

# NeuroField: A Neural Field Theory simulation toolbox

P.K. Fung\*, R.G. Abeysuriya, X. Zhao, P. M. Drysdale, 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). The key features of neural field models are captured by the three key equations governing general neural field theory

---

\*Corresponding author. Tel. +61 9036 7274

Email address: [ffung@physics.usyd.edu.au](mailto:ffung@physics.usyd.edu.au) (P.K. Fung)

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

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

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

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

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

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

8 The most challenging part of applying neural field theory is the implementa-  
 9 tion of the numerical solver. Several factors contribute to making the numerical  
 10 integration of neural field equations difficult. In particular, propagation delays  
 11 between neural populations result in delay-differential equations that require

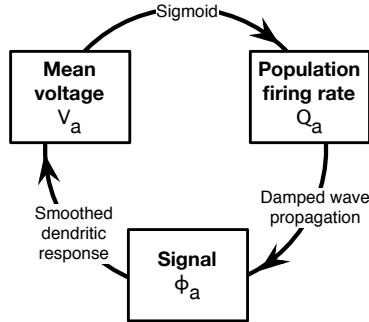


Figure 1: Schematic overview of the key dynamic quantities of neural field models, and the relationships between them.

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 receives 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) 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.

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

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

## 57 *2.2. Data structures*

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

63 The high-level classes **Solver** and **Array** serve as containers to drive the  
 64 simulation, and to store collections of simulation elements, respectively. The  
 65 **outlet** object serves as a modular container for writing variables into the output  
 66 file. Creating an **outlet** object enables any variable (including new user-defined  
 67 quantities) to be included in the output.

The main objects (**Couple**, **Dendrite**, **Qresponse**, **Population** and **Propag**)

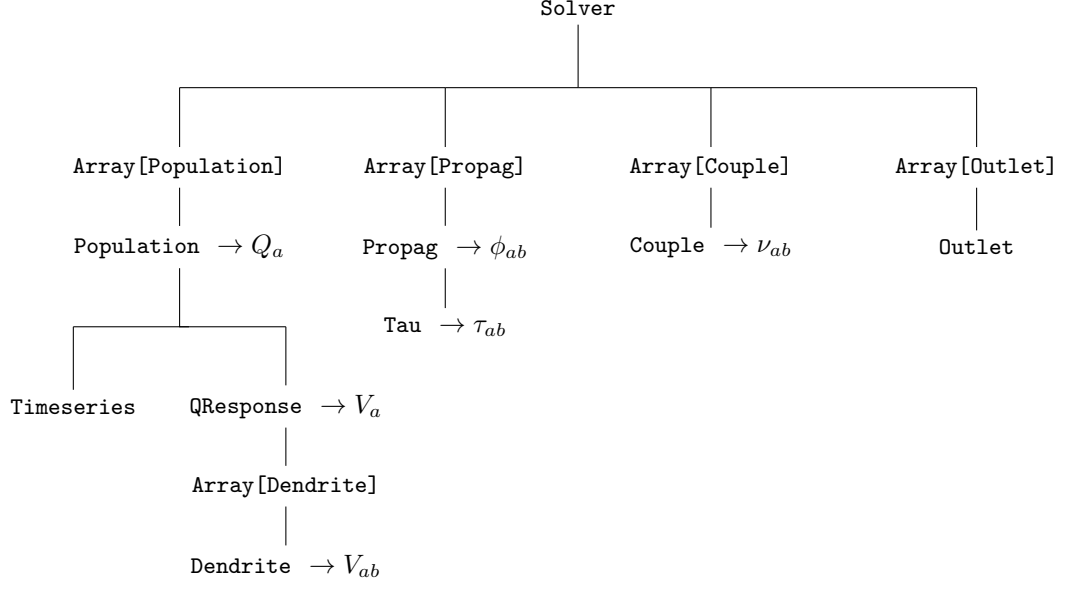


Figure 2: Schematic diagram showing key NeuroField objects, their hierarchical relationships, and their principal associated dynamic quantities.

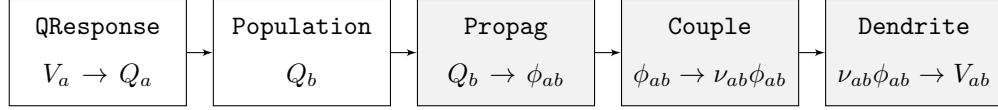


Figure 3: Schematic diagram showing the relationship between fundamental NeuroField objects. The white blocks conceptually relate to neural populations, and the shaded blocks relate to connections.

are each responsible for one part of the neural field model.

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

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

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

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

### 68 2.2.1. Populations

69 A **Population** object represents a neural population, which is primarily  
70 characterized by a firing rate. A stimulus population is one that has no incoming  
71 connections, instead firing according to a pre-programmed selection (e.g., white  
72 noise, or pulsed activity). Other populations receive connections from other  
73 populations, which are specified as a set of **Couple** objects. Each population  
74 contains two subsidiary objects, an array of **Dendrite** objects (one for each  
75 **Couple**), and a **QResponse**. The signal arriving through a **Couple** object is  
76 passed to a corresponding **Dendrite** which implements the synaptodendritic  
77 effects in Eq. (4). The contribution  $V_{ab}$  from each presynaptic population is then  
78 summed to provide the soma potential  $V_a$ . The population's **QResponse** object  
79 then provides the implementation of Eq. 6 which calculates the population's  
80 resulting firing rate. Another common alternative for the firing response is a  
81 linear function, which is suitable for small perturbations to a steady state. These  
82 behaviours are all specified within the **QResponse** object.

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

89 [More from XL about this](#)

### 90 2.2.2. Propagators

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

98 in Eq. 5, which is the full wave propagator.

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

Do we have a derivation or something for the actual stencil equation? (FE-  
 LIX)

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

More from XL/John about this

114 Finally, the propagator object also encapsulates any time delays, which typi-

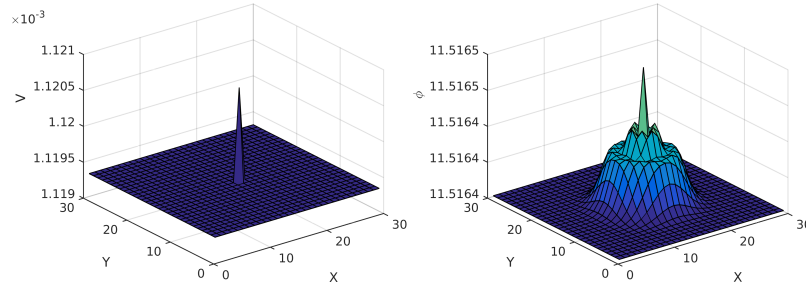


Figure 4: Effect of wave propagation in a single-population model. The stimulus is a single short pulse at the center node. The left panel shows population voltage  $V_a$ , which shows the spatial localization of the input signal. The right panel shows the signal  $\phi_a$  after wave propagation.

115 cally arise due to spatial separation of neural populations (for example, between  
 116 cortical and thalamic populations). By storing the time delay internally in the  
 117 **Propag** object, all other parts of the simulation are able to simply query the  
 118 **Propag** to obtain  $\phi_b$ , and the **Propag** will return the retarded value where appli-  
 119 cable. Thus customizing other parts of the model requires no special handling  
 120 of time delays. The delays may also vary spatially, so that different parts of the  
 121 system have different time delays.

### 122 2.2.3. *Couples*

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

132 [More from Felix about this](#)

### 133 2.2.4. *Input and output*

134 NeuroField input and output are written in plain text files, which are hu-  
 135 man readable and simple to construct and parse programatically. An simple  
 136 configuration file is shown in Fig. 5.

137 The configuration file first starts with a specification of the simulation du-  
 138 ration and time step. The total number of nodes is specified, and assumed to  
 139 correspond to a square grid - in this case, 30x30 nodes. It is also possible to  
 140 specify a rectangular grid at this point.

141 Next, a list of connections is provided, formatted as a human-readable con-  
 142 nection matrix. Each connection is assigned a number at this point. The dimen-  
 143 sion of the matrix corresponds to the number of populations being simulated,



144 and the number of nonzero entries in the matrix corresponds to the number of  
145 couplings, propagators and dendrites.

146 Following the connection matrix is a detailed specification for each popu-  
147 lation. The spatial grid resolution is determined automatically based on the  
148 number of nodes and the physical dimensions of the population. The Courant  
149 condition for numerical stability is automatically checked. The firing response  
150 is specified for each population, with parameters delegated to the appropriate  
151 **QResponse** object. The parameters for each afferent dendrite are specified at  
152 this point as well. Stimulus populations have no dendrites. Instead, the stim-  
153 ulus is specified by a **TimeSeries** object, a container that returns a firing rate  
154 as a function of time. In Fig. 5, the stimulus is of type **Pulse**, and acts at only  
155 a single node of the simulation. Different stimuli can be applied at different  
156 times, and multiple stimulus populations can be used to investigate the effect  
157 of superimposing stimuli.

158 For each connection, the propagator and coupling type need to be specified.  
159 In Fig. 5, one of the propagators is given by the wave equation, and the other  
160 is a direct mapping. The time delay, axonal range, and damping rate are all  
161 specified independently for each propagator.

162 The final block in the configuration file selects the output from NeuroField.  
163 The output can include all of the nodes, or a user-defined subset of nodes.  
164 The start of the output can be delayed to remove transients from the output.  
165 To ensure numerical stability, the equations need to be integrated with a very  
166 small time step (e.g.,  $1 \times 10^{-4}$  s = 10000 Hz). This is much smaller than typically  
167 required for analysis; EEG is typically sampled at just 500 Hz or less. To reduce  
168 file size and make handling the output less computationally intensive, an output  
169 sampling interval can be specified, downsampling the output from NeuroField.

170 **Stuff about the output file**

```

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 Interval: 1e-4
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.

171 *2.3. Visualization and analysis*

172 *2.3.1. Helper scripts*

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

175 *2.3.2. Reading output files*

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

178 *2.3.3. Writing config files*

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

182 *2.3.4. Calculating power spectra*

183 The power spectrum can be obtained by FFT, but correct normalization  
184 and calculation of the power spectrum including multiple spatial modes can  
185 be challenging to implement. We have implemented a 3D FFT algorithm that  
186 correctly normalizes the output and includes volume conduction effects that  
187 selective attenuate spatial modes depending on their wavenumber. The result  
188 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.

189 *2.3.5. Visualizing output*

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

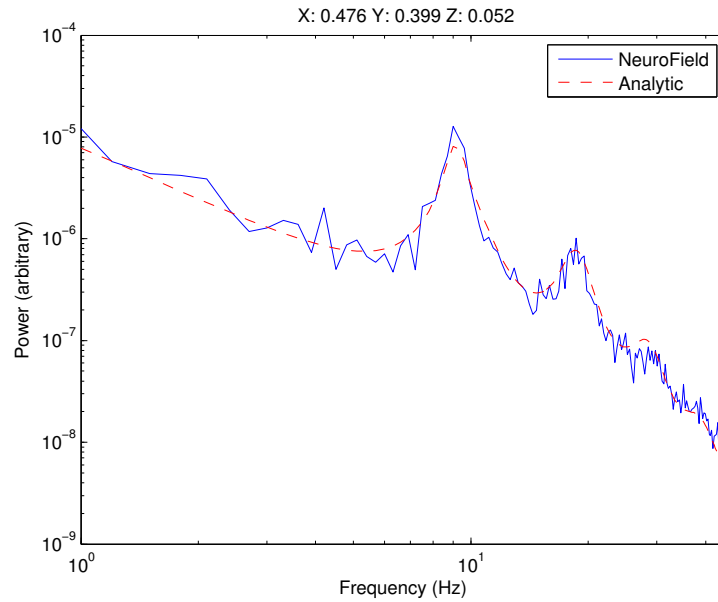


Figure 7: Comparison of linear analytic spectrum with the power spectrum computed using the NeuroField package analysis tools.

### 3. Results

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

#### 3.1. Corticothalamic model (Romesch)

NeuroField has been extensively tested with a recent neural field corticothalamic model of the brain (Robinson et al., 2002, 2004, 2005, 2001, Rowe et al., 2004) that we 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 structure of the model is shown in Fig. 8. Much of our work is in the linear regime and utilizes analytic results for small perturbations to steady states. Steady states can be obtained by setting the differential operators (4) and (5) to unity, and solving for the the steady state firing rate  $\phi_e^{(0)}$  Robinson et al. (2004)

$$S^{-1}(\phi_e^{(0)}) - (\nu_{ee} + \nu_{ei})\phi_e^{(0)} = \nu_{es}S \left\{ \nu_{se}\phi_e^{(0)} + \nu_{sr}S \left[ \nu_{re}\phi_e^{(0)} + (\nu_{rs}/\nu_{es}) \left( S^{-1}(\phi_e^{(0)}) - (\nu_{ee} + \nu_{ei})\phi_e^{(0)} \right) \right] + \nu_{sn}\phi_n^{(0)} \right\}, \quad (11)$$

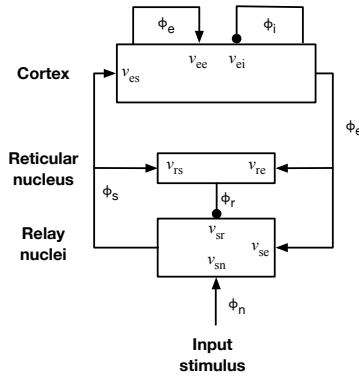


Figure 8: Schematic overview of the corticothalamic model.

where  $S^{-1}$  denotes the inverse sigmoid function. To linearize the model, we take a Taylor expansion of the sigmoid and retain only the first term:

$$Q_a(\mathbf{r}, t) = Q_a^{(0)} + S' \left( V_a^{(0)} \right) \left[ V_a(\mathbf{r}, t) - V_a^{(0)} \right] \quad (12)$$

This can be written in terms of a perturbation by relabeling  $Q_a - Q_a^{(0)} \rightarrow Q_a$ ,  $V_a - V_a^{(0)} \rightarrow V_a$ , yielding

$$Q_a(\mathbf{r}, t) = \rho_a^{(1)} V_a(\mathbf{r}, t), \quad (13)$$

where  $\rho_a^{(n)} = S^{(n)} \left( V_a^{(0)} \right)$  and  $S^{(n)}$  is the  $n$ th derivative of the sigmoid function at the steady state value of  $Q_a$ . Taking the Fourier transform of Eqs. 1–5,13 and solving for  $\phi_e/\phi_n$  yields a transfer function that maps stimulus to cortical response in the frequency domain

$$T = \frac{\phi_e(\mathbf{k}, \omega)}{\phi_n(\mathbf{k}, \omega)} = \left( D_e(\mathbf{k}, \omega) - \frac{1}{1 - G_{ei}L(\omega)} \left\{ L(\omega)G_{ee} + \frac{[L(\omega)^2G_{ese} + L(\omega^3G_{erse}]e^{i\omega t_0}}{1 - L(\omega^2G_{srs})} \right\} \right)^{-1} \frac{L(\omega^2G_{esn})}{(1 - L(\omega)^2G_{srs})} \quad (14)$$

202 where the gains  $G_{ab}$  are given by  $G_{ab} = \rho_a \nu_{ab}$ ,  $G_{ese} = G_{es}G_{se}$ ,  $G_{erse} =$   
 203  $G_{es}G_{sr}G_{re}$ ,  $G_{srs} = G_{sr}G_{rs}$ , and  $G_{esn} = G_{es}G_{sn}$ .

204 The transfer function can be related to the EEG power spectrum

205 However, some phenomena are strongly nonlinear, and these must be solved  
 206 by numerical integration. We have used NeuroField to model results relating  
 207 to seizures (?), visually evoked potentials (Roberts and Robinson, 2012), and  
 208 sleep spindles (Abey Suriya et al., 2014, ?).

$$P(\omega) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \Delta k_x \Delta k_y |T(\mathbf{k}, \omega)|^2 |\phi_n(\mathbf{k}, \omega)|^2 F(k) \quad (15)$$

$$k^2 = \left( \frac{2\pi m}{L_x} \right)^2 + \left( \frac{2\pi n}{L_y} \right)^2 \quad (16)$$

209 where the size of the two-dimensional rectangular cortex is  $L_x \times L_y$ .

$$F(k) = e^{-k^2/k_0^2}, \quad (17)$$

Prediction of the EEG power spectrum involves two additional steps - first, summation over

- Sleep spindles
- Wake alpha peak
- Volume conduction

*3.2. Plasticity (Felix)*

*3.3. Bursting (XL)*

*3.4. Seizures (XL)*

## 4. 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.

*4.1. White noise stimulus*

- White noise requires stochastic DE integrator, effectively Euler (FELIX)

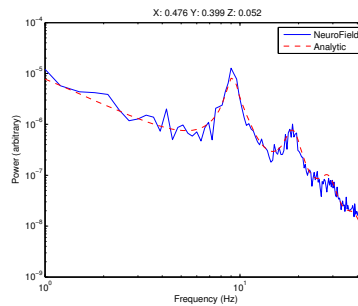


Figure 9: Wake EC, same params as already published

- 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 (ROMESH)

#### 4.2. Performance

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

Additional considerations?

## 5. 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.

## 6. References

- Abeyesuriya RG, Rennie CJ, Robinson PA. Prediction and verification of non-linear sleep spindle harmonic oscillations. *J Theor Biol* 2014; 344:70–77.
- 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.



251 Kerr CC, Rennie CJ, Robinson PA. Model-based analysis and quantification of  
252 age trends in auditory evoked potentials. *Clin Neurophysiol* 2011; 122:134–  
253 147.

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

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

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

261 Roberts JA, Robinson PA. Quantitative theory of driven nonlinear brain dy-  
262 namics. *NeuroImage* 2012; 62:1947–1955.

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

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

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

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

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

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

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