

Speeding up Cardiac Digital Twinning using a GPU-based Eikonal Solver

Augusto De Filippo Cavalini¹, Rodrigo Weber dos Santos¹, Lucas Arantes Berg¹

¹ Departamento de Ciência da Computação (DCC)
Universidade Federal de Juiz de Fora (UFJF)
Juiz de Fora – MG – Brazil

augustocavalini.filippo@estudante.ufjf.br

rodrigo@dcc.ufjf.br, berg@ice.ufjf.br

Abstract. *Cardiac digital twins are patient-specific computational models that enable personalized diagnosis and treatment planning. A fundamental step in their construction is the simulation of electrical activation in the myocardium, commonly approximated using Eikonal-based propagation models. However, repeatedly solving the Eikonal equation can become a computational bottleneck during the personalization process. In this work we investigate the use of a GPU-based Fast Iterative Method (FIM) solver to accelerate the computation of cardiac activation maps in the digital twin pipeline. Preliminary results show a reduction in execution time, achieving a speedup of approximately $1.7\times$ compared to a CPU solver, while preserving activation patterns and ECG morphology when compared to clinical data.*

Resumo. *Gêmeos digitais cardíacos são modelos computacionais específicos de pacientes que permitem diagnóstico personalizado e planejamento de tratamentos. Uma etapa fundamental na construção desses modelos é a simulação da ativação elétrica no miocárdio, frequentemente aproximada por modelos baseados na equação de Eikonal. Entretanto, a resolução repetida dessa equação pode se tornar um gargalo computacional no processo de personalização. Neste trabalho investigamos o uso de um solucionador baseado no método Fast Iterative Method (FIM) executado em GPU para acelerar o cálculo de mapas de ativação cardíaca no pipeline de geração de gêmeos digitais. Resultados preliminares indicam redução no tempo de execução, alcançando um speedup de aproximadamente $1,7\times$ em relação a um solucionador de CPU, mantendo padrões de ativação e morfologia de ECG comparáveis aos observados em dados clínicos.*

1. Introduction

Cardiac digital twins are computational models designed to represent patient-specific cardiac behaviour and support diagnosis, treatment planning and in silico experimentation. A key step in these models is the simulation of electrical activation across the myocardium, commonly approximated using the Eikonal equation, which provides an efficient representation of wavefront propagation in cardiac electrophysiology [Keener 1991].

Although Eikonal-based models are less expensive than detailed reaction–diffusion formulations, their repeated solution during personalization and parameter optimization may still become a computational bottleneck. GPU-based algorithms, such as the Fast Iterative Method (FIM), are well suited for this task because they exploit parallel hardware [Fu et al. 2013].

In this work, we investigate a GPU-based FIM solver as a replacement for the conventional CPU-based Eikonal solver in a cardiac digital twin pipeline. Preliminary results compare execution time, activation map similarity and ECG reconstruction error.

2. Methods

2.1. Cardiac activation modeling

Electrical activation in cardiac tissue can be modeled at different levels of physiological detail. While biophysically detailed reaction–diffusion models provide accurate representations of electrophysiology, they are computationally expensive and often unsuitable for rapid patient-specific simulations.

As an efficient alternative, many cardiac digital twin pipelines approximate wavefront propagation using the Eikonal equation [Keener 1991]. This formulation models the arrival time of the depolarization wavefront across the myocardium while neglecting the full ionic dynamics. The activation time $\phi(\mathbf{x})$ at position \mathbf{x} satisfies the anisotropic Eikonal equation

$$\sqrt{\nabla\phi^T M \nabla\phi} = 1, \quad (1)$$

where M is a symmetric positive-definite tensor that encodes the anisotropic conduction properties of the cardiac tissue. In cardiac models this tensor reflects the preferential conduction direction along the myocardial fiber architecture.

Solving this equation over a cardiac mesh yields activation maps that describe the propagation of the electrical wavefront throughout the heart. These activation maps can then be used to compute simulated electrocardiograms (ECGs) and to guide the personalization of cardiac digital twins.

2.2. GPU-based Fast Iterative Method

To accelerate the computation of activation maps, we employ the Fast Iterative Method (FIM), a numerical algorithm designed for efficiently solving Eikonal equations on unstructured meshes [Fu et al. 2013]. The method is based on an iterative update scheme that maintains a list of active nodes representing the current propagation front.

During each iteration, activation times are locally updated using a causal upwind scheme until convergence is reached. Unlike methods such as the Fast Marching Method (FMM), which rely on strict ordering of updates, FIM allows asynchronous updates within a narrow band of active nodes. This property makes the algorithm highly suitable for parallelization on modern architectures.

In particular, the algorithm can be efficiently mapped to GPU hardware by updating multiple nodes simultaneously and exploiting the massive parallelism of Single Instruction, Multiple Data (SIMD) processors. Previous work has shown that this

approach can significantly accelerate the solution of Eikonal equations on tetrahedral meshes [Fu et al. 2013].

In this work we employ the *fimpy*¹ library, an open-source Python implementation of the Fast Iterative Method that provides GPU-enabled solvers for Eikonal equations. The library allows efficient computation of activation times on large meshes and can be integrated into existing cardiac simulation pipelines.

2.3. Simulation Pipeline

To evaluate the proposed approach within a realistic cardiac digital twin workflow, the simulations were performed using a pipeline adapted from the *Cardiac-Digital-Twin*² framework [Camps et al. 2024]. This repository provides tools for cardiac electrophysiology simulation and digital twin personalization, including an implementation of an Eikonal-based activation model using the DijkstraTet solver.

The cardiac mesh used in this preliminary evaluation was obtained from the data and modeling framework associated with the work of Camps et al. [Camps et al. 2024]. The experiments were conducted using one cardiac model configuration, including its corresponding mesh, fiber architecture and stimulation sites.

In the present study, this original Eikonal solver was used as the reference method. The FIM-based solver provided by *fimpy* was then integrated into the same pipeline in order to replace the original activation solver while keeping the remaining components of the simulation workflow unchanged. The same cardiac mesh, fiber architecture and stimulation sites were used for both solvers, allowing a direct comparison between the GPU-based FIM approach and the original CPU-based Eikonal implementation within the same cardiac modeling framework.

2.4. Hardware and Computational Setup

All simulations were performed on Linux Fedora using an NVIDIA GeForce RTX 4070 Ti SUPER GPU, a 12th Gen Intel(R) Core(TM) i5-12400F CPU and 16 GB of RAM. The software environment included Python 3.11.4, CuPy 14.0.1 through the `cupy-cuda12x` package, matplotlib 3.10.7, numba 0.62.1, pypm-pypi 0.5.0, SALib 1.4.7, pyDOE2 1.3.0, meshio 5.3.5 and fim-python[gpu] 1.2.2. The *fimpy* GPU kernels were executed with 128 parallel blocks, thread blocks of size (1024, 1, 1) and multiple CUDA streams.

2.5. Evaluation metrics

The evaluation considered three criteria: execution time, measured as the average runtime over 10 simulations; activation map similarity, computed as the RMSE between activation times across all mesh nodes; and ECG reconstruction error, computed as the RMSE between simulated and clinical ECG signals across all time samples and leads.

3. Preliminary Results

This section presents preliminary results evaluating the proposed GPU-based Fast Iterative Method (FIM) solver in comparison with the reference Eikonal solver. The evaluation focuses on two aspects: computational performance and similarity between activation maps.

¹<https://github.com/thomgrand/fim-python>

²<https://github.com/juliacamps/Cardiac-Digital-Twin>

3.1. Execution Time

To evaluate the computational performance of the proposed approach, both the GPU-based FIM solver and the reference Eikonal implementation based on the DijkstraTet algorithm were executed 10 times under the same simulation conditions in order to obtain stable statistics for runtime measurements. Table 1 summarizes the average execution time and standard deviation for both methods..

Table 1. Execution time comparison over 10 runs

Method	Mean time (s)	Standard deviation (s)	Speedup
FIM GPU	0.433075	0.009799	1.699944×
Eikonal DijkstraTet	0.736204	0.021028	1.00×

The results show that the GPU-based implementation outperforms the CPU-based DijkstraTet solver in terms of runtime, achieving an average speedup of approximately 1.7×. Although the observed acceleration is moderate, it demonstrates the potential benefits of exploiting GPU parallelism for solving Eikonal equations on cardiac meshes. The low standard deviation observed in both methods also indicates that the execution times are stable across multiple runs, suggesting that the GPU solver provides reliable performance under the tested conditions. From a practical perspective, even moderate speedups can have a significant impact on cardiac digital twin pipelines, where activation simulations may be executed many times during parameter optimization and personalization procedures.

3.2. Activation Map Comparison

Figure 1 presents a visual comparison between activation maps obtained using the GPU-based FIM solver and the reference Eikonal solver based on the DijkstraTet method. The color scale represents the activation time distribution across the cardiac mesh.

Both methods produce highly similar activation patterns, indicating that the GPU-based solver preserves the global propagation structure of the electrical wavefront. The spatial distribution of early and late activation regions is consistent between the two approaches, suggesting that the anisotropic conduction properties of the myocardial tissue are correctly captured by both solvers.

Small differences can nevertheless be observed in the local activation structure. Activation maps generated with the FIM-based solver appear slightly smoother near the root nodes, whereas the Dijkstra-based solution exhibits sharper activation gradients. As a result, nodes farther from the root nodes tend to be activated slightly earlier in the FIM-based solution.

This effect is also reflected in the maximum activation times observed in the simulations. The latest activation time obtained with the FIM-based solver was approximately 76 ms, whereas the DijkstraTet solver reached about 79 ms, explaining the slightly darker red region observed in the reference activation map.

To quantitatively evaluate the differences between both activation maps, the Root Mean Squared Error (RMSE) between activation times computed by the two solvers was calculated across all mesh nodes. The resulting RMSE was 2.708 ms, indicating close numerical agreement between the GPU-based FIM solution and the reference solver.

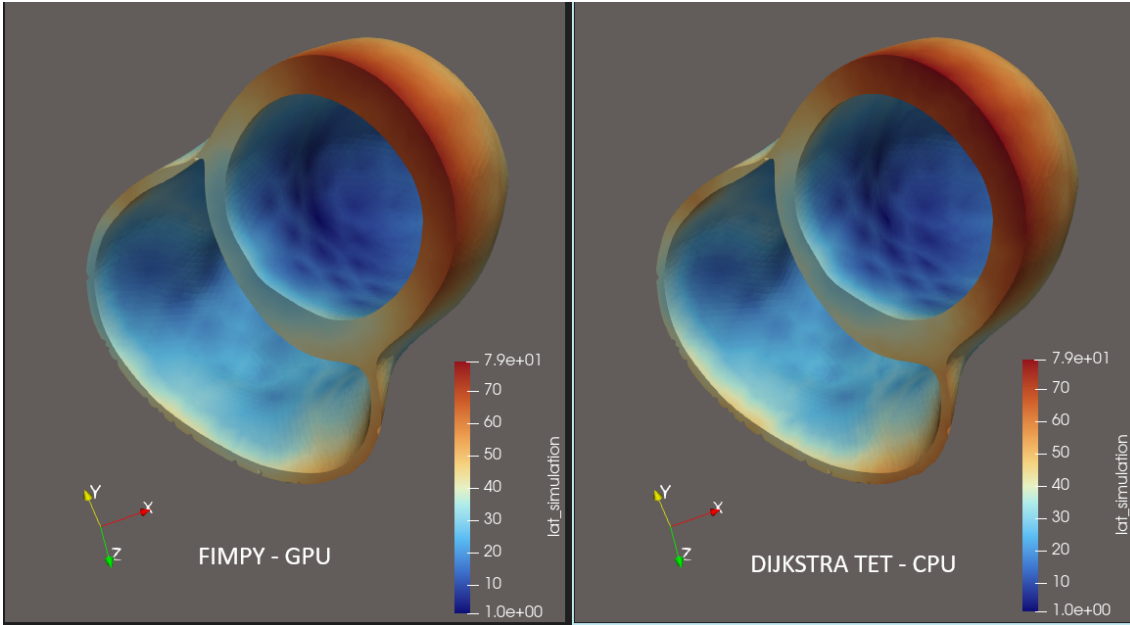


Figure 1. Comparison between activation maps generated using the FIMPY solver (Left) and the Eikonal DijkstraTet solver (Right).

In addition to the activation map comparison, the accuracy of the resulting simulated ECG signals was evaluated by computing the RMSE between simulated ECG signals and the clinical ECG across all time samples and leads. Table 2 summarizes the obtained results.

Table 2. ECG reconstruction error

Method	RMSE
FIM GPU	0.6836
Eikonal DijkstraTet	0.6202

Although the Eikonal solver achieved a slightly lower ECG error, the difference between both approaches remains relatively small. Together with the low activation map RMSE, these results indicate that replacing the classical Eikonal solver with the GPU-based FIM implementation does not significantly alter the electrophysiological behavior of the simulated cardiac activation.

4. Discussion

The preliminary results indicate that the proposed GPU-based Fast Iterative Method (FIM) solver can reproduce activation patterns and ECG signals that are largely consistent with those obtained using the reference Eikonal solver.

From the computational perspective, the GPU-based implementation demonstrated a measurable reduction in execution time. The experiments indicated an average speedup of approximately $1.7\times$ when compared to the reference CPU-based DijkstraTet solver. Although this speedup is moderate, it already represents a meaningful improvement for cardiac digital twin pipelines, where activation simulations are typically executed repeatedly during parameter optimization and model personalization.

The comparison between activation maps indicates that both methods produce highly similar spatial activation patterns across the cardiac mesh. The RMSE between activation times computed by the two solvers was approximately 2.7 ms, which is small relative to the overall activation duration of the ventricles (approximately 70–80 ms). This suggests that the GPU-based FIM solver provides a numerically consistent approximation of the reference Eikonal solution. Consistently, the reconstructed ECG signals also present similar quantitative agreement with the clinical ECG. Although the reference Eikonal solver achieved a slightly lower RMSE, the difference between both approaches remains relatively small, indicating that the GPU-based implementation preserves the electrophysiological characteristics of the simulated activation.

Such computational gains can contribute to reducing the total time required for generating patient-specific cardiac models and may help facilitate the integration of cardiac digital twin simulations into clinical research workflows.

5. Conclusion

This work investigated the use of a GPU-based Fast Iterative Method (FIM) solver to accelerate the generation of cardiac activation maps within cardiac digital twin pipelines. The proposed approach replaces a conventional CPU-based Eikonal solver with a parallel implementation designed for GPU architectures.

The preliminary experiments demonstrate that the GPU-based solver can reduce execution time while preserving the electrophysiological characteristics of the simulated activation. In particular, the results showed a speedup of approximately $1.7\times$ compared to the reference DijkstraTet solver, while maintaining close agreement in both activation maps and ECG reconstruction metrics.

Future work will include a more extensive evaluation of computational performance across larger cardiac meshes, the analysis of additional metrics for activation map similarity, and further validation of ECG reconstruction accuracy using larger patient datasets. Additionally, future investigations will explore the impact of different GPU configuration parameters, such as kernel launch configurations, thread-block dimensions and memory usage strategies within the *fimpy* implementation, in order to further optimize the performance of the FIM solver.

References

- Camps, J., Berg, L. A., Wang, Z. J., Sebastian, R., Riebel, L. L., Doste, R., Zhou, X., Sachetto, R., Coleman, J., Lawson, B., et al. (2024). Digital twinning of the human ventricular activation sequence to clinical 12-lead ecgs and magnetic resonance imaging using realistic purkinje networks for in silico clinical trials. *Medical Image Analysis*, 94:103108.
- Fu, Z., Kirby, R., and Whitaker, R. (2013). A fast iterative method for solving the eikonal equation on tetrahedral domains. *SIAM Journal on Scientific Computing*, 35:C473–C494.
- Keener, J. P. (1991). An eikonal-curvature equation for action potential propagation in myocardium. *Journal of mathematical biology*, 29:629–651.