# Efficient Emulation of Large-Scale Neuronal Networks

BENG/BGGN 260 Neurodynamics

University of California, San Diego

Week 6

# Reading Material

- W. Gerstner and W. Kistler, *Spiking Neural Models: Single Neurons, Populations, Plasticity*, Cambridge Univ. Press, 2002, Ch. 4, pp. 93-145.
- C. Koch, *Biophysics of Computation*, Oxford Univ. Press, 1999, Ch. 14, pp. 330-349.
- P. Dayan and L. Abbott, *Theoretical Neuroscience*, MIT Press, 2001, Ch. 7.2, pp. 231-240.
- Michelle Rudolph and Alain Destexhe, "Analytical Integrate-and-Fire Neuron Models with Conductance-Based Dynamics for Event-Driven Simulation Strategies", *Neural Computation*, vol. 18, pp. 2146-2210, 2006.
- E. Ros, R. Carrillo, E. Ortigosa, B. Barbour, and R. Agis, "Event-Driven Simulation Scheme for Spiking Neural Networks Using Lookup Tables to Characterize Neuronal Dynamics", *Neural Computation*, vol. 18, pp. 2959-2993, 2006.

# Efficient Event-Driven Emulation of Spiking Networks

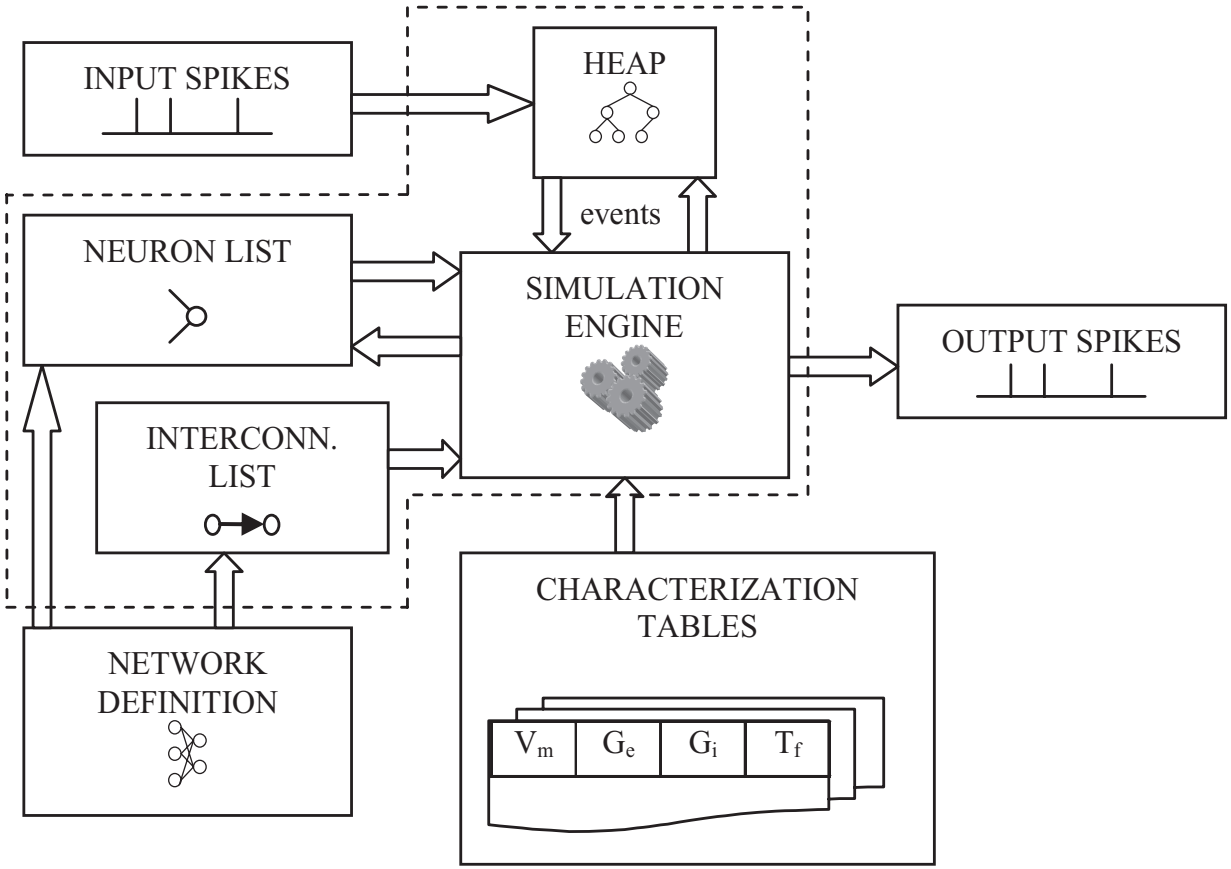



**Figure 1: Main structures of the ED-LUT simulator**

Input spikes are stored in an input queue and are sequentially inserted into the spike heap. The network definition process produces a neuron list and an interconnection list, which are consulted by the simulation engine. Event processing is done by accessing the neuron characterization tables to retrieve updated neuronal states and forecast spike firing times.

Ros *et al*, 2006

# Efficient Event-Driven Emulation of Spiking Networks

While $t_{sim} < t_{end}$
{
Extract the event with a shortest latency in the spike heap

If it is a **firing event**
If it is still a valid event and the neuron is not under a refractory period
Update the neuron state: $V_m$, $g_{exc}$, $g_{inh}$ to the postfiring state.
Prevent this neuron from firing during the refractory period.
(Once this is done, update the neuron time label to $t_{sim} + t_{refrac}$).
Predict if the source neuron will fire again with the current neuron state.
If the neuron will fire:
Insert a new *firing event* into the spike heap.
Insert the *propagated event* with the shortest *latency* (looking at the output connection list).

If it is a **propagated event**
Update the target neuron state: $V_m$, $g_{exc}$, $g_{inh}$ looking at the *characterization tables*, before the event is computed.
Modify the conductances ($g_{exc}$, $g_{inh}$) using the connection weight ($G_{exc,i}$, $G_{inh,i}$) for the new spike.
Update the neuron time label to $t_{sim}$.
Predict if the target neuron will fire.
If it fires:
Insert the *firing event* into the spike heap with the predicted time.
Insert only the next *propagated event* with the next shortest *latency* (looking at the output connection delay table).
}

**Figure 2: Simulation algorithm**

This pseudocode describes the simulation engine. It processes all the events of the spike heap in chronological order.

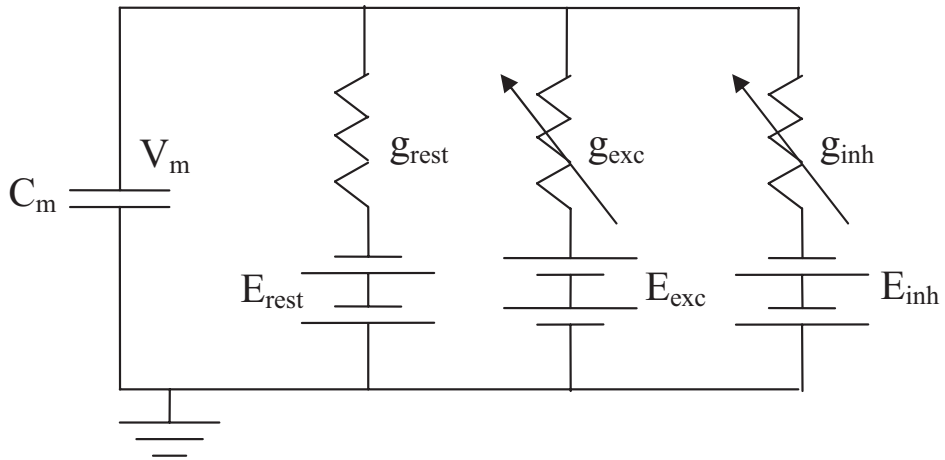
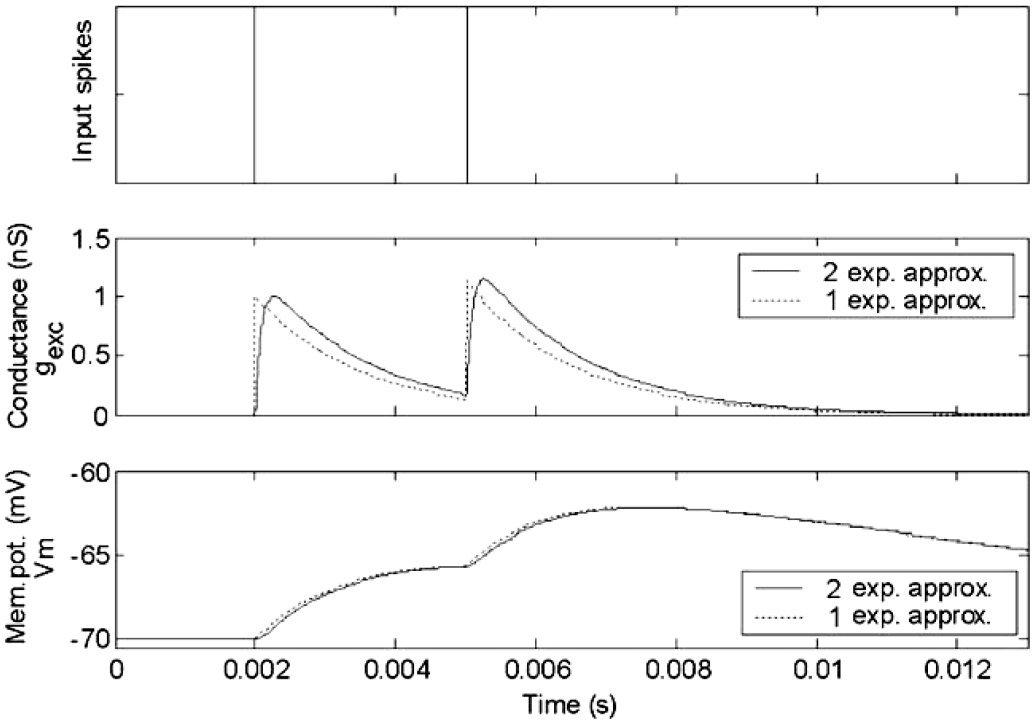



**Figure 3: Equivalent electrical circuit of a model neuron**

$g_{exc}$ and $g_{inh}$ are the excitatory and inhibitory synaptic conductances, while $g_{rest}$ is the resting conductance, which returns the membrane potential to its resting state ($E_{rest}$) in the absence of input stimuli.

Ros *et al*, 2006

# Exponential Postsynaptic Conductance Model




$$g_{exc}(t) = \begin{cases} 0, t < t_0 \\ G_{exc} \cdot e^{-(t-t_0)/\tau_{exc}}, t \geq t_0 \end{cases}$$

$$g_{inh}(t) = \begin{cases} 0, t < t_0 \\ G_{inh} \cdot e^{-(t-t_0)/\tau_{inh}}, t \geq t_0 \end{cases} \quad (1)$$

**Figure 4. A postsynaptic neuron receives two consecutive input spikes**

The evolution of the synaptic conductance is the middle plot. The two excitatory postsynaptic potentials (EPSPs) caused by the two input spikes are shown in the bottom plot. In the solid line plots, the synaptic conductance transient is represented by a double-exponential expression (one exponential for the rising phase, one for the decay phase). In the dashed line plot, the synaptic conductance is approximated by a single-exponential expression. The EPSPs produced with the different conductance waveforms are almost identical.

Single decaying exponential model for synaptic conductance is adequate for accurate emulation of postsynaptic membrane dynamics.

Ros *et al*, 2006

# Event-Driven, Table-Based Estimation of Postsynaptic Action Potential Timing

- For single-exponential conductance dynamics, arrival of a new excitatory presynaptic spiking 'event' at time $t$ signifies a one-time increment in the conductance:

$$g_{exc}(t) = G_{exc,j} + e^{-(t-t_{previousspike})/\tau_{exc}} g_{exc}(t_{previousspike}) \qquad (2)$$

  and similar for an inhibitory event.
- Postsynaptic time-to-firing (in the absence of other 'events') is then tabulated for different values of initial excitatory and inhibitory conductance:

$$C_m \frac{dV_m}{dt} = g_{exc}(t_0) e^{-(t-t_0)/\tau_{exc}} (E_{exc} - V_m) + g_{inh}(t_0) e^{-(t-t_0)/\tau_{inh}} (E_{inh} - V_m) + G_{rest}(E_{rest} - V_m) \qquad (3)$$

- In the event of an earlier other 'event', the conductance values are updated according to (2).

Ros *et al*, 2006

# Accuracy of Event-Driven, Table-Based Emulation

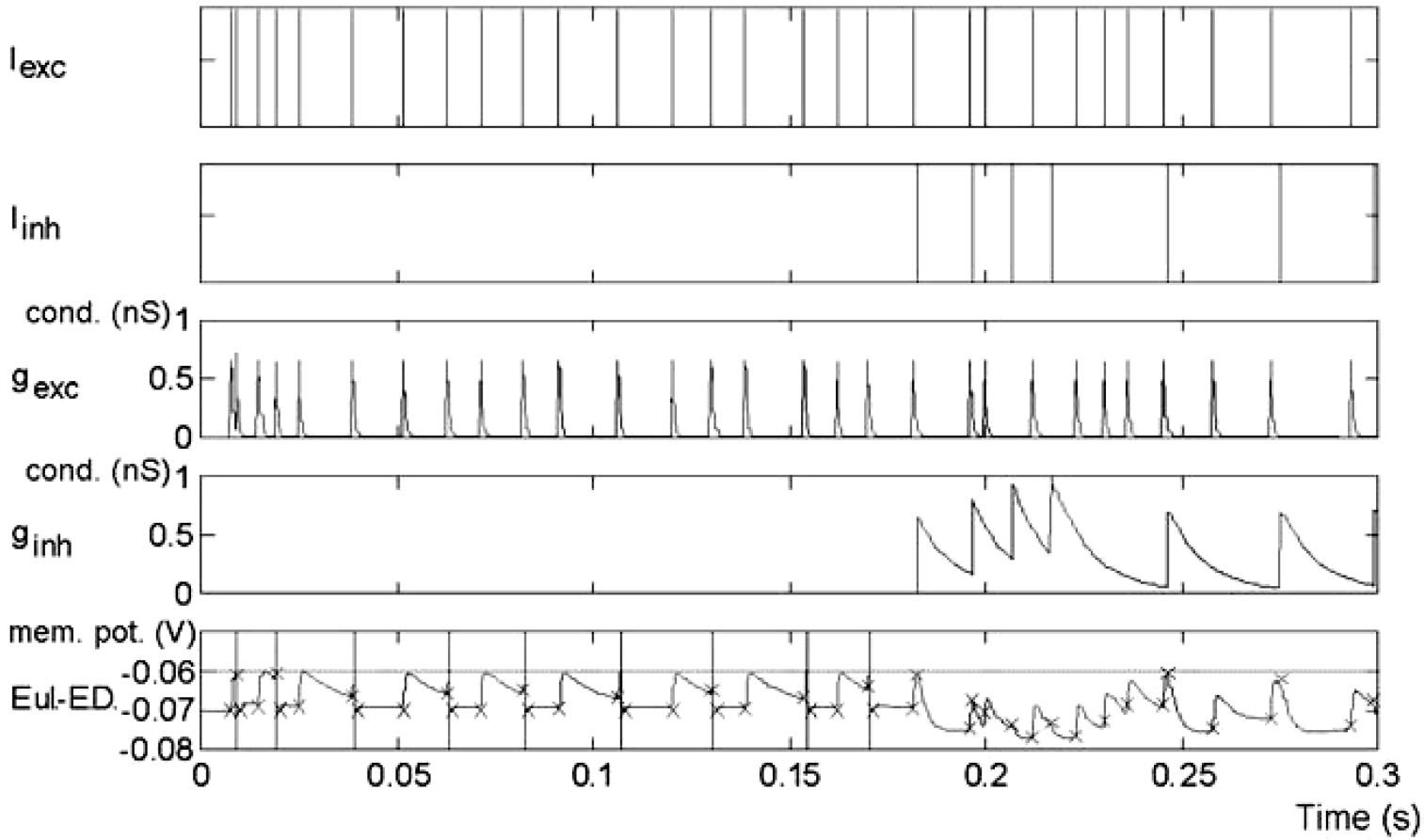



**Figure 9: Single neuron simulation**

Excitatory and inhibitory spikes are indicated on the upper plots. Excitatory and inhibitory conductance transients are plotted in the middle plots. The bottom plot is a comparison between the neural model simulated with iterative numerical calculation (continuous trace) and the event-driven scheme, in which the membrane potential is updated only when an input spike is received (marked with an x).

Ros *et al*, 2006

# Notes on the Event-Driven Approach

- The table-based, event-driven approach can be extended to multiple excitatory and inhibitory synapse types with different synaptic strengths, time constants, and reversal potentials.
  - *Each additional and distinct time constant requires an additional variable dimension in the table lookup. Additional synapse types with time constant identical to another existing synapse type do not incur additional variable dimensions, independent of synaptic strength or reversal potential.*
  - *The complexity of table lookup is exponential in the number of variable dimensions, and hence does not scale to large numbers of different time constants in the synaptic dynamics.*

- In the case of a single synaptic time constant, a simple and exact closed-form expression for the postsynaptic time-to-firing replaces the table lookup for efficient implementation [Rudolph and Destexhe, 2006].