Fully-Asynchronous Fully-Implicit Variable-Order Variable-Timestep Simulation of Neural Networks

Bruno Magalhães[1], Michael Hines[2], Thomas Sterling[3], and Felix Schürmann[1]

[1] Blue Brain Project, École Polytechnique Fédérale de Lausanne
Biotech Campus, 1202 Genève, Switzerland
[2] Department of Neuroscience, Yale University, New Haven, CT 06510, USA
[3] Department of Intelligent Systems Engineering, School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN 47404, USA

Abstract. State-of-the-art simulations of detailed neurons follow the Bulk Synchronous Parallel execution model. Execution is divided in equidistant communication intervals, with parallel neurons interpolation and collective communication guiding synchronization. Such simulations, driven by stiff dynamics or wide range of time scales, struggle with fixed step interpolation methods, yielding excessive computation on intervals of quasi-constant activity and inaccurate interpolation of periods of high volatility in solution. Alternative adaptive timestepping methods are inefficient in parallel executions due to computational imbalance at the synchronization barriers. We introduce a distributed fully-asynchronous execution model that removes global synchronization, allowing for long variable timestep interpolations of neurons. Asynchronicity is provided by point-to-point communication notifying neurons' time advancement to synaptic connectivities. Timestepping is driven by scheduled neuron advancements based on interneuron synaptic delays, yielding an *exhaustive yet not speculative* execution. Benchmarks on 64 Cray XE6 compute nodes demonstrate reduced number of interpolation steps, higher numerical accuracy and lower runtime compared to state-of-the-art methods. Efficiency is shown to be activity-dependent, with scaling of the algorithm demonstrated on a simulation of a laboratory experiment.

Keywords: Simulation of Neural Networks · Asynchronous computing · Variable timestep · Parallel computing · Distributed computing.

1 Introduction

Simulation of the electrical activity of large networks of biologically detailed neuron models is a major impact scientific problem, allowing for a better understanding of the brain. State-of-the-art neuron models follows from the Hodgkin-Huxley (HH) formalism [5], modeling the electrical currents passing though connecting sections of neuron morphologies (spatially discretized as a tree of cylindrical leaky capacitors, henceforth referred to as **compartments**). Neurons are coupled via electro-chemical transductors, denominated **synapses**. When the voltage at a (**pre-synaptic**) neuron soma reaches a specific action potential threshold, it **spikes** (or **fires**), leading to a chain of biological reactions that changes the

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

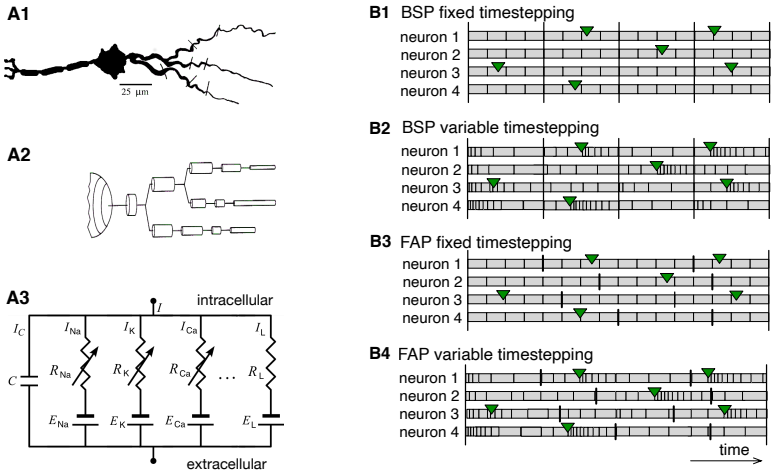

Fig. 1. Left: A1: a sample neuron morphology; **A2:** the spatially-discretized model of a compartmental tree of neuron dendrites; **A3:** The RC circuit representing the electrical activity of a compartment represented by the extended Hodgkin-Huxley model, with Sodium (Na), Potassium (K), Calcium (Ca) and leak (L) currents. **Right:** The four interpolation methods discussed, illustrated by left-to-right execution timelines of the simulation of four neurons. Gray cells represent the duration of interpolation steps. Inverted green triangles represent delivery of synaptic events. **B1:** Bulk Synchronous Parallel (BSP) model with fixed timestepping and collective synchronization. Vertical bars across all neurons represent collective communication; **B2:** BSP model with variable timestepping and collective synchronization implemented in NEURON [8]; **B3:** Fully-asynchronous parallel (FAP) fixed-step method, pioneered by our previous work [11]. Vertical bars across single neurons represent limit of stepping dictated by synaptic dependencies; **B4:** FAP variable-step method presented in this document.

voltage at their **post-synaptic** counterparts. Due to the long simulation time required to express biological phenomena such as learning and synaptic plasticity, the acceleration of the simulation of neural networks is a relevant problem. Existing acceleration efforts follow the Bulk Synchronous Parallel (BSP) execution model, computing several neurons simultaneously via synchronized multithredead and distributed execution [4,7] . Execution time is divided in communication intervals equivalent to the time duration of the shortest **synaptic delay** across all neuron pairs in the network. Equidistant synchronous collective communication calls performs both synchronization of stepping and synaptic exchange. Interpolation of neurons is performed independently within the boundaries of each intervals, typically with fixed timestep methods, as illustrated in Fig. 1 B1). Variable timestep interpolation on the BSP execution model (Fig. 1 B2) has been presented on single compute nodes, with speculative interpolation of individual neurons [8]. Further acceleration can be achieved with finer-grained parallelism of individual neuron models via graph-parallelism of Ordinary Differential Equations (ODEs) [9] and branch-parallelism of neuron topology sections

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

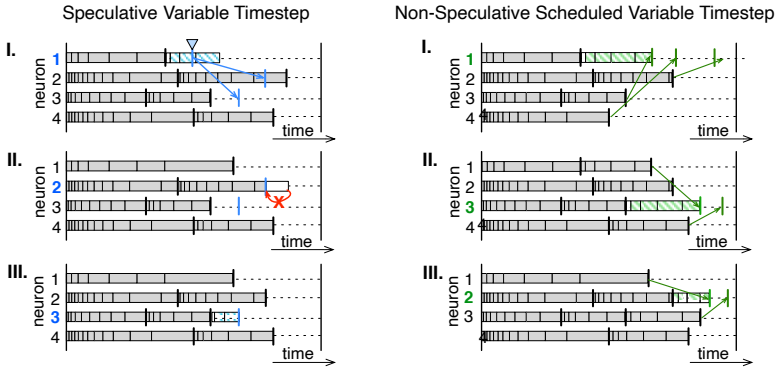

Fig. 2. Illustrative workflow of the two methods for variable timestep (vardt) interpolation described. **Left:** Speculative interpolation (state-of-the-art): **I.** Neuron 1 performs a step (blue area) and spikes during the step interval, with spike time marked with an inverted triangle, and delivery times marked with blue arrow heads; **II.** The current interpolation time of neuron 2 exceeds the spike delivery time. A back stepping to the delivery time follows (red arrow); **III.** Neuron 3 interpolates until the next (spike) event time. **Right:** Non-Speculative scheduled time-stepping. **I.** Neuron 1 advances to the earliest time instant allowed (green area), given by the time instant of pre-syn. neurons 2, 3 and 4 and the shortest synaptic delays 2→1, 3→1, and 4→1, respectively (green arrow heads); **II.** Neuron 3 is now the earliest neuron in time, and follows analogously based on the delays of neurons 1 and 4; **III.** Neuron 2 advances similarly.

[10]. Cache-efficient acceleration has been demonstrated via a barrier-free fixed-timestep simulation on a fully-asynchronous parallel (FAP) execution model [11] (Fig. 1 B3), henceforth referred to as **our previous work**.

We introduce a method for the distributed fully-asynchronous variable-order variable-timestep implicit interpolation of detailed neuron models, benefiting from cache-efficient barrier-free synchronization and performing long variable timesteps on the FAP execution model, as illustrated in Fig. 1 B4). We study the numerical instability and performance dependency of our methods on the biological activity of the network. An implementation of our methods on the core kernel of the NEURON simulator [4] is detailed and benchmark on 64 Cray XE6 compute nodes. Distributed asynchrony and multicore executions on a global memory address space are provided by the HPX runtime system [13]. Benchmarks demonstrate lower time to solution, higher numerical accuracy, the removal of speculative computing and synchronization barriers, and good scaling properties, tested on a simulation of a laboratory experiment.

The methods presented provide insights for the exploration of modern asynchronous runtime systems on large networks of compute nodes, and for the redesign of future simulators across a wide range of scientific domains, driven by large systems of ODEs and highly-heterogeneous activity.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

2 Methods

2.1 Mathematical Model

The RC circuit that models the electrical current passing through the membrane of a compartment n is modelled as:

$$C\frac{dV_n}{dt} = -\sum_i g_i x_i (V_n - E_i) + I(t) + \sum_{c:p(c)=n} \frac{V_c - V_n}{r_c} - \frac{V_n - V_{p(n)}}{r_{p(n)}} \quad (1)$$

where g_i and E_i describe the conductance and reversal potential of the ionic channels. x_i models the opening probability of the transmembrane ion channel currents, typically described by a voltage-gated ODE, and omitted for brevity. Synaptic currents or injected current stimuli, if any, are included in $I(t)$. The branching contributions are provided by Ohm's Law and the neuronal cable theory: the subscript $p(n)$ refers to the index of the parent compartment of n, and c to an iterator over the indices of its children in the compartmental tree. The variable r defines the axial resistance as a function of the diameter and the cytoplasmic resistivity. The RC circuit underlying the current passing through a compartment is illustrated in Figure 1, layout A3). The solution of this system is solved numerically. The complexity of the spatio-temporal model of the neuron activity is reduced by performing a spatial discretization of the neuronal morphology, from biologically inspired to HH-based compartmental representation, and assume the spatial discretization to be small enough, so that the state across compartments' length is constant (Fig. 1, A1 and A2). Thus, interpolation of solution is performed for consecutive discrete time intervals only. The resolution follows a fixed step defined as *small enough* to capture the currents with fastest dynamics, set to 0.025 milliseconds. The fastest synaptic delay across our network model has been measured as $0.1ms$ or equivalently 4 computation steps, accounting for 0.13% of the total synapses.

Simple and Complex Neuron Models: A problem specific optimization allows for a speed-up on the resolution of **simple neuron models** such as the Hodgkin-Huxley, where state variables are described by linear ODEs and depend only on the voltage V, and vice-versa. An implicit resolution based on interleaved timestepping of voltage and states, by solving voltages at a given time t and states at time $t + \Delta t/2$, allows for the resolution of the system of ODEs as a system of linear equations. Resolution of **complex models**, including non-linear and/or correlated state equations (such as synaptic plasticity presented by Graupner et al. [3]) cannot be resolved with the aforementioned method, and require a fully-implicit (non-staggered) resolution. Reliable resolutions rely on fixed-step iterative implicit methods such as Backward Euler. Alternatively, an implicit variable timestep method with variable order is possible. Its implementation to our use case is detailed next.

Variable Step Implementation: The CVODE (C Variable-step solver for ODEs, [2]) is an implementation of the Backward Differentiation Formula (BDF) for

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

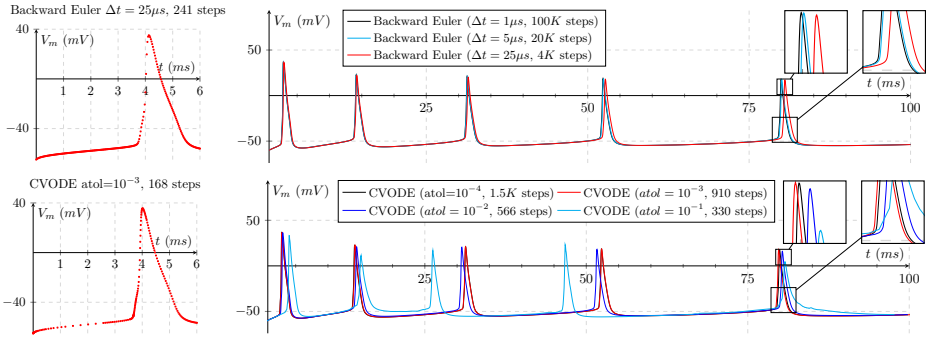

Fig. 3. Voltage potential at soma and interpolation steps for a sample neuron during $6ms$ (left) and $100ms$ simulation (right) of a $1.3mA$ continuous current injection, interpolated with Backward Euler (top) and CVODE (bottom) methods. The reference implementations are the Backward Euler with $\Delta t = 1\mu s$ and CVODE with absolute tolerance 10^{-1}, presented in black and considered indistinguishable. The standard NEURON step size and tolerance are $\Delta t = 25\mu s$ and 10^{-3} and are presented in red.

the variable-step multistep implicit method solving the Initial Value Problem (IVP, $\dot{y} = f(t, y)$, $y(t_0) = y_0$ where $y \in \mathbb{R}^N$) for ODEs as:

$$\sum_{i}^{q} \alpha_{n,i} y_{n-i} + \Delta t_n \beta_n \ \dot{y} = 0 \tag{2}$$

where $y = [..., V_{k-1}, V_k, V_{k+1}, .., x_{i-1}, x_i, x_{i+1}, ...]$ is a vector representing the state variables of a neuron or a set of neurons following the variable notation in Equation 1, q is the order of the current iteration, and α and β are the q-dependent BDF-method coefficients. BDF-1 is the Backward Euler. In brief, CVODE returns the Δt_n and y_n that solve BDF-q for an user-provided tolerance **(atol)**. The computation is performed iteratively, with a suggested step size for each iteration based on the solution gradient and order q. Given and user-provided function that computes the ODE right-hand side and a Jacobian $j = \partial f / \partial y$ (or an approximation to it), the resolution relies on Newton iterations, with a stop condition based on the test $\|y_{n(m)} - y_{n(0)}\| \leq \epsilon$ for iteration m in step n: If error is greater than threshold, a reiteration follows with a smaller $\Delta t_{n(m+1)}$; if error is smaller, proceeds to step $n + 1$ with larger $\Delta t_{(n+1)(0)}$.

2.2 Asynchronous Timestepping of Neuron Networks

Control of neurons time advancement on synchronized distributed executions is a solved problem, by enforcing a BSP-like synchronization barrier [4], as in layouts B1 and B2 in Figure 1. Barrier-free variable-timestep interpolation are possible with a speculative interpolator on a single compute node and small networks of neurons. Neurons are described by individual interpolators, and advance in time under the best assumption that no **discontinuity** of solution (synaptic current) will arrive with a delivery time earlier than the neuron current time. Discontinuities lead to a **reset of the IVP** problem and interpolator state history, and

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

consequently to small steps in the following iterations. When a discontinuity is required to be delivered in a past instant in time, a **backstepping** operation must precede, in order to reset the recent step and interpolate neuron state back to a time instant of confidence (the time of the discontinuity). Simulations of small neuron networks on single compute nodes are possible and have previously shown a substantial runtime acceleration utilising this model [8]. Distributed executions and/or large neural network are infeasible with this method due large number of IVP resets, the complexity of backstepping cascades of events across several compute nodes, large amount of time spent on speculative stepping, and computational imbalance at synchronization barriers.

An alternative approach was implemented, based on the non-speculative asynchronous stepping methodology detailed in our previous work [11]. Neurons hold a map storing the time instant of their pre-synaptic connectivities. The map is updated by stepping notifications received actively at a certain frequency, throughout the stepping of its pre-synaptic dependencies. Neurons step to the maximum time allowed by their synaptic connectivities. This guarantees synapses to be delivered in future time instants, thus removing backstepping and reversion of sent synaptic spikes. The method is improved with an *earliest neuron steps first* scheduler at each compute node that keeps track of neurons advancement, and picks the earliest neuron in time as the next to interpolate. This guarantees the maximisation of the step length and provides a larger variable step interval, with the benefit of reduced communication and computation, the removal of solution resets and backstepping, and larger stepping intervals. For completion, both approaches described are pictured in Figure 2.

2.3 Implementation details

Our methods were implemented on the core kernel of the NEURON simulator[7]. Single Instruction Multiple Data (SIMD) capabilities — supported only by fixed-step method — were added to variable-step implementations. Communication, synchronisation control objects, memory allocation, threading, distributed memory space, distributed execution and parallelism were implemented with HPX [13], the runtime system for the Parallex execution model [6]. The Implementation details have been covered in our previous manuscript [11], and are omitted for brevity. Efficient point-to-point communication and remote direct access memory is provided with specialized Infiniband network hardware.

3 Results

3.1 Numerical Accuracy

We compare the numerical accuracy of both fixed and variable step models by measuring the time difference of the main unit of interest in the activity of spiking neuron networks — the spiking time instants. Figure 3 presents the voltage trajectory and number of steps of a $1.3mA$ current clamp experiment for

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

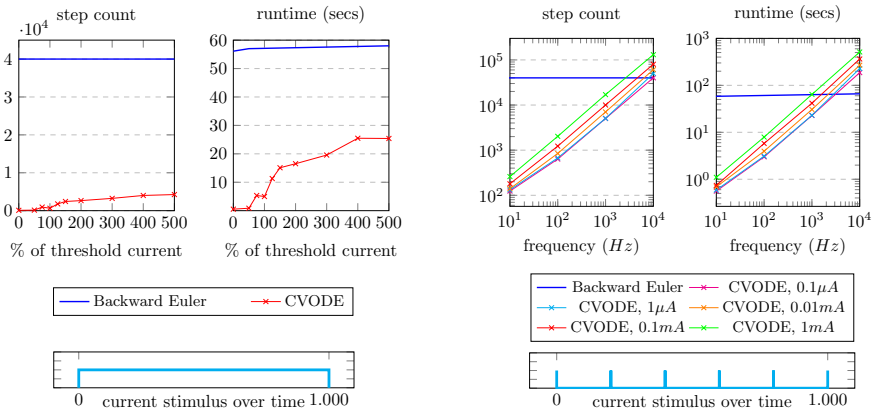

Fig. 4. Interpolation steps and runtime for $1000ms$ simulation of a pyramidal cell on an Intel core i5 at 1.6 GHz. Left: injection of a continuous current as a percentage of the threshold current ($0.206mA$). Right: injection of short $1\mu s$ current pulses of different amplitudes I at different frequencies. Results presented for the Backward Euler with $\Delta t = 25\mu s$ and CVODE with atol=10^{-3}.

a single ($6ms$) and several ($100ms$) spikes of a layer 5 pyramidal cell. Results on the single spike voltage trajectory ($6ms$) display a reduced step count and better adaptation to trajectory change when comparing CVODE to Euler method. The rationale behind the better performance is adaptive stepping is gradient sensitive, thus better adapting to the trajectory of a neuron voltage. CVODE displays less steps during long periods of low gradient (e.g. $1 - 2.5ms$), and greater number of steps for steep trajectories (the spike trajectory). The $100ms$ simulation displays a phase shift in solution (measured as the time difference between peak voltage values of reference and benchmark curves) that increases with the increase of the step size on the Euler methods. In practice, the timestep determines the fastest reaction time of the system, thus large timesteps will inevitably cause the system dynamics to be *slow*. The analysis show that a CVODE tolerance value of 10^{-2} approximates the resolution of the default Euler method (step size $25\mu s$), with a reduction of $7\times$ in step count. At longer runs, the variable step demonstrated to be more precise, due to no accumulation of phase shift, with the maximum trajectory shift measured at approximately $1.1ms$. On the other hand, a tolerance value of 10^{-3} approximates closely the optimal solution with 40% less steps, and with a margin of error similar to its $5\mu s$ Euler counterpart for the period of $100ms$, while yielding $22\times$ less interpolations.

3.2 Performance Dependency on Stiffness and Discontinuities

We measured the response of both stepping methods to spiking frequency. Performance was measured in terms of steps count and time to solution on an Intel i5 at 1.6 GHz. Changes in trajectory were enforced by injecting a continuous current of a given amplitude on a neuron during $1000ms$. Current intensity is measured as a percentage of the **threshold current**, the minimum continuous

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

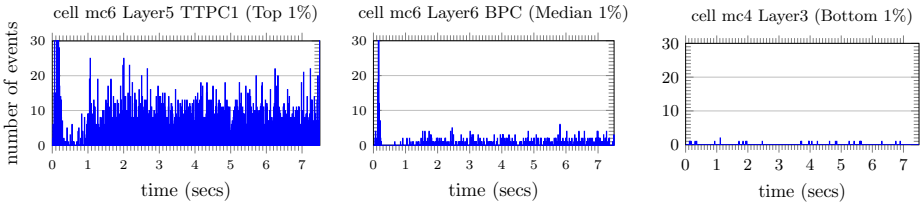

Fig. 5. Number of incoming spikes (bin size 0.25ms) measured throughout 7.5 secs of simulation, for a sample neuron collected from the top, median and bottom 1% on a network of 219K neurons.

current value that needs to be injected to force a neuron to spike. Results are presented in Figure 4 (left), and demonstrate that high dynamics of the solution degrade the CVODE performance. This is due to CVODE requiring smaller steps on high trajectory variations in order to respect the absolute tolerance value. For the range of tested scenarios, the measured CVODE to Euler reductions were: (1) 434× in step count and 98× in runtime for injected currents below 50% of the threshold current; (2) 62× steps and 11.6× runtime for 100%; and (3) 9.4× and 2.5× runtime for 500%, a worst case scenario of little prob. of occurrence.

We measured the effect of discontinuities on both methods by injecting several current pulses at a fixed frequency on a neuron soma, mimicking synaptic events. The experiment results are displayed in Figure 4 (right) and suggest that the CVODE performance depends on the trajectory change incurred by each discontinuity, i.e. the amplitude of the current injected: the larger the voltage increase, the larger the change in trajectory gradient, thus the more interpolation steps are required. As expected, results demonstrate that the number of discontinuities plays a major role in performance. CVODE is shown to deliver a reduction of steps in the order of $153 - 322\times$ for a frequency of 10 discontinuities per second for current values of $1mA$ to $0.1\mu A$. The step count equilibrium between Euler and CVODE method lies in the interval of $10^{3.2} - 10^{4.0}Hz$ for similar currents interval. The runtime demonstrates a similar dependency on the injected current, yielding a speed-up of $51\times$ for $10Hz$ decreasing linearly up to the speed-up equilibrium value at $1000Hz$ for the strongest current. For the lightest current injected, a speed-up of $100\times$ is visible for a $10Hz$ discontinuity rate, decreasing to an Euler matching value at circa 1600 events/sec ($10^{3.2}Hz$).

3.3 Simulation of a Laboratory Experiment

We tested the suitability of variable step methods to our problem by measuring the spiking activity of a simulation of 7.5 secs of electrical activity mimicking a laboratory experiment. The experimental set-up performs a fixed step simulation of the spontaneous activity of 219.247, detailed in section *Simulating Spontaneous Activity* in [12]. A representative distribution of discontinuity events for three groups of neurons — organized by highest 1%, median 1%, and lowest 1% number of discontinuities — is displayed in Fig. 5. The simulation incurred a total of circa 155 million events, with the following distribution: (a) top 1%

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

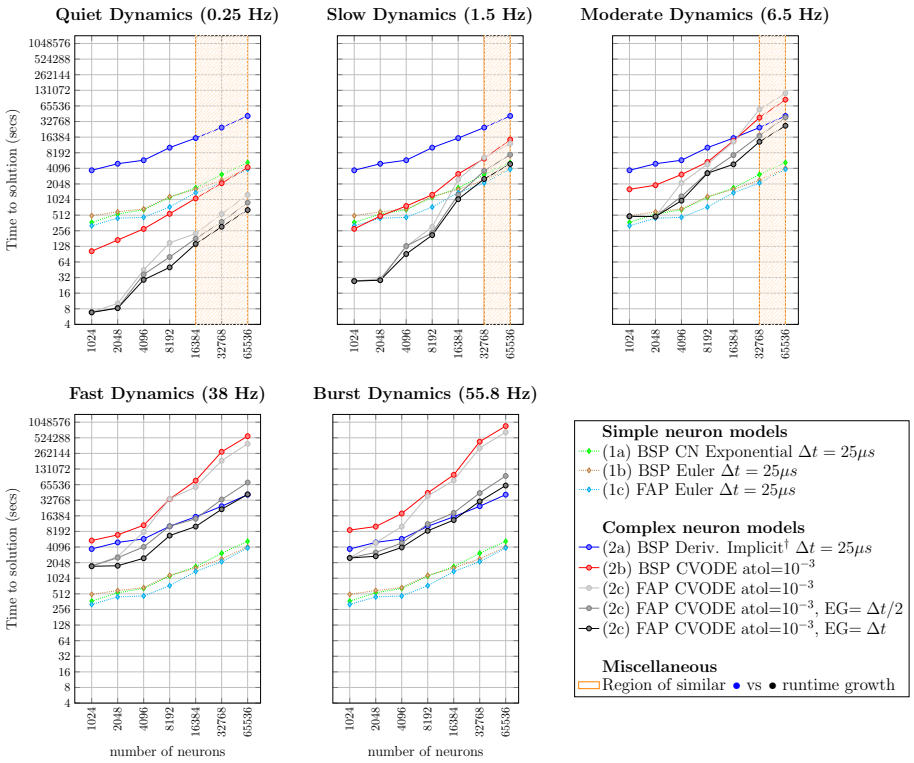

Fig. 6. Runtime for the simulation of one second of biological activity described by five spiking rate dynamics, measured for increasing input network sizes, on 64 Cray XE6 compute nodes. **Key:** BSP: Bulk Synchronous Parallel; FAP: Fully-Asynchronous Parallel; atol: absolute tolerance; EG: event grouping interval; †: able to solve non-linear ODEs implicitly, and unable to solve correlated mechanism states implicitly.

of neurons, between 3040 and 6146 events in 7.5 secs, or 405-820 Hz; (b) median 1%, from 541 to 558 events (72-74.4 Hz); and (c) bottom 1%: less than 100 events ($\leq$10 Hz). The average number of events was of 707 events for the 7.5 secs of simulation, or equivalently, 94 Hz, significantly below the $1000Hz$ threshold discussed in the previous section. Moreover, the results on the distributions of time interval between discontinuities, plotted in red on the right, display large periods of *silence* between events arrival in the median and bottom use cases, but not on the top, suggesting the suitability of adaptive stepping to most (but not all) neurons in the population.

3.4 Large-Scale Benchmark

We simulate one second of the electrical activity of a digitally reconstructed neural network extracted from the model of Markram et al. [12]. Execution times were collected on 64 Cray XE6 compute nodes, powered by an AMD Opteron 6380 with 16 cores at 2.5 GHz, 64 GB of RAM and 256-bit floating point units. CVODE was defined to utilise the default maximum BDF order value of 5.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

On the set-up of the test bench, it is relevant to mention that neuronal activity is highly dependent on the mammal specie, brain region and momentary activity. Simulations must approximate real use cases, as spiking activity affects heavily the performance of variable step methods, as shown previously. Thus, our test bench benchmarks the efficiency of five different brain dynamics described in literature: **(1)** a model of *quiet dynamics* with a mean spiking rate of 0.25 Hz per neuron, representative of circa 90% of neurons in the human brain during regular activity; **(2)** *slow dynamics* at 1.5Hz, representing the lower bound of active neurons; **(3)** *moderate dynamics* at 6.5 Hz, an approximation of the regime of slow oscillations regime from the Brunel network model [1] and an upper limit to the rat frontal cortex; **(4)** *fast dynamics* at 38 Hz, characterizing neuronal activity during periods of high vigilance; and the inhibition-dominated model of the Brunel Network [1]; and **(5)** *burst dynamics* at 55.8 Hz, typically a byproduct of strong current injections, similar to the first instants of simulation in Fig. 5; and the fast spiking regime of the Brunel Network [1]. Neurons activity is triggered by a constant current injection in all neurons throughout the whole duration of the simulation, strong enough to approximate the spiking rate to the regimes described. The input neural networks are retrieved from layers 4 and 5 of the rodent brain, where the longest dendritic trees and densest synaptic connectivity exist, thus representing a worst-case scenario for variable-step methods, and favourable to fixed-step methods. Thus, the results presented are a lower bound of possible acceleration. For complete coverage of the topic, we include the following state-of-the-art solvers for simple neuron models (labelled 1a to 1c, and restrained to linear ODEs with uncorrelated states) and complex models (2a-2c): **(1a)** the *cnexp* fixed step solver in NEURON, with added SIMD, providing an interleaved resolution of current and states as linear equations, with an analytical resolution of first-order ODEs describing state variables; **(1b)** the *Euler* solver in NEURON, with added SIMD, resolving the current-states dependency with an explicit Euler method with staggered timestepping, and as a linear equations; **(1c)** the same *Euler* method on a FAP execution model, presented in our previous work (Fig. 1 B3); **(2a)** the BSP fixed step *derivimplicit* solver available in NEURON, with added SIMD, with interleaved-timestep resolution of current as a linear equation, and implicit resolution of individual mechanism state ODEs; **(2b)** the BSP variable step method in NEURON with added SIMD and a collective communication barrier (Fig. 1 B2); and **(2c)** the SIMD-enabled FAP with variable timestepping introduced in this paper (Fig. 1 B4). We tested our methods in neural networks ranging from 1024 to 65536 neurons, a scale that approximates two columns in the rodent neocortex, and the maximum allowed due to memory requirements of the BDF order. The benchmark results are presented in Figure 6. The FAP variable step method (2c●) is presented alongside two variants — labelled 2c● and 2c● — that group and deliver instantly the discontinuity events within an interval equivalent to the timestep $\Delta t/2$ and Δt of the interleaved- and fixed timestep methods, respectively. This approach yields a level of reduced precision in the delivery of events — similar to fixed step methods — while maintaining the same high variable-order variable-step accu-

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

racy during continuous periods of activity and reducing significantly the number in IVP resets. CVODE-based executions are displayed for an absolute tolerance (atol) of 10^{-3}, the default value in the NEURON simulator. For brevity, we omit CVODE executions with an absolute tolerance of 10^{-2} — tested and delivering a runtime reduction of 5%-8%) — and the analysis comparing only simple fixed-step solvers (1a-1c), covered in our previous work [11].

4 Discussion

4.1 Fixed- vs Variable-Timestep Interpolators

Fixed step methods do not yield significantly-different execution times across different spiking regimes. This is due to the homogeneous computation of neuron state updates throughout time, and the light computation attached to synaptic events and collective communication not yielding a substantial increase of runtime. As expected, variable step executions are penalized on regimes with high discontinuity rates. Runtimes of fixed- and variable-step solvers approximate as we increase the spiking rate, i.e. the increase of runtimes with the input size is steeper for variable timestep (2b● and 2c●●●) compared to fixed timestep methods (2a●). This is due to discontinuities in variable-step being delivered throughout a continuous time line, compared to the discrete delivery instants of the fixed-step methods, — therefore increase the number of interpolation steps; and the iterative model of the variable timestep reinitializing the state computation with small step sizes on each IVP reset. A remarkable performance is visible on the quiet dynamics use case, where our fully-implicit ODE solver of complex models (with Newton iterations), still runs faster than the simple solver resolving only a system of linear equations. The underlying rationale is that — despite the inherent computation cost of Newton iterations in the variable step methods — the low level of discontinuities allow for very long steps, that surpass the simulation throughput of fixed step methods. The measured speed-up of our reference method (2c●) compared to the reference fixed step method (2a●) was of 544-65× across input sizes for the quiet dynamics, down to 7.7-1.8× to the moderate dynamics. The fast dynamics presented a speed-up of twofold for the dataset of 1024 neurons, and a similar runtime for the 66K neurons. The burst dynamics, although of very unlikely probability of occurrence, demonstrated an acceleration of 1.5× for 1024 neurons and a deceleration of 1.5× for 66K neurons.

4.2 Variable Step Event Grouping

On the analysis of the performance of the CVODE with grouping of events within half fixed timestep (2c●), when applied to the largest dataset tested, the previous acceleration was reduced to 47× for quiet, 4.4× for slow, and 1.2× for moderate dynamics, with an inferior performance on the remaining regimes. A further reduction of speed-up to 33× for quiet and 1.9× for slow dynamics was noticeable on the CVODE implementation without events grouping (2c●), with lower performance for the remaining spike regimes. Although being more precise and solving correlated states implicitly, this method runs slower than

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

the reference implicit fixed step method 2a● in the use cases characterized by a high number of neurons and/or strong network activity. This goes in line with the conclusions in Section 3.3, confirming that performance is activity dependent, and the performance depends on the network connectivity. The speed-up introduced across FAP CVODE variants (2c●●●) increases with the amount of discontinuities in the system — correlated to high network activity or size — as the efficiency of the event grouping method is related to the amount of events in the same grouping interval that are delivered at once.

4.3 Fully-Asynchronous vs Bulk-Synchonous Execution Models

We study the performance difference between the BSP and FAP execution models. Results show that runtimes of both implementations approximate with an increase of input. This is visible by comparing the fixed step trajectories 1b♦ and 1c♦, and the variable step trajectories 2b● and 2c●. For small network sizes, the difference in runtime is few orders of magnitude higher than for larger network sizes. On large models, the runtimes are similar. This property was demonstrated in our previous work: in brief, an increase of network leads to a higher number of network connectivity, reducing the maximum stepping interval per neuron, and approximating it to the communication delay in BSP methods. On fixed step methods, it is noticeable a similar runtime on large (66K) networks of neurons, as timesteps are computationally homogeneous. On variable step methods, similar runtimes are only noticeable when significant network activity is present (moderate, fast and burst dynamics), as little network activity leads to few discontinuities and analogously large variable step intervals.

4.4 Runtime Dependency on Input Size and Spike Activity

It is known that, on simulations of small networks, variable-timestep methods yield a significant acceleration in time to solution compared to fixed timestep methods [8], due to little interneuron connectivity. However, the larger networks yield up to 10 thousand synapses per neuron, with the number of discontinuities in the system being related to the network activity. The question lies now on which conditions are required for similar computation complexity in both interpolators. To that extent, we measured the regions of similar runtime growth for the reference fixed step (2b●) and our variable step methods (2c●). The region is labelled as ▭ in Figure 6. As expected, fixed step methods yield a quasi-linear runtime growth with the increase of the input size, and are independent of the spiking regime, due to almost ideal scaling of the algorithm in the BSP model. On the other hand, the runtime of variable timestep methods — dependent on the number of discontinuities — demonstrates a rapidly increasing growth with the input size outside the region of similar growth, and almost linearly inside. Moreover, as it depends on the network activity, the lower limit of the region increases with the spiking rate, and is delimited at 16.4K, 32.8K and 32.8K neurons or more for the quiet, slow and moderate dynamics, while not visible in the fast and burst dynamics. In the three spiking regimes where such region exists,

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

the similar growth in both approaches provides a confidence of the scaling capabilities of our methods in larger network models. This is of high importance as it provides an estimation of runtime upper bound in simulations combining neurons with heterogeneous spiking rates, as discussed next.

4.5 Overall Runtime Speed-up Estimation

To conclude our analysis, we computed an estimation of the performance acceleration on a simulation combining several spiking regimes. We measured the the distribution of neuron spike rates and neurons per spiking regime, following the laboratory experiment simulation described in Section 3.3. Estimations were collected from 2-4s of simulation time from the central minicolumn (31.3K neurons) of a 219K neurons network, to avoid boundary-effects from reduced connectivity and the initial artificial synaptic burst from the current injection. The measured percentage of neurons on each regime is 31.43%, 38.44%, 27.02%, 3.10% and 0.01%, relating to 68.9K, 84.3K, 59.2K, 6.8K and 22 neurons. Following the runtimes described in Section 4.1, the speed-up range for the interval of 1024-66K neurons when comparing our methods with the state-of-the-art solver for complex models (2a●) are estimated as: 224.5-11.9x for the variable step method with precise event delivery (2c●); 225.1-17.1x for the similar implementation with delivery of events within the next half timestep (2c●); and 228.5-24.6x for the use case with full-timestep event group delivery (2c●). Since the quiet, slow and moderate dynamics regimes weight over 95% in the runtime calculation, and as for datasets above 32.8K the reference vs benchmark runtimes have a similar runtime growth in those regimes, we believe the overall runtime and scaling properties are almost fully-preserved on larger networks.

Summary and Closing Remarks

This paper presented a distributed simulation of detailed neuron models with variable-order variable-timestep methods on a fully-asynchronous execution model, yielding asynchronous computation, communication and synchronisation. We detailed state-of-the-art approaches based on the Bulk Synchronous Parallel execution model (BSP), their limitations on the numerical resolution of complex neuron models, computation load imbalance, and speculative computing in variable-step simulations. We simulate five spiking regimes that characterize several dynamics of the mammal brain, on up to 65536 neurons on 64 Cray XE6 compute nodes, and compare our methods against five state-of-the-art numerical solvers. Results demonstrate higher numerical accuracy, with a speed-up of 544-65× for a quiet spiking regime of 0.25Hz representing a majority of neurons in regular brain activity, down to 7.7-1.8× to a moderate regime of 6.5Hz, and 2× to no acceleration for 38Hz, a pattern of unlike occurrence or short duration. An analysis of performance achievable on the simulation of a laboratory experiment demonstrates a speed-up of 224.5-11.9x for an execution with precise delivery of events, increasing to 225.1-17.1x and 228.5-24.6x for two optimized alternatives that group events delivery in the next half and full timestep. with the scaling properties of our methods preserved on larger networks of neurons.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8

As a final remark, although being applied to a network of neurons, most methods presented are problem-independent and do not require intrinsic knowledge of the problem domain, therefore opening the prospectus for the acceleration of a wide domain of scientific problems modelled by complex systems of ODEs.

Acknowledgements

This study was supported by funding to the Blue Brain Project, a research center of the École polytechnique fédérale de Lausanne, from the Swiss government's ETH Board of the Swiss Federal Institutes of Technology. The computing infrastructures were provided by Indiana University. A portion of Michael Hines efforts was supported by NINDS grant R01NS11613.

References

1. Brunel, N.: Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. Journal of computational neuroscience **8**(3), 183–208 (2000)
2. Cohen, S.D., Hindmarsh, A.C.: Cvode, a stiff/nonstiff ode solver in c. Computers in physics **10**(2), 138–143 (1996)
3. Graupner, M., Brunel, N.: Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location. Proceedings of the National Academy of Sciences p. 201109359 (2012)
4. Hines, M.L., Carnevale, N.T.: The neuron simulation environment. Neural computation **9**(6), 1179–1209 (1997)
5. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of physiology **117**(4), 500–544 (1952)
6. Kaiser, H., Brodowicz, M., Sterling, T.: Parallex an advanced parallel execution model for scaling-impaired applications. In: Parallel Processing Workshops, 2009. ICPPW'09. International Conference on. pp. 394–401. IEEE (2009)
7. Kumbhar, P., Hines, M., Fouriaux, J., Ovcharenko, A., King, J., Delalondre, F., Schürmann, F.: Coreneuron : An optimized compute engine for the neuron simulator (2019)
8. Lytton, W.W., Hines, M.L.: Independent variable time-step integration of individual neurons for network simulations. Neural computation **17**(4), 903–921 (2005)
9. Magalhaes, B., Hines, M., Sterling, T., Schuermann, F.: Exploiting implicit flow graph of system of odes to accelerate the simulation of neural networks. Proceedings of IEEE International Parallel & Distributed Processing Symposium (IPDPS)
10. Magalhaes, B., Hines, M., Sterling, T., Schuermann, F.: Asynchronous simd-enabled branch-parallelism of morphologically-detailed neuron models. Frontiers in Neuroinformatics (2019)
11. Magalhaes, B., Hines, M., Sterling, T., Schuermann, F.: Fully-asynchronous cache-efficient simulation of detailed neural networks. Proceedings of International Conference on Computational Science (ICCS) pp. 421–434 (2019)
12. Markram, H., Muller, E., Ramaswamy, S., Reimann, M.W., Abdellah, M., Sanchez, C.A., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., et al.: Reconstruction and simulation of neocortical microcircuitry. Cell **163**(2), 456–492 (2015)
13. Sterling, T., Anderson, M., Bohan, P.K., Brodowicz, M., Kulkarni, A., Zhang, B.: Towards exascale co-design in a runtime system. In: Exascale Applications and Software Conference. Stockholm, Sweden (Apr 2014)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50426-7_8