

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)

7 have previously used to investigate the alpha rhythm (O'Connor and Robinson,
8 2004, Robinson et al., 2003), age-related changes to the physiology of the brain
9 (van Albada et al., 2010), evoked response potentials (Kerr et al., 2011, Rennie
10 et al., 2002), seizures (Breakspear et al., 2006), and many other phenomena.

11 The key features of neural field models are captured by the three key equa-
12 tions governing general neural field theory

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

$$Q_a(\mathbf{r}, t) = S_a\left[\sum_b V_{ab}(\mathbf{r}, t)\right], \quad (2)$$

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

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

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

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

and the sigmoid population response S_a is given by

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

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

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

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

29 2. Method and Results

30 2.1. Key features/Basic functions

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

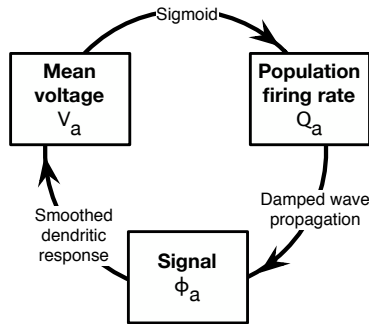


Figure 1: Schematic overview of the key dynamic quantities of neural field models, and the relationships between them.

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

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

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

63 *2.2. Data structures*

64 NeuroField is an object-oriented program where classes are used to encaps-
65 ulate different components of the simulation. This structure makes it simple
66 to write new components to customize parts of the simulation, that can be eas-
67 ily integrated into the rest of the simulation engine. An overview of the class
68 structure is illustrated in Fig. 2.

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

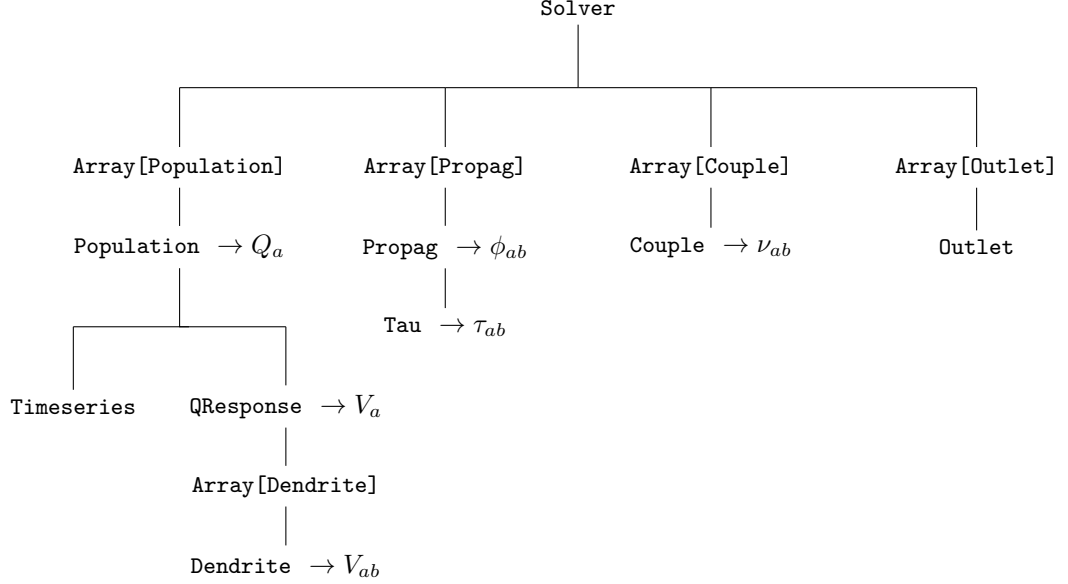


Figure 2: Schematic diagram showing key NeuroField objects, their hierarchical relationships, and their principal associated dynamic quantities.

71 **outlet** object serves as a modular container for writing variables into the output
 72 file. Creating an **outlet** object enables any variable (including new user-defined
 73 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.

$$P = \nu_{ab}\phi_{ab}, \quad \text{Couple} \quad (7)$$

$$D_{ab}V_{ab} = P, \quad \text{Dendrite} \quad (8)$$

$$Q_a = S_a \left[\sum_b V_{ab} \right], \quad \text{QResponse/Pop} \quad (9)$$

$$\mathcal{D}_{ab}\phi_{ab} = Q_b, \quad \text{Propag} \quad (10)$$

74 2.2.1. Populations

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

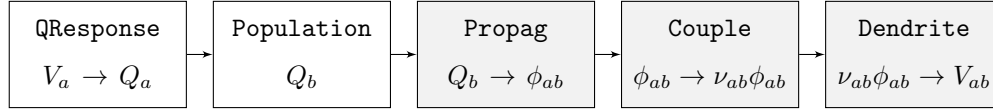


Figure 3: Schematic diagram showing the relationship between fundamental NeuroField objects. The white blocks conceptually relate to neural populations, and the shaded blocks relate to connections.

noise, or pulsed activity). Other populations receive connections from 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.

[More from XL about this](#)

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 derivatives 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.

Do we have a derivation or something for the actual stencil equation?

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.

More from XL/John about this

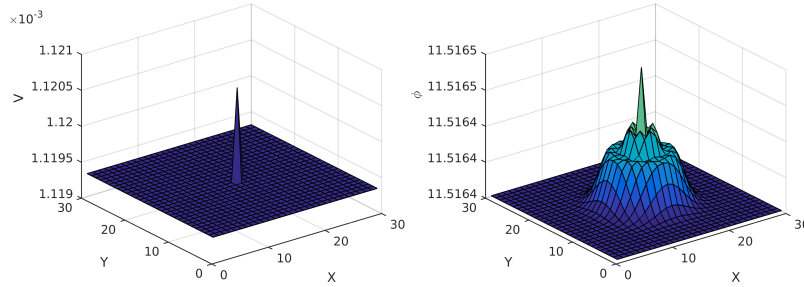


Figure 4: Effect of wave propagation in a single-population model. The stimulus is a single short pulse at the center node. The left panel shows population voltage V_a , which shows the spatial localization of the input signal. The right panel shows the signal ϕ_a after wave propagation.

120 Finally, the propagator object also encapsulates any time delays, which typi-
121 cally arise due to spatial separation of neural populations (for example, between
122 cortical and thalamic populations). By storing the time delay internally in the
123 **Propag** object, all other parts of the simulation are able to simply query the
124 **Propag** to obtain ϕ_b , and the **Propag** will return the retarded value where appli-
125 cable. Thus customizing other parts of the model requires no special handling
126 of time delays. The delays may also vary spatially, so that different parts of the
127 system have different time delays.

128 2.2.3. *Couples*

129 The coupling strength ν_{ab} scales the incoming signals, weighting the contri-
130 bution from different presynaptic sources. The strength of the synapse is stored
131 in a **Couple** object. Typically the strength is constant, reflecting the number
132 and strength of the relevant synaptic connections. However, modulation of the
133 coupling strengths forms the basis of neural plasticity. In NeuroField, this is
134 achieved by modifying the **Couple** to take into account time-varying connec-
135 tions. We have used this functionality in previous studies to model a wide
136 range of plasticity effects including spike-timing dependent plasticity (STDP)
137 and calcium dependent plasticity (CaDP).

138 [More from Felix about this](#)

139 2.2.4. *Input and output*

140 NeuroField input and output are written in plain text files, which are hu-
141 man readable and simple to construct and parse programatically. An simple
142 configuration file is shown in Fig. 5. The configuration file first starts with a
143 specification of the simulation duration and time step. The total number of
144 nodes is specified, and assumed to correspond to a square grid - in this case,
145 30x30 nodes.


```

Time: 0.15 Deltat: 0.0001
Nodes: 900

    Connection matrix:
From:  1  2
To 1:  1  2
To 2:  0  0

Population 1: Excitatory
Length: 0.5
Q: 10.98
Firing: Sigmoid - Theta: 0.01292 Sigma: 0.0038 Qmax: 340
    Dendrite 1: alpha: 83.33333333 beta: 769.2307692
    Dendrite 2: alpha: 83.33333333 beta: 769.2307692

Population 2: Stimulation
Length: 0.5
    Stimulus: Pulse - Onset: 0 Node: 465 Amplitude: 1 Width:
                1e-3

Propag 1: Wave - Tau: 0 Range: 0.2 gamma: 30
Propag 2: Map - Tau: 0

Couple 1:  Map - nu: 1e-4
Couple 2:  Map - nu: 1e-4

Output: Node: All Start: 0
Population: 1
Dendrite:
Propag: 1
Couple:

```

Figure 5: Example config file from NeuroField, for a simple system consisting of one neural population, and a stimulus. This configuration was used to generate Fig. 4.

146 *2.3. Visualization and analysis*

147 *2.3.1. Helper scripts*

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

150 *2.3.2. Reading output files*

151 `nf_read()` allows users to parse the output file from NeuroField into a MAT-
152 LAB struct object. `nf_grid` reshapes the output for handling matrices.

153 *2.3.3. Writing config files*

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

157 *2.3.4. Calculating power spectra*

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

Time	Pop . 1 . V	Dendrite . 1 . V
	1	1
5.00e-03	-1.020386100923e-03	2.589927678839e-02
1.00e-02	-1.020386100890e-03	2.589927678842e-02

Figure 6: Example output from NeuroField, showing two time series from different parts of the system. The plain text file contains a row for each time, and a column for each quantity requested.

164 *2.3.5. Visualizing output*

165 The `nf.extract()` function makes it easy to select data for plotting from a
166 `NeuroField` object. `nf.movie` can plot an animation of the output

167 **3. Results**

168 In this section, we present examples of `NeuroField` applied to recent research.

169 *3.1. Corticothalamic model (Romesch)*

- 170 • Sleep spindles
- 171 • Wake alpha peak
- 172 • Volume conduction

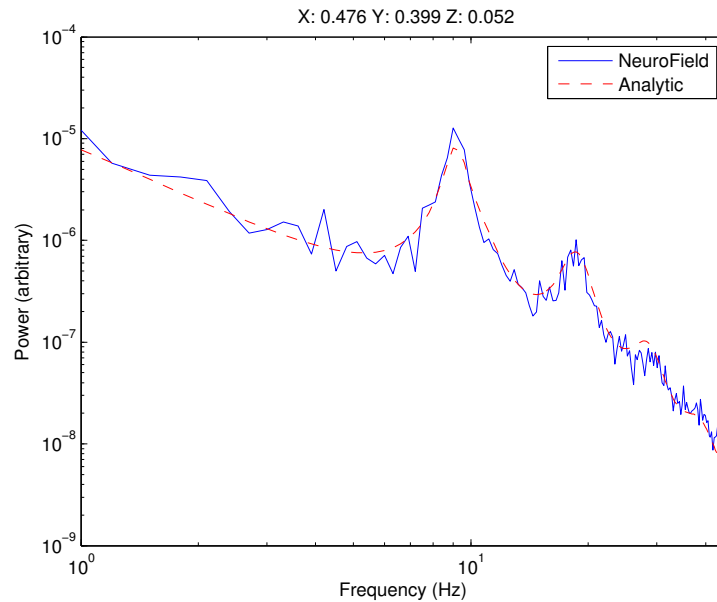


Figure 7: Comparison of linear analytic spectrum with the power spectrum computed using the `NeuroField` package analysis tools.

173 3.2. *Plasticity (Felix)*

174 3.3. *Bursting (XL)*

175 3.4. *Seizures (XL, Romesh)*

176 4. Discussion

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

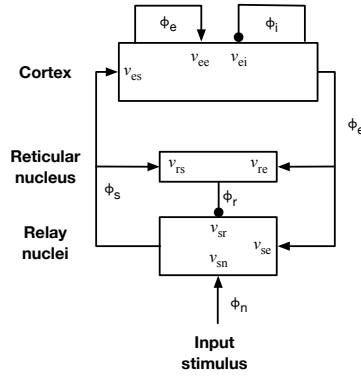


Figure 8: Wake EC, same params as already published

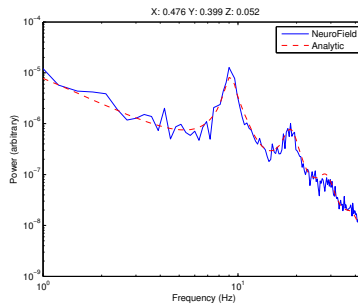


Figure 9: Wake EC, same params as already published

182 4.1. *White noise stimulus*

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

187 4.2. *Performance*

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

192 5. Acknowledgements

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

196 6. References

- 197 Breakspear M, Roberts JA, Terry JR, Rodrigues S, Mahant N, Robinson PA. A
198 unifying explanation of primary generalized seizures through nonlinear brain
199 modeling and bifurcation analysis. *Cereb Cortex* 2006; 16:1296–1313.
- 200 Deco G, Jirsa VK, Robinson PA, Breakspear M, Friston KJ. The dynamic brain:
201 from spiking neurons to neural masses and cortical fields. *PLoS Comput Biol*
202 2008; 4:e1000092.
- 203 Kerr CC, Rennie CJ, Robinson PA. Model-based analysis and quantification of
204 age trends in auditory evoked potentials. *Clin Neurophysiol* 2011; 122:134–
205 147.

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

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

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

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

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

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

220 Robinson PA, Rennie CJ, Wright JJ, Bahramali H, Gordon E, Rowe DL. Pre-
221 diction of electroencephalographic spectra from neurophysiology. *Phys Rev*
222 *E* 2001; 63:21903.

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

226 Rowe DL, Robinson PA, Rennie CJ. Estimation of neurophysiological param-
227 eters from the waking EEG using a biophysical model of brain dynamics. *J*
228 *Theor Biol* 2004; 231:413–433.

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