Romesh Abeysuriya 7/8/12

1 Noise and Neurofield

Correctly calculating the noise amplitude for NeuroField is critical when comparing the power spectrum in NeuroField to analytic predictions. In the analytic work, we work almost exclusively in Fourier space, and use white noise with

$$\phi_n(k,\omega) = C$$

where C is some value for the noise. The zero frequency, spatially uniform component $\phi_n([0,0],0) = 1$ is typically used when solving the steady state equation. The product $\nu_{sn}\phi_n$ determines the amplitude of noise going into the system. With ν_{sn} on the order of mV, we find $\phi_n(k,\omega)$ in the range of $1\times10^{-3} - 1\times10^{-2}\text{s}^{-1}$ for physiologically realistic power spectra (based on comparing spindle harmonic amplitude in experimental data with the analytic nonlinear spectrum).

In NeuroField, white noise is drawn from a Gaussian distribution at each node, at each point in time. There is a relationship between the mean and standard deviation of the Gaussian distribution used in NeuroField, and between $\phi_n^{(0)}$ and $\phi_n(k,\omega)$ in the analytics. The mean of the Gaussian distribution equals $\phi_n^{(0)}$. These notes are concerned with the other components $\phi_n(k,\omega)$ which are related to the standard deviation of the NeuroField Gaussian distribution.

The two central issues to deal with are firstly correctly normalizing the power spectrum, and secondly determining the correct noise amplitude in the time domain.

1.1 1D FFT Normalization

Correctly normalizing the output of the FFT routine in Matlab is critical to determining the correct noise amplitude. Correct normalization comes from Parseval's theorem which is essentially a statement of conservation of energy. For a 1D discrete signal $x(n)$ with N elements , if the Fourier components c_k are defined

$$c_k = \frac{1}{N}\sum_{k=0}^{N} x(n)e^{-2\pi ikn/N}$$

then Parseval's theorem for a discrete, periodic signal (which is the case when using a numerical FFT routine) refers to power:

$$\frac{1}{N}\sum_n |x(n)|^2 = \sum_k |c_k|^2$$

with units [V^2]. Physically, the LHS of this equation is the mean of the signal squared, which is the expectation value of the power. The summation of the RHS is the power in the Fourier domain. In Matlab, the built-in FFT function does not include the factor of $1/N$ when calculating c_k and therefore this normalization is achieved by dividing by the output of the FFT function by N

$$c_k = \frac{\text{FFT}(x)}{N}$$

The sum of the Fourier components as written above is a measure of the energy per period contained in the signal. The energy per period can be calculated from the power spectral density (which is predicted analytically in the model) by integrating over frequency. The power spectral density is therefore calculated from the Fourier components by dividing them by the spectral bin size Δf, which scales the power spectrum so that the integral provides the correct energy.

A simple test of this is the FFT of a sine wave. Suppose time series is defined by

$$x(t) = 3\sin(10\times 2\pi t)$$

which has an amplitude of 3V and a frequency of 10 Hz, with duration T and sampled at a rate f_s. The power per period is given by the average value of $|x(t)|^2$over an integer number of cycles and is 4.5V^2. Now, we expect that there should only be non-zero Fourier components at ± 10Hz, so we can deduce that these two components must each have a magnitude of 1.5 V s so that $\sum |c_k|^2 = 4.5$. Finally, dividing the components by

$$\Delta f = \frac{1}{T}$$

provides the power spectral density function (which we refer to as the power spectrum)

$$P(f) = \frac{|c_k|^2}{\Delta f}$$

with units [V^{2}s], which when integrated from the range of frequencies $-f_s/2$ to $f_s/2$ has the correct energy of 4.5V^2. This result is independent of the sampling rate f_s, although this was numerically verified with $f_s = 500$ Hz. The power spectrum includes the factor $1/\Delta f$ so that energy is obtained by integration of $P(f)$ rather than summation of $|c_k|^2$.

The end result of all this is that the output from Matlab's FFT function needs to be scaled by a factor $1/N$ in order to correctly calculate the Fourier components. After this, no additional scaling factors are required to calculate the power spectral density, only the frequency bin size which is uniquely defined by the sampling rate and the number of samples.

1.1.1 1D noise and sampling rate

We now consider the relationship between standard deviation and $\phi_n(\omega)$, in one dimension. The quantity that we require setting is the spectral power density $\phi_n^2(\omega)$ which is the term that appears in the transfer function used to calculate the power spectrum

$$P = \int \frac{d^2k}{(2\pi)^2} |T_{en}(\omega)|^2 |\phi_n(\omega)|^2$$

A critical step in the following derivation is considering $|\phi_n^2(\omega)|$ to be the power spectral density of the white noise.

Analytically, the Fourier transform of a stationary random process does not exist, so we obtain the power spectral density of the white noise using the Wiener–Khinchin theorem which states that the power spectral density of a stationary random process is equal to the Fourier transform of its autocorrelation function. However, this formulation assumes that the FFT has been defined

$$c_k = \sum_{k=0}^{N} x(n) e^{-2\pi i k n}$$

The autocorrelation of the signal $x(t)$ is given by

$$r(\tau) = E[x(t)x^*(t-\tau)]$$

For real valued $x(t)$ the complex conjugate $x^*(t)$ is simply $x(t)$ and white noise generated from a Gaussian distribution with standard deviation σ can be defined by its autocorrelation function which is

$$r(\tau) = \begin{cases} \sigma^2 & \tau = 0 \\ 0 & \tau \neq 0 \end{cases}$$

The autocorrelation is thus a delta function at the origin, and therefore power spectral density will have a value

$$P(f) = \sigma^2$$

over all frequencies. This theoretical result is of limited utility however, because it considers the ideal case where there is no sampling, infinite bandwidth, and the power spectrum has infinite energy when integrated over all frequencies. We can consider the discrete case using Parseval's theorem, from which we know that the sum of the discrete Fourier components c_k must equal the expectation value of the original function in the time domain. For white noise, the expectation value is is σ^2. We therefore have the equality

$$\sigma^2 = \int_{-f_s/2}^{f_s/2} P(f)df$$

But because white noise has the same power spectral density for all f, we obtain

$$P(f) = \phi_n^2(f) = \phi_{n,f}^2$$

the additional subscript f emphasising the difference between the constant-valued $\phi_n^2(\omega)$ and $\phi_{n,f}^2(f)$ which differ by a factor of 2π. Note that the reason these must differ by a factor of 2π is because the total energy in the power spectrum integrated over frequency or angular frequency must be the same $\int P(\omega)d\omega = \int P(f)df$. *We are taking $\phi_{n,f}^2$ to be the hypothetical constant value of $P(f)$ that satisfies Parseval's theorem, which we then solve for, consistent with the earlier claim that $|\phi_n^2(\omega)|$ is the power spectral density of the noise.* We thus obtain

$$\begin{aligned}
\sigma^2 &= \int_{-f_s/2}^{f_s/2} \phi_{n,f}^2 df \\
&= \sum_N \phi_{n,f}^2 \Delta f \\
&= N\phi_{n,f}^2 \Delta f \\
&= \phi_{n,f}^2 f_s
\end{aligned}$$

with

$$\Delta f = \frac{f_s}{N}$$

Thus, for a particular choice of standard deviation, the resulting power spectral density depends on the sampling rate, such that the power (mean squared voltage) of the noise is conserved. This dependence arises because the bandwidth of the noise depends on the sampling rate.

Finally, we need to compute the value of $\phi_{n,f}^2$ based on $\phi_n(\omega)$. When a plot of the power spectrum is made in Matlab, dividing by the transfer function gives a constant value for the noise. This constant value is $\phi_n^2(\omega)$ in the analytic solution. This can be converted to $\phi_n^2(f)$ by multiplying by 2π (because $df = \frac{d\omega}{2\pi}$). Also, this value corresponds to the single sided power spectral density. In NeuroField, we require the full power spectrum including negative frequency components. Therefore, the value of $\phi_n^2(f)$ must be halved because each frequency component is doubled for the single sided spectrum. Thus, $\phi_{n,f}^2 = \pi\phi_n^2(\omega)$.

Therefore, once we have chosen $\phi_n^2(\omega)$ we are able to calculate the appropriate noise amplitude in NeuroField using the relation

$$\begin{aligned}
\sigma &= \sqrt{f_s \pi \phi_n^2(\omega)} \\
&= \sqrt{\frac{\pi\phi_n^2(\omega)}{\Delta t}}
\end{aligned}$$

From this relation, we can see that

- For the same sampling rate, increasing σ corresponds to increasing $\phi_n^2(\omega)$ (as expected).

- For the same value of σ, if the sampling rate is increased, the corresponding value of $\phi_n^2(\omega)$ is smaller. This means that the power spectral density will be smaller, although the total power will be the same, because there will be a larger bandwidth in the Fourier transform.
- Similarly, if the sampling rate is decreased, the corresponding value of $\phi_n^2(\omega)$ will be larger, and the power spectral density will have larger values at each frequency bin.

1.1.2 Cortex only network

Consider a network consisting of

- Excitatory population of neurons with a wave propagator, $\gamma = 116$, $r_e = 0.0086$, and dimensions 50 x 50cm
- An inhibitory population of neurons for which the local inhibition approximation applies
- A population of stimulus neurons directly connected to both cortical populations

Connectivities are the same as in the EIRS model ($\nu_{ee} > 0$,$\nu_{ei} < 0$, matched to $\nu_{ie}$and ν_{ii} by the random connectivity assumption) and with $\alpha = 83$s^{-1} and $\beta = 770$s^{-1}. The steady state voltage is given by solutions to

$$V_e - (\nu_{ee} + \nu_{ei})S(V_e) - \nu_{en}\phi_n^{(0)} = 0$$

and the power spectrum is calculated as in the EIRS case but with cortical transfer function

$$T_{en}(k,\omega) = \frac{LG_{en}}{\left[\left(1 - \frac{i\omega}{\gamma_e}\right) + k^2 r_e^2\right](1 - LG_{ei}) - LG_{ee}}$$

$$L = \left(1 - \frac{i\omega}{\alpha}\right)^{-1}\left(1 - \frac{i\omega}{\beta}\right)^{-1}$$

In the 1D case, $k = 0$ and we only use the single, spatially uniform mode. The power spectrum is given by

$$P(\omega) = |T_{en}(0,\omega)|^2|\phi_n(\omega)|^2$$

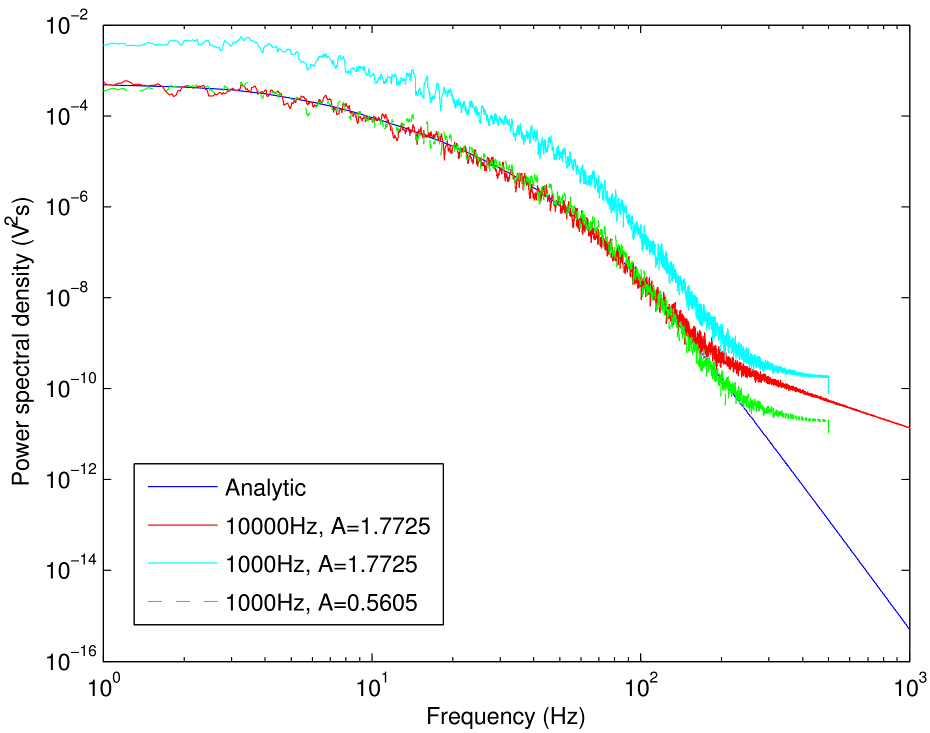
For $\phi_n(\omega) = 1 \times 10^{-2}$s^{-1} we can calculate the corresponding standard deviation for the noise in NeuroField using the previously derived correspondance. At a sampling rate of 10000 Hz, we find

$$\sigma = \sqrt{f_s \pi \phi_n^2(\omega)} = 1.7725$$

Alternatively, if we sample at 1000 Hz, we find

$$\sigma = \sqrt{f_s \pi \phi_n^2(\omega)} = 0.5605$$

The figure below compares the power spectrum calculated analytically with $\phi_n(\omega) = 1 \times 10^{-2}$s^{-1} compared with NeuroField using the above noise amplitud and sampling rates. In this cortical network, the parameter values used were $\nu_{ee} = 1$ mVs, $\nu_{ei} = -1.5$ mVs, and $\nu_{en} = 1$ mVs. The FFT has been taken using only the scaling factors arising from Parseval's theorem, and the results both predict the same power spectral density. This verifies the method of calculating the noise amplitude from $\phi_n(\omega)$ is correct in the 1D case. A 15 point moving average smoothing has been used to reduce noise in the spectrum.



1.2 2D and 3D FFT normalization

We now consider the generalization of FFT normalization into the 2D and 3D cases, in the simple case of 2D and 3D sinusoids defined by

$$A(x,y) = 3\sin(10 \times 2\pi x + 5 \times 2\pi y)$$

$$A(x,y,z) = 3\sin(10 \times 2\pi x + 5 \times 2\pi y + 20 \times 2\pi z)$$

Note that the power in these signals is each still 4.5V^2, the mean value of the signal squared over the grid. Consider the coordinates in each direction to be

$$x, y, z = 0 : 0.002 : 0.9980$$

which corresponds to $T = 1$ s,$N = 500$, $f_s = 500$ Hz. Suppose we label the number of points in each direction of the grid as M, N, P so that the 2D case has $M \times N$ points, and the 3D case has $M \times N \times P$ points (with $M, N, P = 500$ in this case). To satisfy Parseval's theorem, we need to divide the Fourier components by the number of points in the transform. This provides

$$\frac{1}{MN}\sum_m \sum_n |A_{mn}|^2 = \sum_m \sum_n |c_{mn}|^2 = 4.5\mathrm{V}^2$$
$$\frac{1}{MNP}\sum_m \sum_n \sum_p |A_{mnp}|^2 = \sum_m \sum_n \sum_p |c_{mnp}|^2 = 4.5\mathrm{V}^2$$

which is a simple extension of the discrete form of Parseval's theorem used in the 1D case- the mean squared power in the spatial and temporal domain is equal to the sum of the squared magnitude of the Fourier components in the Fourier domain, and the Fourier components are calculated in Matlab by dividing by the total number of points in the transform.

Now, the frequency resolution is given by $1/T$ which in this case is 1 Hz. As there is a single sine wave in the original signal, there will be only two nonzero Fourier components, which have magnitude 1.5 V s for the same reason as discussed in the 1D case. Similarly, if we examine the grid of Fourier components, the zero frequency component corresponds to the first row and column of the matrix. Therefore in Matlab, if we represent the matrix of Fourier components as S, the nonzero components are located at

$$|S_{2D}(1+5,1+10)| = 1.5$$
$$|S_{3D}(1+5,1+10,1+21)| = 1.5$$

Interestingly, note that the X and Y coordinates are swapped- the rows of the matrix correspond to Y and the columns correspond to X. The conversion from Fourier components to power spectral density progresses in exactly the same way as in 1D. Because integration requires multiplying the power spectral density by the bin size and summing, the power spectral density is retrived from the Fourier components by dividing by the bin size. The bin size can be specified in angular units or in regular units, as long as a consistent choice is made for the variable being integrated over. If the integration is performed immediately, and the power spectral density is not explicitly examined, then it is sufficient to simply sum the Fourier components.

The end result is that Parseval's theorem is satisfied in Matlab as a direct generalization of the 1D case. The same as in 1D, the Fourier components need to be divided by the number of points in the Fourier transform. Similarly, to convert from Fourier components to power spectral density, it is necessary to divide by the size of the bin in whichever dimension integration will be performed in order to recover the correct power.

1.2.1 3D noise and sampling rate

The same problem exists when considering spatial sampling. The size of the cortex is assumed to be 50 x 50cm, which is discretized in NeuroField when solving the wave equation for propagation of ϕ_e in the cortex. NeuroField allows the user to pick the number of nodes in each direction along the grid, subject to the constraints

- CFL condition must be met, where $\Delta t < \frac{\Delta x}{v}$ to ensure that the timestep is small enough to propagate waves with velocity v
- $\Delta x < \frac{r_e}{2}$ which ensures that the spatial discretization is small enough relative to the axon length

However, an additional effect is that the noise amplitude depends on the number of nodes. As shown in the figure below, as the number of nodes increases, the effective amount of noise $\phi_n(k,\omega)$ decreases, in the same way that $\phi_n^2(\omega)$ decreased as the sampling rate increased for the same standard deviation of noise in NeuroField. This is in fact exactly the same scenario, because when the physical size of the cortex is fixed, changing the number of nodes alters the spatial sampling frequency (samples per meter). The challenge is therefore to relate $\phi_n^2(k,\omega)$ to the temporal sampling rate, and the number of nodes.

We can proceed with the 3D case in exactly the same way as was used in the 1D case. We start with Parseval's theorem, noting that the mean squared power of the noise in 3D is still σ^2, the same as in the 1D case. Parseval's theorem states

$$\sigma^2 = \int_f \int_{k_x} \int_{k_y} P(f,k_x,k_y)\, df\, dk_x\, dk_y$$

and supposing $\phi_{n,f}^2$ is the hypothetical constant value of $P(f,k_x,k_y)$,converting this into a sum as from before gives us

$$
\begin{aligned}
\sigma^2 &= \sum_{M,N,P} \phi_{n,f}^2 \Delta f \Delta k_x \Delta k_y \\
&= MNP\phi_{n,f}^2 \frac{1}{P\Delta T}\frac{2\pi}{M\Delta x}\frac{2\pi}{N\Delta y} \\
&= \frac{4\pi^2\phi_{n,f}^2}{\Delta T \Delta x \Delta y}
\end{aligned}
$$

where the angular wave number bin size is given by multiplying the wave number bin size by 2π as

$$\Delta k = \frac{2\pi}{L_x} = \frac{2\pi}{M\Delta x}$$

which is directly analogous to computing the frequency bin size. Once again we need to perform the conversion from $\phi_n^2(k,\omega)$ to $\phi_{n,f}^2$ but notice that again, we accumulate a single factor of 2π because we have bins Δf and Δk (and so we only need to convert from f to ω, with k remaining unchanged). We also only consider the single sided spectrum in the temporal frequency direction- we retain all of the wave number components separately and sum over all of them when computing the power spectral density $P(f)$. Therefore we retain the relationship $\phi_{n,f}^2 = \pi\phi_n^2(\omega)$. Finally then, we obtain

$$\sigma = \sqrt{\frac{4\pi^3\phi_n^2(\omega)}{\Delta T \Delta x \Delta y}}$$

1.2.2 Cortex only network (3D)

We now consider the same cortical-only network, only now allowing for spatial variation. This was achieved by running NeuroField with 400 nodes, such that $\Delta x = 0.025$m. For a noise amplitude of $\phi_n(k,\omega) = 3 \times 10^{-5}$ the above formula provides $\sigma = 1.3364$. Numerical integration was performed using NeuroField in exactly the same way as the 1D case, and the power spectral density was evaluted consistent with the scalings required by Parseval's theorem (dividing by total number of elements, dividing by the frequency bin size, summing over angular wavenumber so no division by Δk was required).

When different spatial modes are considered, the formula for the analytic power spectrum includes a summation over the spatial modes

$$
\begin{aligned}
P(\omega) &= \int\int dk^2 |T_{en}(k,\omega)|^2 |\phi_n(\omega)|^2 \\
&= \sum_{m,n} |T_{en}(k_{m,n},\omega)|^2 |\phi_n(\omega)|^2 \Delta k^2
\end{aligned}
$$

Note that the transfer function only depends on the value of k_{mn}^2 (which appears in the denominator of T_{en}) and the values of k_{mn}^2 are given by

$$k_{mn}^2 = \left(\frac{2\pi m}{L_x}\right)^2 + \left(\frac{2\pi n}{L_y}\right)^2$$

The numerical result was compared to the analytic result for two different numbers of spatial modes- first considering spatial modes up to $|m,n| = 2$ and second considering modes up to $|m,n| = 4$. This corresponds to having 25 and 81 terms respectively in the sum for $P(\omega)$. The figure below compares the analytic result and the numerical result, with no additional scaling factors. The results agree well in terms of both shape and total power, and furthermore both approaches predict the same changes to the power spectrum when the number of wave modes included changes. This result verifies that the method of calculating the noise amplitude from $\phi_n(k,\omega)$ is valid in the 3D case with both spatial and temporal variations.

Interestingly, it is worth noting that moving average smoothing was not required to produce these results-summation over several different modes also has the effect of reducing noise in the spectrum. Segmenting the data in the time direction and averaging multiple segments together (directly analogous to the 1D case) provides a further reduction in noise at the expense of frequency resolution, without the arbitrary nature of a moving average.