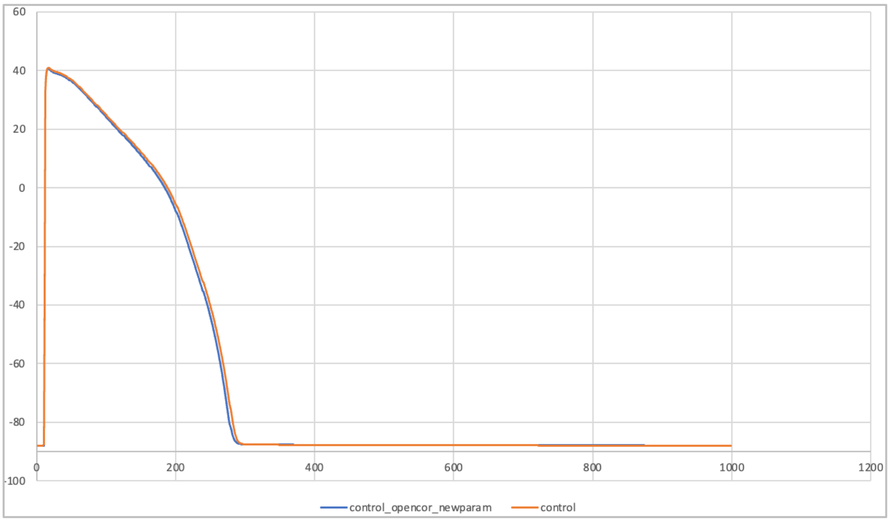
solutions generated from OpenCOR. Visual comparisons of action potential time-series data confirmed a close alignment between the two simulation platforms.

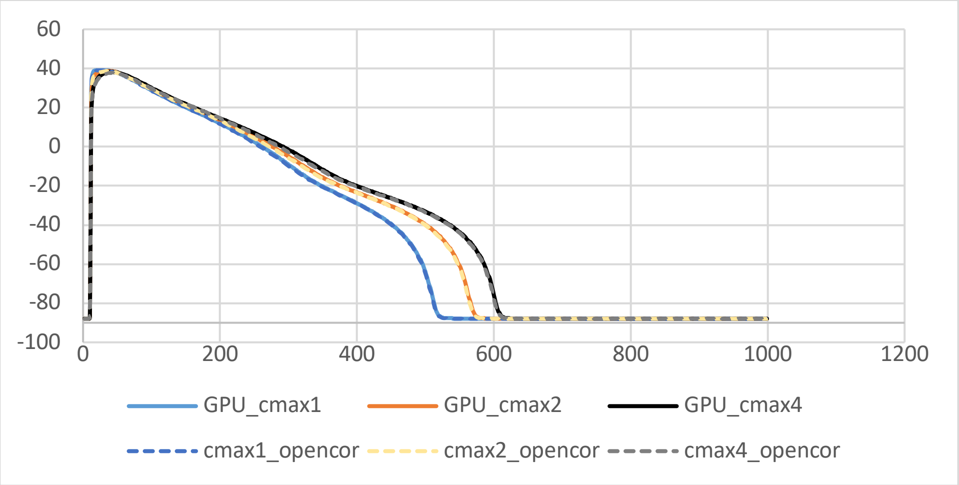


[Figure 3. 4] Action Potential (mV) Shape of both CPU (blue) and GPU (orange) Result Using ORd 2017

Key biomarker such as action potential duration, were analysed under identical no-drug conditions to ensure the physiological fidelity of the GPU-based results. Better compared to ORd 2011 model, the findings demonstrated that the GPU simulation similarly replicated the results from CPU with MAE value of 0.004 mV that peaks around 4 mV at 0.055 ms. The numerical outputs showed no significant differences, confirming the reliability and accuracy of the GPU implementation for the ORd 2017 model. Figure 3.4 provides a visual comparison of the action potentials produced by the GPU and CPU simulations, illustrating their near-identical behaviour.

3.2.2 Result Validation Under Drug

In this section, the accuracy of the GPU-based simulation for the ORd 2017 cell model was evaluated under drug conditions. The simulation incorporated drug-induced effects by modifying ionic current parameters based on IC50 and Hill coefficient values. These adjustments were applied uniformly across both the GPU and CPU (OpenCOR) simulations to ensure consistency in the drug response modelling. The validation process involved comparing action potential traces and key electrophysiological biomarker, such as action potential, between the GPU and CPU simulations.



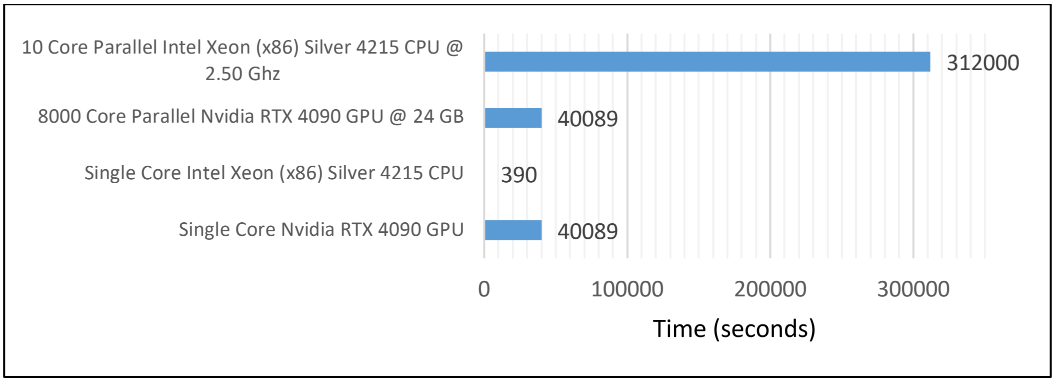
[Figure 3. 5] Action Potential Shape (mV) of both CPU (dashed) and GPU under drug effect Using ORd 2017

Despite the added complexity of drug effects, the GPU simulation produced outputs that were nearly identical to those of the CPU-based simulations, as shown in figure 3.5. Simulation from GPU only differs by 0.002 mV in average, with peak difference of 4.2 mV at 12 ms. This validation further establishes the robustness and reliability of the GPU-based simulation for scenarios involving drug effects.

3.2.3 Computational Time and Efficiency Analysis

This section examines the computational performance of the GPU-based simulation for the ORd 2017 cell model, compared against a 10-core Intel Xeon (x86) Silver 4215 CPU @ 2.50 GHz. The GPU simulations were executed using an NVIDIA RTX 4090 with 24 GB of memory. Both hardware setups processed 8000 samples (2000 samples per drug at four different concentrations). The computation time on CPUs scales linearly with the number of samples due to sequential processing limitations, even when multiple cores are utilised. However, the GPU’s parallel processing architecture allows it to maintain consistent computational times regardless of sample size, effectively minimising linear growth in execution time. While GPUs generally operate at lower clock speeds compared to CPUs, their ability to handle large-scale parallel tasks offers a significant advantage. The GPU's simulation time is nearly constant regardless of the number of samples due to its parallel processing capability.

For this simulation, the GPU takes approximately 40,089 seconds to simulate any number of samples, whether it's 1 or 8,000. On the other hand, the CPU system, using 10 CPUs in parallel, takes about 390 seconds to simulate one sample on each CPU. This means that the 10-core CPU system can complete 10 samples in 390 seconds. Therefore, to simulate all 8,000 samples, the CPU system would require a total of 312,000 seconds. Based on this information, it is calculated that for fewer than 1,028 samples, the CPU system with 10 cores is more efficient, as its total computation time (less than 40,089 seconds) would be faster than the GPU's fixed simulation time.



[Figure 3.6] Simulation time comparison between GPU and CPU in ORd 2017

For the ORd 2017 cell model, the GPU-based simulation achieved a speedup of up to 7.78 times compared to the 10-core CPU implementation. As shown in figure 3.6, GPU parallelisation is not as dominant compared to GPU parallelisation in ORd 2011. This lower speedup compared to the ORd 2011 cell model is attributed to the use of the Forward Euler method, which is computationally more demanding than the Rush-Larsen method. Nevertheless, the GPU remains significantly more efficient for large-scale simulations.

3.3 GPU Simulation Result Using ToR-ORd Cell Model

This section highlights the results obtained from GPU simulations using the ToR-ORd cell model. Similar to the ORd 2017 model, the forward Euler method was employed as the ODE solver. While this method provides adequate numerical stability and robustness for the ToR-ORD model, it results in longer computational times compared to the more efficient Rush-