



HAL
open science

The MICROCARD project

Mark Potse

► **To cite this version:**

Mark Potse. The MICROCARD project. EuroHPC Summit 2024 - The European High Performance Computing, Mar 2024, Antwerp, Belgium. hal-04713117

HAL Id: hal-04713117

<https://inria.hal.science/hal-04713117v1>

Submitted on 30 Sep 2024

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.



Distributed under a Creative Commons Attribution - NoDerivatives 4.0 International License

The MICROCARD project

This poster was authored for the MICROCARD consortium by Mark Potse (IHU Liryc, Bordeaux, France; CARMEN team, Inria, France; and Institut de mathématiques de Bordeaux, UMR5251, Talence, France) for presentation at the 2024 EuroHPC Summit meeting which took place from 18 to 21 March 2024 in Antwerp, Belgium.

This project has received funding from the European High-Performance Computing Joint Undertaking EuroHPC (JU) under grant agreement No 955495. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Italy, Germany, Austria, Norway, and Switzerland. The project was co-funded by the French National Research Agency ANR, the German Federal Ministry of Education and Research, the Italian ministry of economic development, the Swiss State Secretariat for Education, Research and Innovation, the Austrian Research Promotion Agency FFG, and the Research Council of Norway.

The MICROCARD project

Modeling the heart cell by cell on exascale supercomputers

The contraction of the heart is coordinated electrically thanks to conducting links between the cells. Structural damage, due to aging or disease, can break these links and cause dangerous arrhythmia.

To better understand these arrhythmia we need models in which the individual cells and their connections are resolved. For a whole heart, with billions of cells, this means trillions of model elements with many degrees of freedom each.

The MICROCARD project aims to run such models on exascale computers. It covers every aspect of this challenge from code transformations to numerical schemes, mesh creation, and real-life use cases.

✉ mark.potse@inria.fr

🌐 microcard.eu

📘 @Project MICROCARD

🐦 @P_Microcard

Cardiac electrical activity can be modeled with a large **reaction-diffusion** system. Traditionally this is done with a homogenization approach so that model elements average the behaviour of hundreds of cells. **The goal of MICROCARD is to discretize individual cells and their interconnections.**

The **reaction term** represents the currents generated by ion channels and pumps everywhere in the cell membrane. It is governed by a system of ordinary differential equations with strongly nonlinear coefficients.

$$\begin{aligned}
 J_{Ca} &= G_{Ca} m^2 h_j (V_m - E_{Ca}) \\
 \partial_t m &= (m_{\infty} - m) / \tau_m \\
 \partial_t h &= (h_{\infty} - h) / \tau_h \\
 \partial_t j &= (j_{\infty} - j) / \tau_j \\
 m_{\infty} &= [1 + \exp(-56.86 - V_m) / 9.03]^{-1} \\
 \tau_m &= \alpha_m V_m \\
 \alpha_m &= [1 + \exp(-60 - V_m) / 5]^{-1} \\
 h_{\infty} &= 0.1 / [1 + \exp(V_m + 35 / 5)] \\
 j_{\infty} &= 0.1 / [1 + \exp(V_m - 50 / 200)] \\
 \tau_h &= (\alpha_h h_j)^{-1} \\
 \alpha_h &= \begin{cases} 0 & \text{if } V \geq -40\text{mV} \\ 0.057 \exp(-V_m + 80) / 6.8 & \text{otherwise} \end{cases} \\
 &\text{(10 more lines like that)}
 \end{aligned}$$

The **diffusion term** is a large system of linear equations ($N \sim 10^{12}$)

In our cellular-scale model, an additional difficulty is that the reaction term is an internal boundary condition, as it is present in the membrane of every cell.

MLIR

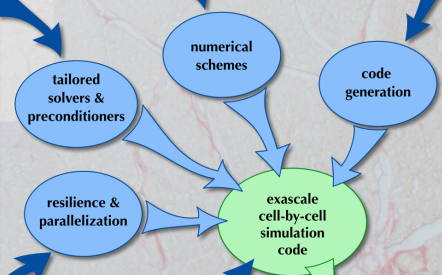
Using models expressed in a domain-specific language and compiled using the multilevel intermediate representations (MLIR) framework, we can now compute membrane currents about 2 times faster on vector CPUs and many times faster on GPUs. The speedup is larger for more compute-intensive models.

Model	Nvidia A100 (GFLOPS)	AVX-512 (GFLOPS)	baseline (GFLOPS)
Abraham	10	10	10
Abraham2	15	15	15
Abraham3	20	20	20
Abraham4	25	25	25
Abraham5	30	30	30
Abraham6	35	35	35
Abraham7	40	40	40
Abraham8	45	45	45
Abraham9	50	50	50
Abraham10	55	55	55
Abraham11	60	60	60
Abraham12	65	65	65
Abraham13	70	70	70
Abraham14	75	75	75
Abraham15	80	80	80
Abraham16	85	85	85
Abraham17	90	90	90
Abraham18	95	95	95
Abraham19	100	100	100
Abraham20	105	105	105
Abraham21	110	110	110
Abraham22	115	115	115
Abraham23	120	120	120
Abraham24	125	125	125
Abraham25	130	130	130
Abraham26	135	135	135
Abraham27	140	140	140
Abraham28	145	145	145
Abraham29	150	150	150
Abraham30	155	155	155
Abraham31	160	160	160
Abraham32	165	165	165
Abraham33	170	170	170
Abraham34	175	175	175
Abraham35	180	180	180
Abraham36	185	185	185
Abraham37	190	190	190
Abraham38	195	195	195
Abraham39	200	200	200
Abraham40	205	205	205
Abraham41	210	210	210
Abraham42	215	215	215
Abraham43	220	220	220
Abraham44	225	225	225
Abraham45	230	230	230
Abraham46	235	235	235
Abraham47	240	240	240
Abraham48	245	245	245
Abraham49	250	250	250
Abraham50	255	255	255
Abraham51	260	260	260
Abraham52	265	265	265
Abraham53	270	270	270
Abraham54	275	275	275
Abraham55	280	280	280
Abraham56	285	285	285
Abraham57	290	290	290
Abraham58	295	295	295
Abraham59	300	300	300
Abraham60	305	305	305
Abraham61	310	310	310
Abraham62	315	315	315
Abraham63	320	320	320
Abraham64	325	325	325
Abraham65	330	330	330
Abraham66	335	335	335
Abraham67	340	340	340
Abraham68	345	345	345
Abraham69	350	350	350
Abraham70	355	355	355
Abraham71	360	360	360
Abraham72	365	365	365
Abraham73	370	370	370
Abraham74	375	375	375
Abraham75	380	380	380
Abraham76	385	385	385
Abraham77	390	390	390
Abraham78	395	395	395
Abraham79	400	400	400
Abraham80	405	405	405
Abraham81	410	410	410
Abraham82	415	415	415
Abraham83	420	420	420
Abraham84	425	425	425
Abraham85	430	430	430
Abraham86	435	435	435
Abraham87	440	440	440
Abraham88	445	445	445
Abraham89	450	450	450
Abraham90	455	455	455
Abraham91	460	460	460
Abraham92	465	465	465
Abraham93	470	470	470
Abraham94	475	475	475
Abraham95	480	480	480
Abraham96	485	485	485
Abraham97	490	490	490
Abraham98	495	495	495
Abraham99	500	500	500
Abraham100	505	505	505

Ginkgo

The Ginkgo parallel linear algebra package now has the solvers and preconditioners we need, with support for multi-GPU nodes. This graph shows how a Ginkgo BICGStab solver scales on the Frontier supercomputer (AMD MI250X accelerators).

Ω_p (pericardium), Ω_i^k (intercalated disc), Ω_j^l (cell membrane), Γ_m^k (cell membrane)

$$\begin{cases}
 \nabla \cdot (\sigma^k \nabla \phi^k) = I_i & \text{on } \Omega_i^k \\
 \nabla \cdot (\sigma^j \nabla \phi^j) = 0 & \text{on } \Omega_j^l \\
 \phi^k = \phi^j = \phi_p & \text{on } \Gamma_m^k \\
 -\sigma^k \nabla \phi^k \cdot \mathbf{n} = G_m \nabla \phi^j \cdot \mathbf{n} = C_m \partial_t V_m^k + J_{ion}(V_m^k, \mathbf{y}) + I_i & \text{on } \Gamma_m^k \text{ (transmem. currents)} \\
 -\sigma^j \nabla \phi^j \cdot \mathbf{n} = -\sigma^k \nabla \phi^k \cdot \mathbf{n} = -C_m \partial_t V_m^k + \sigma(\phi^k - \phi^j) & \text{on } \Gamma_m^k \text{ (intercalated discs)} \\
 \sigma^k \nabla \phi^k \cdot \mathbf{n} = J_e & \text{on } \Gamma_e^k \text{ (electrode surface)} \\
 \partial_t \mathbf{y} = \mathbf{F}(V_m, \mathbf{y}) & \text{on } \Omega_j^l
 \end{cases}$$


StarPU

The StarPU runtime scheduler is used to balance load between CPU and GPU.

openCARP

The code is based on the widely used openCARP software for cardiac simulation. We enhanced openCARP with a Ginkgo back-end, asynchronous parallel output, and many other upgrades.

MEGWARE

HPC manufacturer MEGWARE provides a development platform that allows fine-grained measurement of energy consumption.

Mmg PLATFORM

We improved the Mmg remeshing software so that it can handle our complex multi-domain meshes. This allowed us to make virtual cardiac tissue samples upto 1 mm³ volume (7843 cells, 352 million tetrahedra, 115 thousand internal boundaries). We are working to achieve this also with the parallel version of Mmg, and then to scale up further.