**Parallel Discrete Event Simulation of Neurons**

Carl Tropper

The NEURON simulator provides support for simulating neurons and networks of neurons across multiple spatial and temporal scales.

In small spatial domains, such as spines, the volume is low and the total number of molecules present is correspondingly low. At this spatial scale, stochastic effects play key roles in the dynamics. Numerous strategies exist in the literature for simulating these dynamics. We have added a "plugins" submodule to NEURON to connect to external stochastic solvers; this submodule has been tested with NTW (Neuron Time Warp), which performs parallel 3D stochastic simulations. External solvers that work with the plugins submodule receive the morphology, diffusion, and reaction dynamics from NEURON. The user can switch between the built-in deterministic simulations and external solvers by activating the external solver; specification of reaction kinetics is thus made simulator independent.

In small spatial domains, such as spines, the volume is low and the total number of molecules present is correspondingly low. At this spatial scale, stochastic effects play key roles in the dynamics. Numerous strategies exist in the literature for simulating these dynamics. We have added a "plugins" submodule to NEURON to connect to external stochastic solvers; this submodule has been tested with NTW (Neuron Time Warp), which performs parallel 3D stochastic simulations. External solvers that work with the plugins submodule receive the morphology, diffusion, and reaction dynamics from NEURON. The user can switch between the built-in deterministic simulations and external solvers by activating the external solver; specification of reaction kinetics is thus made simulator independent.
In some cases, such as after ischemic shock, neurons and glia affect each other by altering extracellular concentrations. Although the extracellular space is inhomogeneous, molecular dynamics there is also governed by reaction-diffusion kinetics. We describe our progress toward supporting these dynamics: molecules may react and diffuse anisotropically in the extracellular space, concentrations in the extracellular space are modulated by ion channel and pump activity and themselves modulate electrical activity by altering the Nernst potentials. Concentrations may be further modulated by extracellular sources such as blood vessels.

We make use of NTW (Neuron Time Warp) to simulate a Ca wave model. NTW makes use of the Next Subvolume Method, an extension of the Gillespie algorithm as a basis for stochastic simulation. In NSM the region to be simulated (e.g. an apical dendrite) is divided into small regions, or sub volumes. Chemical reactions occur within the sub volumes and molecular diffusion takes place between them. NTW is a parallel discrete event simulator which makes use of Time Warp, a widely used optimistic algorithm to parallelize the stochastic simulator. In Time Warp, the simulator is divided into processes, which are allocated among the compute nodes of a parallel machine. The processes then communicate either by message passing (MPI) if they are on different compute nodes or by shared memory if they are on the same node.

We developed load balancing algorithms for NTW in order to balance the computational load and the communications load for the simulator. The load balancing algorithms made use of reinforcement learning, a technique from artificial intelligence. In reinforcement learning an action is chosen at each iteration of the algorithm from a group of possible actions. A reward table for each action is maintained and updated each time an action is executed. In our load balancing algorithm the actions correspond to transferring sub volumes between processes on different processors. The reward for our load balancing algorithm is based on the number of events executed since the last time load was transferred between processes.

 We used NTW to simulated a calcium wave on a hyppocampal pyramidal neuron (from NeuroMorphOrg). Our geometry consisted of a 3D mesh grid of cubes (the sub-volumes).We made use of 14,749 cubes, each of which had a length of 0.5 µm. We present a figure below comparing the execution time of of our algorithm with and without load balancing algorithms which we have developed. The simulation was done on a one node of compute cluster consisting of two Dual Intel(R) Sandy Bridge EP E5-2670 2.6 GHz CPUs, 8 cores per processor, 8 GB of memory per core, and a Non blocking QDR InfiniBand network with 40 Gbps between nodes.