Journal of Computational Neuroscience 8, 19–50 (2000)

A Population Density Approach That Facilitates Large-Scale Modeling of Neural Networks: Analysis and an Application to Orientation Tuning

DUANE Q. NYKAMP
Courant Institute of Mathematical Sciences, New York University, New York, NY 10012
nykamp@cims.nyu.edu

DANIEL TRANCHINA
Department of Biology, Courant Institute of Mathematical Sciences, and Center for Neural Science, New York University, New York, NY 10003
tranchin@cims.nyu.edu

Received December 29, 1998; Revised March 15, 1999; Accepted April 30, 1999

Action Editor: John Rinzel

Abstract. We explore a computationally efficient method of simulating realistic networks of neurons introduced by Knight, Manin, and Sirovich (1996) in which integrate-and-fire neurons are grouped into large populations of similar neurons. For each population, we form a probability density that represents the distribution of neurons over all possible states. The populations are coupled via stochastic synapses in which the conductance of a neuron is modulated according to the firing rates of its presynaptic populations. The evolution equation for each of these probability densities is a partial differential-integral equation, which we solve numerically. Results obtained for several example networks are tested against conventional computations for groups of individual neurons.

We apply this approach to modeling orientation tuning in the visual cortex. Our population density model is based on the recurrent feedback model of a hypercolumn in cat visual cortex of Somers et al. (1995). We simulate the response to oriented flashed bars. As in the Somers model, a weak orientation bias provided by feed-forward lateral geniculate input is transformed by intracortical circuitry into sharper orientation tuning that is independent of stimulus contrast.

The population density approach appears to be a viable method for simulating large neural networks. Its computational efficiency overcomes some of the restrictions imposed by computation time in individual neuron simulations, allowing one to build more complex networks and to explore parameter space more easily. The method produces smooth rate functions with one pass of the stimulus and does not require signal averaging. At the same time, this model captures the dynamics of single-neuron activity that are missed in simple firing-rate models.

Keywords: neural networks, modeling, population density, orientation tuning, visual cortex

1. Introduction

The ability to model realistic networks of neurons on the computer is severely restricted by the computer time required. Even a simple model of a small part of the brain could contain tens of thousands of neurons and hundreds of thousands of synapses. Although computers are rapidly growing faster, these simple models are still being constrained by available computational time; one is forced to cut corners by reducing the numbers of neurons and synapses. Ideally, one would like to enlarge these models to more closely approximate the networks in the brain. This presently is not feasible since these computations would require unrealistic amounts of computer time.

The population density approach addresses this issue in two ways. First, it is a computationally efficient method for simulating large networks of neurons. Second, it is a method where one does not need to cut corners in the numbers of neurons and synapses simulated. One can simulate networks that are, in effect, of unlimited size.

Our approach is based on ideas and methods introduced by Knight, Manin, and Sirovich (1996) and further developed in Omurtag, Knight, and Sirovich (2000). Although theories based on the notion of a probability-density function have a long history in theoretical neuroscience (Wilbur and Rinzel, 1983; Kuramoto, 1991; Abbott and van Vreeswijk, 1993; Gerstner, 1995), the particular way of applying this theory to model interactions of large populations of sparsely connected neurons introduced by Knight and colleagues (Knight et al., 1996; Omurtag et al., 2000; Sirovich et al., 1999; Knight, 2000) appears to be novel. A similar approach has been taken recently by Brunel and Hakim (1999).

In the population density approach, integrate-and-fire neurons are grouped into large populations of similar neurons. For each population, we form a probability density that represents the distribution of neurons over all possible states. In the simple version of this model presented here, the state of a neuron is completely described by its voltage.

We derive partial differential-integral equations for the evolution of population density functions of coupled populations and corresponding population firing rates. We also develop a reduction of the equations to a diffusion equation that produces similar results over a wide range of parameters.

We compare the results of a simple network of population densities with a similarly organized network of many individual integrate-and-fire neurons. We find that the firing rates of population density simulations closely match the average firing rates of individual neuron populations when the individual neuron populations contain sufficient numbers of neurons. The population density results were obtained in a fraction of the time required for the individual neuron simulations.

As a demonstration of the population density approach, we apply the method to modeling orientation tuning in visual cortex neurons. Our population density model is based on the recurrent feedback model of a hypercolumn in cat visual cortex of Somers et al. (1995). We simulate the response to flashed oriented bars. As in the Somers model, a weak orientation bias provided by feed-forward LGN input is transformed by intracortical circuitry into sharper orientation tuning that is independent of stimulus contrast. We demonstrate the speed at which the simulation runs and the similarity of the population density results to those of an equivalent individual neuron network.

In Section 2 we outline the integrate-and-fire point neuron model that underlies our population density formulation. In Section 3 we introduce the population density approach and derive the population density evolution equations. We present in Sections 4 and 5 the results of single population simulations and simple population network simulations, respectively. We demonstrate a population density model of orientation tuning in Section 6. We discuss the results in Section 7. Details on our diffusion approximation, numerical methods, and parameters are presented in the appendices.

2. The Integrate-and-Fire Neuron

Our implementation of the population density approach is based on an integrate-and-fire point (single compartment) neuron. Although the approach could be generalized to other neuron models, (see Omurtag et al., 2000; Knight, 2000), the population density based on an integrate-and-fire neuron is low-dimensional and thus can be computed efficiently. In this presentation, we make a further simplification that the synaptic time courses are fast, which will allow us to form a one-dimensional population density.

2.1. *General Model of an Integrate-and-Fire Neuron*

The integrate-and-fire point neuron, schematized by its equivalent circuit diagram in Fig. 1, contains three conductances: a fixed resting conductance g_r, a time-varying excitatory conductance $g_e(t)$, and a time-varying inhibitory conductance $g_i(t)$. When the conductances are nonzero, they draw the transmembrane voltage $V(t)$ of the neuron toward their respective equilibrium potentials ($\mathcal{E}_r$, $\mathcal{E}_e$, and $\mathcal{E}_i$, where $\mathcal{E}_i < \mathcal{E}_r < \mathcal{E}_e$).

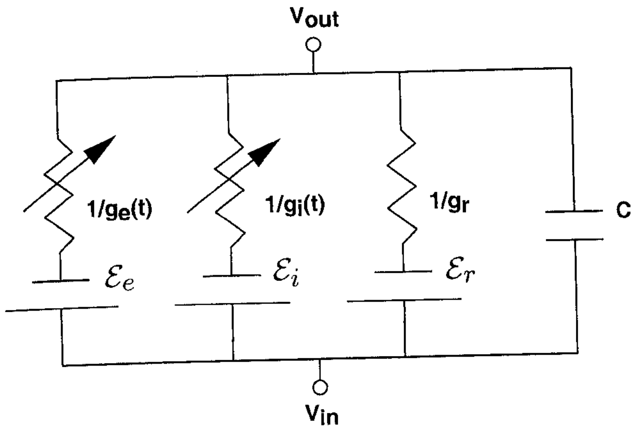

Figure 1. The equivalent circuit of an integrate-and-fire point neuron. The transmembrane voltage of the neuron is $V(t) = v_{in} - v_{out}$, where v_{in} is the voltage inside the neuron and v_{out} is the voltage outside the neuron. C is the capacitance of the neuron's membrane. The neuron has three ion channels modeled as a resistor and battery in series. The resting conductance g_r is fixed, while the excitatory and inhibitory conductances, $g_e(t)$ and $g_i(t)$ are time-varying functions of the neuron's input. $\mathcal{E}_r$, $\mathcal{E}_e$, and $\mathcal{E}_i$ are the resting, excitatory, and inhibitory equilibrium potentials, respectively, with $\mathcal{E}_i < \mathcal{E}_r < \mathcal{E}_e$.

The evolution of V in time t is specified by the equation

$$\tau \frac{dV}{dt} + (V - \mathcal{E}_r) + G_e(t)(V - \mathcal{E}_e) + G_i(t)(V - \mathcal{E}_i) = 0, \tag{1}$$

where $\tau = C/g_r$ is the membrane time constant, C is the membrane capacitance, and $G_{e/i}(t) = g_{e/i}(t)/g_r$. Equation (1) holds as long as $V(t) < v_{th}$, where v_{th} is a fixed threshold voltage, and $\mathcal{E}_r < v_{th} < \mathcal{E}_e$.

When the voltage of a neuron reaches v_{th}, the neuron is said to fire a spike. The output of the neuron is the set of times at which the neuron spikes. After each spike, the voltage of the neuron is reset to the reset voltage v_{reset}, $\mathcal{E}_i < v_{reset} < v_{th}$. In this way, the voltage remains in the range $\mathcal{E}_i < V(t) < v_{th}$, provided it starts in that range.

2.2. *Synaptic Input*

The inputs to an integrate-and-fire neuron are the times of excitatory and inhibitory synaptic inputs, which determine the synaptic conductances, $g_e(t)$ and $g_i(t)$. When a neuron is at rest, the synaptic conductances are zero. An excitatory or inhibitory synaptic input transiently increases the corresponding synaptic conductance.

To simplify the population density equations, we assume that the synaptic time courses are fast—that is, much shorter than the membrane time constant τ. We can then replace the conductance change caused by a single synaptic input with a delta function of magnitude equal to the integral of the original conductance change. Using normalized conductances as in (1), an excitatory/inhibitory synaptic input at time $T^k_{e/i}$ will produce a conductance change similar to a delta function of magnitude $A^k_{e/i}$ given by

$$A^k_{e/i} = \int \hat{G}^k_{e/i}(t)\,dt, \tag{2}$$

where $\hat{G}^k_{e/i}(t)$ is the change in $G_{e/i}(t)$ due to the synaptic input at time $T^k_{e/i}$.

With this approximation, the voltage jumps when a neuron receives synaptic input. Solving Eq. (1) for the time interval from immediately preceding to immediately following the synaptic input $(T^{k-}_{e/i}, T^{k+}_{e/i})$, we calculate that the jump size $\Delta v = V(T^{k+}_{e/i}) - V(T^{k-}_{e/i})$ is

$$\Delta v = \left(1 - e^{-\Gamma^k_{e/i}}\right)\left[\mathcal{E}_{e/i} - V\left(T^{k-}_{e/i}\right)\right], \tag{3}$$

where $\Gamma^k_{e/i} = A^k_{e/i}/\tau$. The size of the voltage jump is proportional to

$$\Gamma^{*k}_{e/i} = 1 - \exp\left(-\Gamma^k_{e/i}\right), \tag{4}$$

which depends on the size of the conductance change. Note that in the limit of a very large synaptic conductance change $\Gamma^k_{e/i} \to \infty$ and $\Gamma^{*k}_{e/i} \to 1$. In this limit, $V(T^{k+}_{e/i}) \to \mathcal{E}_{e/i}$.

An example of the evolution of an integrate-and-fire neuron with delta function conductance change is shown in Fig. 2. An important consequence of our delta function approximation is that the state of a neuron is completely determined by its voltage $V(t)$, $\mathcal{E}_i < V(t) < v_{th}$. We do not need to track the state of its synaptic conductances.

2.3. *Two Sources of Randomness*

We introduce two sources of randomness in our model: the arrival times of synaptic inputs and the size of synaptic conductance changes.

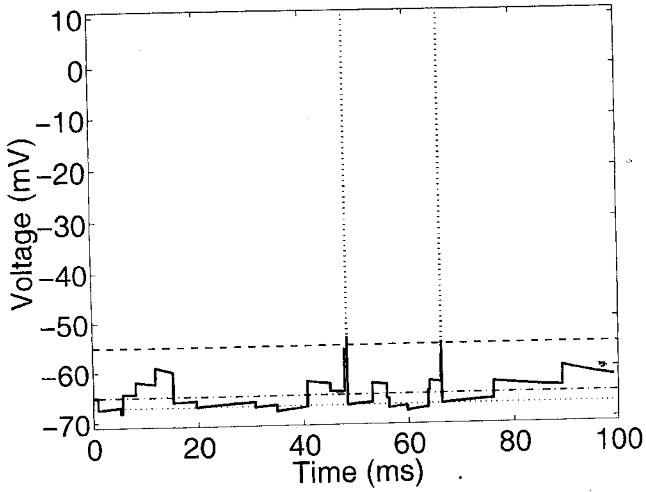

Figure 2. The evolution of the voltage of an integrate-and-fire neuron with delta function unitary synaptic conductances. Between synaptic inputs, the voltage decays toward $\mathcal{E}_r$ (dot-dash line). The voltage jumps upward or downward on receiving excitatory or inhibitory synaptic input, respectively. When the voltage reaches v_{th} (dashed line), the neuron is said to have fired a spike and its voltage is reset to v_{reset} (horizontal dotted line). The dotted vertical lines are shown just to illustrate these spike times. The arrival times of the synaptic input and the size of the resulting voltage jumps are random. The jump sizes are shown atypically large for illustration. Parameters used were $\mathcal{E}_i = -70$ mV, $\mathcal{E}_r = -65$ mV, $\mathcal{E}_e = 0$ mV, $v_{reset} = -67$ mV, $v_{th} = -55$ mV, and $\tau = 20$ ms.

In this model, only the synaptic input rates, which we denote $\nu_e(t)$ and $\nu_i(t)$, are specified. In general, these rates are determined by the firing rates of presynaptic neurons. The precise arrival times of the synaptic inputs, $T^k_{e/i}$, are assumed to be given by a modulated Poisson process with mean rate $\nu_{e/i}(t)$.

We let the size of the synaptic conductance changes, $\Gamma^k_{e/i}$, be random numbers with some given distribution. Thus the $\Gamma^{*k}_{e/i}$ are also random numbers with a distribution function determined by that of $\Gamma_{e/i}$. We define the complementary cumulative distribution function for $\Gamma^*_{e/i}$,

$$\tilde{F}_{\Gamma^*_{e/i}}(x) = \Pr\left(\Gamma^*_{e/i} > x\right), \tag{5}$$

which is some given function of x that depends on the chosen distribution of $\Gamma^k_{e/i}$. The derivation of the model is independent of the choice of $\tilde{F}_{\Gamma^*_{e/i}}(x)$. The choice of $\tilde{F}_{\Gamma^*_{e/i}}(x)$ for our simulations is given in Appendix D.

With these sources of randomness, a neuron will respond differently to multiple presentations of the same input rates. If one repeated the presentation of the same input many times and analyzed the neuron's response, one could map out the relative frequency that a neuron is in a small neighborhood of a given voltage at each time. The relative frequency per unit voltage given by this procedure would be an estimate of the probability density function of the neuron. Figure 3 illustrates the relationship between the individual responses of an integrate-and-fire neuron and its probability density function.

3. The Population Density Approach

The direct calculation of the probability-density function forms the basis of the population-density approach. We ignore the trajectories of individual neurons and instead model the evolution of a probability density that is defined by[1]

$$\rho(v,t)\,dv = \Pr(V(t) \in (v, v+dv)), \quad \mathcal{E}_i \le v \le v_{th}. \tag{6}$$

The key to modeling large neural networks with this approach is the observation that for a population of many similar neurons, this probability density can be interpreted as a *population density* (Knight et al., 1996; Omurtag et al., 2000; Kuramoto, 1991; Abbott and van Vreeswijk, 1993; Gerstner, 1995):

$$\rho(v,t)\,dv = \text{Fraction with } V(t) \in (v, v+dv). \tag{7}$$

Thus, for a fixed time t, the population density $\rho(v,t)$ describes the distribution of neurons over all possible states. We emphasize that, since the state of our model neuron is completely determined by its voltage, this distribution is described by the one dimension of voltage.

3.1. *The Assumptions of the Population Density Approach*

The population density interpretation assumes that there are a large number of similar neurons in each population. These neurons have the same biophysical properties, which in the current model means the same capacitance, resting conductance, and distribution of synaptic conductances. This assumption is more restrictive than the more general view presented in Omurtag et al. (2000), where the population density is based explicitly on the notion of an ensemble average (of replica systems).

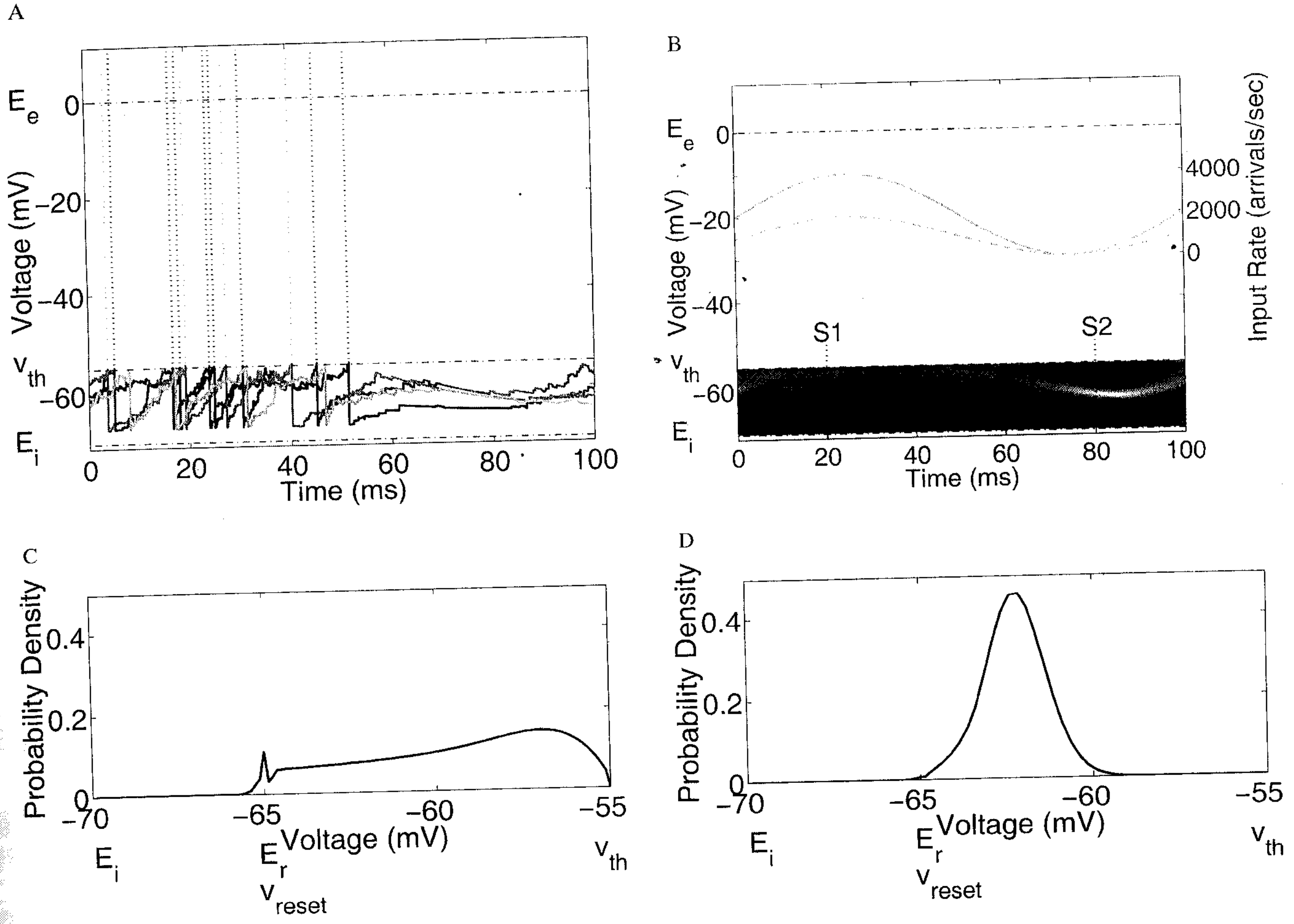

Figure 3. Illustration of the relationship between the individual responses of an integrate-and-fire neuron and its probability density. For this example, we let the excitatory input rate (in arrivals/second) be $\nu_e(t) = 2000(1 + \sin(2\pi f t))$ and the inhibitory input rate be $\nu_i(t) = 1000(1 + \sin(2\pi f t))$, $f = 10$ Hz. **A:** An example of the random responses of a neuron to four presentations of the same input rate. Note that the range of voltages is broader at a high firing rate (around $t = 20$ ms) than at a low firing rate (around $t = 80$ ms). **B:** A grayscale plot of the evolution of the neuron's probability density function for the same input rates. A dark shade indicates a low probability; a light shade indicates high probability. The probability density function can be viewed as the limit of a large number of responses as in panel A. S1 and S2 indicate locations of the snapshots for panels C and D. The excitatory (solid line) and inhibitory (dashed line) input rates are plotted in gray using the righthand scale. **C:** A snapshot of the probability density function during a time of high firing rate. The high probability near v_{reset} is due to the reset of the voltage of neurons that have fired. **D:** A snapshot of the probability density function during a time of low firing rate. Parameters used were $\mathcal{E}_i = -70$ mV, $\mathcal{E}_r = -65$ mV, $\mathcal{E}_e = 0$ mV, $v_{reset} = -65$ mV, $v_{th} = -55$ mV, and $\tau = 20$ ms.

In addition, we assume that each neuron in the population receives excitatory and inhibitory synaptic input with the same average rate (which we denote $\nu_e(t)$ and $\nu_i(t)$, respectively). We assume that the arrival times of synaptic events are random variables given by a modulated Poisson process (Knight et al., 1996). These assumptions are analyzed further in the discussion.

3.2. *Derivation of the Evolution Equations for a Single Population*

We first derive the evolution equation for the probability density function for a single population when the synaptic input rates, $\nu_e(t)$ and $\nu_i(t)$, are given functions of time. This development is similar to that in a number of previous papers (see Discussion).

The evolution equation for $\rho(v, t)$ is based on conservation of probability. We first look at the evolution of the probability contained in an interval (a, b):

$$\Pr\{V(t) \in (a, b)\} = \int_a^b \rho(v', t)\, dv'. \tag{8}$$

The probability contained in that interval can change only through the flux of probability across the endpoints of the interval. Probability flux is a signed quantity. By convention, a positive/negative flux at a point v is interpreted as the probability per unit time of crossing v from below/above. If we let $J(v, t)$ = flux of probability across v at time t, then

$$\begin{aligned} J(a,t) - J(b,t) &= \frac{\partial}{\partial t} \Pr\{V(t) \in (a,b)\} \\ &= \frac{\partial}{\partial t} \int_a^b \rho(v',t)\,dv'. \end{aligned} \tag{9}$$

Letting $b = v$, and differentiating (9) by v, we have

$$\frac{\partial \rho}{\partial t}(v,t) = -\frac{\partial J}{\partial v}(v,t), \quad \mathcal{E}_i < v < v_{th}. \tag{10}$$

We modify (10) to account for the firing and subsequent resetting of the voltage in integrate-and-fire neurons. Integrate-and-fire neurons that cross v_{th} fire a spike. In the probability model, movement of a neuron across a voltage corresponds to flux across that voltage. Thus, the population firing rate is the flux across threshold:

$$r(t) = J(v_{th}, t). \tag{11}$$

The population firing rate is not a temporal average but is an average across all neurons in the population.

When an integrate-and-fire neuron fires, its voltage is reset to v_{reset}. This reset of neurons produces a source of probability at v_{reset}. To reflect this source, (10) is modified to[2]

$$\frac{\partial \rho}{\partial t} = -\frac{\partial J}{\partial v} + \delta(v - v_{reset}) J(v_{th}, t), \quad \mathcal{E}_i < v < v_{th}. \tag{12}$$

To include a refractory period of length τ_{ref}, we can modify (12) to

$$\frac{\partial \rho}{\partial t} = -\frac{\partial J}{\partial v} + \delta(v - v_{reset}) J(v_{th}, t - \tau_{ref}). \tag{13}$$

What remains is to calculate an expression for the flux $J(v,t)$. In the rest of this section, we derive the following equations for $J(v,t)$:

$$J(v,t) = J_l(v,t) + J_e(v,t) + J_i(v,t)$$

$$J_l(v,t) = -\frac{1}{\tau}(v - \mathcal{E}_r)\rho(v,t)$$

$$J_e(v,t) = \nu_e(t) \int_{\mathcal{E}_i}^{v} \tilde{F}_{\Gamma_e^*}\left(\frac{v - v'}{\mathcal{E}_e - v'}\right) \rho(v',t)\,dv'$$

$$J_i(v,t) = -\nu_i(t) \int_{v}^{v_{th}} \tilde{F}_{\Gamma_i^*}\left(\frac{v - v'}{\mathcal{E}_i - v'}\right) \rho(v',t)\,dv'.$$

3.2.1. Components of the Flux. We calculate the flux of probability $J(v,t)$ based on (1), the subthreshold equation for $V(t)$. We break the flux into three components, corresponding to the three conductances of the integrate-and-fire neuron:

$$J(v,t) = J_l(v,t) + J_e(v,t) + J_i(v,t). \tag{14}$$

$J_l(v,t)$ is the leakage flux toward $\mathcal{E}_r$ due to the resting conductance; $J_e(v,t)$ is the excitation flux toward $\mathcal{E}_e$ due to the excitatory conductance; and $J_i(v,t)$ is the inhibition flux toward $\mathcal{E}_i$ due to the inhibitory conductance.

We calculate the equation for each component of the flux separately.

3.2.2. Leakage Flux. For an integrate-and-fire neuron, the voltage evolution due to leakage alone is (1) with $G_{e/i} = 0$:

$$\frac{dV}{dt} = -\frac{1}{\tau}(V - \mathcal{E}_r). \tag{15}$$

The voltage decays exponentially toward the resting potential, $\mathcal{E}_r$.

This movement of the neuron's state gives a leakage flux of probability in the population density

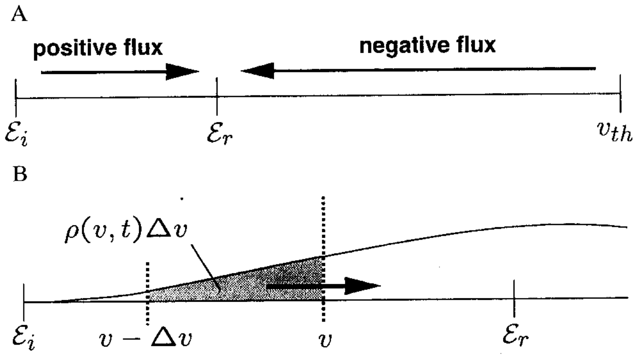

Figure 4. Diagram of the source of leakage flux. **A:** Neurons with $V(t) < \mathcal{E}_r$ move upward due to leakage, creating positive flux of probability. Neurons with $V(t) > \mathcal{E}_r$ move downward due to leakage, creating negative flux of probability. **B:** In the positive flux region from A, neurons cross a fixed voltage v from below. A sample $\rho(v,t)$ is plotted. The shaded region indicates the probability of crossing v in the time interval Δt.

model, as illustrated in Fig. 4. For $V(t) \in (\mathcal{E}_i, \mathcal{E}_r)$, $V(t)$ increases, creating a positive leakage flux across all $v \in (\mathcal{E}_i, \mathcal{E}_r)$. Similarly, the decreasing voltage will create a negative leakage flux across all $v \in (\mathcal{E}_r, v_{th})$. For simplicity, this derivation of the leakage flux will focus on the positive flux regime, $v \in (\mathcal{E}_i, \mathcal{E}_r)$.

Since for $v \in (\mathcal{E}_i, \mathcal{E}_r)$, the only movement of neurons across v is upward, the leakage flux at v is the probability per unit time that a neuron crosses v from below. We calculate this by first looking at the probability that a neuron will cross v in some short time Δt, as schematized in Fig. 4b.

The voltage of a neuron will cross v in the time interval $(t, t + \Delta t)$ if $V(t) \in (v - \Delta v, v)$, where

$$\Delta v = \frac{dV}{dt}\Delta t + O(\Delta t^2). \tag{16}$$

The probability that a neuron is in that interval is

$$\rho(v, t)\Delta v = \rho(v, t)\frac{dV}{dt}\Delta t + O(\Delta t^2). \tag{17}$$

Therefore, the leakage flux, the probability per unit time that a neuron crosses v, is

$$\begin{aligned} J_l(v, t) &= \frac{\rho(v, t)\Delta v}{\Delta t} + O(\Delta t) \\ &= \rho(v, t)\frac{dV}{dt} + O(\Delta t). \end{aligned} \tag{18}$$

Letting $\Delta t \to 0$ and substituting the value for $\frac{dV}{dt}$ from (15), we obtain the following expression for the leakage flux:

$$J_l(v, t) = -\frac{1}{\tau}(v - \mathcal{E}_r)\rho(v, t). \tag{19}$$

3.2.3. *Excitation Flux.*

Since the conductance change due to synaptic input is modeled as a delta function, the voltage of a neuron will jump upward on excitatory input. This jump will create a positive flux of probability across the jumped voltages, which we call the excitation flux.

To calculate the excitation flux across a fixed voltage v, we first calculate the probability that a neuron with voltage $V(t) = v'$, $v' < v$, will cross v, given that the neuron received an excitatory synaptic input (see Fig. 5a).

On receiving an excitatory synaptic input, the neuron with initial voltage $V(t) = v'$ will jump to the voltage $\tilde{v}$, where

$$\tilde{v} = v' + \Gamma_e^*(\mathcal{E}_e - v'), \tag{20}$$

(see Eq. (3)).

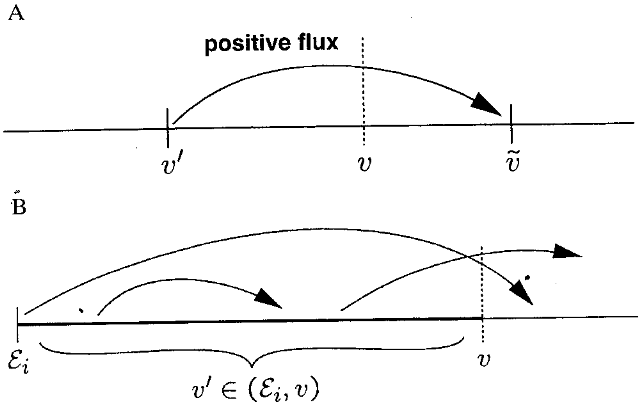

Figure 5. Diagram of the source of excitation flux. **A:** If a neuron receives an excitatory input that causes its voltage to jump over v, it will create a positive flux of probability across v. **B:** Neurons could potentially cross v from any voltage $v' \in (\mathcal{E}_i, v)$, if they received an excitatory input with a large enough conductance change.

If $\tilde{v} > v$, the excitatory input will cause the neuron to cross the voltage v, creating positive flux at v. This condition is equivalent, by (20), to

$$\Gamma_e^* > \frac{v - v'}{\mathcal{E}_e - v'}. \tag{21}$$

Since Γ_e^* is a random variable, it has the following probability of meeting condition (21):

$$\tilde{F}_{\Gamma_e^*}\left(\frac{v - v'}{\mathcal{E}_e - v'}\right), \tag{22}$$

where $\tilde{F}_{\Gamma_e^*}(x) = \Pr(\Gamma_e^* > x)$, as in Eq. (5).

Expression (22) is thus the probability that a neuron with voltage $V(t) = v'$, $v' < v$, will cross v, given that the neuron received an excitatory synaptic input. The rest of the calculation of the excitation flux follows quickly from (22) as follows.

The excitatory synaptic input rate, $\nu_e(t)$, is the probability per unit time that a neuron will receive excitatory input. Note that this probability is independent of past history by the Poisson assumption (Section 3.1).[3] Since the random input times are given by a modulated Poisson process and are independent of Γ_e^*, the probability per unit time that a neuron with voltage v' will cross v is simply the product of $\nu_e(t)$ and (22):

$$\nu_e(t)\tilde{F}_{\Gamma_e^*}\left(\frac{v-v'}{\mathcal{E}_e-v'}\right). \tag{23}$$

For a neuron to cross v from v', the neuron must start with the voltage $V(t)=v'$. The probability of $V(t)\in(v', v'+dv')$ is $\rho(v',t)dv$. The total flux of probability from $V(t)\in(v', v'+dv')$ across v is therefore

$$\nu_e(t)\tilde{F}_{\Gamma_e^*}\left(\frac{v-v'}{\mathcal{E}_e-v'}\right)\rho(v',t)\,dv'. \tag{24}$$

A neuron with voltage $V(t)$ anywhere in the interval $(\mathcal{E}_i, v)$ could contribute to the flux across v if it received a large enough excitatory synaptic input (Fig. 5b). The total excitation flux at v is thus given by summing (24) over all infinitesimal intervals from $\mathcal{E}_i$ to v:

$$J_e(v,t)=\nu_e(t)\int_{\mathcal{E}_i}^{v}\tilde{F}_{\Gamma_e^*}\left(\frac{v-v'}{\mathcal{E}_e-v'}\right)\rho(v',t)\,dv'. \tag{25}$$

3.2.4. Inhibition Flux.

The inhibition flux is determined in the same way as the excitation flux, with two important differences (see Fig. 6). First, a neuron can cross the voltage v only if its voltage $V(t)$ is in the interval (v, v_{th}). Second, since neurons cross v by moving to lower voltages, they contribute negative flux to the inhibition flux. Thus, the total inhibition flux is

$$J_i(v,t)=-\nu_i(t)\int_{v}^{v_{th}}\tilde{F}_{\Gamma_i^*}\left(\frac{v-v'}{\mathcal{E}_i-v'}\right)\rho(v',t)\,dv', \tag{26}$$

where $\tilde{F}_{\Gamma_i^*}(x)=\Pr(\Gamma_i^*>x)$.

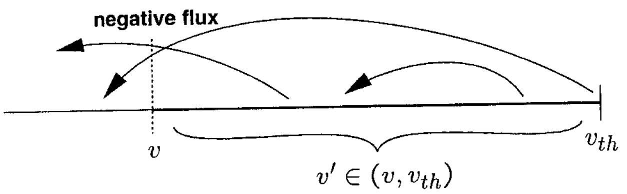

Figure 6. Diagram of the source of inhibition flux. A neuron could possibly cross v from any voltage $v'\in(v, v_{th})$ if it received an inhibitory input with large enough condutance change. The downward movement would create a negative flux of probability across v.

3.3. Derivation of Network Equations

Up to this point, we have looked only at a single population density, which represents neurons that are all similar. Furthermore, the input rate to the single population was a given function. Our primary interest, though, is modeling networks of neurons.

Deriving network equations involves only two more steps. First, we group neurons into many populations and create a population density for each group. Second, we connect the populations together via their firing rates to form networks of population densities.

3.3.1. Grouping Neurons into Populations.

To simulate a network of neurons, the first step is to group the neurons into populations. The groups must be chosen to satisfy the assumptions in Section 3.1. The neurons must be similar so that they can be described by one population density. In addition, the populations must be large enough so that the population density interpretation (7) is applicable. More details on the restriction of these assumptions are given in the discussion.

We form a population density for each group of neurons: $\rho^k(v,t)$, $k=1,2,\ldots,N$, where N is the number of populations. As outlined in the single-population case, each population has a firing rate, $r^k(t)$. Each population evolves according to the population density model Eqs. (12), (14), (19), (25), and (26).

The difference between network equations and the single-population equations is that the synaptic input rates for each population in the network, $\nu_e^k(t)$ and $\nu_i^k(t)$, are not given functions. Instead, the input rates are determined by the firing rates of the presynaptic populations as well as any external input rate. The calculation of these input rates thus depends on the connectivity of the network.

3.3.2. The Coupling of Populations.

We denote the connectivity of the network by W_{jk}, $j,k=1,2,\ldots,N$. W_{jk} is the number of presynaptic neurons from population j that project to each postsynaptic neuron in population k.

Each population, with density $\rho^k(v,t)$, $k=1,2,\ldots,N$, is either excitatory or inhibitory. We denote the set of excitatory indices by Λ_E and the set of inhibitory indices by Λ_I—that is, $\{\rho^k(v,t)\mid k\in\Lambda_{E/I}\}$ is the set of excitatory/inhibitory populations.

If the excitatory and inhibitory external input rates to population k are $\nu_{e,o}^k(t)$ and $\nu_{i,o}^k(t)$, then the total input rates to population k are

$$\nu_{e/i}^k(t) = \nu_{e/i,o}^k(t) + \sum_{j\in\Lambda_{E/I}} W_{jk} \int_0^\infty \alpha_{jk}(t') r^j(t-t')\, dt', \tag{27}$$

where $\alpha_{jk}(t')$ is the distribution of latencies of synapses from population j to population k. The choice of $\alpha_{jk}(t')$ used in our simulations is given in Appendix D.

With synaptic coupling, the input is a combination of outputs from other neurons. Since the output of a single neuron is certainly not guaranteed to be Poisson distributed, the assumption that the input is Poisson distributed (see Section 3.1) needs justification. This justification is given in the discussion.

3.4. *Summary of Population Density Model Equations*

Combining our equations for the flux (19, 25, 26) with the probability conservation Eq. (12), we have a partial differential-integral equation for the evolution of a population with synaptic input rates $\nu_e(t)$ and $\nu_i(t)$. In a network with populations $\rho^k(v,t)$, these input rates are given by (27). The firing rates of each population are given by (11).

The following summarizes the equations of the population density approach with populations $k = 1, 2, \ldots, N$:

$$\frac{\partial \rho^k}{\partial t}(v,t) = -\frac{\partial J^k}{\partial v}(v,t) + \delta(v - v_{reset}) J^k(v_{th}, t) \tag{28}$$

$$J^k(v,t) = J_l^k(v,t) + J_e^k(v,t) + J_i^k(v,t) \tag{29}$$

$$J_l^k(v,t) = -\frac{1}{\tau}(v - \mathcal{E}_r)\rho^k(v,t) \tag{30}$$

$$J_e^k(v,t) = \nu_e^k(t) \int_{\mathcal{E}_i}^{v} \tilde{F}_{\Gamma_e^*}\left(\frac{v-v'}{\mathcal{E}_e - v'}\right) \rho^k(v',t)\, dv' \tag{31}$$

$$J_i^k(v,t) = -\nu_i^k(t) \int_{v}^{v_{th}} \tilde{F}_{\Gamma_i^*}\left(\frac{v-v'}{\mathcal{E}_i - v'}\right) \rho^k(v',t)\, dv' \tag{32}$$

$$r^k(t) = J^k(v_{th}, t) \tag{33}$$

$$\nu_{e/i}^k(t) = \nu_{e/i,o}^k(t) + \sum_{j\in\Lambda_{E/I}} W_{jk} \int_0^\infty \alpha_{jk}(t') r^j(t-t')\, dt'. \tag{34}$$

The boundary conditions for these partial differential-integral equations are that $\rho^k(\mathcal{E}_i, t) = \rho^k(v_{th}, t) = 0$.

In general, the parameters $\mathcal{E}_{e/i/r}$, $v_{th/reset}$, and τ, as well as the functions $\tilde{F}_{\Gamma_{e/i}^*}$, could depend on k.

3.5. *Diffusion Approximation to the Model Equations*

The above partial differential-integral equations for the population density model can be solved efficiently on a computer. Even so, one can make a rigorous approximation to each equation that turns it into a diffusion equation, which can be solved even more quickly.

The diffusion approximation is based on an assumption that the synaptic conductance changes $\Gamma_{e/i}$ are small. The voltage then makes many small jumps due to synaptic input, leading to movement of probability that is similar to diffusion. However, as shown in Section 4.3, the diffusion approximation gives good results even with voltage jumps of moderate size.

The condition that the $\Gamma_{e/i}$ are small is equivalent to the condition that $\tilde{F}_{\Gamma_{e/i}^*}(\gamma) \ll 1$ except for small γ. Under this condition, we obtain the following diffusion approximation (see Appendix A) for (28) to (33):

$$\begin{aligned}\frac{\partial \rho}{\partial t} = {} & \frac{\partial}{\partial v}\left[\left(\frac{v-\mathcal{E}_r}{\tau} - \nu_e(t)c_{1e}(v) + \nu_i(t)c_{1i}(v)\right)\rho\right] \\ & + \frac{\partial}{\partial v}\left[(\nu_e(t)c_{2e}(v) + \nu_i(t)c_{2i}(v))\,\frac{\partial \rho}{\partial v}\right] \\ & + \delta(v - v_{reset})r(t)\end{aligned} \tag{35}$$

$$r(t) = -\nu_e(t)c_{2e}(v_{th})\frac{\partial \rho}{\partial v}(v_{th}, t), \tag{36}$$

where the $c_{1e/2e}$ and $c_{1i/2i}$ are given in Appendix A. Here, we have dropped the dependence on k. To obtain the final form of the diffusion equation, we used the fact that $\rho(v_{th}, t) = 0$ and $c_{2i}(v_{th}) = 0$. For the excitatory population parameters given in Appendix D, the values of $\tilde{F}_{\Gamma_{e/i}^*}$, $c_{1e/2e}$ and $c_{1i/2i}$ are plotted in Fig. 7.

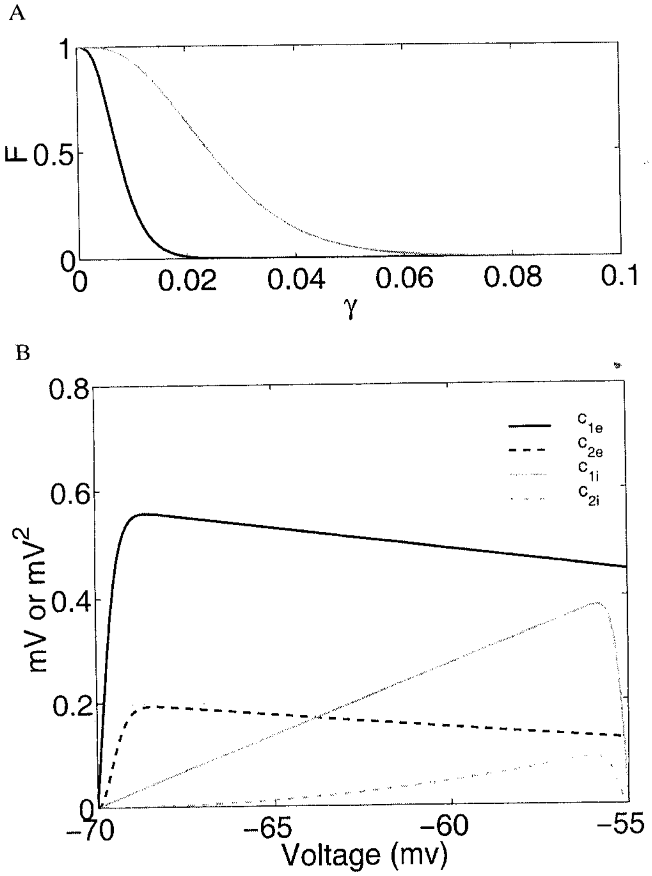

Figure 7. Complementary cumulative distribution functions for $\Gamma^*_{e/i}$ and diffusion coefficients for excitatory population parameters given in appendix D. **A:** $\tilde{F}_{\Gamma^*_e}(\gamma)$ is plotted with a black line and $\tilde{F}_{\Gamma^*_i}(\gamma)$ is plotted with a gray line. Already for γ near 0.1, both functions are nearly zero. For $\gamma > 0.1$ (not shown), both functions are practically zero. **B:** Coefficients for the diffusion equation (35). For c_{1e} and c_{1i}, the scale is in mV. For c_{2e} and c_{2i}, the scale is in mV^2.

4. Single-Population Results

In this section, we demonstrate simulations of a single, uncoupled population density. This population represents uncoupled neurons receiving input from an external source. The population firing rate and distribution of neurons over voltage are compared to corresponding estimates obtained from various numbers of individual neurons.

The population density equations are solved numerically using the method given in Appendix B, using a delta function modification specific for the case where $v_{reset} = \mathcal{E}_r$. The parameters we used are given in Appendix D. For each simulation, we show the results after the populations have settled into their periodic states, so the results do not depend on initial conditions.

The computation times listed are from simulations run on a Silicon Graphics Octane computer with 1 195 MHz MIPS R10000 processor.

4.1. *A Single-Population Density*

We simulated the response of a single population to the following excitatory and inhibitory synaptic input rates:

$$\nu_{e/i}(t) = \bar{\nu}_{e/i}(1 + \sin(2\pi f t)), \tag{37}$$

where the mean excitatory input rate $\bar{\nu}_e = 2000$ arrivals/second, the mean inhibitory input rate $\bar{\nu}_i = 1000$ arrivals/second, and $f = 10$ Hz. Using the parameters given in Appendix D, the average voltage jump due to excitatory and inhibitory input was about 0.5 mV and $-\frac{1}{3}$ mV, respectively.

Figure 8 shows the response of the population density model to these synaptic input rates. The population density, which is the distribution of neurons over voltage states, is plotted in Fig. 8a, and the resulting population firing rate is plotted in Fig. 8b along with the input rates (37).

The high peak in the population density around $t = 80$ ms corresponds to a time of low firing rate; most of the neurons have voltages near $\mathcal{E}_r = -65$ mV. During periods of high firing rate (such as around $t = 20$ ms), many of the neurons are closer to $v_{th} = -55$ mV. Additionally, since the neurons that have crossed threshold and fired are reset to v_{reset}, there is a sharp peak of probability at v_{reset} during this period.

4.2. *Comparison with Population of Individual Neurons*

The population density represents the fraction of neurons per unit voltage in the mathematical limit of an infinite number of neurons. Since such numbers aren't encountered in practice, the usefulness of the population density depends on its ability to represent accurately the behavior of populations of a finite number of individual neurons. We thus created populations of various numbers of integrate-and-fire neurons to compare with the population density. Details of how we computed the activity of the individual neurons are described in Appendix C.

Figure 9 compares the population density with individual neuron populations of 10, 100, 1,000, and

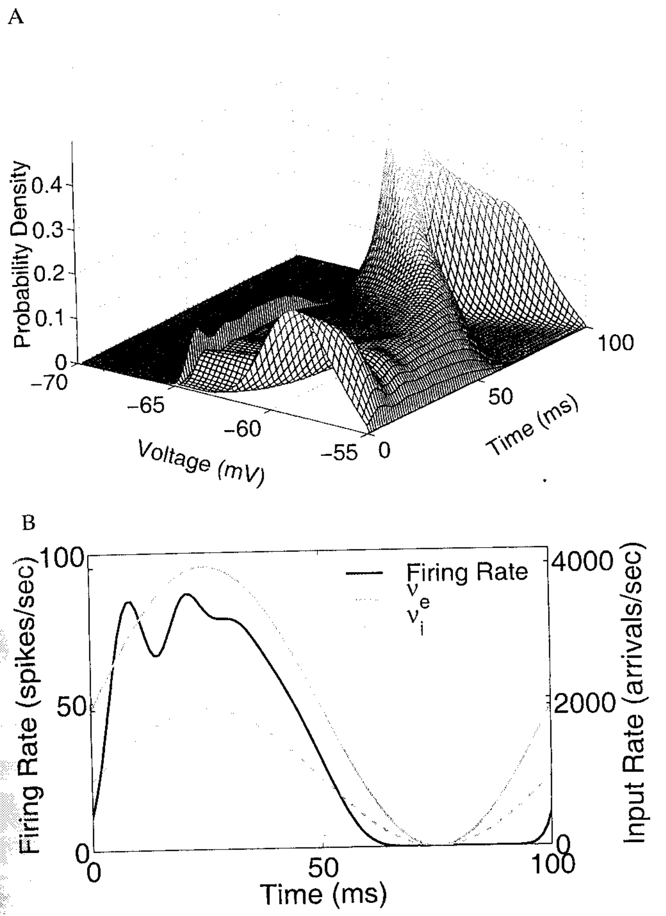

Figure 8. The evolution of the population density and its corresponding firing rate in response to sinusoidally modulated input rates. **A:** Distribution of neurons over voltage states as a function of time. Note that the periods of high firing rate (see panel B) correspond to periods with many neurons near threshold (−55 mV). The high peak near $t = 80$ ms corresponds to a low firing period when most of the neurons are near $\mathcal{E}_r = -65$ mV. Snapshots of similar high and low firing-rate distributions were shown in Fig. 3c, d. Neurons that are in the refractory state are not shown. **B:** Resulting firing rate of the population are plotted using the lefthand scale; the synaptic input rates ($\nu_e(t)$ and $\nu_i(t)$) are plotted using the righthand scale. Parameters used were $\mathcal{E}_i = -70$ mV, $\mathcal{E}_r = -65$ mV, $\mathcal{E}_e = 0$ mV, $v_{reset} = -65$ mV, $v_{th} = -55$ mV, $\tau = 20$ ms, and $\tau_{ref} = 3$ ms.

10,000 neurons. Each neuron received synaptic input that was a modulated Poisson process with the rates given in (37). The left column compares snapshots of the probability density at time $t = 20$ ms, and the right column compares the firing rates during the whole period. The results are obtained from one pass of the stimulus.

The distribution across voltages for 10 neurons (Fig. 9a) is too sparse to compare well to the population density model, but already with 100 neurons (Fig. 9c) the distributions are pretty similar. The firing rates, too, are already close to each other when the individual neuron population has 100 neurons (Fig. 9d). Both the distribution of neurons and the firing rate have almost converged with a population of 1,000 neurons (Fig. 9e,f).

We made no attempt to smooth the results of the individual neuron populations. We used histograms with fixed bin sizes (1/4 mV for the distribution across voltages, 2 ms for the firing rates). The small population results would have appeared even closer with smoothing.

This comparison serves as a verification of the population density method since all our assumptions, including the Poisson assumption, are satisfied in this simulation. This setup corresponds to the situation where the number of neurons in a population is vast, but one is observing only a limited number of them. Thus, Fig. 9 allows us to determine how many individual neurons must be followed computationally to estimate the population results at a given level of accuracy.

In addition, the comparison enables us to benchmark the computational efficiency of the population density approach. With a single population, all the times are small, but they shed light on relative differences. A simulation of two periods with the population density method took 0.1 seconds of computational time with a time step of 1 ms. The individual neuron simulation took 1.1 seconds with 100 neurons per population and 11 seconds with 1,000 neurons per population. The individual neuron results begin to give a smooth estimate around 100 neurons per population. Thus, even without synaptic interactions, the population density approach is between 10 and 100 times faster than the individual neuron simulations. Further savings of computational time can be achieved with the diffusion approximation of the population density equations.

4.3. *Comparison with Diffusion Approximation*

The diffusion approximation speeds up the population density simulations at the expense of minor differences in the firing rate and distribution of neurons.

A comparison between the diffusion approximation results and full model results is illustrated in Fig. 10. Snapshots of the distribution of neurons over voltages at $t = 20$ ms are shown in Fig. 10a. The firing rates over the whole stimulus period are shown in Fig. 10b.

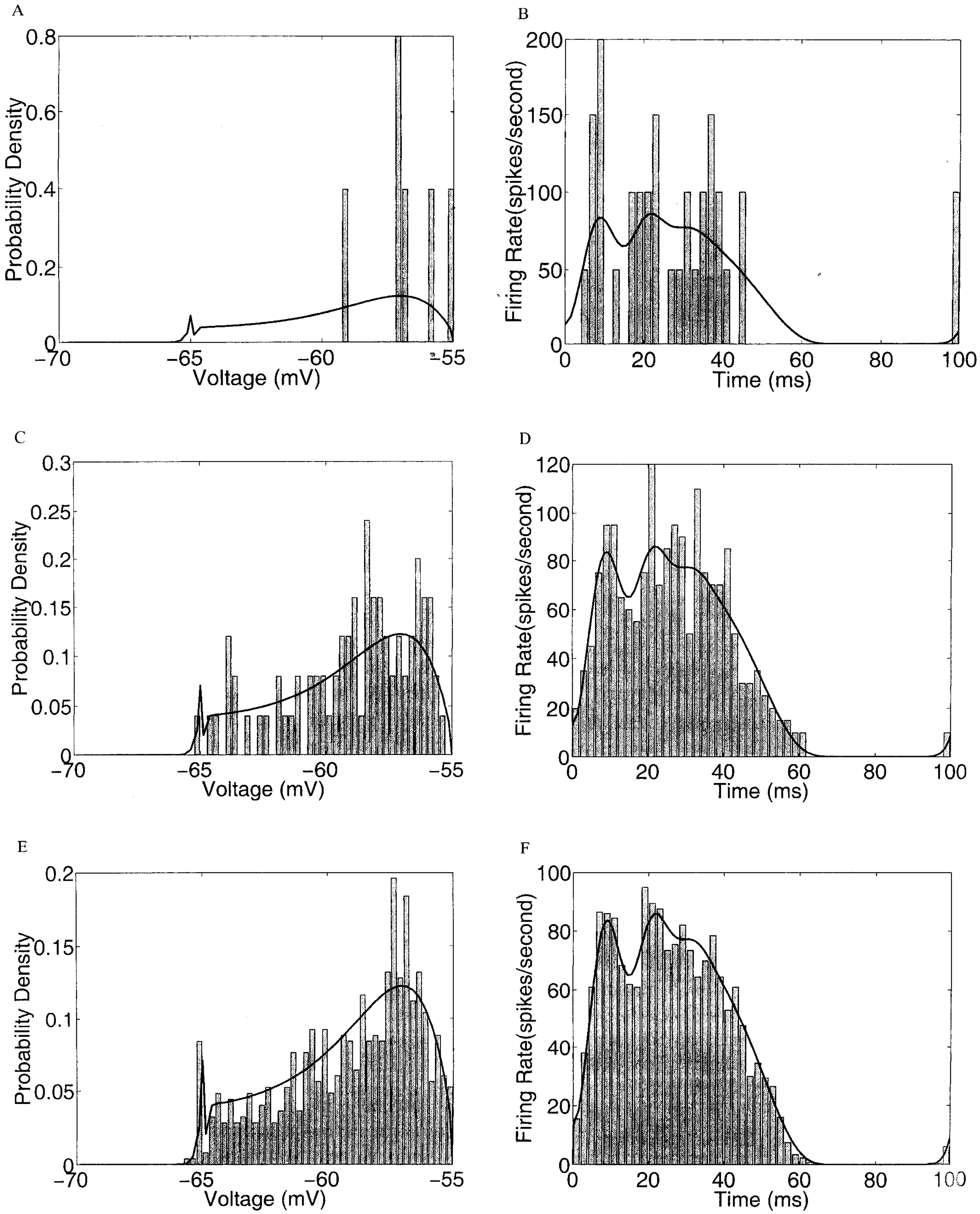

Figure 9. Comparison of the population density model with populations of individual neurons. The response to a single pass of the synaptic input rates from Fig. 8) is shown. The population density results (from Fig. 8) are shown by the solid line. The individual neuron results are shown by the histograms. Both a snapshot of the probability density at $t = 20$ ms (left column) and the firing rates (right column) are shown. The comparisons are with four population sizes for the individual neurons: **A, B:** 10 neurons, **C, D:** 100 neurons, **E, F:** 1,000 neurons, and **G, H:** 10,000 neurons. In the snapshots, neurons in the refractory state are not shown. Parameters are the same as in Fig. 8.

(Continued)

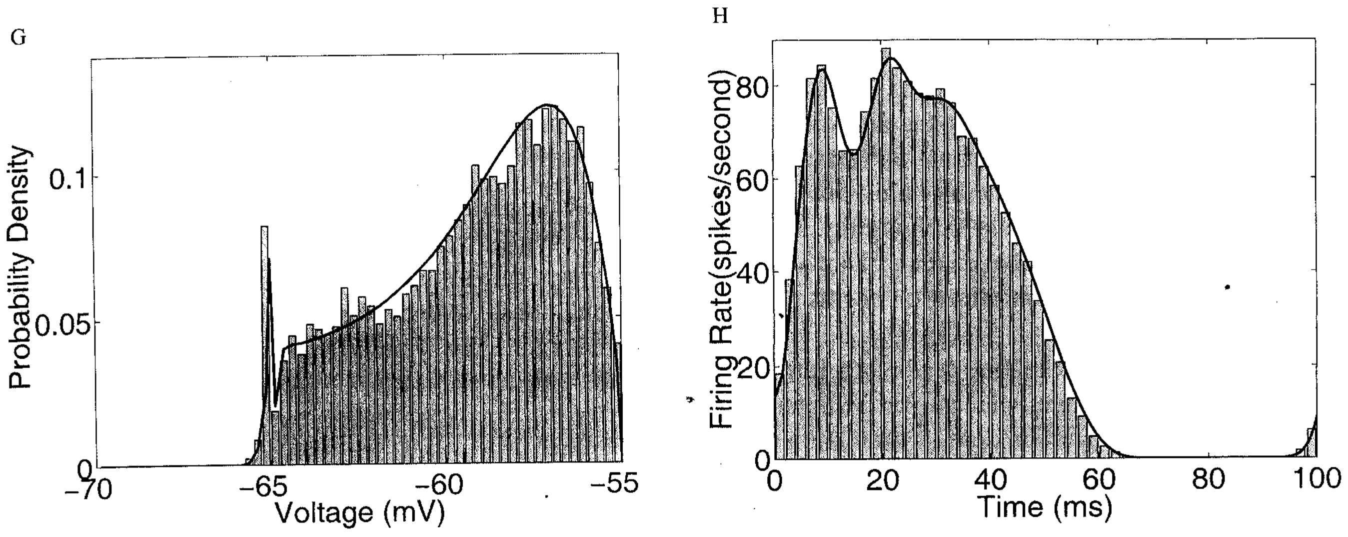

Figure 9. (*Continued*)

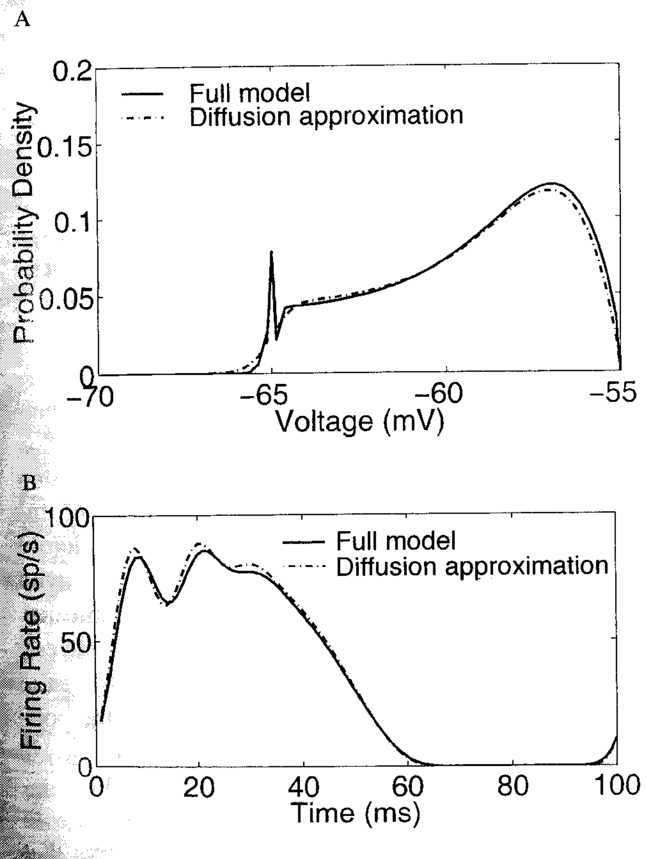

Figure 10. Comparison of the full partial differential-integral equation model with its diffusion approximations under same input and parameters as Fig. 8. **A:** A snapshot of the distribution of neurons over voltage states at time $t = 20$ ms. The full model snapshot is the $t = 20$ ms cross-section of Fig. 8a. The diffusion approximation includes the peak of probability at −65 mV because the equations are not completely diffusion (see text). **B:** The resulting firing rates of the populations. The full model firing rate is the same as Fig. 8b.

The population density snapshots of Fig. 10a show that the versions are similar. The diffusion approximation contains a peak at $v_{reset} = -65$ mV because the equations are not completely diffusion equations. As explained in Appendix B, we calculate separately the evolution of a delta function component of $\rho(v, t)$ at v_{reset}.

The resulting population firing rates shown in Fig. 10b are also similar. The diffusion approximation tends to overshoot the peaks in the firing rate. The root mean squared error of the diffusion approximation rate is less than 5%.

Figure 11 shows how the error in firing rates increases as the strengths of the synapses, and thus the

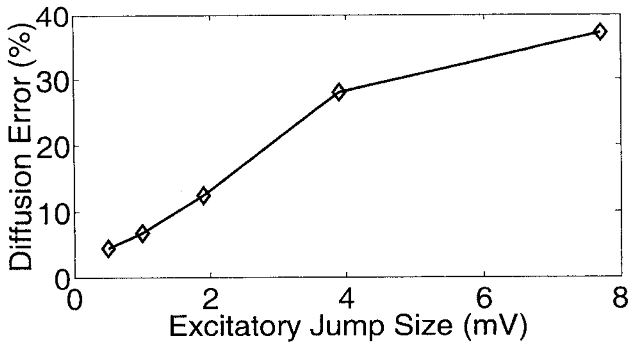

Figure 11. Root mean squared errors of the diffusion approximation rates compared with the full partial differential-integral equation rates. The error is plotted against the average voltage jump due to an excitatory synaptic input with a starting voltage halfway between $\mathcal{E}_r$ and v_{th}. Average inhibitory jumps were approximately half the excitatory jumps. The first data point is from Fig. 10. For each subsequent data point, the synaptic conductance changes were doubled and the input rates halved.

synaptic voltage jumps, are increased. Although the evolution of the voltage may not be similar to diffusion for excitatory voltage jumps as large as 2 mV, the diffusion approximation is still close to the full solution. We thus view the diffusion approximation as mathematical tool to speed the computation rather than a claim that the evolution of the voltage is physically similar to diffusion. For excitatory voltage jumps of 4 mV, the accuracy of the diffusion approximation begins to break down.

The diffusion approximation leads to a further increase in computational efficiency. For the parameters used in Fig. 10, a population density simulation of two periods was reduced from 0.1 seconds with the full model to 0.03 seconds with the diffusion approximation. With larger synaptic conductance changes, the difference between the diffusion and full model speeds becomes greater, although the diffusion approximation becomes less accurate.

5. Network Results

5.1. A Simple Network

To illustrate the performance of the population density approach, we constructed a simple network of two populations. A schematic of the network is shown in Fig. 12. The network contained one excitatory (E) and one inhibitory (I) population. Each element of the connectivity matrix, W_{jk}, $j, k = E, I$, gives the number of synapses made by population j onto each individual neuron in population k (see Section 3.3.2). W was chosen so that all connections were equal, except that the excitatory-excitatory weight was half the others:

$$W = \bar{W}\begin{pmatrix} 0.5 & 1 \\ 1 & 1 \end{pmatrix}, \qquad (38)$$

where the parameter $\bar{W}$ determines the overall strength of the synaptic connections.

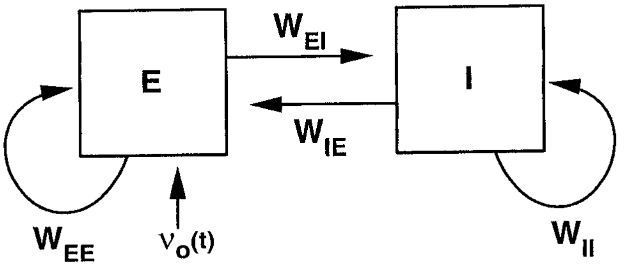

Figure 12. Schematic of the network architecture used in the simulations. The network contained one excitatory population (E) and one inhibitory population (I). The synaptic connection weights were $W_{EE} = \bar{W}/2$, $W_{EI} = W_{IE} = W_{II} = \bar{W}$. The external excitatory input rate to the excitatory population (E) was $\nu_o(t) = \bar{\nu}_o(1 + \sin(2\pi f t))$, $f = 10$ Hz.

In addition to the synaptic connections between the populations, the excitatory population received an external excitatory synaptic input at the rate

$$\nu_{e,o}(t) = \nu_o(t) = \bar{\nu}_o(1 + \sin(2\pi f t)), \qquad (39)$$

where $f = 10$ Hz and $\bar{\nu}_o$ is a parameter determining the overall strength of the external input. The inhibitory population received no external input, $\nu_{e,o}(t) = 0$.

With this network structure, we solved Eqs. (28) to (34) to obtain the output of the network for various values of $\bar{W}$ and $\bar{\nu}_o$.

5.2. Individual Neuron Network for Comparison

The simple network can be used to test how well the population density approach models the interactions of finite populations. The Poisson assumption may not be completely satisfied with finite populations, and thus the population density results may not match finite-population results.

To test the accuracy of the population density network, we implemented an individual neuron simulation of the network in Fig. 12. We created one population of M_e excitatory neurons and one population of M_i inhibitory neurons. We connected the neurons randomly so that the number of connections a neuron received from a particular population was given by the same matrix, W, from Eq. (38). The synaptic delays between each pair of neurons were chosen randomly from the same distribution of latencies $\alpha_{jk}(t)$ used for the population density simulations (see Appendix D).

The excitatory population received external excitatory input that was a modulated Poisson process with rate $\nu_o(t)$ from Eq. (39). The other synaptic input that neurons received was not forced to be Poisson; a neuron received a synaptic conductance change some delay after one of its presynaptic neurons fired. Inhibitory neurons received no input with an imposed Poisson distribution. In this way, a comparison between the population density and the individual neuron implementation serves as a test of the Poisson assumption.

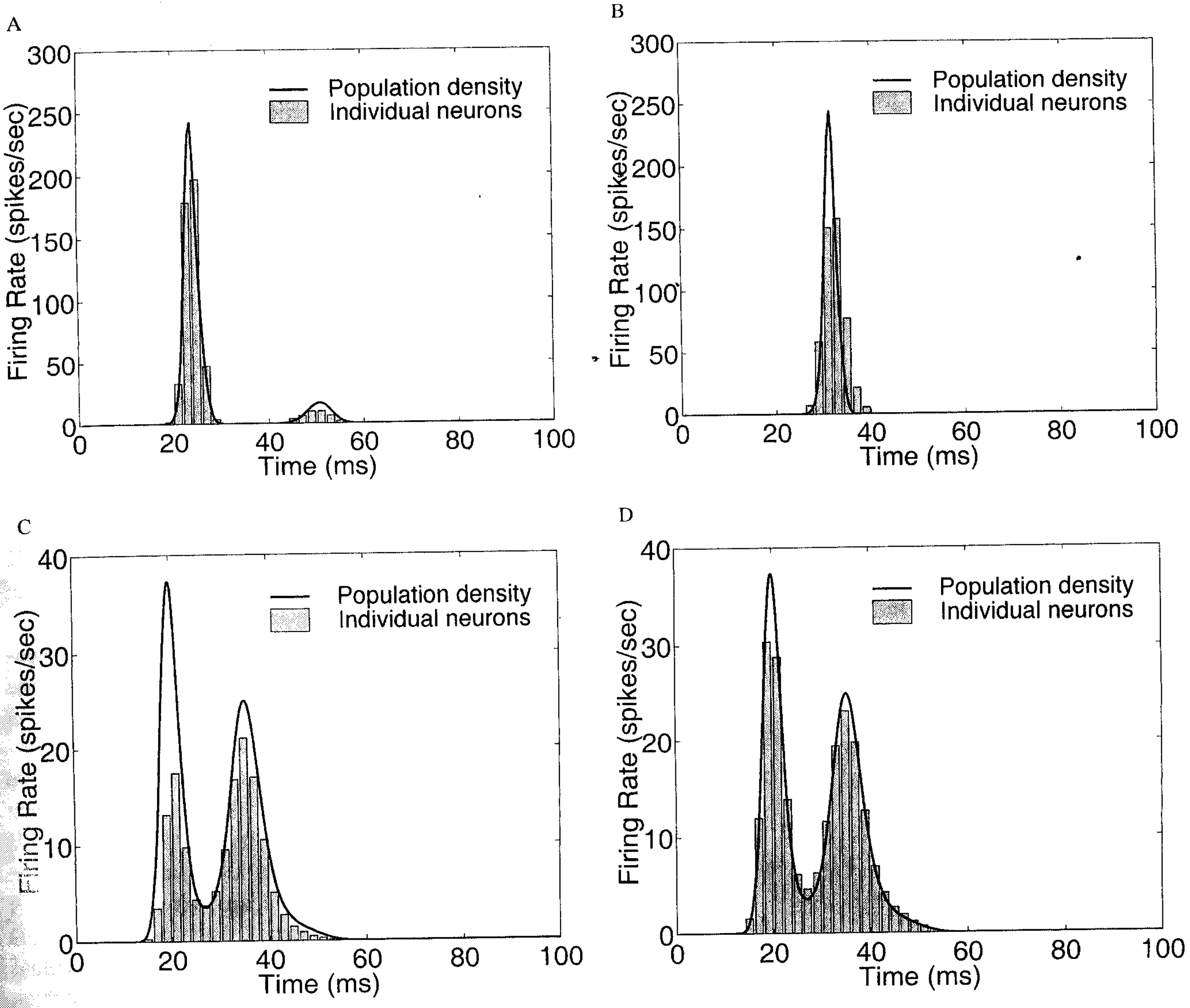

Figure 13. Comparison of the inhibitory firing rates for population density and individual neuron implementations of the network in Fig. 12. The firing rates of the population density are shown by solid lines. Average firing rates of the individual neurons are shown by the histograms. **A:** A sparse network with $M = 100$, $\bar{W} = 30$ and $\bar{\nu}_o = 700$ arrivals/second. **B:** A nonsparse network with $M = 100$, $\bar{W} = 70$, and $\bar{\nu}_0 = 500$ arrivals/second. **C:** A sparse network with $M = 100$, $\bar{W} = 10$, and $\bar{\nu}_0 = 1000$ arrivals/second. **D:** The same network as in C but with random numbers of synapses.

5.3. *Comparison of Results*

We compared the firing rates of the population density and individual neuron implementation of the simple network. We looked at two sizes of individual neuron populations. In both cases $M_e = M_i = M$. In one case $M = 100$ neurons/population, and in the other $M = 1000$ neurons/population.

To determine precisely how well the results matched, we averaged over many periods of $\nu_o(t)$ (500 when $M = 100$, 50 when $M = 1000$) for the individual neurons. The errors given below are root mean square errors between the individual neuron histograms and the corresponding areas under the population density curve.

Figure 13 compares the firing rates given by the inhibitory population density with the firing rates given by a single realization of an individual neuron network for two sets of parameters. We focus on the inhibitory population since the inhibitory population received no external input. The match between the excitatory populations (not shown) was always as good as the match between the inhibitory populations. We show the results from individual neuron networks with $M = 100$.

Increasing M to 1000 only improved the match (not shown).

Figure 13a shows the results with $\bar{W} = 30$ and $\bar{\nu}_o = 700$ arrivals/second. The average individual neuron firing rates are close to the population density firing rates with an error of less than 15%. For the excitatory populations, the firing rates were within 10% of each other (not shown). The same simulation with $M = 1000$ produced similar results (not shown). The close fit of the two models demonstrates that, at least for this set of parameters, the Poisson assumption is satisfied well enough.

The population density approach assumes sparse coupling since it represents an infinite number of neurons with a finite number of synapses per neuron. This sparse coupling allowed us to argue that the inputs to each neuron are conditionally independent as required when we approximate the input to each neuron as a modulated Poisson process (see discussion). Thus, one might expect that in an individual neuron network where the coupling was not sparse, the results would begin to diverge from that of the population density simulation.

An example where the coupling is not sparse is shown in Fig. 13b. In this figure, $\bar{W} = 70$ and $\bar{\nu}_0 = 500$ arrivals/second. Since the number of neurons/population $M = 100$, the coupling is not sparse and the error has increased to almost 40%. Increasing M to 1000 restores the sparseness, and the error drops below 20% (not shown).

The Poisson assumption also requires a sufficient number of converging synapses. For the parameters $\bar{W} = 10$ and $\bar{\nu}_0 = 1000$ arrivals/second, $\bar{W}$ is not large enough and the error exceeds 40% for both $M = 100$ (Fig. 13c) and $M = 1000$ (not shown). However, in realistic networks, the number of converging synapses is likely to be larger. Moreover, these results change significantly with randomization in the number of synapses that each neuron receives.

5.4. *A Random Number of Synapses*

In the derivation of the population density equations, we required that each neuron in a population k received exactly W_{jk} synapses from neurons in population j. This requirement was met in the simulations above. However, this requirement is unrealistic since one would expect the number of synapses to vary among neurons in each population.

We tested the importance of this restriction by creating individual neuron networks with random numbers of synapses and comparing the results to the population density network. In the random network, W_{jk} is interpreted as the expected number of synapses from neurons in population j onto each neuron in population k. For each neuron in population k, the precise number of synapses from population j was chosen from a binomial distribution with parameters $p = 1/M_k$ and $n = W_{jk}M_k$ (thus, the mean $= W_{jk}$).

To our surprise, we found that the introduction of this type of randomness did not hurt and sometimes improved the match between the firing rates of the individual neuron and the population density networks. The improvement was dramatic when $\bar{W}$ was very small, as shown in Fig. 13c,d. In this case, the error dropped from over 40% to below 10%. The drop in error was noticable but less dramatic for almost all the other examples mentioned. For example, with $\bar{W} = 30$, the error dropped from almost 15% to almost 10% (not shown). However, for the nonsparse network of $\bar{W} = 70$ and $M = 100$, the error changed very little.

Clearly, an equal number of synapses for each neuron in a population is not crucial for the validity of the population density results. Although we don't have a completely satisfactory explanation for the almost universal improvement of the error with randomness, we propose an explanation for the improvement with very low connectivity (e.g., $\bar{W} = 10$) in the discussion.

In our subsequent individual neuron networks, we use a random number of synapses. Not only does this test the population density model against a more realistic network, it also allows us to implement individual neuron networks where W_{jk} is not an integer, such as in the following orientation tuning example.

6. Orientation Tuning in the Visual Cortex

As a further test and as a demonstration of the computational efficiency of the population density approach, we implemented a population density model of one hypercolumn of visual cortex. Our network model is based loosely on the model of orientation tuning in cat visual cortex by Somers et al. (1995), which we paraphrased and recast in the population density framework.

6.1. *Overview of Orientation Tuning*

Figure 14 is a schematic illustration of the response of a typical cortical neuron to the presentation of an oriented dark bar on a uniform background. This hypothetical

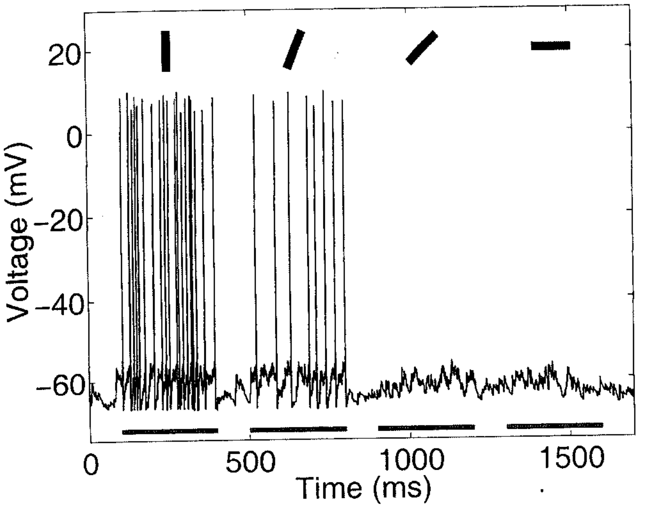

Figure 14. Illustration of the orientation tuning of a cortical neuron. This neuron fires most rapidly in response to a vertical bar. (Adapted from Somers et al., 1995.)

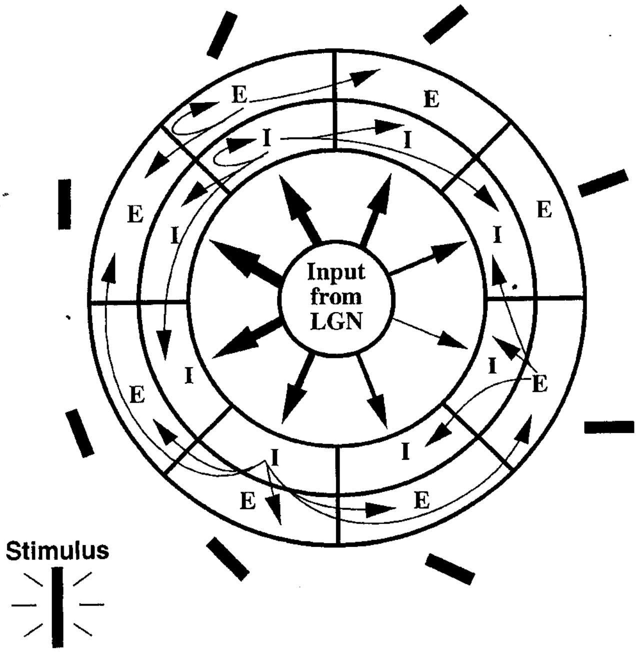

Figure 15. Schematic of our model network of one hypercolumn of visual cortex. Layer IV excitatory (E) and inhibitory (I) neurons of cat primary visual cortex are grouped into populations of neurons with similar orientation preference (denoted by oriented bars). The cortex receives feed-forward input from the LGN with weak orientation bias. Cortical interaction strength depends on the difference in preferred orientation. Inhibitory interactions are longer range than excitatory interactions. Typical interactions are shown by the arrows.

neuron is said to have a vertical preferred orientation because it responds most strongly to vertically oriented stimuli. The neuron displays sharp orientation tuning—that is, the response falls off quickly as the orientation changes from vertical. A salient feature of orientation tuning in visual cortex (not shown) is that the sharpness of tuning is almost independent of stimulus contrast. The origin of this sharp, contrast-independent orientation tuning is a controversial subject (cf. review by Shapley and Sompolinsky, 1997). Our focus in the present article is to illustrate the population density method and test its accuracy with various practice problems rather than to address underlying neural mechanisms. In the Somers model, and thus in ours, this contrast-independent orientation selectivity is a result of sharpening by the cortical network. Similar mechanisms underlie the models of others (Ben-Yishai et al., 1995; Hansel and Sompolinsky, 1996; Carandini and Ringach, 1997).

6.2. *A Population Version of the Somers et al. Model*

The structure of our network is very similar to the network of Somers et al. (1995). We made some modifications to a neuron's response to synaptic input and the number of synaptic connections. In this section, we describe the details of the model, including variations from the Somers et al. model. More details on parameters are given in Appendix D.

We modeled one hypercolumn of layer IV neurons in cat primary visual cortex. The neurons in the hypercolumn received input from the same part of visual space but had different preferred orientations that span the full set of 180 deg. In this model cortical neurons have broad orientation tuning even in the absence of cortical-cortical interactions as a consequence of anisotropic projection from the lateral geniculate nucleus (LGN) to each cortical column.

We grouped the neurons into populations with similar preferred orientation, dividing the orientation space into a set of discrete orientations, θ_j, $j = 1, 2, \ldots, N$, as schematized in Fig. 15. We made an arbitrary choice of $N = 18$, while Somers et al. took N to be 11. At each preferred orientation, θ_j, we formed one excitatory (E) and one inhibitory (I) population density function, $\rho^{E/I}(v, t, \theta_j)$, and corresponding firing rates, $r^{E/I}(t, \theta_j)$. Somers et al. (1995) estimated that there are 2,500 to 4,500 neurons in layer IV of one hypercolumn. According to this estimate, each of our columns in reality would contain ~200 excitatory and ~50 inhibitory neurons.

As in the Somers et al. model, cortical interaction strength depended on differences in preferred orientation. The synaptic connectivity was a gaussian

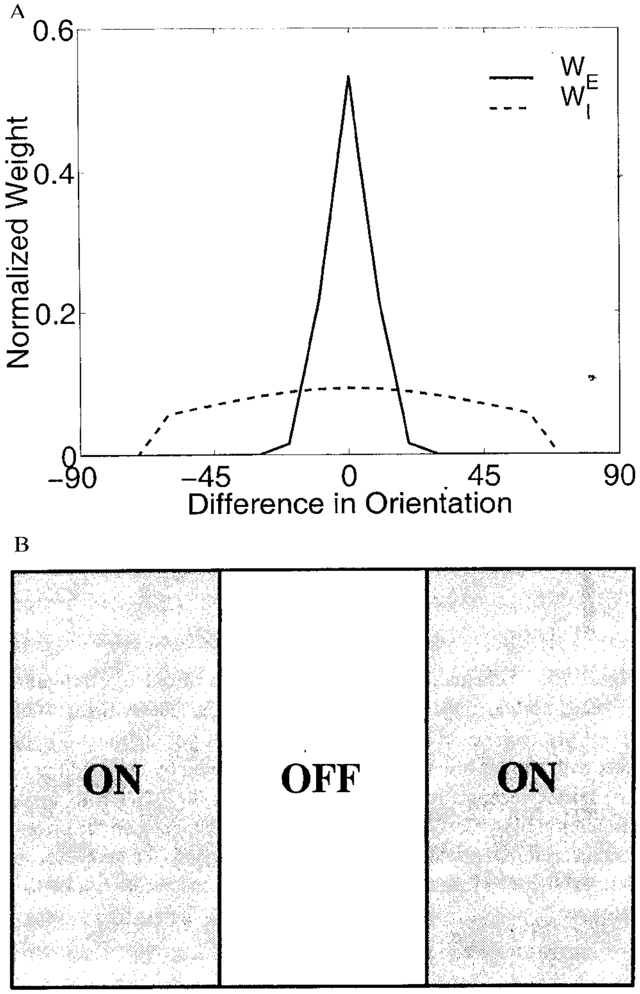

Figure 16. **A:** Normalized synaptic connectivity functions. W_E and W_I were used for excitatory and inhibitory connections, respectively. **B:** The rectangular center-off subfield flanked by center-on subfields of the LGN → cortex weighting function.

function of the preferred orientation difference, with the maximum connectivity being within the same population. Inhibitory interactions were longer range than excitatory interactions, $\sigma_I = 60\,\text{deg}$, $\sigma_E = 7.5\,\text{deg}$. The gaussian functions were clipped at 60 deg. The synaptic connectivity functions are shown in Fig. 16a.

In our model, the total number of excitatory synapses onto each excitatory and inhibitory neuron was 72 and 112, respectively, and the total number of inhibitory synapses onto each excitatory and inhibitory neuron was 48 and 16, respectively. Each excitatory neuron received 48 excitatory synaptic inputs from the LGN, and each inhibitory neuron received 32. These numbers are twice those used in the Somers et al. model. The number of synapses they used is a vast underestimate of the actual number of synapses. In reality, each excitatory neuron receives 3,000 to 6,000 synapses, though only some fraction of these come from other layer IV cells and the LGN. Somers et al. chose few synapses for sake of computational ease and compensated to some extent by also choosing large unitary postsynaptic conductance magnitudes. In the population density framework, a larger number of synapses from one population to another has no effect on computational complexity. Therefore, we easily doubled the number of synapses Somers et al. used and reduced the size of the unitary postsynaptic conductance magnitudes by half.

Synaptic delays between cortical populations were gamma distributed with a mean of 3 ms and a standard deviation of 1 ms. This is similar to Somers et al. zero-bounded gaussian function.

The integrate-and-fire point neuron model underlying our population density equations is less elaborate than that of Somers et al. In particular, we omitted the elevation and decay of spike threshold following a spike and omitted the activation of the after-hyperpolarizing conductance following a spike. Instead we simply reset the voltage to v_{reset}, which we took to be equal to the resting voltage $\mathcal{E}_r$.

In the model of Somers et al., the time courses of the unitary postsynaptic conductances were fast on the time scale of the membrane time constant for both inhibitory and excitatory neurons. Therefore, our approximation of the conductance waveforms by delta functions (see Section 2.2) was justified.[4] In an effort to be realistic, we made the unitary postsynaptic conductances in our model random with a coefficient of variation of 0.5 (see Appendix D). The average size of the jumps in voltage due to each type of synaptic input was similar to half the size of the postsynaptic potentials in the model of Somers et al. Starting from halfway between $\mathcal{E}_r$ and v_{th}, our average excitatory postsynaptic potentials in excitatory and inhibitory neurons were 0.5 and 1.2 mV, respectively. Average inhibitory postsynaptic potentials were -0.3 and -0.7 mV in excitatory and inhibitory neurons, respectively.

The weak orientation bias in cortical input from the LGN was produced, as in the model of Somers et al., by an elongated rectangular center-off subfield flanked by two identical center-on subfields. In our model each cortical neuron received input from every LGN neuron in the three subfields. In the model of Somers et al., the field lengths along the major axis were random with a mean ratio of major to minor subfield length of 2:1, and the LGN → cortex connections were random.

The various preferred orientations of the inputs to the various cortical columns were obtained by simply rotating the LGN → cortex weighting function shown in Fig. 16b. Using the weighting function in this way, we transformed the spatial structure of the stimulus (as reflected in the LGN response) to a bias in the orientation coordinates of our cortical model.

In our model, as in the model of Somers et al., an individual LGN neuron simply mirrored its input from a single retinal neuron. We generated the firing rate of an LGN neuron/retinal ganglion cell by first convolving a spatiotemporal separable impulse response function with the stimulus and then multiplying by a contrast dependent scale factor. Then this computed signal was half-wave rectified (giving only the positive part) to give the rate. The spatial part of the spatiotemporal impulse response function was the same difference of gaussians as that used by Somers et al. The temporal part was not the simple exponential function of Somers et al. Instead, we used the model of Victor (1987) for the dynamics of the center mechanism of a cat retinal ganglion cell.

6.3. *Population Equations*

As described above, our model of one hypercolumn has $2N$ probability densities, $\rho^{E/I}(v, t, \theta_j)$, and corresponding firing rates, $r^{E/I}(t, \theta_j)$, $j = 1, 2, \ldots, N$. For each population density, we have one set of Eqs. (28) to (34).

The external input to our network is the excitatory input from the LGN to excitatory and inhibitory populations. Denote the corresponding input rates to excitatory and inhibitory populations with preferred orientation θ_j by $\nu^E_{e,o}(t;\theta_j)$ and $\nu^I_{e,o}(t;\theta_j)$, respectively. There is no external inhibitory input ($\nu^{E/I}_{i,o} = 0$).

In this notation, Eq. (34) for the excitatory input rate to the inhibitory population at θ_j, for example, becomes

$$\nu^I_e(t;\theta_j) = \nu^I_{e,o}(t;\theta_j) + \sum_k W_{EI}(\theta_j - \theta_k) \times \int_0^\infty \alpha(\tau) r^E(t-\tau;\theta_k)\,d\tau, \qquad (40)$$

where $W_{EI}(\theta_j - \theta_k)$ is the number of synapses from the excitatory neurons at θ_k onto inhibitory neurons at θ_j. The distribution of synaptic delays, $\alpha(\tau)$, is given in Appendix D.

We numerically solve this system of equations under various stimulus conditions, where the external input $\nu^{E/I}_{e,o}(t;\theta_j)$ is generated from the response of the LGN to the stimulus.

6.4. *Orientation Tuning Network Results*

Figures 17 and 18 show the firing rates of the excitatory population to the presentation of a flashed dark bar

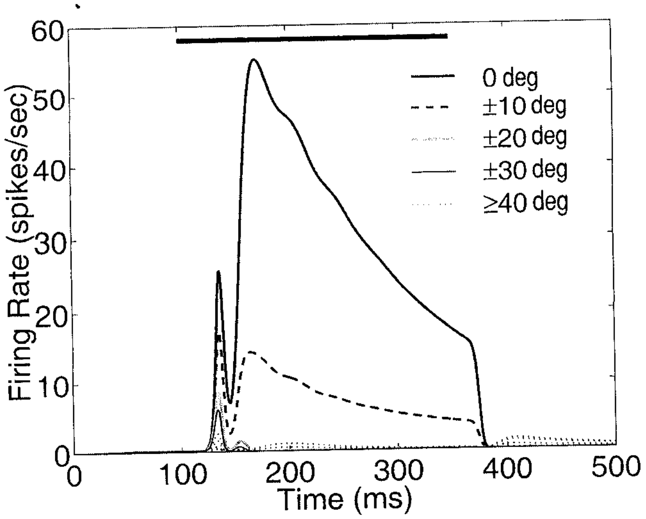

Figure 17. Responses of populations of excitatory neurons with different preferred orientations to a flashed bar at 0 deg. The bar is flashed for 250 ms, beginning at 100 ms, as shown by the horizontal line. After the initial transient, neurons with preferred orientation ≥20 deg from zero are silent.

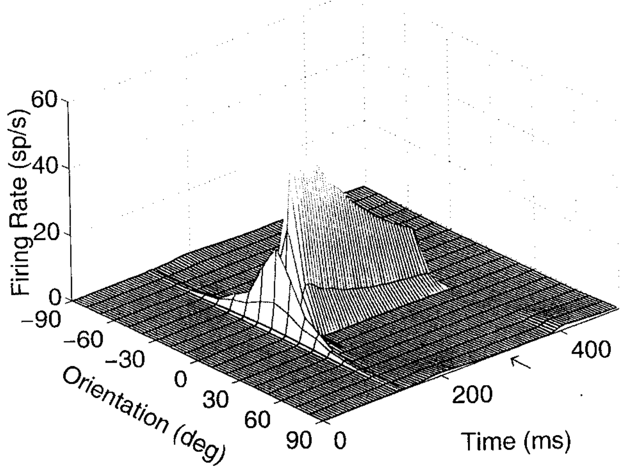

Figure 18. A 3D view of the same excitatory responses shown in Fig. 17, demonstrating the orientation tuning of the neurons. Firing rate across preferred orientations is plotted against time. The initial transient reflects the orientation bias of the input from the LGN (not shown). Subsequently, the tuning sharpens dramatically. The stimulus onset is at 100 ms, as in Fig. 17. The arrow indicates the time of the snapshot in Fig. 19.

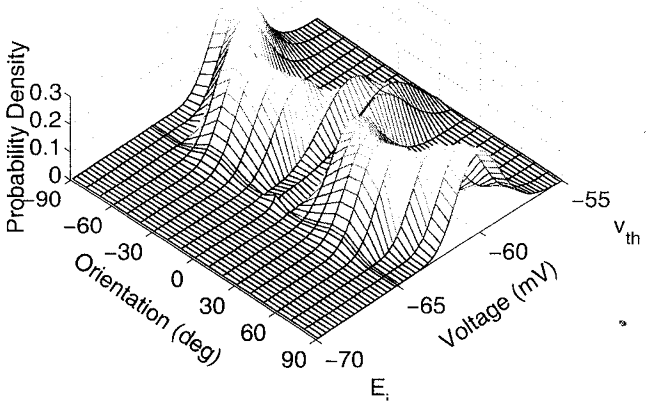

Figure 19. Snapshot of the distribution of excitatory neurons over voltage. The probability density across voltage is plotted against preferred orientation. This snapshot is from $t = 300$ ms in Fig. 18. Neurons with preferred orientation near 0 deg are distributed closer to v_{th}, as they are the only ones firing.

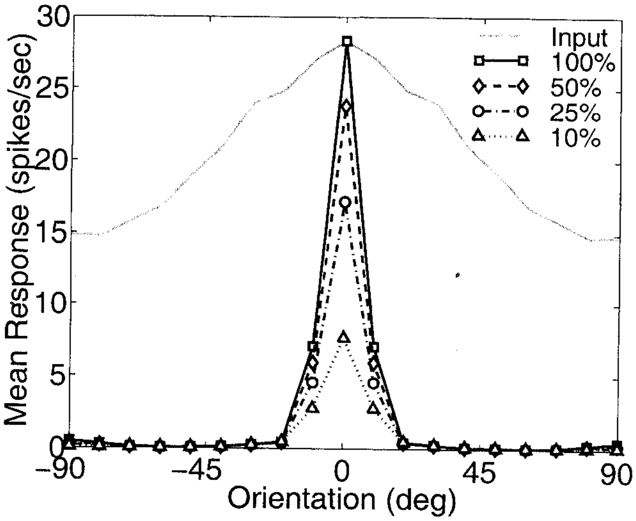

Figure 20. Orientation tuning as measured by the mean firing rate during the 250 ms that the flashed bar is visible. Cortical excitatory population response (symbols) is plotted against preferred orientation for contrasts ranging from 10% to 100%. The orientation bias of the input from the LGN (gray line) is plotted on an arbitrary scale.

oriented at 0 deg. The bar was flashed for 250 ms beginning at $t = 100$ ms with 100% contrast. The initial transient of the response is broadly tuned across the preferred orientations. This response reflects the weak orientation bias of the input from the LGN (not shown). However, the later response is much more sharply tuned; only neurons within 20 deg of zero fire significantly.

The full set of excitatory probability densities for $t = 300$ ms is plotted in Fig. 19. At that time, the firing rate response in Fig. 18 was sharply tuned around 0 deg. Correspondingly, the probability densities near 0 deg in Fig. 19 are closer to v_{th} (-55 mV) than the other probability densities. The excitation of these neurons across v_{th} created the tuning of the firing rate seen in Fig. 18.

Further details of the orientation tuning produced by the cortical network are shown in Fig. 20. For different contrasts of the flashed bar, the mean firing rate remains sharply tuned around 0 deg (symbols). Note that the tuning width is roughly independent of stimulus contrast. The tuning is much sharper than the orientation bias of the input from the LGN (gray line).

6.5. *Comparison with Network of Individual Neurons*

We implemented an individual neuron network of this model of a hypercolumn of visual cortex in the same way we implemented an individual neuron version of the simple network in Section 5.1. The parameters and network structure of the individual neuron network were the same as the population density network given in Section 6.2. Excitatory and inhibitory neurons were each grouped into 18 populations with identical orientation preference. Each excitatory/inhibitory neuron with a preferred orientation of θ_j received excitatory external input that was an independent modulated Poisson process with the mean rate $\nu_{e,o}^{E/I}(t; \theta_j)$, above.

For all simulations, we let each population contain the same number of neurons, which we denoted M, giving a total of $36M$ neurons. We created a single realization of the individual network with $M = 100$ and a second with $M = 1{,}000$, both with a random number of synapses as in Section 5.4. We compared the average firing rates of each individual neuron population with the firing rate of the corresponding population density. To obtain the firing rates for individual populations, we averaged over 20 passes of the stimulus for $M = 100$ and 2 passes of the stimulus for $M = 1{,}000$. Additional passes would have improved the match between the population density and individual neuron firing rates. However, since we were interested in comparing running times, we did not wish to slow the individual neurons simulations down to improve the results.

The comparison of populations with 0 deg and 90 deg orientation preference is shown in Fig. 21a–d. The individual neuron results are from the network with $M = 100$. For both orientation preferences, the

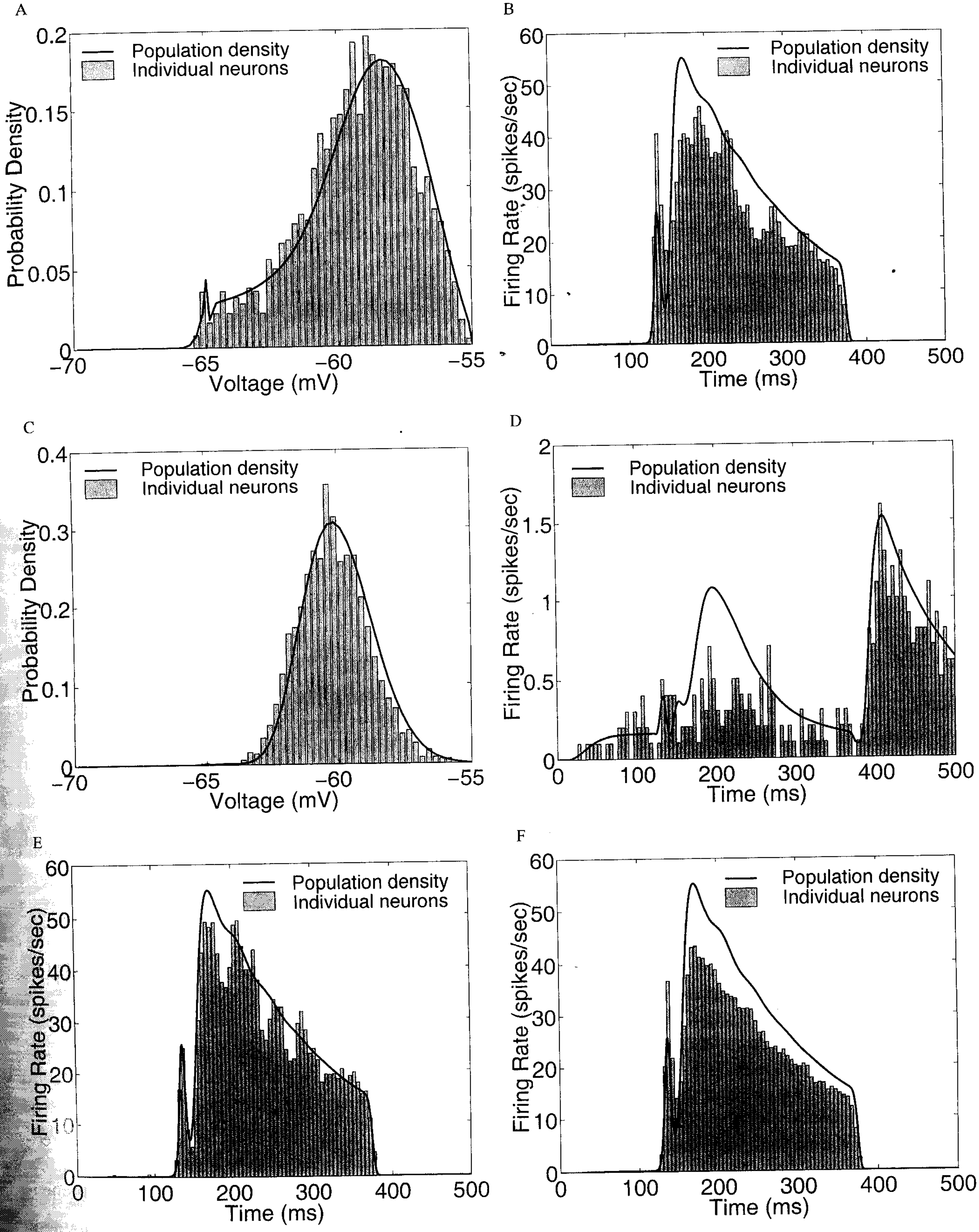

Figure 21. Comparison of excitatory neuron responses for the population density and individual neuron networks of a hypercolumn of visual cortex. Population density results are shown by solid lines; average individual neurons results are shown by the histograms. **A:** Probability density snapshot at $t = 300$ ms of excitatory neurons with 0 deg. The population density line is the 0 deg cross-section of Fig. 19. The individual neuron network contained $M = 100$ neurons at each orientation. **B:** Firing rates of the same networks in A. The population density line is the 0 deg cross-section of Fig. 18. **C, D:** Same as panels A and B, but at 90 deg. **E:** Same as panel B but with $M = 1000$. **F:** Same as panel B, except that individual neuron results are averaged over 10 realization of the network.

probability density snapshots at $t = 300$ ms for the two networks match well (panels A and C). The individual neuron firing rate matches most of the qualitative features of the population density firing rate (panels B and D). Much of the time, the results match quantitatively, as well. However, for 0 deg the individual neuron simulation overshoots the initial transient and undershoots the following maximum firing rate. For 90 deg the firing rates are so low that the individual neurons haven't fired enough spikes to obtain a good estimate of the firing rate. Nonetheless, the results match well, except for undershooting around $t = 200$ ms.

Increasing the size of the individual populations to $M = 1{,}000$ improves the match with the population density. As shown in Fig. 21e, the initial transient for the 0 deg population is reproduced well, and all other discrepancies are smaller than for $M = 100$. For the 90 deg population (not shown), the results differ little from the $M = 100$ case.

The 0 deg and 90 deg populations were representative of the other populations. Since the individual neuron results were based on one realization of the network, there was some variation due to the random connections. To test which differences for $M = 100$ were due to the random connectivity, we ran an individual neuron simulation with $M = 100$ and averaged over 10 passes of the stimulus for each of 10 realizations of the network. As seen in the comparison with the population density in Fig. 21f, there still are systematic difference between the population density network and the individual neuron network with $M = 100$. This difference is significantly reduced, but not eliminated, for $M = 1{,}000$ (not shown).

The population density simulations with 36 populations took 30 seconds for a 500 ms run and 0.5 ms time step. The equivalent individual neuron simulation took 530 seconds for 20 passes and $M = 100$; for 2 passes and $M = 1{,}000$, it required 1,090 seconds. Thus the individual neurons simulations took 17 and 35 times longer than the population density. The diffusion approximation sped the population density simulations to 5 seconds at the cost of the differences shown in Fig. 22. Thus, the diffusion approximation was over 100 times faster than the individual neuron simulation with $M = 100$ and over 200 times faster than the simulation with $M = 1{,}000$.

7. Discussion

We have explored a novel population density method for modeling large groups of interacting neurons that was introduced by Knight et al. (1996). A similar approach has recently been taken by Brunel and Hakim (1999). Two salient differences of Brunel and Hakim's approach are that they based their work on neurons that were driven by current rather than conductance changes, and they used a Fokker-Planck (diffusion) approximation for the partial differential-integral equation. The method of Knight and colleagues is further elaborated in Omurtag et al. (2000), Sirovich et al. (1999), and Knight (2000).

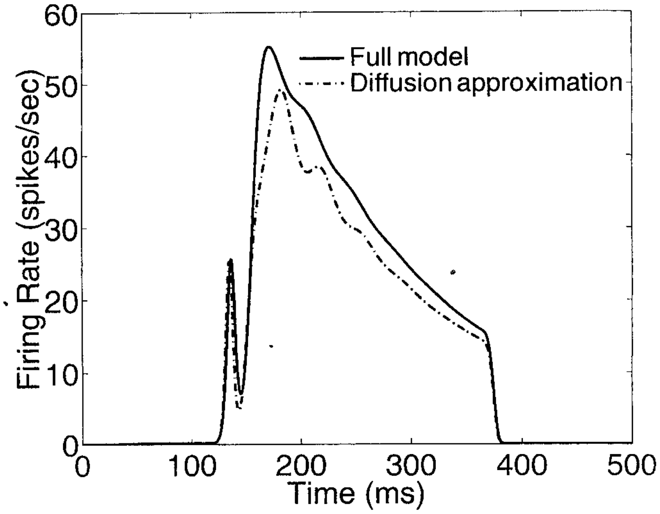

Figure 22. Comparison with the diffusion approximation. The firing rates of the excitatory population density with a 0 deg orientation preference are shown for the full model, the diffusion approximation. After the initial transient, the diffusion approximation fires more slowly than full model.

Our emphasis in the present article was on testing the population density method against direct individual neuron computations for various networks. We also presented conservative, second-order-accurate schemes for solving the partial differential-integral equations and associated, approximate diffusion-like equations in Appendices A and B.

7.1. *Previous Population Models*

Population density theory has a rich history in theoretical neuroscience. We mention below only a few examples and do not attempt to present a comprehensive review of the field.

Until the early 1990s the population density function had been applied, in large part, to analyzing the statistics of spike trains in individual neurons with prescribed synaptic input (cf. review in Tuckwell, 1988). However, the ideas behind modeling large populations of neurons go back to the early 1970s (e.g. Wilson and

Cowan, 1972, 1973; Knight, 1972a, 1972b; Amari, 1974). The seminal work of Wilson and Cowan (1972, 1973) was based on statistical mechanical-like reasoning and dealt explicitly with important nonlinearities in the dynamical behavior of neural networks. Knight (1972a, 1972b) related the activity of a single integrate-and-fire neuron to that of a population of such neurons and investigated the nonlinear behavior intrinsic to such neurons.

In the early 1990s, investigators began applying population density theory to populations of neurons that were coupled in an all-to-all manner. These studies were concerned with stability of the steady state, synchronous versus asynchronous firing activity, and collective oscillations (Kuramoto, 1991; Strogatz and Mirollo, 1991; Abbott and van Vreeswijk, 1993; Treves, 1993). An early application of the population density function to modeling the responses of interacting populations of sensory neurons is the study of Chawanya et al. (1993). They presented a model for feature linking through collective oscillations of neurons in the orientation columns of primary visual cortex. In this study, any pair of coupled populations was coupled in an all-to-all manner.

A novel integral equation method for modeling population activity has been analyzed by Gerstner and colleagues (Gerstner and van Hemmen, 1994; Gerstner, 1995, 1999). Gerstner's integral equation is a generalization of the integral equation of Wilson and Cowan; it assumes all-to-all coupling and is applicable for certain kinds of noise when synaptic input is modeled as current injection. Very recently Pham et al. (1998) analyzed activity in a sparsely connected excitatory network by a probability approach in which states and time were discretized. Tanabe et al. (1998) used a Fokker-Planck approximation to analyze information transfer by a population of leaky integrate-and-fire neurons. Barna et al. (1998) simulated rhythmic activity in a hippocampal CA3 slice using probability density functions that described the states of both neurons and propagating spikes; this probability density approach was subsequently used by Adorján et al. (1999) to model orientation selectivity in the primary visual cortex. Finally, the paper of Brunel and Hakim (1999) concerns itself with global oscillations in a sparsely connected population of integrate-and-fire neurons.

The methods of Knight and colleagues (Knight et al., 1996; Omurtag et al., 2000; Sirovich et al., 1999; Knight, 2000) and Brunel and Hakim (1999) are unusual in the way the interacting populations are coupled. The coupling mechanism provides a natural, intrinsic source of noise that gives asynchronous behavior over a wide range of conditions. Thus, the model has no external, independent source of noise. However, other noise sources could easily be accommodated.

7.2. *Analysis of Assumptions*

We have demonstrated that, when the assumptions underlying the population density method (Section 3.1) are met, the method of Knight et al. gives the same results as direct individual neuron computations in a fraction of the time. Furthermore, our sample computations show that the population density results remain valid over a larger range of conditions than implied by these assumptions. Some assumptions, however, are fundamental to the current approach, and deviations from the assumptions would require modifications of the population density equations to guarantee good agreement between the population density and individual neuron behavior.

The derivation of the population density equations was based on four assumptions. First, each population contained a large number of identical integrate-and-fire point neurons. Second, each neuron in the population received excitatory and inhibitory input with the same average rate. Third, the arrival times of the synaptic events were conditionally independent random variables given by a modulated Poisson process. As explained below, this assumption implies that the networks are sparsely coupled. Fourth, we used a simple single compartment neuron model where the synapses were fast compared to the membrane time constant.

7.2.1. A Large Number of Identical Neurons. We have shown a close similarity between population density results and direct individual neuron results when we performed computations on only one realization of the random individual neuron network with as few as 100 neurons per population. It thus appears that the population density method provides a good approximation to the behavior of networks even when the number of neurons in each group is not vast.

7.2.2. Identical Input Rates. In the Knight et al. population framework, each member of a target population is assumed to receive synaptic inputs from neurons in a source population at a rate that is proportional to the population firing rate of the source. The constant of proportionality is the number of synapses made by the source population on a single target neuron. The input rate, and thus the number of synapses, for each neuron in a given population is assumed to be identical.

However, our individual neuron computations have shown that this assumption is not necessary. The match between the firing rates of the population density and individual neuron simulations is not harmed by the introduction of randomness in the number of synaptic connections. In most of our test cases, the introduction of randomness improved the match. This result is puzzling because the assumption of a common input rate for each neuron in a given population can be regarded as a mean-field approximation for synaptic input rates. We would expect that this mean-field approximation would introduce additional errors.

We do not have a completely satisfactory explanation for the almost universal improvement of the error with randomness. For networks with very low connectivity (e.g. $\bar{W} = 10$ in (38)), we posit the following explanation. The error with very low connectivity is due to insufficient numbers of converging inputs to achieve a modulated Poisson process. With random numbers of synapses, some neurons receive a larger number of synpases. These additional converging inputs would cause the input to these neurons to be closer to Poisson. If the additional inputs were excitatory, these neurons would fire faster and contribute more to the population firing rate than other neurons, helping the firing rate be closer to that of the population density. It is unclear why this positive effect would be stronger than the adverse effect of neurons firing faster due to a decreased number of inhibitory inputs. By the above reasoning, those neurons should have input that is further from Poisson. Nonetheless, this argument does explain why the largest improvement is observed in networks with very low connectivity.

7.2.3. The Poisson Assumption. Despite the common synaptic input rate, neurons in a target population receive inputs that differ in detail. The arrival times of excitatory/inhibitory synaptic inputs are assumed to be governed by a modulated Poisson process at a rate that is the net rate of excitatory/inhibitory inputs. This net rate is obtained by summing over source populations. In this method, the input Poisson processes for all neurons in a given population are assumed to be conditionally independent—that is, independent given the average rate. This conditional independence can be satisfied exactly only if target neurons in each population do not share inputs from any individual source neurons. Hence, sparse connectivity is an implicit assumption in the model. A similar argument was used by Brunel and Hakim (1999).

Sparse connectivity is implicit for another reason as well. In the model, the postsynaptic conductance amplitudes are chosen to give postsynaptic potential amplitudes in the physiological regime. This choice of conductance amplitudes is incompatible with dense or all-to-all coupling because the population density represents the limit of an infinite number of neurons. A dense coupling would imply that, in the limit, each neuron would receive an infinite number of synaptic connections and thus would receive input at an infinite rate. This limit cannot give a sensible limiting evolution equation for the population density function unless the unitary postsynaptic conductances approach zero in such a way that the product of the input rate and the unitary postsynaptic conductance remains order 1.

Sparse connectivity is a Feature of our implementation of the population density approach because we have used the resulting Poisson distributed input as the major source of noise in our networks. In many of the models mentioned above, other local noise sources were added to all-to-all coupled models. The addition of other noise sources to our model may improve the match with direct individual neuron simulations when the connectivity is not sparse.

The assumption of Poisson input to each neuron may seem troubling at first glance because there is no guarantee that each neuron fires action potentials in a Poisson manner. However, a limit theorem of point processes (Çinlar, 1972) says that a composite point process formed by superposing a number of uniformly sparse, conditionally independent non-Poisson point processes approaches a Poisson process as the number of components approaches infinity. Thus, if each neuron receives enough conditionally independent synaptic inputs, the overall input point process should be close to Poisson. The fact that a typical neuron has thousands of synaptic inputs argues in favor of the validity of the Poisson assumption.

The good match between population density results and individual neuron results for several neural networks presented in this article suggests that the Poisson approximation is good even for a modest number of synaptic inputs from sources with some overlap. However, the assumptions of the model do not allow one to model faithfully a network composed of small subpopulations in which neurons in any subpopulation share a substantial fraction of their synaptic inputs.

7.2.4. ***Simple Neuron Model.*** Our implementation of the population density theory in the present article employed a very simple model for the individual neurons in the populations. The individual neuron, which is in reality spatially extended with extensively branched dendritic tree and axonal arbor, was collapsed into a single isopotential compartment (point neuron). The complex time- and voltage-dependent conductances that underlie the action potential were modeled crudely by a simple conductance-driven integrate-and-fire mechanism.

Furthermore, the unitary postsynaptic conductance waveform was modeled as a delta function with random magnitude. We have only shown so far that the population density method is much faster than the point neuron method for integrate-and-fire neurons with instantaneous postsynaptic conductance waveforms. The approximation of instantaneous postsynaptic conductances is not always appropriate. For example, the time course of conductances governed by class B γ-aminobutyric acid (GABA$_B$) receptors can be quite slow (Howe et al., 1987). Accurate modeling of neurons containing such conductances by the population density method would require introducing at least one additional argument into the population density function. If the inhibitory conductance is assumed to jump up instantaneously on arrival of a single synaptic input and then to decay exponentially, the state of the neuron would be completely described by its voltage and the value of its inhibitory conductance.

It is presently unclear how computation time for the population density method will compare with that of the point neuron method for more complicated underlying models for the individual neuron, such as the one above. The dimension of the population density increases with the number of variables used to describe the neuron. Thus, the computation time required for the population density simulations increases rapidly as the complexity of the neuron model increases. For more complex neurons, additional techniques, such as the principal components analysis method by Knight and colleagues (see Section 7.4), may be required for the population density method to retain its speed advantages.

7.3. A Comparison to Rate Models

The population density approach captures dynamics of neurons that cannot be contained in equations involving only the mean rate of each population. For example, firing rates in the population density model can be dramatically altered by changes that leave the expected synaptic input unchanged and thus would not affect firing rate equations. When the unitary postsynaptic conductances and input rates are scaled in inverse proportion such that the expected synaptic input remains unchanged, the population density changes its behavior. The effect of such a scale change can be seen very clearly in the diffusion approximation for the partial differential-integral equation in which the diffusion coefficient is scaled by the synaptic conductance scale factor.

Another difference between the population density model and firing-rate models is that firing-rate models give only mean firing rate, while the population density model also contains information on fluctuations. The population density model allows one to compute any moment of the spike count in any time interval.

A simple example is provided by specializing to the steady state for the model considered in this paper. In this case, spike firing is a renewal process in which all moments of the random number of events (spike count in this case) in any time interval are completely determined by the probability density function for the interevent interval (interspike interval in this case). The probability density function for the interspike interval can be computed as follows. First one solves for the steady-state firing rates. Then, using those steady input rates and starting with a delta function probability density function at v_{reset}, one solves the partial differential-integral equation for a given population without reseting neurons after they fire. Neurons that fire are simply removed from the population. The flux across threshold gives the probability per unit time that a neuron crosses threshold (fires a spike), and, in this case, every neuron that fires a spike is firing for the first time since reset. Thus, the flux across threshold can be interpreted as the probability density function for the interspike interval.

Specifically, the population density method captures dynamics, such as rapid transients, that are missed by the classical Wilson-Cowan-like differential equation (Wilson and Cowan, 1972, 1973) for firing rate of each population:

$$\tau_r \frac{dr^k(t)}{dt} + r^k(t) = g\left(\int_0^\infty \sum_{j=0}^{N} C_{jk}(s) r^j(t-s)\, ds\right) \tag{41}$$

for $k = 1, 2, \ldots, N$, $N =$ the number of populations. Here $r^0(t)$ is the firing rate of external sources, and $r^k(t)$ for $k \geq 1$ is the firing rate of the kth population. Input from population j to population k is filtered through the kernel $C_{jk}(s)$; $g(\cdot)$ is the transfer function of the population, and τ_r is the time constant at which the firing rate evolves.

The limitations of Eq. (41) have been discussed by others (Wilson and Cowan, 1972; Abbott and van Vreeswijk, 1993; Gerstner, 1995, 1999). Gerstner (1999) demonstrated that the transient response of the population changes at the time scale of the postsynaptic conductances while the longer-time average response evolves more slowly. An equation of the form of (41) cannot capture those two time scales. A demonstration of a single population to a step in input rate is shown in Fig. 23 (designed after Gerstner, 1999, Fig. 5). Since in our model, the postsynaptic conductance time course is modeled as a delta function, the firing rate changes instantaneously with the synaptic input before settling to a new steady state. This instantaneous dependence results from the fact that the population firing rate is equal to the excitation flux evaluated at the threshold voltage and the fact that this flux (31) depends instantaneously on the excitatory synaptic input rate $\nu_e(t)$.

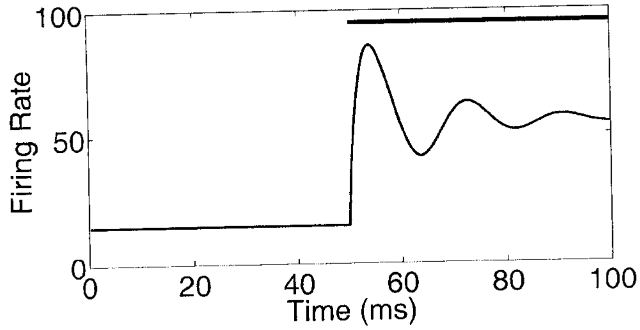

Figure 23. The transient response of the population firing rate to a rapid change in input. Beginning at $t = 50$, the excitatory input rate to the a single population of uncoupled neurons is stepped up from 1,000 Hz to 2,000 Hz (shown by black bar). The firing rate transiently jumps up before returning to a new steady-state response.

7.4. *Computational Speed*

The population density method was dramatically faster than the individual neuron method even when we solved the exact system of partial differential-integral equations. A further reduction in computation time was achieved by a numerical method giving a diffusion-like approximation. Furthermore, these savings in computation time were obtained without any attempt to find a fast algorithm for solving the population density equations. Knight and colleagues (Knight, 2000) have found that computation time can be substantially reduced even further by using principal-component analysis to approximate the population density function by a superposition of components in a function space of low dimension (~10).

The speed of the Knight et al. population density approach will facilitate modeling more complex neural networks than one could realistically simulate with direct individual neuron models. Provided that a neural network can be grouped into populations with hundreds of similar, sparsely coupled neurons, the population density implementation of the network can quickly produce accurate population firing rates and distributions of neurons across voltage. Thus, the population density approach may be an important tool in the implementation of truly large-scale models of the networks in the brain.

Appendix A. Derivation of Diffusion Approximation

The diffusion approximation is based on the assumption that synaptic inputs are small and thus the voltage jumps due to synaptic inputs are small. Correspondingly, in the synaptic flux Eqs. (31) and (32), the integrals at a voltage v are determined mainly by the values of $\rho(v', t)$ with $v' - v$ small. These integrals can thus be replaced with expressions involving the derivatives of $\rho(v, t)$, giving a diffusion equation for the evolution equation for $\rho(v, t)$.

The diffusion approximation assumes that ρ is sufficiently smooth so that we can approximate the value of $\rho(v', t)$ with the first two terms of a Taylor series centered around v:

$$\rho(v', t) = \rho(v, t) + (v' - v)\frac{\partial \rho}{\partial v}(v, t) + \frac{(v' - v)^2}{2}\frac{\partial^2 \rho}{\partial v^2}(\xi, t), \tag{42}$$

where ξ is between v' and v.

The error that we make by neglecting the third term will be less than

$$M(v' - v)^2, \tag{43}$$

where

$$M = \frac{1}{2} \max_{v,t} \left| \frac{\partial^2 \rho}{\partial v^2}(v,t) \right|. \tag{44}$$

To calculate the approximation for the excitation flux, we substitute the first two terms of the Taylor series for $\rho(v',t)$ (42) into the flux Eq. (31) to obtain

$$J_e(v,t) = \nu_e(t) \left[c_{1e}(v)\rho(v,t) - c_{2e}(v)\frac{\partial \rho}{\partial v}(v,t) \right], \tag{45}$$

where

$$c_{1e}(v) = \int_{\mathcal{E}_i}^{v} \tilde{F}_{\Gamma_e^*}\left(\frac{v - v'}{\mathcal{E}_e - v'} \right) dv', \tag{46}$$

and

$$c_{2e}(v) = \int_{\mathcal{E}_i}^{v} \tilde{F}_{\Gamma_e^*}\left(\frac{v - v'}{\mathcal{E}_e - v'} \right) (v - v')\, dv'. \tag{47}$$

The error in this approximation is less than (see (43))

$$\nu_e(t) M \int_{\mathcal{E}_i}^{v} \tilde{F}_{\Gamma_e^*}\left(\frac{v - v'}{\mathcal{E}_e - v'} \right) (v' - v)^2\, dv'. \tag{48}$$

The error term can be simplified by changing variables in the integral to $\gamma = \frac{v - v'}{\mathcal{E}_e - v'}$, giving

$$\begin{aligned} &\nu_e(t) M (\mathcal{E}_e - v)^3 \int_0^{\frac{v - \mathcal{E}_i}{\mathcal{E}_e - \mathcal{E}_i}} \tilde{F}_{\Gamma_e^*}(\gamma) \frac{\gamma^2 d\gamma}{(1-\gamma)^4} \\ &\quad < \nu_e(t) M (\mathcal{E}_e - \mathcal{E}_i)^3 \int_0^1 \tilde{F}_{\Gamma_e^*}(\gamma) \frac{\gamma^2 d\gamma}{(1-\gamma)^4}, \end{aligned} \tag{49}$$

since $\mathcal{E}_i < v < \mathcal{E}_e$.

The integral on the right in (49), which we denote by I, can be simplified through integration by parts:

$$\begin{aligned} I &= \int_0^1 \tilde{F}_{\Gamma_e^*}(\gamma) \frac{\gamma^2 d\gamma}{(1-\gamma)^4} \\ &= -\frac{1}{3} + \int_0^1 f_{\Gamma_e^*}(\gamma) \frac{(1-\gamma)^3 + \gamma^3}{3(1-\gamma)^3} d\gamma \\ &= \int_0^1 f_{\Gamma_e^*}(\gamma) \left[\frac{(1-\gamma)^3 + \gamma^3}{3(1-\gamma)^3} - \frac{1}{3} \right] d\gamma \end{aligned} \tag{50}$$

$$= \int_0^1 f_{\Gamma_e^*}(\gamma) \frac{\gamma^3 d\gamma}{3(1-\gamma)^3}, \tag{51}$$

where

$$f_{\Gamma_e^*}(\gamma) = -\frac{\partial}{\partial \gamma} \tilde{F}_{\Gamma_e^*}(\gamma) \tag{52}$$

is the probability density function for Γ_e^*. Note that

$$\int_0^1 f_{\Gamma_e^*}(\gamma)\, d\gamma = 1 \tag{53}$$

since $0 < \Gamma_e^* < 1$. Also, implicit in the calculation of the boundary term is the fact that

$$\tilde{F}_{\Gamma_e^*}(0) = 1 \tag{54}$$

and the assumption that

$$\lim_{\gamma \to 1^-} \frac{\tilde{F}_{\Gamma_e^*}(\gamma)}{(1-\gamma)^3} = 0. \tag{55}$$

Substituting the value of the integral I from (51) into expression (49) for the error, we find that the error in the diffusion approximation is less than

$$\nu_e(t) \frac{M}{3} (\mathcal{E}_e - \mathcal{E}_i)^3 \int_0^1 f_{\Gamma_e^*}(\gamma) \frac{\gamma^3 d\gamma}{(1-\gamma)^3}. \tag{56}$$

If the excitatory synaptic conductances are small, then $f_{\Gamma_e^*}(\gamma)$ is almost zero except for small γ, and the value of the integral in the error bound (56) is the third moment of Γ_e^* to smallest order in γ. Thus, the error is proportional to the third moment of Γ_e^*, which is small when the excitatory synaptic conductances are small.

In the same way, the full diffusion approximation for the inhibition flux is

$$J_i(v,t) = -\nu_i(t) \left[c_{1i}(v)\rho(v,t) + c_{2i}(v)\frac{\partial \rho}{\partial v}(v,t) \right], \tag{57}$$

where

$$c_{1i}(v) = \int_{v}^{v_{th}} \tilde{F}_{\Gamma_i^*}\left(\frac{v - v'}{\mathcal{E}_i - v'} \right) dv', \tag{58}$$

and

$$c_{2i}(v) = \int_{v}^{v_{th}} \tilde{F}_{\Gamma_i^*}\left(\frac{v - v'}{\mathcal{E}_i - v'} \right) (v' - v)\, dv'. \tag{59}$$

The error bound for the inhibition flux is

$$\nu_i(t)\frac{M}{3}(v_{th}-\mathcal{E}_i)^3\int_0^1 f_{\Gamma_i^*}(\gamma)\frac{\gamma^3 d\gamma}{(1-\gamma)^3}, \tag{60}$$

where $f_{\Gamma_i^*}(\gamma)$ is the probability density function for Γ_i^*.

Combining Eqs. (45) and (57) with Eqs. (28) to (30), we obtain a diffusion equation for the evolution of $\rho(v,t)$ (35). This diffusion equation can be solved numerically more quickly than the full partial differential-integral equation.

Appendix B. Numerical Method for Solving Partial Differential-Integral Equation

To solve Eqs. (28) to (32) numerically, we discretized $\frac{\partial J}{\partial v}$ in voltage and solved the resulting set of ODEs using the trapezoid rule. For the diffusion approximation, this resulted in the Crank-Nicolson method (Crank and Nicolson, 1947). We used the second order in voltage discretization described below for the partial differential-integral equation.

B.1. *Discretization of the Flux*

We divided $(\mathcal{E}_i, v_{th})$ into K intervals so that $\mathcal{E}_i$, $\mathcal{E}_r$, and v_{th} were half grid points. With interval lengths $\Delta v = (v_{th}-\mathcal{E}_i)/K$, the grid points were numbered $v_k = \mathcal{E}_i + (k-\frac{1}{2})\Delta v$, for $k = 1, 2, \ldots, K$. We let $\rho_k = \rho(v_k, t)$.

We discretized $\frac{\partial J_l}{\partial v}$, $\frac{\partial J_e}{\partial v}$, and $\frac{\partial J_i}{\partial v}$ separately.

B.1.1. Discretization of the Leakage Flux. For $v < \mathcal{E}_r$, the leakage moves probability upward and for $v > \mathcal{E}_r$, the leakage moves probability downward. To ensure stability, we used a downward second-order approximation for the derivative for $v < \mathcal{E}_r$ and a upward approximation for $v > \mathcal{E}_r$.

For $v_k < \mathcal{E}_r$, we let

$$\begin{aligned}\frac{\partial J_l}{\partial v}(v_k,t) = {}& [-(v_{k-2}-\mathcal{E}_r)\rho_{k-2} + 4(v_{k-1}-\mathcal{E}_r)\rho_{k-1}\\ &-3(v_k-\mathcal{E}_r)\rho_k]/(2\tau\Delta v),\end{aligned} \tag{61}$$

and for $v_k > \mathcal{E}_r$, we let

$$\begin{aligned}\frac{\partial J_l}{\partial v}(v_k,t) = {}& [3(v_k-\mathcal{E}_r)\rho_k - 4(v_{k+1}-\mathcal{E}_r)\rho_{k+1}\\ &+(v_{k+2}-\mathcal{E}_r)\rho_{k+2}]/(2\tau\Delta v).\end{aligned} \tag{62}$$

B.1.2. Discretization of Excitation Flux. To obtain a second-order scheme, we used a centered difference on the half grid points to discretize the spatial derivatives of J_e at the grid points:

$$\frac{\partial J_e}{\partial v}(v_k,t) = \frac{J_e\left(v_{k+\frac{1}{2}},t\right) - J_e\left(v_{k-\frac{1}{2}},t\right)}{\Delta v}. \tag{63}$$

To complete our second-order scheme, we needed a third-order approximation to the integral $J_e(v,t)$, since it would be divided by Δv in the derivative approximation. To obtain this, we changed variables to

$$\gamma = \frac{v-v'}{\mathcal{E}_e - v}.$$

To compute the resulting $\rho(\frac{v-\gamma\mathcal{E}_e}{1-\gamma})$ that appears in the integral, we interpolated ρ from the known values at neighboring grid points with a quadratic polynomial. This procedure gave a third-order approximation of ρ. The resulting integral was independent of ρ, and we computed it using a midpoint rule in γ, decreasing the discretization interval in γ until we achieved a relative error of less than 10^{-5}.

B.1.3. Discretization of Inhibition Flux. We discretized the inhibition flux in the same way as the excitation flux. We used a centered difference with the half grid points to discretize the spatial derivatives and computed the integrals using the change of variables:

$$\gamma = \frac{v-v'}{\mathcal{E}_i - v'}.$$

ρ was interpolated with quadratic polynomials from the known values at grid points, and the resulting integrals, which were independent of ρ, were computed to a relative error of 10^{-5}.

B.2. *Delta Function Modification*

When $v_{reset} = \mathcal{E}_r$, the flux reset to v_{reset} does not advect away. Thus the delta function source in Eq. (28) builds up, forming a delta function component of ρ at v_{reset}. Convergence estimates of our numerical methods depend on existence of derivatives of ρ, which clearly wouldn't exist at v_{reset}.

To improve our numerical method so that it can handle this situation, we divide ρ into two components, a smooth component ρ_s and a delta function at v_{reset} with weight ρ_δ:

$$\rho(v,t) = \rho_s(v,t) + \delta(v - v_{reset})\rho_\delta(t). \quad (64)$$

ρ_s should then have sufficient derivatives in v, allowing our methods to converge well.

The evolution of $\rho_s(v,t)$ is determined by the components of $\frac{\partial J}{\partial v}$ that are smooth. The evolution of $\rho_\delta(t)$ is determined by the explicit delta function $\delta(v - v_{reset})J(v_{th}, t)$ as well as delta function components of $\frac{\partial J}{\partial v}$. In this section, we derive these modified evolution equations.

Since the delta function sits at $\mathcal{E}_r$, it is not affected by the leakage. Thus the equation for the leakage flux (30) contains only a smooth component:

$$J_l(v,t) = -\frac{1}{\tau}(v - \mathcal{E}_r)\rho_s(v,t). \quad (65)$$

Combining the new expression for ρ (B4) with the excitation flux (31), we obtain a new expression for the excitation flux:

$$J_e(v,t) = J_{e,s}(v,t) + \mathcal{H}(v - v_{reset})J_{e,\delta}(v,t), \quad (66)$$

where

$$J_{e,s}(v,t) = \nu_e(t)\int_{\mathcal{E}_i}^{v} \tilde{F}_{\Gamma_e^*}\left(\frac{v - v'}{\mathcal{E}_e - v'}\right)\rho_s(v',t)\,dv', \quad (67)$$

$$J_{e,\delta}(v,t) = \nu_e(t)\tilde{F}_{\Gamma_e^*}\left(\frac{v - v_{reset}}{\mathcal{E}_e - v_{reset}}\right)\rho_\delta(t), \quad (68)$$

and $\mathcal{H}(y)$ is the Heaviside unit step function:

$$\mathcal{H}(y) = \begin{cases} 1 & \text{if } y > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (69)$$

Similarly, the new expression for the inhibition flux is

$$J_i(v,t) = J_{i,s}(v,t) + \mathcal{H}(v_{reset} - v)J_{i,\delta}(v,t), \quad (70)$$

where

$$J_{i,s}(v,t) = -\nu_i(t)\int_{v}^{v_{th}} \tilde{F}_{\Gamma_i^*}\left(\frac{v - v'}{\mathcal{E}_i - v'}\right)\rho_s(v',t)\,dv' \quad (71)$$

and

$$J_{i,\delta}(v,t) = -\nu_i(t)\tilde{F}_{\Gamma_i^*}\left(\frac{v - v_{reset}}{\mathcal{E}_i - v_{reset}}\right)\rho_\delta(t). \quad (72)$$

Plugging these modified flux expressions (65), (66), and (70) into the original evolution Eq. (28) will result in both smooth and delta function components on the right-hand side. (Some of the delta function components arise from the fact that $\frac{d\mathcal{H}}{dv}(v) = \delta(v)$.) Using Eq. (64) and separating the smooth components and the coefficients of $\delta(v - v_{reset})$ into two equations gives the following evolution equations:

$$\frac{\partial \rho_s}{\partial t} = -\frac{\partial}{\partial v}(J_l + J_{e,s} + J_{i,s}) - \mathcal{H}(v - v_{reset})\frac{\partial J_{e,\delta}}{\partial v} - \mathcal{H}(v_{reset} - v)\frac{\partial J_{i,\delta}}{\partial v}, \quad (73)$$

$$\frac{d\rho_\delta}{dt} = -[\nu_e(t) + \nu_i(t)]\rho_\delta + r(t), \quad (74)$$

where

$$r(t) = J(v_{th}, t) = J_{e,\delta}(v_{th}, t) + J_{e,s}(v_{th}, t). \quad (75)$$

In deriving these equations, we used the fact that $\tilde{F}_{\Gamma_{e/i}^*}(0) = 0$ and $J(v_{th}, t) = J_e(v_{th}, t)$.

The numerical method above can be easily modified for the new equations.

B.3. *Accuracy of Numerical Method*

To determine the accuracy of the numerical method, we first verified that our method was indeed second-order accurate in both voltage and time. Through this verification process, we also determined how small we needed to make the discretizations to be in the second-order accurate regime. We determined that we already achieved second-order accuracy in time with a time step of $\Delta t = 1$ ms and achieved second-order accuracy in voltage with a voltage discretization of $\Delta v = 0.25$ mV ($K = 60$).

With a voltage discretization of $\Delta v = 0.25$ mV ($K = 60$), the maximum error in the population firing rate was a few spikes/second. We decided to use $K = 60$ for all our population density simulations, including those from Section 5 where we were focusing on accuracy.

We chose a time step Δt so that additional error introduced by the time discretization was less than the error from the voltage discretization. We used $\Delta t = 0.5$ ms for all orientation tuning simulations as well as single population simulations. For Section 5, since the focus of the section was accuracy and not speed, we used a smaller time step of $\Delta t = 0.25$ ms just to ensure that our error percentages were accurate. The error percentages changed by about 1% when we moved from $\Delta t = 0.5$ ms to $\Delta t = 0.25$ ms.

Appendix C. Event-Driven Simulation for Solving Individual Neuron Networks

Since we have reduced the unitary synaptic conductances to delta functions (see Section 2.2, the evolution equation for the voltage of an integrate-and-fire neuron (1) can be easily solved analytically. $G_{e/i}(t) = 0$ between synaptic events, and the voltage decays exponentially to $\mathcal{E}_r$ until the neuron receives synaptic input, when the voltage jumps the amount in Eq. (3). We thus need to update each neuron's voltage only when it receives a synaptic event. This procedure is much faster than a time stepping algorithm and gives an exact solution, up to limits imposed by machine precision.

When a neuron fires, we schedule synaptic events for each of its postsynaptic neurons at a time offset by the delay for each synapse. These events are added to the queue of upcoming events. Since the delays are random, the order of the events differs from the order in which the events were created. An important task is determining which event is next. Thus, the implementation of a priority queue is central to the event-driven simulation.

A priority queue is a queue in which the first item in the queue is always the one with the highest priority, which in this case is the earliest time. We implemented our priority queue using a heap, which obtains the earliest event in $O(n \log n)$ steps, where n is the number of events in the queue.

To run the event-driven simulation, we kept track of the time of each neuron's last synaptic input and the voltage of the neuron after that input. Each step consisted of taking the next event off the priority queue and acting on that event. Each event included the time of the event, the neuron affected, and whether the event was excitatory or inhibitory. Based on that neuron's previous state and the information from the event, we could update the neuron's voltage.

For example, if the next event was an excitatory input for neuron j at time t, and neuron j had a voltage of v_o at its previous input at time t_o, we would update its voltage to

$$v = \mathcal{E}_e + e^{-\Gamma_e}\left[\mathcal{E}_r + (v_o - \mathcal{E}_r)e^{-(t-t_o)/\tau} - \mathcal{E}_e\right]. \quad (76)$$

If $v > v_{th}$, we would then record a spike, reset v to v_{reset}, and schedule events for the postsynaptic neurons of neuron j.

Appendix D. Parameters

For all simulations, we used a gamma distribution for the distribution of sizes of unitary synaptic conductances:

$$f_{\Gamma_{e/i}}(x) = \frac{\exp(-x/a_{e/i})}{a_{e/i}(n_{e/i}-1)!}\left(\frac{x}{a_{e/i}}\right)^{n_{e/i}-1}, \quad (77)$$

where $f_{\Gamma_{e/i}}(x)$ is the probability density function of $\Gamma_{e/i}$. Since $\Gamma^*_{e/i} = 1 - \exp(-\Gamma_{e/i})$, the probability density function of $\Gamma^*_{e/i}$ is

$$f_{\Gamma^*_{e/i}}(x) = \frac{f_{\Gamma_{e/i}}(-\log(1-x))}{1-x}.$$

Therefore, the complementary cumulative distribution function for $\Gamma^*_{e/i}$ is

$$\tilde{F}_{\Gamma^*_{e/i}}(x) = \int_x^1 \frac{f_{\Gamma_{e/i}}(-\log(1-y))}{1-y}dy \quad (78)$$

$$= e^{-u(x)}\sum_{l=0}^{n_{e/i}-1}\frac{u(x)^l}{l!}, \quad (79)$$

where $u(x) = -\log(1-x)/a_{e/i}$.

The average values of $\Gamma_{e/i}$ corresponded to postsynaptic potentials that were half those used in the model by Somers et al. (1995). We used the following average values: $\mu_{\Gamma_e} = 0.008$, $\mu_{\Gamma_i} = 0.027$ for excitatory neurons, and $\mu_{\Gamma_e} = 0.020$, $\mu_{\Gamma_i} = 0.066$ for inhibitory neurons. We chose a coefficient of variation of 0.5 for all Γs. This determined the values of $a_{e/i}$ and $n_{e/i}$ for both excitatory and inhibitory neurons.

We used the following parameters for both excitatory and inhibitory neurons: $\mathcal{E}_i = -70$ mV, $\mathcal{E}_r = v_{reset} = -65$ mV, $v_{th} = -55$ mV, and $\mathcal{E}_e = 0$ mV. For excitatory neurons, $\tau = 20$ ms and $\tau_{ref} = 3$ ms. For inhibitory neurons, $\tau = 10$ ms and $\tau_{ref} = 1$ ms.

For simulations involving a single population, we used the parameters for excitatory neurons.

The distribution of synaptic latencies $\alpha(t)$ were chosen so that the mean latency was 3 ms and the standard deviation was 1 ms, approximately. We used a gamma distribution of order 9 with mean 3 ms, which we truncated for $t > 7.5$ ms:

$$\alpha(t) = \begin{cases} \bar{\alpha} \dfrac{\exp(-t/\tau_\alpha)}{\tau_\alpha (n_\alpha - 1)!} \left(\dfrac{t}{\tau_\alpha}\right)^{n_\alpha - 1} & 0 \leq t \leq 7.5 \text{ ms} \\ 0 & \text{otherwise,} \end{cases} \tag{80}$$

where $n_\alpha = 9$, $\tau_\alpha = 1/3$ ms, and $\bar{\alpha}$ is a constant so that $\alpha(t)$ integrates to one. We used the same $\alpha(t)$ for each type of connection.

Acknowledgments

We gratefully acknowledge the extensive advice on mathematics and physiology given though numerous discussions with David McLaughlin and Robert Shapley. We also thank Charles Peskin and Michael Shelley for helpful input on theory and numerical methods. This material is based on work supported under a National Science Foundation Graduate Fellowship.

Notes

1. Throughout this article, we use an upper-case V to denote the random, fluctuating voltage of a single integrate-and-fire neuron and a lower-case v to denote fixed voltages.
2. This modification is equivalent to subtracting

$$J(v_{th}, t)\mathcal{H}(v - v_{reset})$$

from $J(v, t)$ reflecting the negative flux of neurons moving from v_{th} to v_{reset}. Here $\mathcal{H}(x)$ is the Heaviside step function, $\mathcal{H}(x) = 1$ for $x \geq 1$, $\mathcal{H}(x) = 0$ otherwise. Without modification, (10) does not conserve probability for $v \in (\mathcal{E}_i, v_{th})$, since, in general, $J(v_{th}, t) \neq 0$.

3. The Poisson nature of the random synaptic input times is required because we ignore past history when calculating the probability of an input at time t. Implicit in this calculation is the assumption that the probability of an input in nonoverlapping intervals is independent—that is, the assumption that the inputs are a modulated Poisson process. If one wanted to assume a more complicated distribution of inputs, one would have to track the probabilities of past input sequences. For example, for another renewal process, one could add an extra dimension to the population density (say $\rho(v, s, t)$, where $s =$ the time since the last input), and calculate a more complicated expression for the excitation flux.
4. In reality this may not be a good assumption for inhibitory synapses. See Discussion.

References

Abbott LF, van Vreeswijk C (1993) Asynchronous states in networks of pulse-coupled oscillators. *Phys. Rev. E* 48:1483–1490.

Adorján P, Barna G, Érdi P, Obermayer K (1999) A statistical neural field approach to orientation selectivity. *Neurocomputing*, 26–27:477–482.

Amari S (1974) A method of statistical neurodynamics. *Kybernetik* 14:201–215.

Barna G, Grőbler T, Érdi P (1998) Statistical model of the hippocampal CA3 region, II. The population framework: Model of rhythmic activity in the CA3 slice. *Biol. Cybern.* 79:309–321.

Ben-Yishai R, Lev Bar-Or R, Sompolinsky H (1995) Theory of orientation tuning in visual cortex. *Proc. Natl. Acad. Sci. USA* 92:3844–3848.

Brunel N, Hakim V (1999) Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. *Neural Comput.* 11:1621–1671.

Carandini M, Ringach D (1997) Predictions of a recurrent model of orientation selectivity. *Vision Res.* 21:3061–3071.

Chawanya T, Aoyagi A, Nishikawa I, Okuda K, Kuramoto Y (1993) A model for feature linking via collective oscillations in the primary visual cortex. *Biol. Cybern.* 68:483–490.

Çinlar E (1972) Superposition of point processes. In PAW Lewis, ed. Stochastic Point Processes: Statistical Analysis, Theory, and Applications. Wiley, New York, NY. pp. 549–606.

Crank J, Nicolson P (1947) A practical method for numerical evaluation of solutions for partial differential equations of the heat conduction type. *Proc. Camb. Philos. Sco.* 43:50–67.

Gerstner W (1995) Time structure of the activity in neural network models. *Phys. Rev. E* 51:738–758.

Gerstner W (1999) Population dynamics of spiking neurons: fast transients, asynchronous states, and locking. *Neural Comput.* (to appear).

Gerstner W, van Hemmen JL (1994) Coding and information processing in neural networks. In E Domany, JL van Hemmen, K Schulten, eds. Models of Neural Networks II. Springer-Verlag, New York, NY. pp. 1–93.

Hansel D, Sompolinsky H (1996) Chaos and synchrony in a model of a hypercolumn in visual cortex. *J. Comp. Neurosci.* 3:7–34.

Howe JR, Sutor B, Zieglgansberger W (1987) Baclofen reduces post-synaptic potentials of rat cortical neurones by an action other than its hyperpolarizing action. *J. Physiology* 384:539–569.

Knight BW (1972a) Dynamics of encoding in a population of neurons. *J. Gen. Physiol.* 59:734–766.

Knight BW (1972b) The relationship between the firing rate of a single neuron and the level of activity in a population of neurons: Experimental evidence for resonant enhancement in the population response. *J. Gen. Physiol.* 59:767–778.

Knight BW (2000) Dynamics of encoding in neuron populations: Some general mathematical features. *Neural Comput.* (to appear).

Knight BW, Manin D, Sirovich L (1996) Dynamical models of interacting neuron populations. In EC Gerf, ed. Symposium on Robotics and Cybernetics: Computational Engineering in Systems Applications. Cite Scientifique, Lille, France.

Kuramoto Y (1991) Collective synchronization of pulse-coupled oscillators and excitable units. *Physica D* 50:15–30.

Omurtag A, Knight BW, Sirovich L (2000) On the simulation of large populations of neurons. *J. Comp. Neurosci.* 8:51–53.

Pham J, Pakdaman K, Champagnat J, Vibert JF (1998) Activity in sparsely connected excitatory neural networks: Effect of connectivity. *Neural Networks* 11:415–434.

Shapley R, Sompolinsky H (1997) New perspectives on the mechanisms for orientation selectivity. *Curr. Op. Biol.* 7:514–522.

Sirovich L, Knight BW, Omurtag A (1999) Dynamics of neuronal populations: The equilibrium solution. *SIAM* (to appear).

Somers DC, Nelson SB, Sur M (1995) An emergent model of orientation selectivity in cat visual cortical simple cells. *J. Neurosci.* 15:5448–5465.

Strogatz SH, Mirollo RE (1991) Stability of incoherence in a population of coupled oscillators. *J. Stat. Phys.* 63:613–635.

Tanabe S, Pakdaman K, Nomura T, Sato S (1998) Dynamics of an ensemble of leaky integrate-and-fire neuron models and its response to a pulse input. *Technical Report of IEICE*, NLP98-14:41–48.

Treves A (1993) Mean-field analysis of neuronal spike dynamics. *Network* 4:259–284.

Tuckwell HC (1988) Introduction to Theoretic Neurobiology. Cambridge University Press, New York, NY. Vol. 2, pp. 111–189.

Victor JD (1987) The dynamics of the cat retinal X cell centre. *J. Physiol.* 386:219–246.

Wilbur WJ, Rinzel J (1983) A theoretical basis for large coefficient of variation and bimodality in neuronal interspike interval distributions. *J. Theor. Biol.* 105:345–368.

Wilson HR, Cowan JD (1972) Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical J.* 12:1–24.

Wilson HR, Cowan JD (1973) A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* 13:55–80.