Accelerated Neutrino Oscillation Probability Calculations and Reweighting on GPUs

Richard Graham Calland[1]
[1]University of Liverpool, Department of Physics, Oliver Lodge Bld, Oxford Street, Liverpool, L69 7ZE, UK

DOI: http://dx.doi.org/10.3204/DESY-PROC-2014-05/23

Neutrino oscillation experiments are reaching high levels of precision in measurements, which are critical for the search for CP violation in the neutrino sector. Inclusion of matter effects increases the computational burden of oscillation probability calculations. The independency of reweighting individual events in a Monte Carlo sample lends itself to parallel implementation on a Graphics Processing Unit. The library Prob3++ was ported to the GPU using the CUDA C API, allowing for large scale parallelized calculations of neutrino oscillation probabilities through matter of constant density, decreasing the execution time of the oscillation probability calculations by 2 orders of magnitude, when compared to performance on a single CPU core. Additionally, benefit can be realized by porting some systematic uncertainty calculations to GPU, especially non-linear uncertainties evaluated with response functions. The implementation of a fast, parallel cubic spline evaluation on a GPU is discussed. The speed improvement achieved by using the GPU calculations with the T2K oscillation analysis is also noted.

1 Neutrino Oscillation Probability for Long Baseline Experiments

It is established that neutrinos exhibit oscillation between flavour states [1][2]. The standard formalism describes this phenomena using a unitary transition matrix (PMNS) to compute the probability of a neutrino ν_α to change to ν_β. A neutrino travelling a distance L (km) from its source has a probability to change flavour defined as

$$\begin{aligned} P(\nu_\alpha \to \nu_\beta) = \delta_{\alpha\beta} - 4\sum_{i>j} Re(U^*_{\alpha i}U_{\beta i}U_{\alpha j}U^*_{\beta j})\sin^2(\frac{\Delta m^2_{ij}L}{4E}) \\ +2\sum_{i>j} Im(U^*_{\alpha i}U_{\beta i}U_{\alpha j}U^*_{\beta j})\sin(\frac{\Delta m^2_{ij}L}{2E}) \end{aligned} \quad (1)$$

where E is the neutrino energy, $U_{flavour,mass}$ is the mixing matrix, Δm^2_{ij} is the difference between mass states i,j and $\delta_{\alpha\beta}$ is the Kronecker delta function. However, when the neutrino propagates through matter, an extra potential is added to the equation which manifests from the forward scattering of ν_e on the ambient electrons in matter. This extra potential term requires that the mass matrix of the Hamiltonian be re-diagonalized in order to find the eigenstates

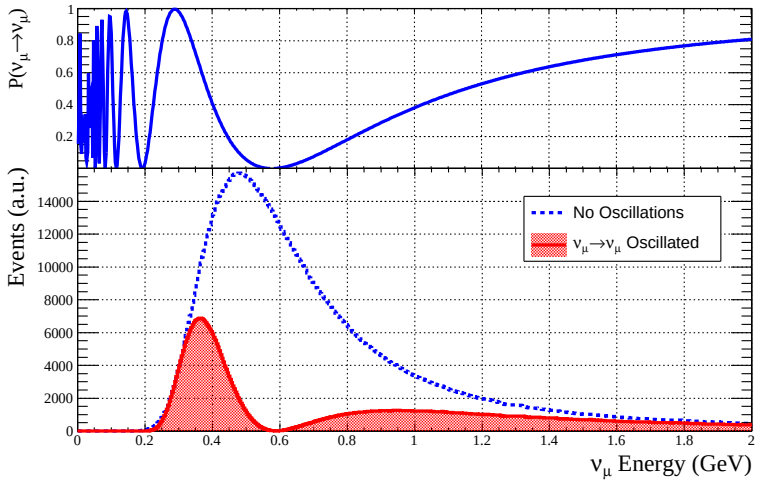

Figure 1: Effect of neutrino oscillation on a mock neutrino energy spectrum. Top plot shows the ν_μ survival probability and the bottom plot shows the mock energy spectrum with and without the survival probability applied.

in matter. The extra computation required to calculate matter effects can be costly, however there is an analytical solution prescribed in [3] which is used in the work presented here.

2 The Tokai-to-Kamioka Experiment

The Tokai-to-Kamioka (T2K) experiment [4] is a second generation long baseline neutrino oscillation experiment, designed to use an intense beam of neutrinos to make precision measurements of the θ_{23} mixing angle, and to look for θ_{13}. The experiment, which is located in Japan, is comprised of 3 sections. The first is located on the east coast of Japan, where the J-PARC accelerator produces a neutrino beam composed of mainly ν_μ by colliding protons onto a graphite target. This beam is measured at the second stage, by a near detector (ND280) situated 280 m from the neutrino source. ND280 can make characterisation measurements of the beam before it has the chance to undergo neutrino oscillation. This stage is important because it can provide valuable constraints on the uncertainties in the neutrino beam flux and cross section models. The final stage, takes place 295 km away at the Super-Kamiokande (SK) 50 kt water Cherenkov detector, where neutrino oscillation parameters are measured by looking for a disappearance of ν_μ-like events and an appearance of ν_e-like events.

Due to the setup of the experiment, there are multiple samples from multiple detectors, which leads to a relatively complicated oscillation analysis involving many systematic uncertainties.

3 Event-by-event Reweighting

Neutrino oscillation analyses in T2K are performed using a large sample of Monte Carlo (MC) simulated events inside the near and far detectors to construct a PDF. The number of MC

events is large due to the multiple interaction modes and reconstruction effects. The events are produced assuming no neutrino oscillation has taken place, so the effect of neutrino oscillation must be calculated as a weight for these MC events. This is visualized in Figure 1. In order to infer the most probable values of the oscillation parameters, the PDF must be reweighted in order to find the response of the detector simulation to varying values of oscillation parameters. This reweighting can be done in two ways. The first, and simplest, is to construct templates for the detector MC. This is essentially using a histogram with a suitable binning, filling the histogram with the MC events and then using the bin centre to calculate a weight for the bin. An alternative method is to keep all MC event information inside memory, and calculating a weight for each MC event. These events can then be binned using the weights to compute a binned likelihood in the same way as using templates. The obvious difference is that the event-by-event method requires orders of magnitudes more calculations, because instead of doing computations per bin, we are instead doing them per event. The advantages of using this method, if one can overcome the computational challenges, are that there is no additional loss of information by compiling events into bins; the shape information of the PDF is retained inside the bin as described in Figure 2. In recent binning schemes used for T2K PDFs, it was found that the difference in predicted number of events from reweighting differs by 1-2% between template and event-by-event methods.

Another advantage of the event-by-event method is in the case where one constructs multiple PDFs from the same set of MC events, as is the case for T2K, where different event topologies are selected from the data based on a few selection criteria. As detector systematic uncertainties are varied, this can cause events to move between sample selections. This systematic effect can easily be modelled using the event-by-event method, but it is difficult to do so with a template method due to the loss of MC information about the event category of each event. This, along with parameters that may shift the kinematics of an event and cause the event to shift between bins in a histogram, is a strong justification to retain all the relevant information about MC events and reweight each event individually.

4 Calculation on GPU

Using the event-by-event method to reweight PDFs (and thus calculate the likelihood) has the consequence of increasing the number of calculations involved by orders of magnitude. In T2K neutrino oscillation analyses, the two largest contributions to CPU time are the evaluation of the oscillation probability including matter effects, and the evaluation of cubic splines that are used to encode the non-linearity of the cross section model on the PDF. These two tasks were offloaded to a GPU using the CUDA toolkit version 5. For the oscillation probability calculation, the library Prob3++ [5] is used to calculate these probabilities on CPU. This library was ported to run on the GPU [6]. The structure of this calculation is as follows. First, the mass and mixing matrices were computed on CPU and copied to GPU memory. Then, an array of neutrino energies taken from the MC was copied to the GPU memory. An array of the same length as the energy array was allocated on GPU memory to store the results. A GPU algorithm (kernel) was then executed with each GPU thread operating on an element of the neutrino energy array. The kernel calculates a weight using the event energy and information from the mass and mixing matrices. The resulting weight is saved to the corresponding element in the weights array, which is copied back to CPU memory when the kernel has finished processing. The time taken to perform this oscillation probability calculation over varying numbers of events in a standalone

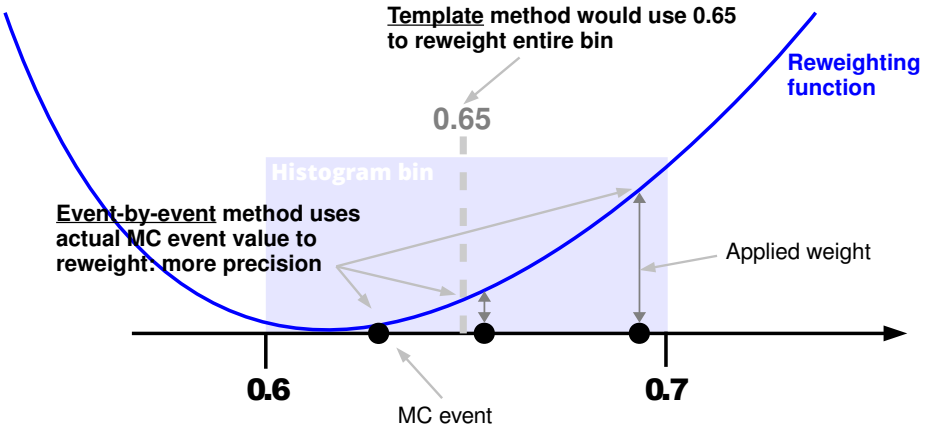

Figure 2: A comparison of template and event-by-event reweighting methods. When calculating weights using a reweighting function, the template method uses the bin centre to calculate a single weight for the whole bin. In contrast, the event-by-event method calculates a weight for each event, using the events energy value. The template method loses information about the shape of the reweighting function, whereas the event-by-event method retains it.

program was studied, with results shown in Figures 3 and 4.

In addition to oscillation probability reweighting, the simulated MC events are also reweighted according to a neutrino cross section model. This model is encoded with cubic splines to efficiently describe the non-linearity of each parameter response without having to rerun the time consuming simulation for every iteration of the fit. When cross section parameters are varied, each response function must be evaluated to produce a new weight, on a per neutrino interaction mode, per event basis. To improve the performance, the array of spline objects used in the CPU code were reformatted into a structure of arrays to better suit the GPU architecture. This structure is on the order of 1 Gb in size, and was copied only once to the GPU as a read-only resource. At each iteration of the analysis, the new parameter values were copied to GPU memory, and were used by the kernel to evaluate a single spline per GPU thread across multiple threads simultaneously. Finally, the resulting weights of the splines were copied back to CPU memory, where they were applied to the events during the construction of the PDF. When compared to the performance of the original array of spline objects structure, the GPU implementation saw a factor of 20 times speed up in a standalone benchmark. This benchmark consisted of evaluating a large number of splines both sequentially (CPU) and simultaneously (GPU) to compare execution times.

A validation study was performed for both the oscillation reweighting and the spline evaluation GPU algorithms. It was found that the relative difference between GPU and CPU versions was on the order of 10^{-12} and 10^{-6} for oscillation reweighting and spline evaluation respectively; both within acceptable tolerance for this application.

The benchmarks found in Figures 3 and 4 were executed on an Intel Xeon E5640 quad-core processor clocked at 2.67 Ghz, using an NVIDIA M2070 GPU which has 448 cores clocked at

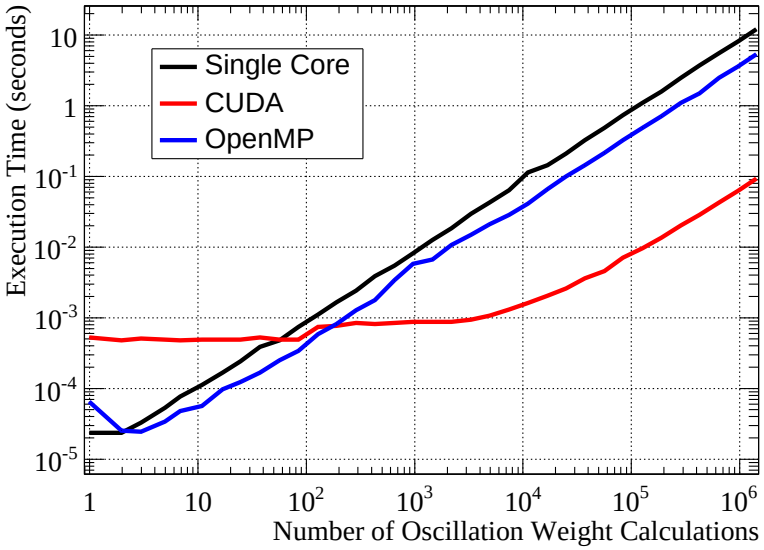

Figure 3: Time taken to evaluate the oscillation probabilities of batches of events, as a function of batch size. Single core CPU time is compared with the CUDA implementation, and also a multi-threaded version using OpenMP.

1.15 GHz. The code was compiled using the CUDA 5 toolkit and gcc version 4.6.3 with the -O2 optimization flag enabled. The OpenMP benchmark was restricted to 4 cores, using the same hardware as the single core benchmark.

5 Conclusion

When offloading both oscillation probability calculations with matter effects, and evaluation of response functions to GPU, the T2K neutrino oscillation analysis saw approximately 20 times speed up in total execution time compared to running everything on a single CPU core. There is much scope to further improve the acceleration of the T2K neutrino oscillation analysis with GPUs. The most obvious way is to perform all reweighting operations on the GPU, instead of offloading two components. However, this may require significant reworking of existing analysis code. Since the generation of the results in this study, the oscillation probability code has been improved to fit the mass and mixing matrices inside constant memory on the GPU. This increased the maximum observed speed increase factor in the standalone benchmark from 130 to 180.

6 Acknowledgments

The author would like to acknowledge the SES (Science Engineering South) Centre for Innovation service (CFI) Emerald HPC cluster, along with the HEP computing staff at the University

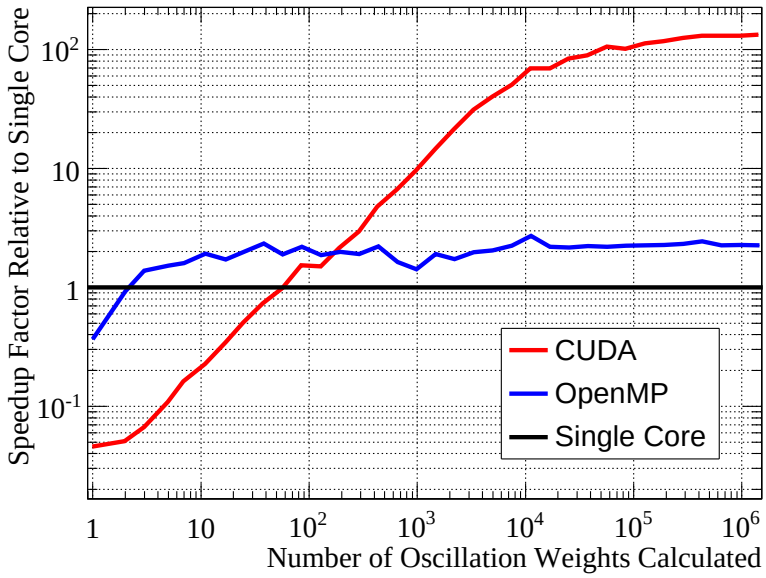

Figure 4: The speedup time relative to single core execution, which shows the speed increase factor for the GPU and OpenMP implementations. The largest speed increase factor seen between CPU and CUDA implementations was 130.

of Liverpool for their support.

References

[1] Y. Fukuda et al. Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.*, 81:1562–1567, Aug 1998.

[2] Q. R. Ahmad et al. Measurement of the Rate of $\nu_e + d \rightarrow p + p + e^-$ Interactions Produced by ^8B Solar Neutrinos at the Sudbury Neutrino Observatory. *Physical Review Letters*, 87(7):071301, August 2001.

[3] V. Barger, K. Whisnant, S. Pakvasa, and R. J. N. Phillips. Matter effects on three-neutrino oscillations. *Phys. Rev. D*, 22:2718–2726, 1980.

[4] T2K Collaboration, K. Abe, N. Abgrall, H. Aihara, Y. Ajima, J. B. Albert, D. Allan, P.-A. Amaudruz, C. Andreopoulos, B. Andrieu, and et al. The T2K experiment. *Nuclear Instruments and Methods in Physics Research A*, 659:106–135, December 2011.

[5] R. Wendell. Prob3++ software for computing three flavor neutrino oscillation probabilities. `http://www.phy.duke.edu/~raw22/public/Prob3++/`, 2012.

[6] RG Calland, AC Kaboth, and D Payne. Accelerated event-by-event neutrino oscillation reweighting with matter effects on a gpu. *Journal of Instrumentation*, 9(04):P04016, 2014.