

028/9/9
• Fully tested
• Alpha

Reference documentation for EEGcode

1 Overview of code function

The code implements the eeg model equations.

- The code currently runs in one of three modes: (1) Initial run, (2) restart calculation and (3) default parameters mode.
 - (1) In initial run mode code takes input from an eegcode.conf file initializes the simulation and then takes specified number of integration steps writing out a eegcode.output file and a eegcode.dump file for restarting. The input filename can be changed on the command line by using the switch “-i newinputfilename”. Similarly the name of the output filename which defaults to eegcode.output can be changed via the switch “-o newoutputfilename”. The dump file which defaults to eegcode.dump can be changed via “-d newdumpfilename”.
 - (2) The code dumps sufficient data about parameter values to restart. In restart mode the simulation will continue where it left off. Restart mode is accessed by running the code with command “eegcode restart” rather than the usual “eegcode” command on the command line. In order to run the code in restart mode it is necessary to take the “eegcode.dump” file which was produced from the run which will be restarted to be renamed as the new “eegcode.conf” file. The new “eegcode.conf” file should also be changed to reflect the new number of timesteps etc. Note:- A restarting “eegcode.conf” file has different content to an initial run “eegcode.conf” and can only be used in restart mode.
 - (3) The default parameters mode is accessed by running the code with the command “eegcode write defaults” on the command line. The code then writes its own default “eegcode.conf” file and uses it for a initial run. The default “eegcode.conf” file can be inspected after the code finishes.
- One integration step of the model implements the following stages: 1) Firing Response/Stimulus Response. 2) Wave Equation Integration Step which includes Q delay processing 3) Coupling Response 4) Dendritic Response 5) Afferent Summation.
- Dendritic Response - is by direct explicit step or Fourth order Runge Kutta. Dendritic Response solves the following equation:

$$\left\{ \frac{1}{\alpha_{ab}(\mathbf{r}, t)\beta_{ab}(\mathbf{r}, t)} \frac{d^2}{dt^2} + \left[\frac{1}{\alpha_{ab}(\mathbf{r}, t)} + \frac{1}{\beta_{ab}(\mathbf{r}, t)} \right] \frac{d}{dt} + 1 \right\} V_{ab}(\mathbf{r}, t) = P_{ab}(\mathbf{r}, t). \quad (1)$$

- Wave Equation integration is by explicit finite difference (9 point) algorithm on a regular square grid. Other parts of code are unaffected by spatial geometry so this can be switched to irregular gridding easily. The wave equation currently solved is

$$\left[\frac{1}{\gamma_{ab}^2} \frac{\partial^2}{\partial t^2} + \frac{2}{\gamma_{ab}} \frac{\partial}{\partial t} + 1 - r_{ab}^2 \nabla^2 \right] \phi_{ab}(\mathbf{r}, t) = Q_b(\mathbf{r}, t - \tau_{ab}). \quad (2)$$

The patchy propagator has been implemented but has not been tested.

Y
seen -

- Two types of coupling response have been implemented. The first is a relatively simple and most commonly used form which is

$$P_{ab}(\mathbf{r}, t) = \nu_{ab}\phi_{ab}(\mathbf{r}, t). \quad (3)$$

The second form is a coupling which implements a time evolving coupling $\nu(t)$ according to the Rennie/Clearwater evolution equation. The form is described later in this document.

- Parameters α_{ab} , β_{ab} , θ_a , σ_a , $Q_{a\max}$, ν_{ab} , Effective range r_{ab} , γ_{ab} are all set at initialization currently. They have been explicitly designed to allow replacement with functional forms allowing evolution in time and space as needed.
- Arbitrary numbers of populations are allowed.
- In order to save memory a connection matrix between the populations is specified at run-time. This ensures that where two populations are not connected the corresponding wave equation and dendritic response are not created in memory. *set up*
- Arbitrary number of stimulus populations are allowed. Stimulus populations are populations of neurons which correspond to different stimulus modes. They have no feedback into them from the other populations. Three simple stimulus patterns are currently implemented. 1) Constant in space and time 2) Gaussian noise of constant amplitude in space and time 3) Sinusoidal variation in time of stimulus with constant amplitude and stimulus is spatially uniform. These correspond to the first three stimulus types in Chris Rennie's code. Other stimulus patterns are to be written shortly.

2 Input files

The input file is a human readable file of the parameters of the simulation. An input file is generated by the default running procedure of the code. This can be useful for understanding the layout of the input file. The default file creates a input file for the standard corticothalamic model. Before reading the input file description it is worthwhile to peruse the default file.

The input file is in a number of sections, first the global simulation information, next the information for each population is given, then the propagation data, then the coupling data and finally the output data.

2.1 Global information

Each bullet point corresponds to a line of the input file.

- The first line of the input file is skipped to allow the user to have a title line
- **Nodes per population:** 2500 This line contains the number of grid points in the spatial dimension per population of neurons. The code has been explicitly designed to have equal number of neurons per population. *Set*
- **Number of neural populations:** 5 This is the number of neural populations and includes the number of stimulus modes. It is denoted n in this document.

Default / example value?

- Number of neural connections: 11 This is the number of neural populations which are connected together. By definition this number lies in the range n and n^2 where n is the number of neural populations.
- A blank line is inserted in the file at this point
- Populations / connections / matrix 1 2 3 4 The code ignores this line but it is useful to provide a way of lining up the data in the connections / matrix.
- Now follows n lines of connection matrix data. On each line the code looks for colon character followed by a number. There should be n colons followed by a number on each line. The row number corresponds to the population ID number of the TO population. The column number corresponds to the population ID number of the FROM population of the connection. Zeros correspond to unconnected populations whilst nonzero number correspond to connected populations. For easy of use it is suggested that each connection be numbered in ascending order when reading down columns as in the default input file. This allows human crosschecking with later sections of the input file.
- Number of integration steps: 100000 Skippoints: 49 Deltat: 0.0001 The parameters are: (1) the number of integration steps the code should step forward during the run, (2) An optional parameter of how many integration steps should be skipped before an time step should be outputted, and (3) the time increment for each time step.
- Q delay depths: 420 :0 :0 :420 :0 This line contains n colon characters each followed by a number which is the Q delay depth. Each number corresponds to the Q delay depth of a neural population and the populations are in the same order as the connection matrix. The Q delay depth is the number of time steps for the signal to propagate from this population to another population. Mathematically for population i the Q delay depth is $\max(\tau_{ai}/\Delta t)$. In the default input file, population 1 (the excitatory cortical neurons) transmit signals to the thalamus with a delay of 42ms and the time step is 0.1ms so the Q delay depth is 420. Similarly, the delay from the relay nuclei in the thalamus to the cortex is also 42ms so the delay depth is also 420.
- Propagator types 1: Waveeqn 2: Waveeqn 3: Waveeqn 4: Waveeqn 5: Mapping \$ 6: Mapping 7: Mapping 8: Waveeqn 9: Waveeqn 10: Mapping 11: Mapping \$ These lines contain n colons preceeded by an index number and followed by a specification of the propagator type. The propagation data should appear in the same order as in the connection matrix when the connection matrix is read top to bottom and then left to right. Three types of propagator are available: 1) "Waveeqn" - is a propagator implementing the usual wave equation as described in the previous section. 2) "Mapping" is a propagator which maps $Q_b(t - \tau)$ directly to $\phi_{ab}(t)$. This propagator is equivalent to cases where spatial spreading is negligible i.e. for a wave equation where numerically r_{ab} is small and γ_{ab} is large. 3) "Eqnset" - is the untested Robinson patchy propagator. 4) "Harmonic" - is a harmonic oscillator propagator i.e. identical with "Waveeqn" operator but with no spatial dependence.
- Coupling types 1: Modulate 2: Simple 3: Simple 4: Simple 5: Simple 6: Simple \$ 7: Simple 8: Simple 9: Simple 10: Simple 11: Simple \$ These lines contain n colons preceeded by an index number and followed by a specification of the

synaptic coupling type. Two types of coupling are available: 1) “Simple” - is a coupling which does not evolve with time and is given in Eq. 3. 2) “Modcouple” - is a Clearwater/Rennie coupling described in detail later in this document.

- A blank line finishes this section of the input file

2.2 Population data

This section contains n repeated population information sections. Each population is presented in the same order as the connections matrix. There are two types of neural populations: 1) ordinary populations and 2) stimulus populations. The two types of populations have different input information sections

2.2.1 Ordinary populations

Each population information section is as follows. Each bullet point corresponds to a line of the population information section.

- The first line is ignored by the code but should be used for a text description of the population’s name eg. **Population 1 - Excitory neurons**.
- **Initial Q:** 8.87145 When the simulation starts an initial value for the firing rate Q is required.
- **Firing Response Theta:** 0.013 **Sigma:** 0.0038 **Qmax:** 250.0 These are the parameters of the population sigmoid firing response. Note the sigma value is in some papers known as sigma-tilde. It is already scaled by $\pi/\sqrt{3}$.
- **Number of Dendritic responses:** 3 The code uses this line purely for crosschecking purposes. It is equal to the number of populations whose connections terminate on the dendrites of the current population i.e. the number of non-zero elements in the column of the connection matrix of this population. We shall denote this number m
- Now follow m lines of dendritic response data. Each Dendritic response data line begins with an optional preamble which is ignored by the code. It is strongly suggested that for human readability this should be of the form **Dendritic Response from population 3**. A complete dendritic response input line is given by

```
Dendritic Response from population 3 V initial: 0.0106457 alpha: 75.0 beta: 285.0.
```

The following three parameters are the initial value of the subvoltage V_{ab} and the parameters α_{ab} and β_{ab} of the dendritic response differential equation. It should be noted the code has a separate dendritic response differential equation for each different population synapsing on the dendritic tree. The subvoltage V_{ab} is the voltage in the dendrite due to incoming connections from population b . In some earlier papers all pulse densities were summed before a single differential equation described the total voltage in the dendritic tree. This code allows different dendritic responses for pulse densities arriving from different populations. If each dendritic response in a population has the same alpha and beta values the earlier case is effectively reproduced. If the initial V_{ab} is given by the initial values of ν_{ab} and ϕ_{ab} via $V_{ab} = \nu_{ab}\phi_{ab}$ as is the case in a steady state condition then for example the dendritic response may be given as

Dendritic Response from population 3 V initial: Steady alpha: 75.0 beta: 285.0.

The “Steady” short hand feature cannot be used if any of the coupling described later in this document are of the ModCouple type.

- A blank line finishes each population information section.

2.2.2 Stimulus populations

The code identifies stimulus populations as populations which have no connections from other populations on their dendritic tree i.e., The column for that population contains no nonzero elements. Each stimulus population information section is as follows. Each bullet point corresponds to a line of the population information section.

- The first line is ignored by the code but should be used for a text description of the population’s name eg. Population 5 - Stimulus neurons.
- Initial Q: 8.87145 When the simulation starts an initial value for the firing rate Q is required.
- The final line depends on the type of stimulus the stimulus population provides. Examples of the current stimulus patterns are provided below.

Pulse stimulus pattern

```
Stimulus mode: 1 Time to start of stimulus: 0.002 Amplitude: 20 Pulse Duration:  
0.02 Pulse repetition period: 0.005
```

White noise stimulus pattern (spatially uncorrelated) where Ranseed is an optional parameter for the random number generator’s initial seed defaulting to -98716872.

```
Stimulus mode: 2 Time to start of stimulus: 0.002 Ranseed:-98716872 Amplitude:  
20
```

Sinusoidal stimulus pattern -Sinusoidally modulated firing rate

```
Stimulus mode: 3 Time to start of stimulus: 0.002 Amplitude: 20 Modulation  
Frequency: 10
```

Coherent white noise stimulus pattern (spatially coherent) where Ranseed is an optional parameter for the random number generator’s initial seed defaulting to -98716872.

```
Stimulus mode: 4 Time to start of stimulus: 0.002 Ranseed:=-98716872 Amplitude:  
20
```

2.3 Propagation data

Each bullet point corresponds to a line of the propagation data section.

- The first line is ignored and is merely a title line Propagation data.
- Now follow n lines of propagation data. There is a line of propagation data for each connection in the connection matrix. The propagation data should appear in the same order as in the connection matrix when the connection matrix is read top to bottom and then left to right. There are three types of propagators each with their own input parameters

2.3.1 WaveEqn

This propagator is the wave equation propagator and has input given by

Propagator 2 - Initial Phi: 8.87145 **Tauab:** 0 **Effective Range:** 0.08 **gamma:** 100.0.

The Initial Phi is the initial value for ϕ_{ab} in the wave equation. Tauab is $\tau_{ab}/\Delta t$ the delay term in the wave equation i.e. number of time steps in the delay term. Optionally Tauab can be replaced by Taauab which is the delay term in seconds rather than time steps. In this case the program automatically rounds to an integer number of time steps. Effective range is r_{ab} in the wave equation. The final parameter can be gamma or velocity in the wave equation. If the initial ϕ_{ab} is equal to the initial Q_{ab} as occurs in the steady state condition the propagator input can for example be given as

Propagator 2 - Initial Phi: Steady **Tauab:** 0 **Effective Range:** 0.08 **velocity:** 10.0.

2.3.2 Mapping

Propagator 5 - Tauab: 0

This propagator is the mapping propagator where spatial spreading is negligible. Its form is given by

$$\phi_{ab}(\mathbf{r}, t) = Q_b(\mathbf{r}, t - \tau_{ab}). \quad (4)$$

The input form has only one parameter Tauab (the delay term). i.e. number of time steps in the delay term. Optionally Tauab can be replaced by Taauab which is the delay term in seconds rather than time steps. In this case the program automatically rounds to an integer number of time steps.

2.3.3 Eqnset *2006*

This is the Robinson patchy propagator. It has not been currently tested.

2.3.4 Harmonic

This is a propagator implementing a harmonic oscillator of the form

$$\left[\frac{1}{\gamma_{ab}^2} \frac{d^2}{dt^2} + \frac{2}{\gamma_{ab}} \frac{d}{dt} + 1 \right] \phi_{ab}(\mathbf{r}, t) = Q_b(\mathbf{r}, t - \tau_{ab}). \quad (5)$$

The input form is given by

Propagator 2 - Initial Phi: 8.87145 **Tauab:** 420 **gamma:** 10.0,

where Tauab is $\tau_{ab}/\Delta t$ the delay term in the wave equation i.e. number of time steps in the delay term. Optionally Tauab can be replaced by Taauab which is the delay term in seconds rather than time steps. In this case the program automatically rounds to an integer number of time steps. An example of the latter form is

Propagator 2 - Initial Phi: 8.87145 **Tauabt:** 0.42 **gamma:** 10.0,

2.4 Coupling data

The coupling data is n lines of coupling data corresponding to each connection in the connection matrix. The coupling data should appear in the same order as in the connection matrix when the connection matrix is read top to bottom and then left to right. There are two types of coupling each with their own input form. A blank line finishes the coupling data section.

2.4.1 Simple

For simple synaptic coupling each Coupling data line contains a single parameter Nu e.g.
Coupling data 1 - Nu: 0.0012.

The parameter Nu (ν_{ab}) is the synaptic coupling parameter. It corresponds to the product of the mean synaptic strength s_{ab} and N_{ab} the mean number of connections from cells of type b to cells of type a .

2.4.2 Modcouple

The modulated synaptic coupling implements the Clearwater/Rennie modulation equation. The modulation is given by:

$$\nu(t) = \nu_0 \left[(1 - \nu_{scal}) e^{h(t)/k} + \nu_{scal} \right] \quad (6)$$

where $h(t)$ is the time low passed filtered form of the neuromodulators concentration $C(t)$ as given by

$$\left\{ \frac{1}{\mu\lambda} \frac{d^2}{dt^2} + \left[\frac{1}{\nu} + \frac{1}{\lambda} \right] \frac{d}{dt} + 1 \right\} h(t) = C(t). \quad (7)$$

The neuromodulator's concentration is in turn given by a user chosen stimulus form analogous to stimulus populations.

The remainder of the input form is specification of the output for ν . This takes an analogous form to the usual output data lines. The ν data is output to a file with filename "eegcode.synaptout.xx" where xx is an index number of the coupling.

An example Modcouple input form is given by

```
Coupling data 1 - Nuzero: 0.0002 Nuscal: 0.02 Mu: .1 Lambda: 1 k: .000001
Concentration mode: 1 Time to start of Concentration: .01 Amplitude: .0001
Pulse Duration: .2 Pulse repetition period: 40
Number of traces: 100 Single/All: All nodes
```

2.5 Output data

Currently the code output a time series of any particular node in any population. The first line of this section contains an optional preamble and a parameter of how many times series to be outputted e.g.

```
Output data - Number of traces : 4.
```

Following the first line each trace is specified on a separate line in the form:

```
Wave Equation Number :1 Single/All Single node number :1.
```

A short cut to output all the traces in a population is given the the input form

```
Wave Equation Number : 11 Single/All nodes: All nodes
```

3 Known bugs or flaws

- Code only runs for initial time set to zero. This only effects the stimulus routines.
- Inadequate range of stimulus patterns currently implemented. *AA*
- Documentation needs to be worked on.

- Code needs more commenting.