GPU-accelerated diffusion-weighted MR simulations on the cloud in Python using Disimpy and freely available hardware

Leevi Kerkelä[a], Marco Palombo[b,c], and Chris A. Clark[a]

[a] UCL Great Ormond Street Institute of Child Health, University College London, London, UK
[b] Cardiff University Brain Research Imaging Centre, School of Psychology and School of Computer Science and Informatics, Cardiff University, Cardiff, UK
[c] UCL Centre for Medical Image Computing, University College London, London, UK

Synopsis

We demonstrate using Disimpy on Google Colaboratory to perform GPU-accelerated diffusion-weighted MR simulations using hardware that is available on the cloud at no cost. Experiments are performed using complex microstructure and the results are compared to Camino. Using the freely available hardware, Disimpy was found to be over 600 times faster than Camino running single-threaded on a modern workstation desktop computer. The performance of Disimpy can be further improved by purchasing access to more powerful GPUs.

Introduction

In diffusion-weighted MR research, the Monte Carlo method is often used to simulate diffusion by generating a large number of random walks from which the signal can be calculated. Since the random walks are independent, generating them in parallel can greatly decrease simulation runtime. With general-purpose graphical processing unit (GPU) computing, it is possible to generate tens of thousands of random walks simultaneously on a single device, enabling the application of complex and large scale simulations[1–4]. Recent advances in both GPU and cloud computing technologies have made powerful GPUs more accessible. For example, Google Colaboratory[5] (Colab) is a Jupyter notebook[6] service that requires no setup to use and provides free access to computing resources including Nvidia GPUs. Disimpy[7] is an open-source diffusion-weighted MR simulator, written in Python, that generates random walks in massively parallel on Nvidia CUDA-capable[8] GPUs. Here, we demonstrate using Disimpy on Colab to accurately simulate diffusion in complex microstructure, and compare it to Camino[9,10], a popular simulator that runs serially on the central processing unit (CPU), in terms of runtime.

Methods

Disimpy was installed on Colab using pip and the simulations were run on Nvidia Tesla K80 that was available at no cost. A tutorial for performing simulations on Colab and reproducing the presented results is available at https://disimpy.readthedocs.io. Camino simulations were performed on Intel Xeon E5-1620 v3 3.5 GHz. Diffusion was simulated inside a computational model of a motor neuron by Palombo et al.[11], shown in Figure 1. A standard pulsed gradient spin echo (PGSE) with $\delta = 20$ ms and $\Delta = 60$ ms was used with eleven b-values uniformly distributed between 0 and 5 ms/μm^2. The diffusion encoding direction was aligned with the x-axis. 10^4 time steps were used with a step length of 0.31 um. Diffusivity was 2 μm^2/ms. The number of random walkers was varied between 1 and 10^6 for Disimpy and between 1 and 10^4 for Camino. The initial positions of the random walkers were sampled from a uniform distribution inside the surface of the neuron model prior to the simulation experiments and passed to the simulators as a parameter.

Results

The simulated signals for 10^4 random are shown in Figure 2, showing an excellent agreement between the the signals generated by Disimpy and Camino. The runtime of each simulation is shown in Figure 3. The runtime of Camino depends linearly on the number of random walkers as the random walks are generated serially on the CPU. Notably, due to massive parallelization, the runtime of Disimpy does not linearly depend on the number of random walkers until it is in the tens of thousands. For 10^6 random walkers, the simulation took 22 minutes with Disimpy and it would have taken 10 days with Camino.

Discussion and conclusion

We demonstrated using Disimpy on Colab to efficiently perform accurate diffusion-weighted MR simulations in massively parallel without the need for anything but a web browser and an internet connection. Using the freely available GPU, the performance gain over Camino was significantly greater than what has been previously demonstrated by running Disimpy on a workstation computer[12]. Camino was used as a benchmark because despite its limitations it continues to be regularly used research[13–15]. Camino simulations are often accelerated by aggregating the results of several simulations performed simultaneously on several threads on a single machine or a cluster. Similarly, Disimpy simulations can be further accelerated by using multiple devices or more powerful GPUs. Finally, it must be mentioned that more modern simulators have been developed and open-sourced by, for example, Lee et al.[16] and Patino et al.[17]. However, they can not be run on Colab like Disimpy because they are not Python packages.

Figures

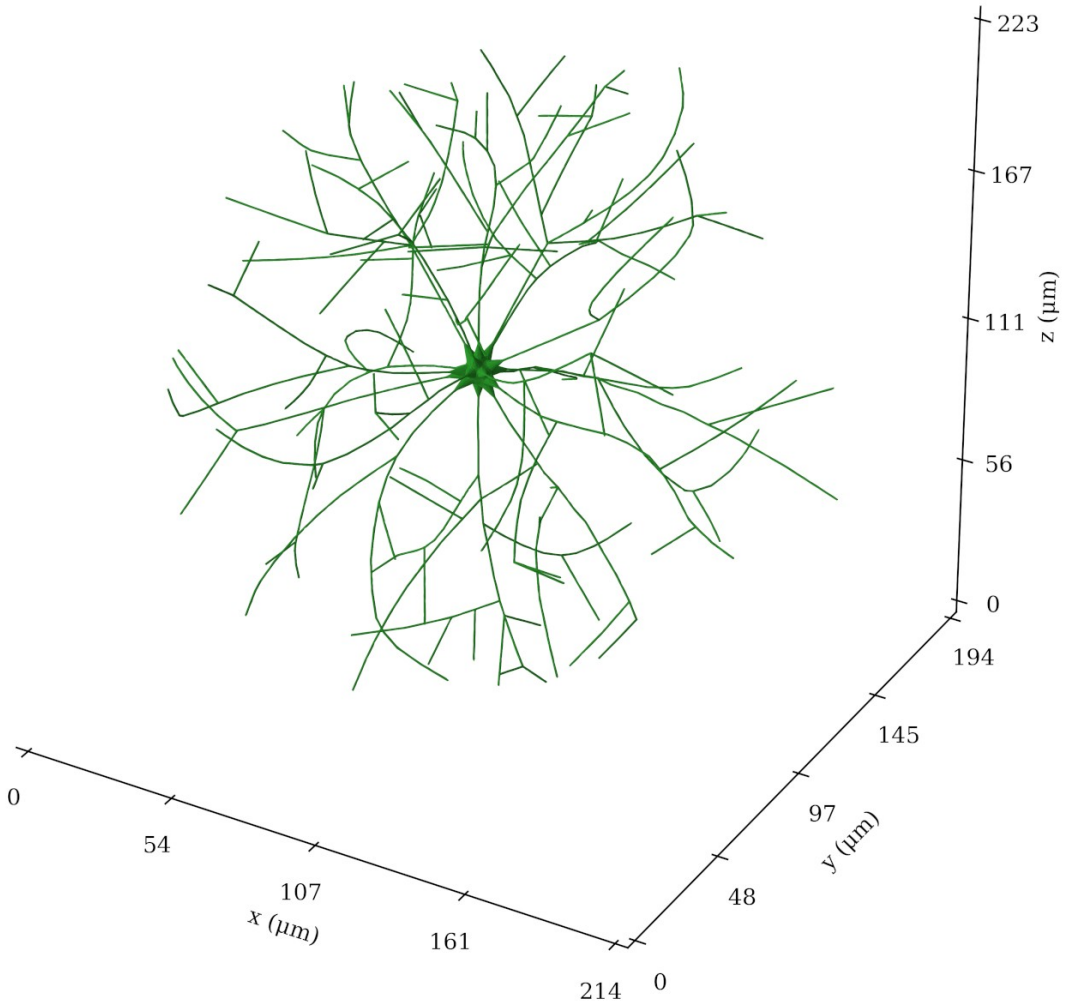

Figure 1: The simulated microstructure mimicking the morphology of a motor neuron. The neuron model was generated using the algorithm by Palombo et al.[11]. The surface is an impermeable surface consisting of 29,688 triangles.

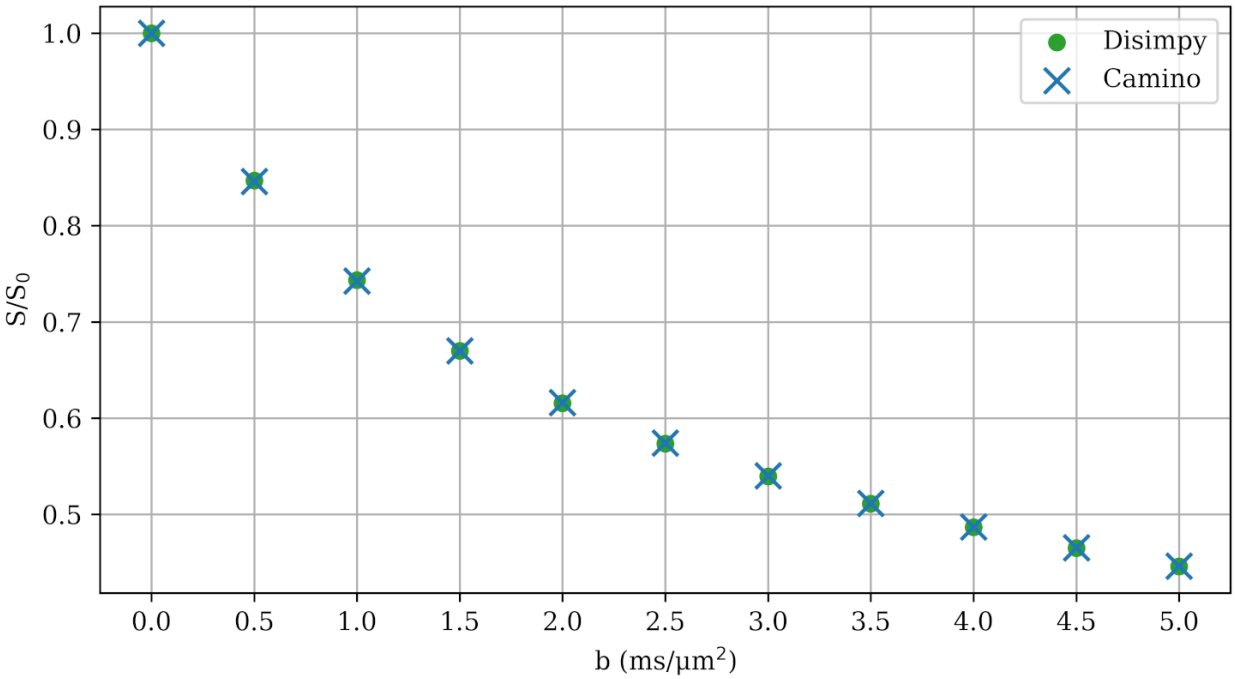

Figure 2: The simulated signals with 10^4 random walkers using both Disimpy and Camino.

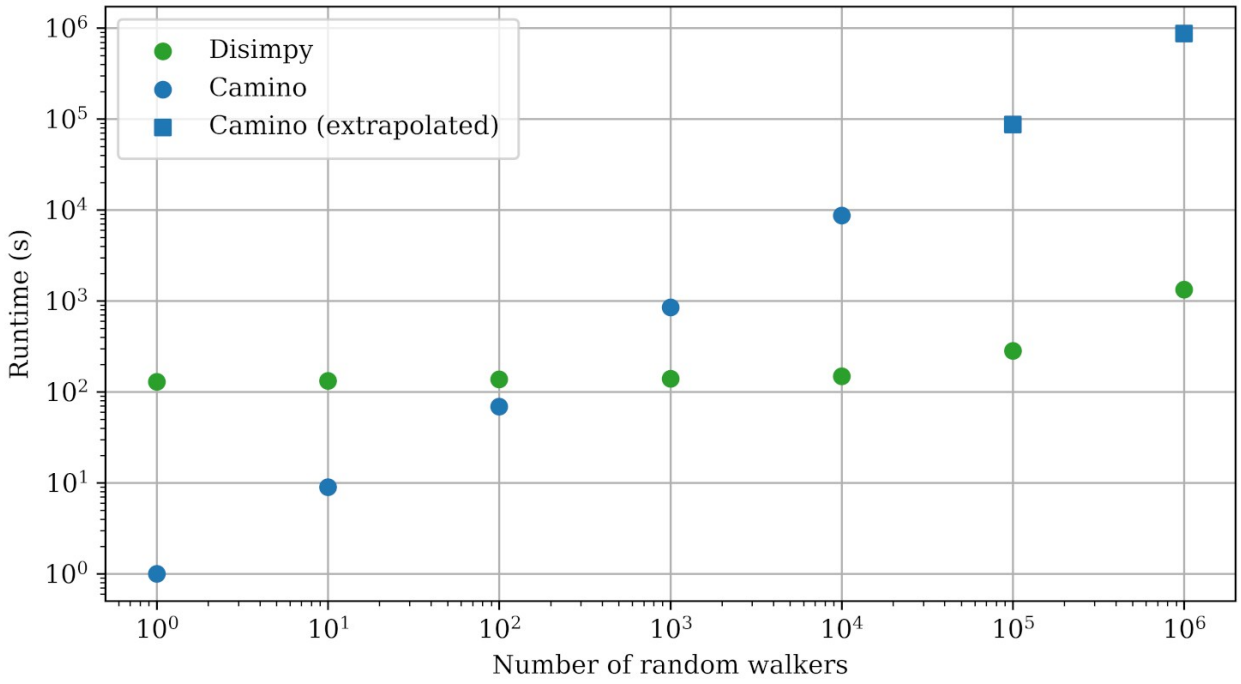

Figure 3: Runtime comparison between Disimpy on Nvidia Tesla K80 and Camino on Intel Xeon E5-1620 v3 3.5 GHz. The runtime of Camino for more than 10^4 random walkers was extrapolated. With 10^6 random walkers, Disimpy was 654 times faster than Camino would have been.

References

1. Nilsson, M. *et al.* An open-source framework for analysis of multidimensional diffusion MRI data implemented in MATLAB. in *Intl. Soc. Mag. Reson. Med* vol. 26 16–19 (2018).

2. Nguyen, K. Van, Hernández-Garzón, E. & Valette, J. Efficient GPU-based Monte-Carlo simulation of diffusion in real astrocytes reconstructed from confocal microscopy. *J. Magn. Reson.* **296**, (2018).

3. Lee, H. H., Papaioannou, A., Kim, S. L., Novikov, D. S. & Fieremans, E. A time-dependent diffusion MRI signature of axon caliber variations and beading. *Commun. Biol.* **3**, (2020).

4. Ligneul, C. *et al.* Diffusion-weighted magnetic resonance spectroscopy enables cell-specific monitoring of astrocyte reactivity in vivo. *Neuroimage* **191**, (2019).

5. Google Colaboratory. https://colab.research.google.com/.

6. Granger, B. E. & Perez, F. Jupyter: Thinking and Storytelling with Code and Data. *Comput. Sci. Eng.* **23**, (2021).

7. Kerkelä, L., Nery, F., Hall, M. & Clark, C. Disimpy: A massively parallel Monte Carlo simulator for generating diffusion-weighted MRI data in Python. *J. Open Source Softw.* (2020) doi:10.21105/joss.02527.

8. Nickolls, J., Buck, I., Garland, M. & Skadron, K. Scalable parallel programming with CUDA. *Queue* (2008) doi:10.1145/1365490.1365500.

9. Cook, P. a *et al.* Camino: Open-Source Diffusion-MRI Reconstruction and Processing. *14th Sci. Meet. Int. Soc. Magn. Reson. Med.* **14**, (2006).

10. Hall, M. G. & Alexander, D. C. Convergence and Parameter Choice for Monte-Carlo Simulations of Diffusion MRI. *IEEE Trans. Med. Imaging* (2009) doi:10.1109/TMI.2009.2015756.

11. Palombo, M., Alexander, D. C. & Zhang, H. A generative model of realistic brain cells with application to numerical simulation of the diffusion-weighted MR signal. *Neuroimage* (2019) doi:10.1016/j.neuroimage.2018.12.025.

12. Kerkelä, L., Henriques, R. N., Hall, M. G., Clark, C. A. & Shemesh, N. Validation and noise robustness assessment of microscopic anisotropy estimation with clinically feasible double diffusion encoding MRI. *Magn. Reson. Med.* (2019).

13. Hill, I. *et al.* Machine learning based white matter models with permeability: An experimental study in cuprizone treated in-vivo mouse model of axonal demyelination. *Neuroimage* **224**, (2021).

14. Cottaar, M. *et al.* Quantifying myelin in crossing fibers using diffusion-prepared phase imaging: Theory and simulations. *Magn. Reson. Med.* **86**, (2021).

15. Ianus, A., Alexander, D. C., Zhang, H. & Palombo, M. Mapping complex cell morphology in the grey matter with double diffusion encoding MR: A simulation study. *Neuroimage* **241**, (2021).

16. Lee, H. H., Fieremans, E. & Novikov, D. S. Realistic Microstructure Simulator (RMS): Monte Carlo simulations of diffusion in three-dimensional cell segmentations of microscopy images. *J. Neurosci. Methods* **350**, (2021).

17. Rafael-Patino, J. *et al.* Robust Monte-Carlo Simulations in Diffusion-MRI: Effect of the Substrate Complexity and Parameter Choice on the Reproducibility of Results. *Front. Neuroinform.* **14**, (2020).