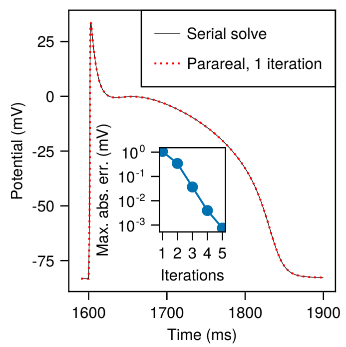
Speeding up Cardiac Simulations with Parallel-in-Time Solvers

Maxfield R Comstock*, Elizabeth M Cherry

Georgia Institute of Technology
Atlanta, GA, USA

Introduction: Cardiac models are an essential tool for understanding the causes of arrhythmias and other types of heart disease. Numerical simulations of these models require small time steps to make accurate predictions, leading to long simulation times even with programs parallelized over space. Parallelization over time presents a promising method to increase the speed of cardiac simulations and fully utilize modern parallel architectures.

Methods: This study evaluates the use of the parareal algorithm, a parallel-in-time integration method for partial differential equations, for simulations of cardiac cells and tissue. The parareal algorithm estimates the system state at fixed times, then iteratively refines these estimates by solving from the previous time estimate in parallel. Thus, fast convergence to an accurate solution is necessary for this method to be viable. Simulations of the Beeler-Reuter (BR) model were performed to evaluate the accuracy and speed of this method in comparison to sequential algorithms.



Comparison of parareal and serial solution. Inset: exponential convergence with number of parareal iterations.

Results: Simulations of the BR model were implemented for a single cell and for one-dimensional tissue using parareal and sequential algorithms. In the single-cell case, the parareal solution converged exponentially to the single-cell solution, as shown in the figure, with a maximum error of $1.1\,\mathrm{mV}$ after one iteration with a speedup factor of 1.9 on a 16-core processor. In the one-dimensional case, the main source of error was the reduced wave speed through the tissue by about $5\,\%$. The resulting action potential duration remains within $0.5\,\%$ of the sequential simulation and achieves a speedup factor of two on a 16-core processor after three iterations.

Conclusion: The parareal algorithm can effectively leverage parallel hardware to increase cardiac simulation speed. Further speedup may be possible through the massive parallelism of graphics cards and spatial parallelization.