**Supplementary Material**

Hideyuki Câteau and Alex Reyes

*Derivation of the two dimensional Fokker-Planck equation and numerical algorithm to integrate it*

A set of Langevin equations accounting for the non-trivial autocorrelation is given as

\[
\begin{align*}
\frac{du}{dt} &= a(u, R) + b(R') (w(t) - \alpha X), \\
\tau_{neg} \frac{dX}{dt} &= -X + w(t).
\end{align*}
\]

Note that the same white noise, \(w(t)\), appears both in \(u\)- and \(X\)-equations. For a later numerical algorithm, it is convenient to transform this set of equations to a diagonalized form in which \(w(t)\) appears in one of the equations. With new variables, \(x = \tau_{neg} b(R') X\) and \(y = u - x\), the equation is diagonalized as

\[
\begin{align*}
\frac{dy}{dt} &= a(y + x, R) + \frac{1 - \alpha}{\tau_{neg}} x, \\
\frac{dx}{dt} &= -x / \tau_{neg} + b(R') w(t).
\end{align*}
\]

(S1)

In this diagonalized form, the corresponding Fokker-Planck equation is straightforwardly derived:

\[
\frac{\partial P(x, y, t)}{\partial t} = -\frac{\partial}{\partial y} \left( a(y + x, R) + \frac{1 - \alpha}{\tau_{neg}} x \right) P + \frac{1}{\tau_{neg}} \frac{\partial}{\partial x} \left( b(R')^2 \right) P + \frac{1}{2} \frac{\partial^2 P}{\partial x^2}.
\]

(S2)

To integrate this equation without developing numerical instability, we need to keep all the coefficients of the second order derivatives to be reasonably large. However, the coefficient to \(\partial^2 / \partial x^2\) can vanish since it is proportional to a firing rate of the previous layer, see Fig. 2. To guarantee this coefficient to be lower-bounded, we add Gaussian white noise, \(b(R_i) w_i(t)\), to the second one of Eq.(S1). This new Gaussian white noise is independent of \(w(t)\). As for the \(y\) variable, there is no second derivative term, which would destabilize the numerical integration. To evade this, we add Poisson background noise at low rate, \(R_y\), to the \(y\)-equation, which is approximated well by...
\[ a(y + x, R_0) + b(R_0) w_0(t) \, . \]

With these additional sources of noise included, the Fokker-Planck equation becomes

\[
\frac{\partial \bar{P}(x,y,t)}{\partial t} = -\frac{\partial}{\partial y} \left( a(y + x, R + R_0) + \frac{1-\alpha}{\tau_{neg}} x \right) \bar{P} + \frac{b \left( R_0 \right)^2}{2} \frac{\partial^2 \bar{P}}{\partial y^2} + \frac{1}{\tau_{neg}} \frac{\partial (x \bar{P})}{\partial x} + \frac{b \left( R' + R_1 \right)^2}{2} \frac{\partial^2 \bar{P}}{\partial x^2}.
\]

The rates of the background noise is set to low enough values \( R_0 = R_1 = N \times 3 \text{Hz} \) compared to a typical firing rate for a signal \( (N \times 37 \text{Hz}) \) to preserve the original behavior of the system.

This diagonalized form is useful in applying the ADI algorithm for the numerical integration. For a numerical integration, we first discretize the x-y space with the same spacing for both directions, \( \Delta x = \Delta y = h_1 \), and discretize time as \( t = n h_0 \). Suppose that we have probability values at discretized coordinates, i, j, at discretized time, n: \( P^n_{ij} = \bar{P}(ih_1, jh_1, nh_0) \), and try to obtain those values at the next time step, n+1. We need to use difference operators in either x or in y but not a mixed difference operator, since no cross-derivative \( \partial^2 / \partial x \partial y \), is involved. Generally, in the ADI algorithm, \( P^n_{ij} \)'s are updated in two time steps, first from n to n+1/2, then from n+1/2 to n+1:

\[
\begin{align*}
P_{ij}^{n+1/2} - P_{ij}^n &= A_y P_{ij}^{n+1/2} + A_x P_{ij}^n, \\
P_{ij}^{n+1} - P_{ij}^{n+1/2} &= A_y P_{ij}^{n+1/2} + A_x P_{ij}^{n+1},
\end{align*}
\]  \tag{S3}

with \( A_x \) and \( A_y \) being difference operators in x and y.

Note that a special combination of time indices, n, n+1/2 and n+1, are taken on the right-hand sides. This particular combination, which leads to the same continuum limit as many other combinations, was taken here to be able to calculate \( P_{ij}^{n+1} \) from \( P^n_{ij} \) by solving two linear algebraic equations consecutively:
\[
\begin{align*}
(1 - A_y) P_{ij}^{n+1/2} &= (1 + A_y) P_{ij}^n, \\
(1 - A_x) P_{ij}^{n+1} &= (1 + A_x) P_{ij}^{n+1/2}.
\end{align*}
\]

In order to account for the boundary condition correctly, Eq.(S3) must be supplemented with matrices determined by probability conservation law, \( \sum_{j} P_{ij}^{n+1} = \sum_{j} P_{ij}^{n} \):

\[
\begin{align*}
P_{ij}^{n+1/2} &= P_{ij}^n + A_y P_{ij}^{n+1/2} + (A_x + L_2 + L_{2y}) P_{ij}^n, \\
P_{ij}^{n+1} &= P_{ij}^{n+1/2} + (A_y + 2 L_{2y}) P_{ij}^{n+1/2} + (A_x + L_2 + L_{2y}) P_{ij}^{n+1}.
\end{align*}
\] (S4)

The domain where \( P \) is defined has been transformed from a rectangular one specified by \( u_{\text{lower limit}} \leq u \leq u_{\text{threshold}} \) and \( -X_{\text{limit}} \leq X \leq X_{\text{limit}} \), to a parallelogram because of the diagonalization. One of the four boundaries corresponds to the firing threshold, so the probability flux through it needs to be re-injected into the line representing the reset potential. Matrices, \( L_{2y} \) and \( L_{2y} \), perform this re-injection. They are non-zero only on this reset line and have values, \( L_{2y} P_{ij} = \frac{h_0}{2} \left( K_{r_0} (R + R_0) + g_y i + g_y \right) P_{ij} \) and

\[
L_{2y} P_{ij} = \frac{h_0}{2} \left( g_{\text{rate}} - \frac{r_0}{2} (i - 1) \right) P_{ij}, \quad \text{where} \quad K_{r_0}(R) = \frac{1}{h_1} a \left( h_1 i, R \right), \quad r_0 = \frac{1}{\tau_{\text{seg}}},
\]

\[
g_{\text{rate}} = \frac{1}{2 h_1^2} b \left( R_{0} \right)^2 \text{ and } g_y = \frac{1}{\tau_{\text{seg}}} (1 - \alpha). \quad \text{Exceptionally, at the lower bound of } i \]

\((i = -m_0), \quad L_{2y} \) assumes different value: \( L_{2y} P_{ij} = \frac{h_0}{2} g_{\text{rate}} P_{ij} \).

Matrix, \( A_x \), acts on i index only. Its (i,i+1), (i,i) and (i,i+1) components are non-vanishing, and are respectively given as

\[
A_x = \left( h_0 \left( g_{\text{rate}} - \frac{r_0}{2} (i - 1) \right), -\frac{h_0}{2} g_{\text{rate}}, \frac{h_0}{2} \left( g_{\text{rate}} + \frac{r_0}{2} (i + 1) \right) \right)
\]

with \( g_{\text{rate}} = \frac{b (R^t + R_1)^2}{2 h_1^2} \). Similarly, \( A_y \) acting on j index has non-zero components of

\[
A_y = \left( \frac{h_0}{2} \left( K_{r_1} (R + R_0) + g_x i + g_y \right), \frac{h_0}{2} \left( -K_{r_1} (R + R_0) - g_x i - 2 g_y \right), \frac{h_0}{2} g_y \right).
\]

The other three boundaries were introduced only to make the lattice size finite. Those boundaries are positioned far enough from the origin of the coordinates to guarantee the fluxes through them very small. The small fluxes through them are re-injected into
the point where they were ejected (reflecting boundary condition). $L_2$ represents the re-injection at these three boundaries and it assumes a non-zero value only there. On the lines representing upper and lower bounds of $i$ ($i = \pm m_0$),

$$L_2 P_{ij} = \frac{\hbar}{2} \left( g_{rate} - \frac{r_j m_0}{2} \right) P_{ij}$$

unless $j$ is also on the lower or upper bounds. For $j$ being the upper and lower bounds,

$$L_2 P_{ij} = \frac{\hbar}{2} g_{rate} P_{ij}$$

unless both $i$ and $j$ are on the lower bounds. On the rest of the boundaries,

$$L_2 P_{ij} = \frac{\hbar}{2} \left( g_{rate} + \frac{r_j i}{2} \right) P_{ij}$$

unless $i = m_0$.

With all the matrices ready, we can solve Eq.(S4) iteratively to determine the temporal evolution of the probability distribution and firing rate.