Chapter 3. Results and Discussion

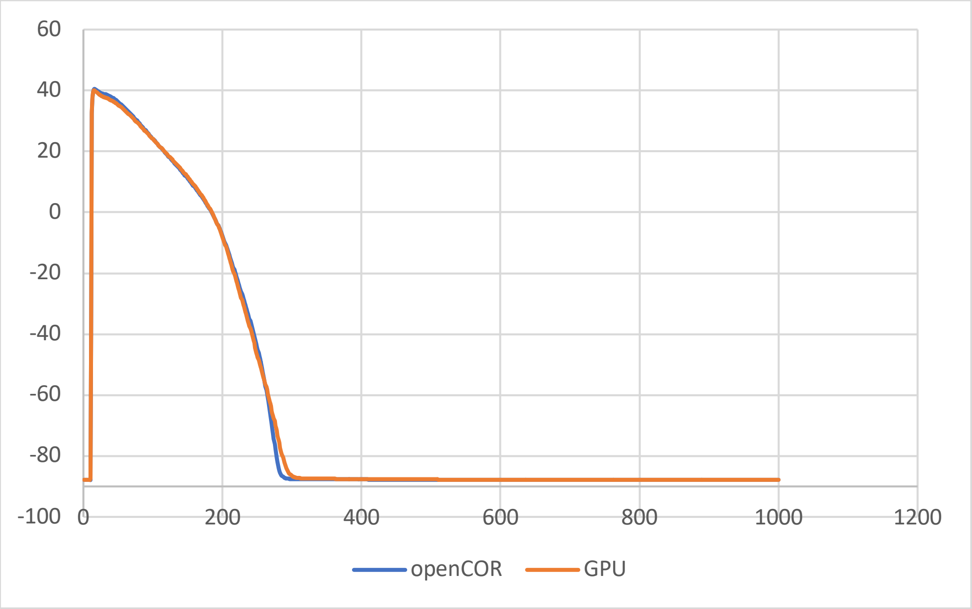
This chapter presents the findings of the GPU-based simulation for three different cell models: ORd 2011, ORd 2017, and ToR-ORd. Each section provides a detailed analysis of the results obtained from each cell model simulation result. In all simulations, the simulation accuracy is validated, the drug-induced changes are analysed, and computational performance is reviewed. There are two kind of simulation tested on these three cell models: Control, and drug-induced simulation. All control simulation results obtained in no-drug situation, and the drug bepridil was simulated to analyse drug-induced changes in the drug-induced simulation. This research uses bepridil with concentration of 33 mMol (cmax 1), 66 mMol (cmax 2), and 132 mMol (cmax 4). All result both with and without drug effect were run for 1000 pacing, and each pace lasts for 1000 milliseconds.

3.1 GPU Simulation Result Using ORd 2011 Cell Model

This section examines the results of GPU-based simulation using the ORd 2011 cell model. The ODE solver in ORd 2011 model was using the Rush-Larsen method which offers faster computational time by optimising the handling of gating variables in the equations. This approach not only accelerates simulations but also ensures sufficient numerical stability for this cell model.

3.1.1 Result Validation

To validate the results of the GPU simulation, this research compared the action potential outputs against reference solutions obtained from the OpenCOR, running under CPU. Visual comparisons of time-series plots for action potentials were performed to ensure qualitative agreement. Key electrophysiological biomarker, such as action potential duration were also compared. These biomarkers from CPU and GPU were compared under a same, no drug conditions, ensuring that the GPU-based simulation accurately reproduces the physiological dynamics from the ORd 2011 model. The findings revealed the GPU simulation result is almost exactly same with its CPU predecessor. Figure 3.1 shows visually action potential from both simulation platforms.

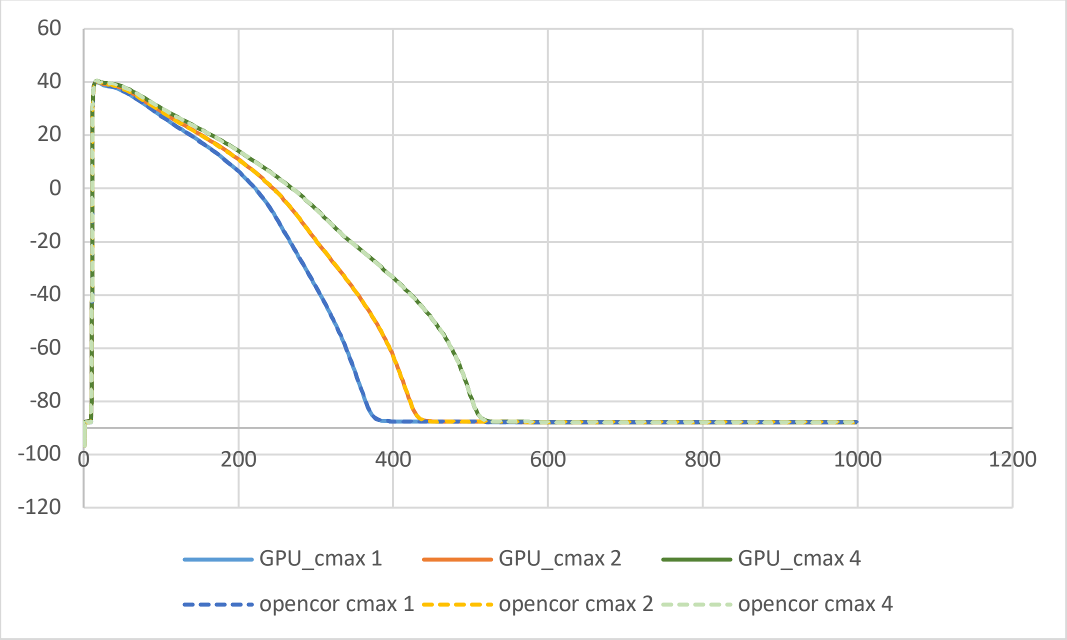


[Figure 3. 1] Action Potential (mV) Shape of both CPU (blue) and GPU (orange) Result Using ORd 2011

As shown, little to no difference from both of the result, indicating a valid result from the GPU-based simulation. Mean Average Error (MAE) measured 0.078 mV in average, and peaks around 9.1 mv at 281 ms. This difference caused by ODE solver in CPU, which not uses Rush-Larsen due to unavailability in OpenCOR.

3.1.2 Result Validation Under Drug

This section evaluates the accuracy of the GPU-based simulation for the ORd 2011 cell model under drug conditions. Drug effects were modelled by adjusting ionic current parameters using IC50 and Hill coefficient values, applied consistently in both GPU and CPU (OpenCOR) simulations. Validation similarly involved by comparing action potential traces.

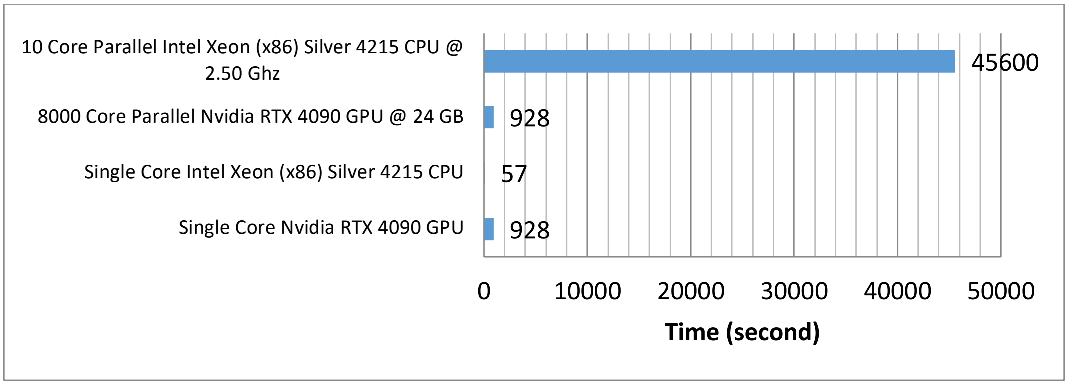


[Figure 3. 2] Action Potential Shape (mV) of both CPU (dashed) and GPU under drug effect Using ORd 2011

The results from figure 3.2 showed that despite the added complexity of drug effects, the GPU simulation closely matched the outputs of the CPU-based OpenCOR simulation. Also similar with drug-free simulation, mean average error measured at 0.071 in average. Highest MAE measured smaller at 8.95 mv at 1.405 ms. Differences are also caused by the different kind of ODE solver.

3.1.3 Computational Time and Efficiency Analysis

This analysis compares computational time between two hardware. GPU based simulation executed using NVIDIA RTX 4090 with 24 GB of memory. CPU based simulation also uses parallel processing of 10 cores Intel Xeon (x86) Silver 4215 CPU @ 2.50 GHz. The computational time compared for 8000 samples (each drug has 2000 samples, and 4 concentrations). In theory, GPU cores operate at lower clock speeds, making them inherently less powerful than CPU cores, which is why CPUs are typically preferred for single-sample simulations. The computation time for CPUs increases linearly with the sample size and pacing, meaning that as the number of samples grows, so does the time require for computation.



[Figure 3. 3] Simulation time comparison between GPU and CPU in ORd 2011

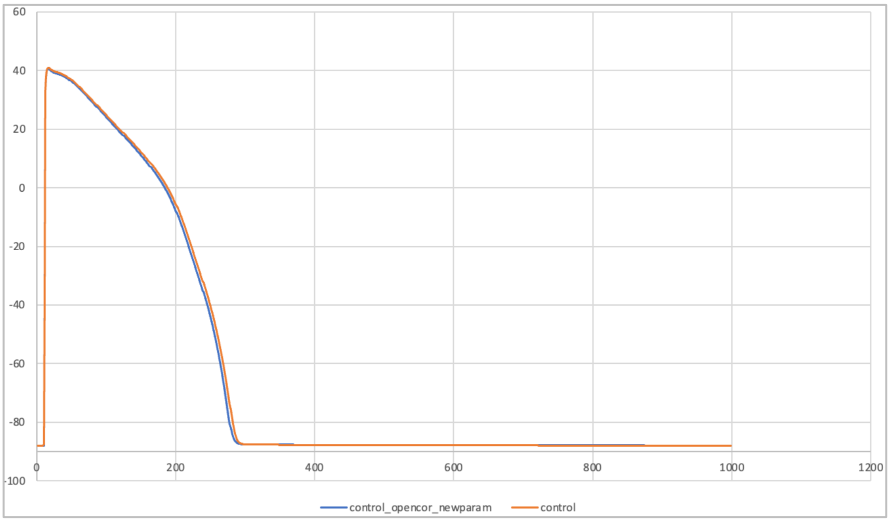
In contrast, GPU parallelisation eliminates this linear growth as seen in the figure 3.3. the time it takes to compute one sample is nearly the same regardless of how many samples are processed, due to its parallel computing architecture. GPU achieved a speedup of up to 40.91 times compared to a 10-core CPU. GPU requires about 928 seconds to complete the simulation, regardless of whether it handles a single sample or 8,000 samples. In contrast, the computation time for the CPU system with 10 cores grows by the number of samples being simulated. Simulating 8000 samples would take the CPU system around 45,600 seconds. From this analysis, the GPU becomes the more efficient option when simulating 163 samples or more, as its fixed runtime of 928 seconds is significantly faster than the increasing runtime of the CPU.

3.2 GPU Simulation Result Using ORd 2017 Cell Model

This section explores the outcomes of the GPU-based simulation using the ORd 2017 cell model. Unlike the ORd 2011 model, which leverages the efficiency of the Rush-Larsen method, the ORd 2017 model uses the forward Euler method for solving ordinary differential equations. While the forward Euler method is straightforward to implement and numerically stable for this model, it results in slower computational times compared to the Rush-Larsen method. This trade-off is necessary due to the instability observed when using the Rush-Larsen method with the ORd 2017 equations.

3.2.1 Result Validation

Similar to previous, validation of GPU simulation results for the ORd 2017 cell model was conducted by comparing outputs with the reference solutions generated from OpenCOR. Visual comparisons of action potential time-series data confirmed a close alignment between the two simulation platforms.

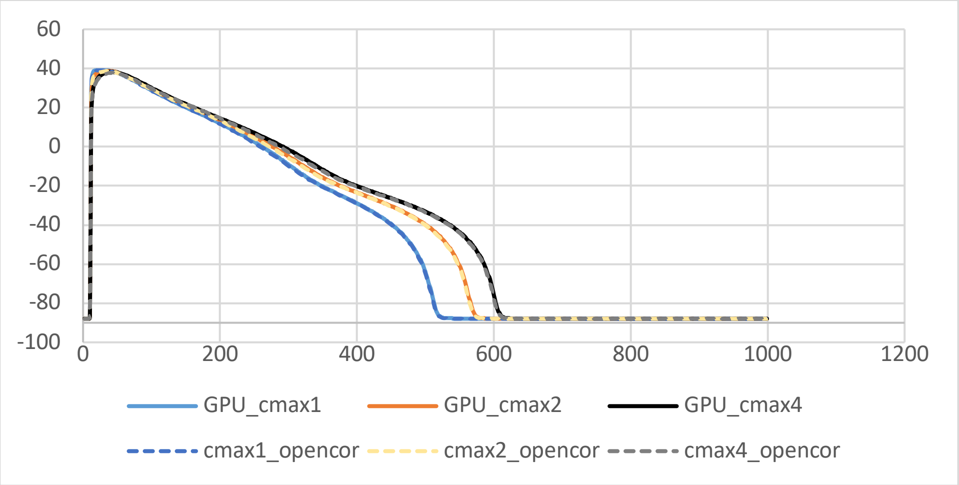


[Figure 3. 4] Action Potential (mV) Shape of both CPU (blue) and GPU (orange) Result Using ORd 2017

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.



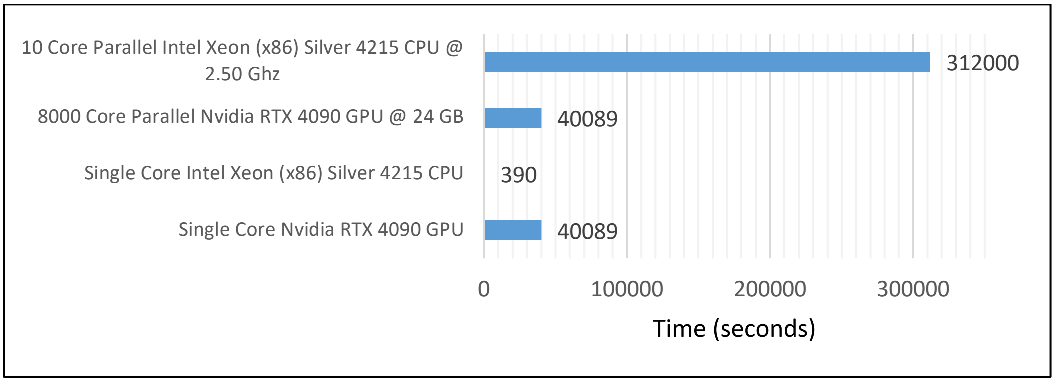
[Figure 3. 5] Action Potential Shape (mV) of both CPU (dashed) and GPU under drug effect Using ORd 2017

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.

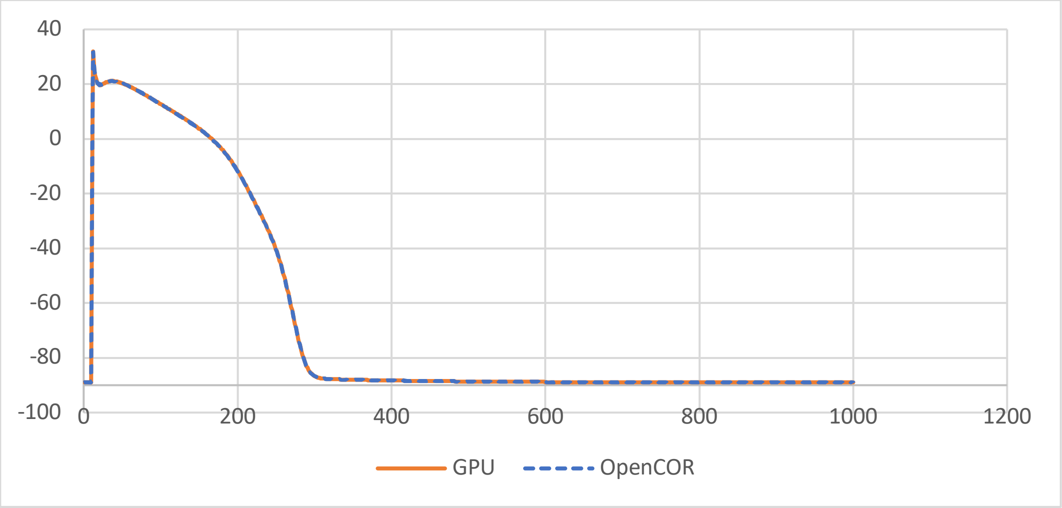
3.3 GPU Simulation Result Using ToR-ORd Cell Model

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-

Larsen method used in the ORd 2011 simulations. Regardless of the computational time drawback, forward Euler method ensure the simulation generates a reliable and usable result.

3.3.1 Result Validation

The validation process for the ToR-ORd cell model involved comparing GPU simulation outputs with the benchmark results obtained from OpenCOR. Time-series plots of action potentials were evaluated for qualitative consistency, and key biomarker, such as action potential duration, were quantitatively assessed under no-drug conditions.

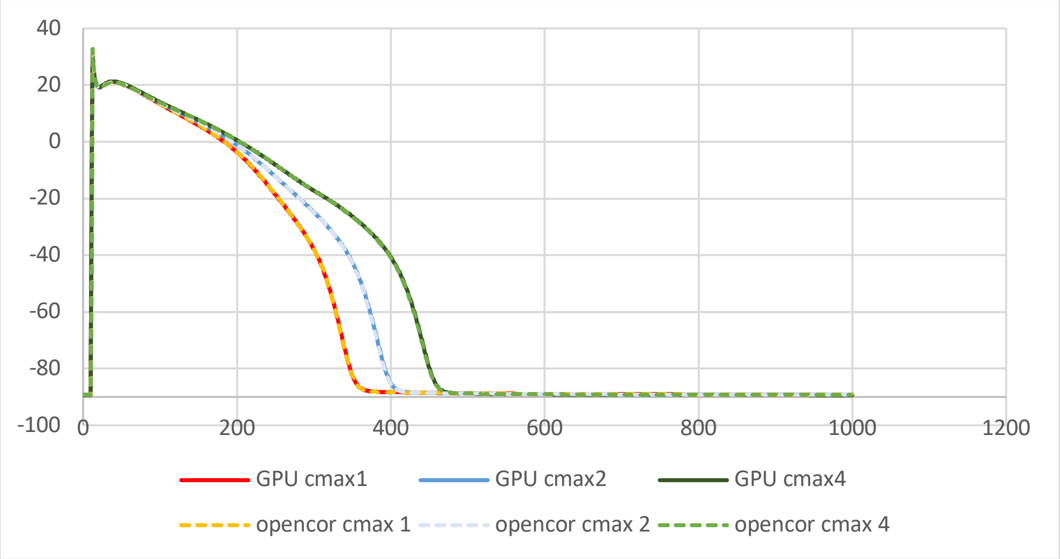


[Figure 3. 7] Action Potential (mV) Shape of both CPU (dashed blue) and GPU (orange) Result Using ToR-ORd

The results from figure 3.7 confirmed the accuracy of the GPU simulation, as it produced outputs that matched the CPU-based OpenCOR simulations without any discernible differences. MAE measured at 0.023 mV in average and with peak difference only 4.1 mV at 10 ms. Figure 3.6 illustrates the action potential traces from both GPU and CPU simulations, demonstrating their near-identical nature.

3.3.2 Result Validation Under Drug

For the validation in ToR-ORd cell model under drug, drug effects were incorporated into the simulation by altering ionic currents based on predefined IC50 and Hill coefficient parameters. Both GPU and CPU (OpenCOR) simulations were configured identically to ensure fair comparison.



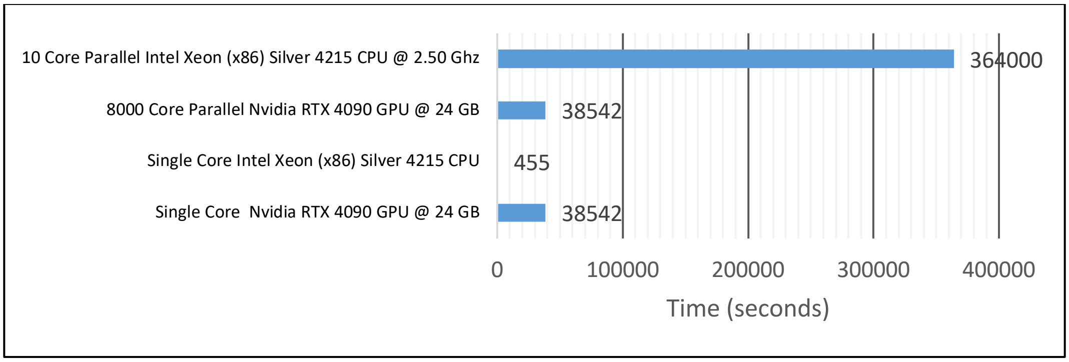
[Figure 3. 8] Action Potential (mV) Shape of both CPU (dashed) and GPU under drug effect Using ToR-ORd cell model

The results in figure 3. 8 showed that the GPU simulation accurately replicated the outputs of the CPU-based simulations, with no significant differences observed in action potential traces or in biomarkers such as APD, ionic current dynamics, and calcium handling. Average difference between CPU and GPU is 0.079 mV, with peak MAE of 43.9 mV at 10 ms.

This huge different spike is caused by ToR-ORd stimulation start point difference (or marked as stim_start variable in the simulation code). The simulation is still reliable even when subjected to drug-induced perturbations.

3.3.3 Computational Time and Efficiency Analysis

The ToR-ORd cell model was also analysed using the same hardware: an NVIDIA RTX 4090 GPU and a 10-core Intel Xeon (x86) Silver 4215 CPU @ 2.50 GHz. The GPU executed the simulations for 8000 samples (2000 samples per drug at four concentrations), matching the experimental conditions of the CPU. Similar to the ORd 2017 cell model, the computational time for the ToR-ORd model on CPUs increased linearly with the number of samples and pacing due to sequential processing. In contrast, the GPU's architecture showed consistent computational performance, demonstrating parallelisation for large-scale simulations.



[Figure 3. 9] Simulation time comparison between GPU and CPU in ToR-ORd cell model

As shown in figure 3.9, the ToR-ORd model GPU-based simulation achieved a speedup of up to 9.44 times compared to the 10-core CPU implementation. It is calculated that for simulations with fewer than 847 samples, the 10-core CPU system is faster, as its total computation time is less than the GPU's fixed time of 38,542 seconds. Beyond this point, the GPU becomes more efficient due to its ability to process a large number of samples simultaneously without increasing computation time. While not as high as the ORd 2011 model, this speedup highlights the GPU’s capability to handle complex simulations efficiently, even with computationally intensive solvers like Forward Euler. Overall, the GPU provides a substantial time-saving advantage across all tested cell models.

A single-core Intel Xeon processor with 2.50 GHz completed the simulation for one sample in 455 seconds. RTX 4090 GPU with only single core took way longer to finish the simulation. This is caused by a lower clock speed in each GPU’s computing core. This phenomenon emphasis the preferability for CPU simulation, which to simulate small number of sample.