E. Multiscale brain modelling. Phil Trans Roy Soc B 2005; 360:1043–1050.

Robinson PA, Rennie CJ, Wright JJ, Bahramali H, Gordon E, Rowe DL. Prediction of electroencephalographic spectra from neurophysiology. Phys Rev E 2001; 63:21903.

Robinson PA, Whitehouse RW, Rennie CJ. Nonuniform corticothalamic continuum model of electroencephalographic spectra with application to split-alpha peaks. Phys Rev E 2003; 68:21922.

Rowe DL, Robinson PA, Rennie CJ. Estimation of neurophysiological parameters from the waking EEG using a biophysical model of brain dynamics. J Theor Biol 2004; 231:413–433.

van Albada SJ, Kerr CC, Chiang AKI, Rennie CJ, Robinson PA. Neurophysiological changes with age probed by inverse modeling of EEG spectra. Clin Neurophysiol 2010; 121:21–38.