RateML: a code generation tool for BrainNetwork Models

Michiel van der Vlag[1,*,†], Marmaduke Woodman[2,†], Jan Fousek[2], SandraDiaz-Pier[1], Aaron Perez Martin[1], Viktor Jirsa[2] and Abigail Morrison[1,3]

[1] Forschungszentrum Jülich, Jülich Super Computingcenter (JSC), Jülich, Germany

[2] Institut de Neurosciences des Systèmes, Aix Marseille Universite, Marseille, France

[3] Institute of Neuroscience and Medicine (INM-6) & Institute for Advanced Simulation(IAS-6) & JARA-Institute Brain

Correspondence*:

Michiel van der Vlag
m.van.der.vlag@fz-juelich.de

INTRODUCTION/MOTIVATION

Understanding the relationship between structure and function in the brain is a highly multidisciplinary endeavour; it requires scientists from different fields to develop and explore hypotheses based on both experimental data and the theoretical considerations from diverse scientific domains [1]. Because of this, simulation platforms have become essential tools to understand different states of the brain and promise, in the future, to provide a way of reproducing enough features of brain activity in order to better understand healthy brain states, diseases, aging, and development [2].

One particularly promising approach is whole-brain simulation based on non-invasive brain imaging techniques suitable for use in human studies [3]. Functional and structural imaging modalities including Electroencephalography (EEG), Magnetoencephalography (MEG), Magnetic Resonance Imaging (MRI), and functional Magnetic Resonance Imaging (fMRI) allow researchers to capture characteristics of the brain primarily at a mesoscopic scale. The brain activity measured by such methods can be mathematically modelled and simulated using The Virtual Brain simulator [4].

One of the strengths of such whole-scale brain simulation is the possibility of personalization of the model for a particular subject [5][6][7]. The basic approach for personalising the simulated model behaviour entails finding the best fit between the numeric solution of the derivative equations, which determine the behaviour of the model, and the patient-specific functional empirical data [8]. Consequently, large parameter exploration are frequently carried out at high performance computing (HPC) centers.

Translating the set of differential equations into a concrete implementation is complex, as several factors can dramatically influence performance and correctness of the simulation.

We therefore conclude that abstracting the modeling from the computational implementation, such that model descriptions can be automatically translated into correct and performant implementations [9], would considerably aid these scientists to exploit the possibilities of whole-brain simulation.

METHODS

To this end, we have developed RateML, a modeling workflow tool that uncouples the specification of Neural Mass Models (NMMs) and Brain Network Models (BNMs) from their implementations as machine code for specific hardware. It is based on the existing domain specific language 'Low Entropy Model Specification' (LEMS) [10], which allows the user to enter declarative descriptions of model components in a concise XML representation. RateML enables users to generate brain models based on an XML format in which the generic features of rate-based neuron models can be addressed, without needing extensive knowledge of mathematical modelling or hardware implementation. In addition to providing code generation of the described models in Python, it is also possible to generate CUDA [11] code in which variables of interest can be designated with a range for parameter exploration. The generated Python code can be directly executed within the TVB simulation framework, whereas the CUDA code has a separate driver module which is also generated before execution.

RESULTS

To chart the simulators behaviour, three CUDA models are benchmarked, namely the Kuramoto [12], WongWang [13] and Epileptor [14], which have respectively 1, 2 and 6 state variables. The benchmarks are ran for 4 seconds of simulated time, 40,000 simulation steps with dt = 0.1 ms and integrated with the Euler method. Parameters size is increased to observe the run-time behaviour and memory scaling. The benchmark experiments are executed on the JuwelsBooster clusters equipped with A100 GPUs with 40 GB of High Bandwidth Memory 2 with a bandwidth of 1,555 GB/s.

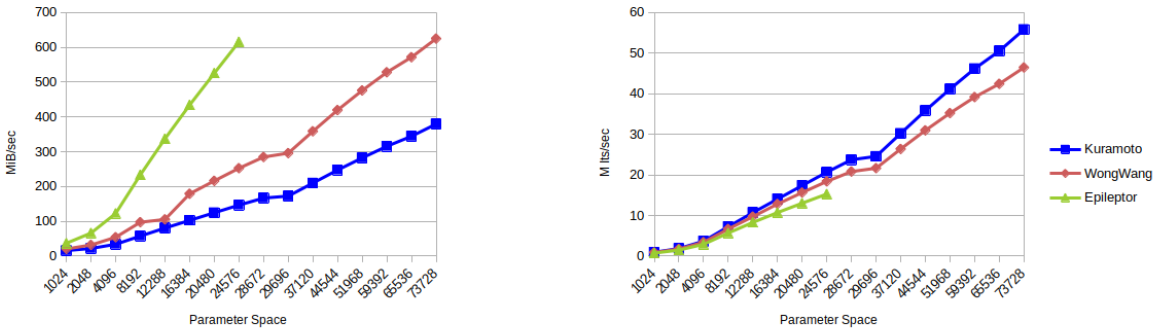

Figure 1. Memory bandwidth consumption and model iterations per second for three TVB CUDAmodels when scaling the parameter space. Results indicate linear scaling when increasing the parameterspace and that the application is memory bound.

From the result shown in figure 1 it can be concluded that the time it takes to execute a model and the memory consumed scale linearly with the size of the parameter space. The results also show at which the maximum number of parameters, the different models can still be simulated. Even simulations with the largest possible parameter space dimension execute under 65 seconds. The CUDA executions show, for these parameter spaces, that they are memory bound. In theory, the parameter space for the Kuramoto, which has a single state, could be doubled before all the memory of the GPU is consumed.

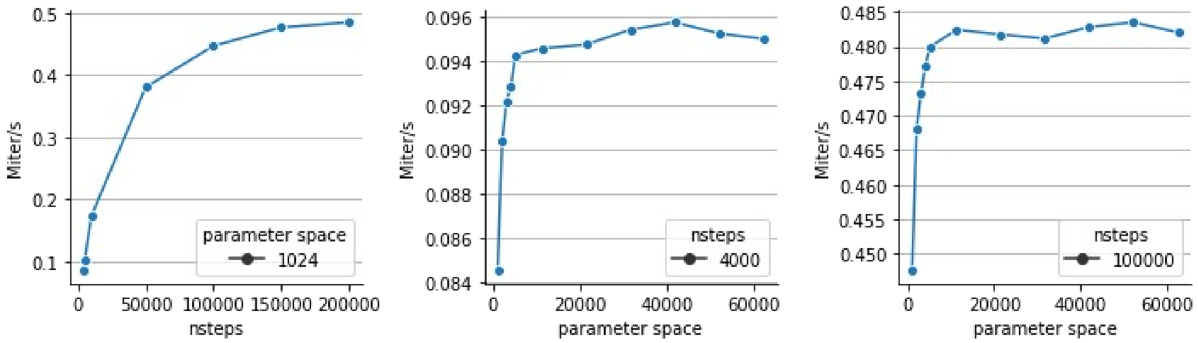

Figure 2. Performance comparison to the code-generated TVB Numba backend (CPU) on a single CrayXC40 node (2*18 cores, 2.10GHz, 128GB RAM).

Finally, for comparison we present the performance of the CPU Numba backend of TVB using template generated code. For this we have configured the simulation with the Montbrió [15] model driven by noise and a connectome of 68 nodes and delays induced by a propagation speed of 2m/s. As this configuration is representative of recent resting-state studies [16], we explored two simulation lengths: 4000 iteration step microbenchmark and a longer experiment spanning several BOLD time points of 100,000 integration steps corresponding to 10s of simulated time. The benchmark was executed on a single Cray XC40 node (Intel Xeon E5-2695 v4, 2*18 cores, 2.10GHz, 128GB RAM) of the multicore partition located in Piz Daint. The results presented in Figure 2 show, that the performance increases with the length of the simulation hinting at better amortization of the simulation initialization overhead and better data locality. Secondly, the peak perfomance is reached already for small parameter space sizes.

DISCUSSION

RateML enables the user to generate complex Python and CUDA neural mass models and to do fast parameters sweeps on the GPU, with identical results as when done with TVB. At the moment some of the variables within the LEMS components used in RateML have been modified to fit specific functionality and match the TVB simulation strategy. Even though at the moment models produced by RateML can not be directly ported to other simulators which support LEMS and are able to simulate BNMs or NMMs, work is being done in collaboration with the LEMS development community to fit all the requirements and perform any required modifications to RateML or extensions to the standard.

Keywords: brain network models, simulation, high performance computing, automatic code generation, domain specific language

REFERENCES

[1]. [Dataset] Peyser, A., Diaz Pier, S., Klijn, W., Morrison, A., and Triesch, J. (2019). Linking experimental and computational connectomics

[2]. Einevoll, G. T., Destexhe, A., Diesmann, M., Gr ¨un, S., Jirsa, V., de Kamps, M., et al. (2019). The scientific case for brain simulations. Neuron 102, 735–744388

[3]. Lynn, C. W. and Bassett, D. S. (2019). The physics of brain network structure, function and control. Nature Reviews Physics 1, 318–332

[4]. Sanzleon, P., Knock, S. A., Woodman, M. M., Domide, L., Mersmann, J., Mcintosh, A. R., et al. (2013). The virtual brain: A simulator of primate brain network dynamics. Frontiers in Neuroinformatics 7. doi:10.3389/fninf.2013.00010

[5]. Falcon, M. I., Jirsa, V., and Solodkin, A. (2016). A new neuroinformatics approach to personalized medicine in neurology: The virtual brain. Current opinion in neurology 29, 429 Furber, S. B., Lester, D. R., Plana, L. A., Garside, J. D., Painkras, E., Temple, S., et al. (2012). Overview of the spinnaker system architecture. IEEE Transactions on Computers 62, 2454–2467

[6]. Bansal, K., Nakuci, J., and Muldoon, S. F. (2018). Personalized brain network models for assessing structure–function relationships. Current Opinion in Neurobiology 52, 42–47

[7]. Hashemi, M., Vattikonda, A., Sip, V., Guye, M., Bartolomei, F., Woodman, M. M., et al. (2020). The bayesian virtual epileptic patient: A probabilistic framework designed to infer the spatial map of epileptogenicity in a personalized large-scale brain model of epilepsy spread. NeuroImage 217, 116839

[8]. Deco, G., McIntosh, A. R., Shen, K., Hutchison, R. M., Menon, R. S., Everling, S., et al. (2014). Identification of optimal structural connectivity using functional connectivity and neural modeling. Journal of Neuroscience 34, 7910–7916386

[9]. Blundell, I., Brette, R., Cleland, T. A., Close, T. G., Coca, D., Davison, A. P., et al. (2018). Code Generation in Computational Neuroscience: A Review of Tools and Techniques. Frontiers in Neuroinformatics 12, 68. doi:10.3389/fninf.2018.00068

[10]. Vella, M., Cannon, R. C., Crook, S., Davison, A. P., Ganapathy, G., Robinson, H. P. C., et al. (2014). libNeuroML and PyLEMS: using Python to combine procedural and declarative modeling approaches in computational neuroscience. Frontiers in Neuroinformatics 8, 38. doi:10.3389/fninf.2014.00038

[11]. [Dataset] NVIDIA, Vingelmann, P., and Fitzek, F. H. (2020). Cuda, release: 10.2.89

[12]. Kuramoto, Y. (1975). International symposium on mathematical problems in theoretical physics. Lecture notes in Physics 30, 420

[13]. Wong, K.-F. and Wang, X.-J. (2006). A recurrent network mechanism of time integration in perceptual decisions. Journal of Neuroscience 26, 1314–1328425

[14]. Jirsa, V. K., Stacey, W. C., Quilichini, P. P., Ivanov, A. I., and Bernard, C. (2014). On the nature of seizure dynamics. Brain 137, 2210–2230

[15]. Montbri ´o, E., Paz ´o, D., and Roxin, A. (2015). Macroscopic description for networks of spiking neurons. Physical Review X 5, 1–15. doi:10.1103/PhysRevX.5.021028

[16]. Rabuffo, G., Fousek, J., Bernard, C., and Jirsa, V. (2021). Neuronal cascades shape whole-brain functional dynamics at rest. eNeuro doi:10.1523/ENEURO.0283-21.2021