



HAL
open science

An Improved Iterative Pace-Mapping Algorithm to Detect the Origin of Premature Ventricular Contractions

Andony Arrieula, Hubert Cochet, Pierre Jaïs, Michel Haïssaguerre, Mark Potse

► To cite this version:

Andony Arrieula, Hubert Cochet, Pierre Jaïs, Michel Haïssaguerre, Mark Potse. An Improved Iterative Pace-Mapping Algorithm to Detect the Origin of Premature Ventricular Contractions. CinC 2020 - Computing in Cardiology, Sep 2020, Rimini / Virtual, Italy. hal-02944731v4

HAL Id: hal-02944731

<https://hal.science/hal-02944731v4>

Submitted on 8 Oct 2020 (v4), last revised 1 Mar 2022 (v5)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Feasibility of Whole-Heart Electrophysiological Models With Near-Cellular Resolution

Mark Potse^{1,2,3}, Emmanuelle Saillard⁴, Denis Barthou^{4,5}, Yves Coudière^{1,2,3}

¹ Université de Bordeaux, IMB, UMR 5251, Talence, France

² CARMEN Research Team, Inria Bordeaux – Sud-Ouest, Talence, France

³ IHU Liryc, fondation Bordeaux Université, Pessac, France

⁴ STORM Research Team, Inria Bordeaux – Sud-Ouest, Talence, France

⁵ LaBRI, Bordeaux INP, Université de Bordeaux, Talence, France

Abstract

Given the opportunity to use a new cluster computer with over a quarter million compute cores we tested the strong and weak scaling of a monodomain reaction-diffusion model of the human ventricles with Ten Tusscher-Panfilov dynamics. Element sizes down to 25 μm and a model size up to 11 billion nodes were tested with both explicit and implicit Euler integration methods. Time steps were 0.01 ms for the implicit method and were resolution-dependent for the explicit method. We found that the weak scaling (increasing model size) was good for both methods. Depending on the model size, strong scaling (speedup at a larger number of cores) was satisfactory for the explicit method, and more limited for the implicit method. The implicit solver was generally slower; only at a resolution of 25 μm and on a relatively small number of cores it was as fast as the explicit solver. We conclude that whole-heart simulations at 25 μm resolution are technically feasible, although not practical yet on currently available systems.

1. Introduction

Reaction-diffusion models of the heart based on detailed representations of membrane electrophysiology are now commonplace. In 1984 Barr and Plonsey estimated that with the computers of the day it would take three thousand years to simulate a single heart beat with such a model [1]. Today, the simulation of just one million nodes which they had in mind can be performed in less than a minute, and it is common to run models of the whole human atria or ventricles for dozens of beats.

Yet, with resolutions of 200 to 500 μm these models can only represent structurally normal tissue, or gross structural abnormalities [2]. Small-scale structural diseases are thought to play an important role in cardiac arrhythmia. In the aging heart they contribute to atrial fibrillation [3], and

in a range of inherited and acquired cardiomyopathies they contribute to sudden cardiac death through ventricular fibrillation, sometimes even in young and apparently healthy subjects [4].

The patterns of tissue damage have different spatial scales, from several millimeters down to the size of a single cell [2, 5]. The more of these scales we can include, the more realistic our models become. Obviously this has an impact on the size and cost of simulations. Each doubling of the resolution implies an eight-fold increase in the number of unknowns and the memory requirements, and a roughly four-fold increase in the volume of inter-process communication of a simulation running on a parallel computer. Moreover, for explicit solvers the increase in spatial resolution imposes a reduction in time step, so that the expected number of operations would grow even faster.

To determine what is feasible on the computers that are available today and those that will arrive soon, we tested the performance of a model of the whole human ventricles with edge lengths of 200 μm down to 25 μm , i.e. model sizes in the order of 10^7 to 10^{10} nodes, on up to 262 144 compute cores.

2. Methods

The tests were performed on a new cluster of 2292 compute nodes, each equipped with two 64-core AMD Rome processors with a clock frequency of 2.6 GHz and 256 GB memory per node, and an InfiniBand HDR100 interconnect. Depending on the memory requirements of the simulations, we ran them on 8 up to 2048 nodes.

Each test consisted of integrating a monodomain reaction-diffusion equation [6] on a model of the human ventricles, for 10 ms simulated time. This was done either with an explicit or an implicit Euler method. Let V_m^n be the transmembrane potential at time step n and I_m the sum of the ionic and stimulation currents. Then the explicit Euler

model	Δx (μm)	Δt (μs) explicit	Δt (μs) implicit	nr. of nodes
M1	200	10	10	$2.16 \cdot 10^7$
M2	100	5	10	$1.68 \cdot 10^8$
M4	50	2	10	$1.33 \cdot 10^9$
M8	25	0.5	10	$1.05 \cdot 10^{10}$

Table 1. Models used for the tests.

method to integrate over a time step Δt reads

$$V_m^{n+1} = V_m^n + \frac{\Delta t}{C_m} (AV_m^n - I_m) \quad (1)$$

where A represents the differential operator $\beta^{-1} \nabla \cdot (G \nabla)$, G the monodomain conductivity tensor [7], C_m is the membrane capacitance, and β is the membrane surface to volume ratio of the tissue. In the implicit Euler method the term AV_m^n is replaced by AV_m^{n+1} . Integration with the implicit method requires a linear system

$$\left(1 - \frac{\Delta t}{C_m} A\right) V_m^{n+1} = V_m^n - \frac{\Delta t}{C_m} I_m \quad (2)$$

to be solved at each time step. We solved this system using a BiCGStab solver without preconditioner.

The implicit Euler method was integrated with a fixed time step of 0.01 ms. For the explicit Euler method we chose the largest time step in the sequence $\dots 1, 2, 5, 10 \mu\text{s}, \dots$ that did not cause instability.

As detailed in Tab. 1 the tests were performed with four models of the same geometry but with edge lengths ranging from 200 down to 25 μm . These models are referred to as M1, M2, M4, and M8. Models M2–M8 were created by scaling M1 appropriately.

Transmembrane ionic currents at each model node were computed with the Ten Tusscher–Panfilov model [8].

No significant output was written during these tests.

The simulations were performed with a recent version of the Propag-5 software [6]. In particular, the code was adapted to perform the initialization phase, in which the geometry is read and partitioned, with fewer processes than the simulation itself. This was necessary because the Parmetis graph partitioner could not be employed with more than a few thousand processes, due to an all-to-all communication that it performs. The smaller number of processes also accelerated the parallel data input.

The Propag-5 code has a hybrid MPI-OpenMP parallelization. Based on past experience we expected the optimal balance between the two to depend on architecture, compiler, MPI and OpenMP implementation, the problem size, and the number of compute nodes employed. Therefore tests were performed with different numbers of OpenMP threads per process. The results presented here refer to the optimal setting.

model	runtime (core-seconds)			
	explicit Euler		implicit Euler	
	per node per step	per node per step	per node per step	per node per step
M1	$5.9 \cdot 10^{-4}$	$5.9 \cdot 10^{-6}$	$3.6 \cdot 10^{-3}$	$3.6 \cdot 10^{-5}$
M2	$7.0 \cdot 10^{-4}$	$3.5 \cdot 10^{-6}$	$2.7 \cdot 10^{-3}$	$2.6 \cdot 10^{-5}$
M4	$1.9 \cdot 10^{-3}$	$3.8 \cdot 10^{-6}$	$3.9 \cdot 10^{-3}$	$3.9 \cdot 10^{-5}$
M8	$7.7 \cdot 10^{-3}$	$3.8 \cdot 10^{-6}$	$7.9 \cdot 10^{-3}$	$7.9 \cdot 10^{-5}$

Table 2. Runtime per model node, and per model node per step, for each of the four models. For each model, the data for the smallest possible number of system nodes were used.

3. Results

Figure 1, panel A, shows the influence of problem size on runtime and scaling with explicit integration. For the first three problem sizes a superlinear scaling phase could be seen, followed by a flattening or even an increase in runtime. For the M8 model this may occur at a larger number of nodes than we had at our disposal. The experimentally-determined Δt (Tab. 1) was 20 times smaller for M8 than for M1.

Figure 1, panel B, shows the scaling results for the implicit solver. For the smallest three models, it took more time on the same number of nodes than the explicit solver, and scaled less well. For M8, the two solvers had similar performance. To see this more clearly, the data for explicit and implicit solvers on M4 and M8 are overlaid in Fig. 2, panel A. For an M4 problem the explicit solver was much faster. For an M8 problem the two were closer, but the explicit solver still won at larger numbers of cores.

Table 2 lists the runtime per model node, and the runtime per model node per step. It shows that, as expected, the runtime per node increases with the model size and spatial resolution. This is also the case, but to lesser extent, for the implicit solver. For the explicit solver, the runtime per node per step varied little.

In Fig. 2, panel B, the performance of the AMD Rome nodes, on which the scaling tests were performed, is compared to two other architectures, Intel Knight’s Landing (KNL) and Skylake. The per-core performance of the AMD processors was close to that of the Skylake processors, and even exceeded it at large numbers of nodes.

4. Discussion

This study shows that simulations of cardiac electrophysiology in the whole human ventricles at a resolution of 25 μm , i.e. with over 10 billion nodes, are possible on existing hardware. The runtime of our explicit solver increased

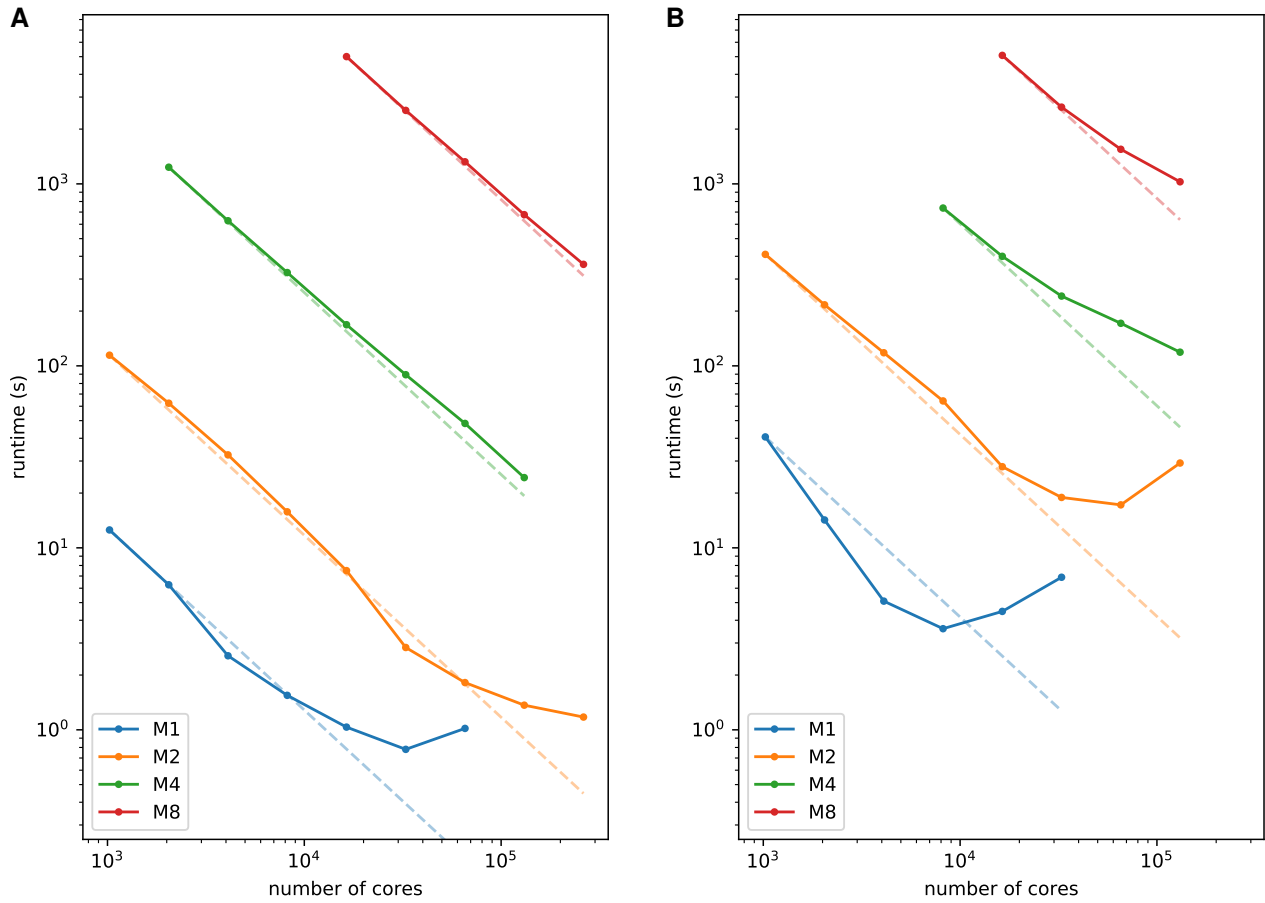


Figure 1. Comparison of the performance and scaling at different model sizes. The dashed lines show the ideal scaling in each case. **A:** using an explicit Euler solver and **B:** using an implicit solver.

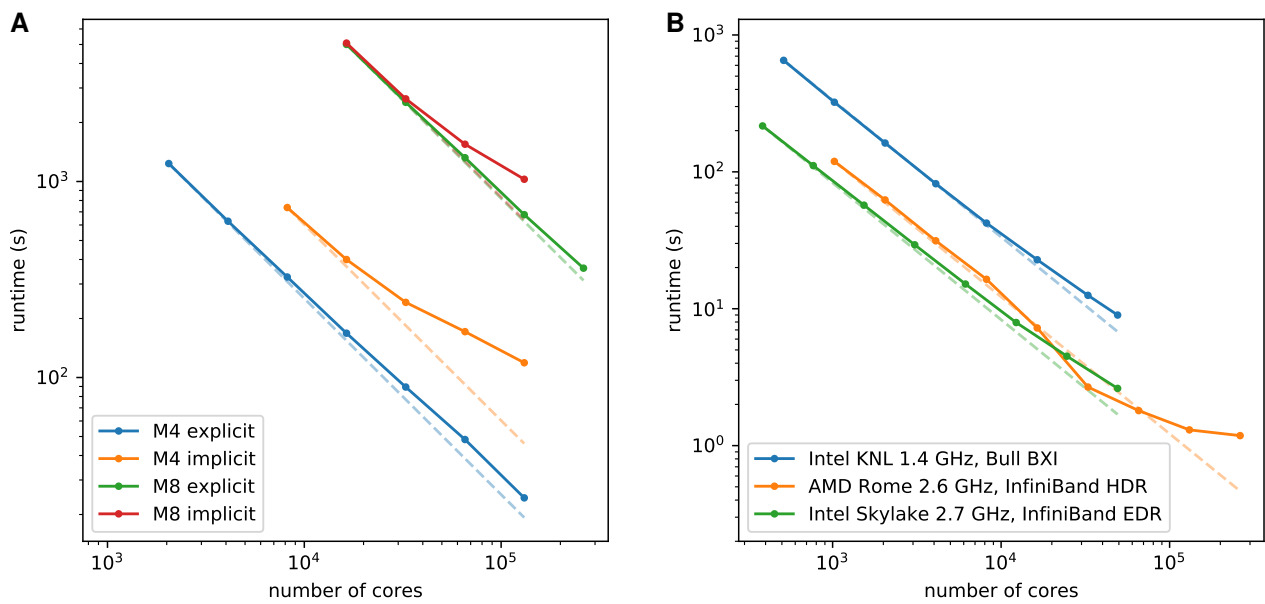


Figure 2. **A:** Comparison of implicit and explicit Euler solvers on models M4 and M8. **B:** The AMD Rome processors compared to two other architectures, using model M2.

as expected with the number of nodes and the increasing number of time steps needed at higher spatial resolution. The implicit solver, which used a time step of $10\ \mu\text{s}$ at any resolution, was much slower at low spatial resolution but closed in at $25\ \mu\text{m}$, so its weak scalability is better.

The strong scalability of the implicit solver was less than that of the explicit solver. This is expected because the implicit solver requires more inter-process communication. The amount of communication increases when the number of processes, and thus the number of domain boundaries, increases. We suspect that on a small number of compute nodes the performance of our code is limited by memory speed, whereas at a larger number of nodes it is limited by communication bandwidth and latency. The implicit solver suffers the most from this because it performs multiple matrix operations per time step. This problem would have been exacerbated if we had used a sophisticated preconditioner. A simpler solver than BiCGStab, which performs two matrix-vector multiplications per iteration, might have improved performance. Finally, our tests were done during the depolarization phase. During repolarization an implicit solver can often make much larger time steps, so it could beat an explicit solver on a full heart cycle.

The results show that our code scales well, in a weak sense to over 10^{10} nodes and in a strong sense to over a quarter million cores. However, for practical applications a number of problems will still have to be solved. First, we currently do not have software that can generate detailed meshes of this size. Secondly, we did not test the ability of the code to output significant amounts of results. The code currently does not have provisions to do this efficiently at such a large scale. Such provisions would include for example dedicated processes to perform output while other processes continue the computations.

Another challenge that will face users of future compute clusters is their expected heterogeneity. The machine that we tested still had a relatively traditional manycore CPU architecture. Other recent and future machines draw much of their compute power from GPUs and other specialized processors. Such machines require major changes in code design to be employed efficiently.

The performance differences between the Rome, Skylake and KNL clusters were probably caused by differences in CPU frequency, architecture and interconnect.

The practical use of bidomain models with an edge length of $25\ \mu\text{m}$, in the order of a myocyte diameter, is limited. For a realistic representation of the tissue at this scale a cell-by-cell model [9, 10] would be needed. This would imply another two or three orders of magnitude increase in model complexity. Much work is to be done before this will be feasible with whole-heart models.

Acknowledgements

This work was supported by the French National Research Agency, grant references ANR-18-CE46-0010-01 EXACARD, ANR-10-IAHU04-LIRYC, and ANR-11-EQPX-0030. This work was granted access to the HPC resources of TGCC under the allocation 2019gch0409 and allocation 2020-A0070307379 made by GENCI.

References

- [1] Barr RC, Plonsey R. Propagation of excitation in idealized anisotropic two-dimensional tissue. *Biophys J* 1984; 45:1191–1202.
- [2] Hoogendijk MG, et al. Mechanism of right precordial ST-segment elevation in structural heart disease: Excitation failure by current-to-load mismatch. *Heart Rhythm* 2010; 7:238–248.
- [3] Gharaviri A, et al. Epicardial fibrosis explains increased endo-epicardial dissociation and epicardial breakthroughs in human atrial fibrillation. *Front Physiol* 2020;11:68.
- [4] Haïssaguerre M, et al. Localized structural alterations underlying a subset of unexplained sudden cardiac death. *Circ Arrhythm Electrophysiol* 2018;11:e006120.
- [5] Spach MS, Boineau JP. Microfibrosis produces electrical load variations due to loss of side-to-side cell connections: A major mechanism of structural heart disease arrhythmias. *PACE* 1997;20:397–413.
- [6] Krause D, et al. Hybrid parallelization of a large-scale heart model. In Keller R, Kramer D, Weiss JP (eds.), *Facing the Multicore-Challenge II*, LNCS Vol. 7174. Berlin: Springer, 2012; 120–132.
- [7] Potse M. Scalable and accurate ECG simulation for reaction-diffusion models of the human heart. *Front Physiol* 2018;9:370.
- [8] Ten Tusscher KHWJ, Panfilov AV. Alternans and spiral breakup in a human ventricular tissue model. *Am J Physiol Heart Circ Physiol* 2006;291:H1088–H1100.
- [9] Tveito A, Jæger KH, Kuchta M, Mardal KA, Rognes ME. A cell-based framework for numerical modeling of electrical conduction in cardiac tissue. *Front Phys* 2017;5:48.
- [10] Bécue PE, Potse M, Coudière Y. A three-dimensional computational model of action potential propagation through a network of individual cells. In Pickett C, Corsi C, Laguna P, MacLeod R (eds.), *Computing in Cardiology*, volume 44. Rennes, France, September 2017; 271.

Address for correspondence:

Mark Potse
CARMEN research team
Inria Bordeaux – Sud-Ouest
200 Avenue de la vieille tour
33450 Talence, France