Cardiovascular Simulations

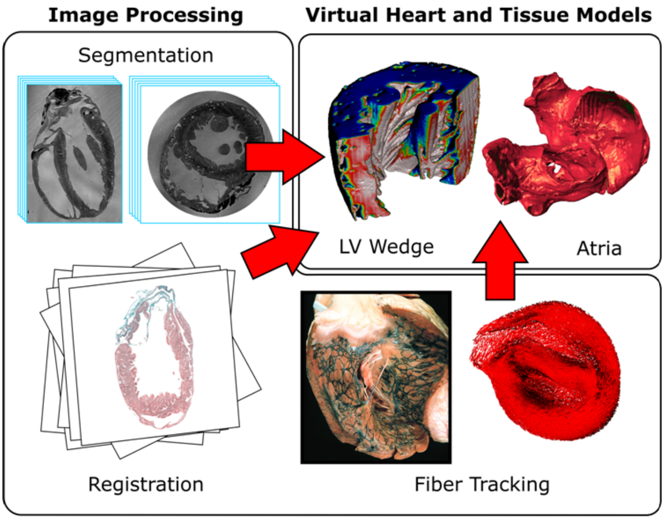
Gundolf Haase

Karl-Franzens University Graz

Graz, Nov 2024

Video

Virtual heart [SFB: G. Plank]

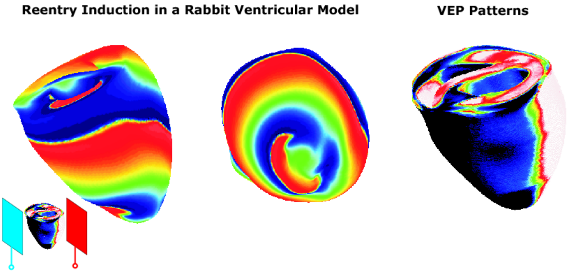


Virtual heart - goals in CARP project [G.Plank]

Gernot Plank [Graz ⟶ John Hopkins (Baltimore) ⟶ Oxford ⟶ Graz]
Ed Vigmond [Calgary ⟶ Bordeaux]

- Simulation of electrical stimulation of virtual heart.
- Including mechanics of the heart.
- Shape the electrical pulse wrt. certain criteria.
- Goal: Minimize the overall energy of the pulse for curing arrhythmia.
- Benefits:
  - Less internal burnings when internal pace makers send electrical pulses against arrhythmia.
  - Surgery planning for scared over tissue after a heart attack.

Rabbit Heart [G. Plank: CARP]



- time-dependent electrical potential, anisotropic coefficients + Purkinje fibers in bidomain equations:
  non–linear system of PDEs and ODEs
- 5.082.272 tetrahedrons with 862.525 FEM-nodes
- Goal: 150 Mill. tetrahedrons using parallel mesh generator Spider by F. Kickinger results in 25 Mill. d.o.f.

Rabbit Heart: bidomain equations

Two components of the electric propagation in the heart:

- ionic flux through the cell membrane: ODEs (3) accounts for over 20 variables,
- electrical model for the tissues: linear PDEs (1)–(2)

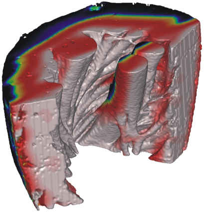
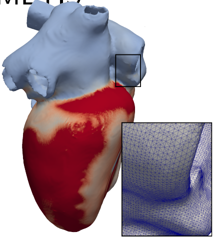
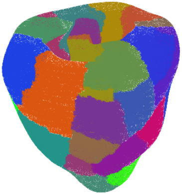
$$\begin{aligned}
\nabla \cdot (\bar{\sigma_i} + \bar{\sigma_e}) \nabla V_e &= -\nabla \cdot \bar{\sigma}_i \nabla V_m & (1)\\
\nabla \cdot \bar{\sigma}_i \nabla V_m &= -\nabla \cdot \bar{\sigma}_i \nabla V_e + \beta I_m & (2)\\
I_m &= C_m \frac{\partial V_m}{\partial t} + I_{ion}(V_m, v) & (3)
\end{aligned}$$

- $\bar{\sigma}_i$, $\bar{\sigma_e}$: intracellular and extracellular conductivity tensors,
- β: surface to volume ratio of the cardiac cells,
- C_m: membrane capacitance per unit area,
- V_m transmembrane voltage,
- I_{ion}: ionic current density flowing through the membrane ionic channel and depends on the transmembrane voltage

Coupled system (1)–(3) is non–linear

Rabbit Heart: Discretization

- Mesh generation by spider/tarantula [F. Kickinger]
- Finite element mesh with tetrahedral and hexahedral elements
- Mesh decomposition by METIS

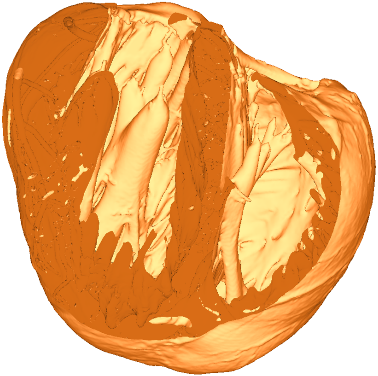
- Solve linear system of equations (**That is our part!**)

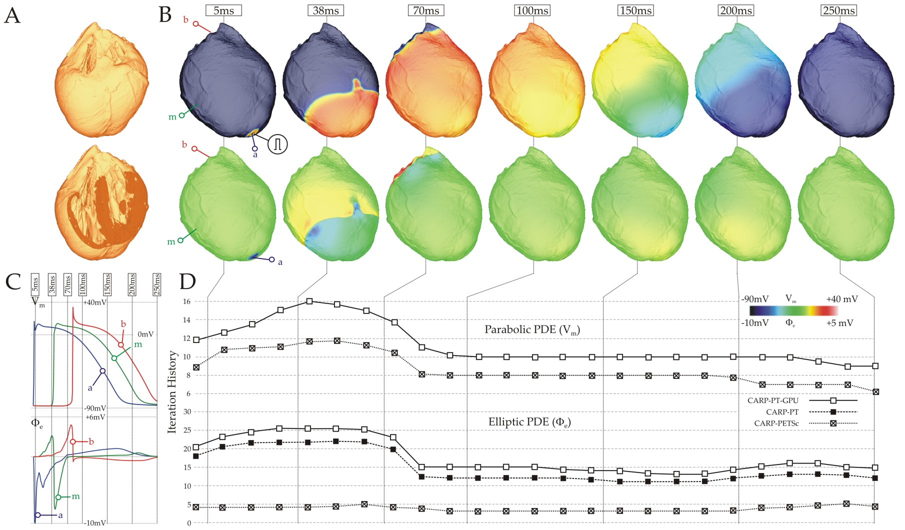
$$Ku = f$$

  in each outer (time/non–linear) iteration.
- System matrix K is sparse but unstructured.

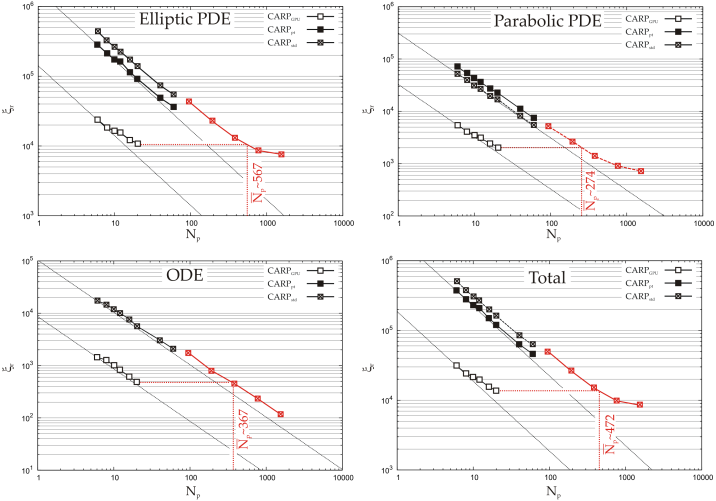
Oxford Benchmark

40.992.163 Elements; 6.901.583 d.o.f; real time duration: 250 ms




A
B
5ms
38ms
70ms
100ms
150ms
200ms
250ms
b
m
a
C
D
V_m
+40mV
0mV
-90mV
+6mV
Φ_e
-10mV
Iteration History
Parabolic PDE (V_m)
Elliptic PDE (Φ_e)
-90mV
+40 mV
-10mV
+5 mV
CARP-PT-GPU
CARP-PT
CARP-PETSc

toolbox (CPU ■/GPU □) vs. Petsc (hypre) ☒ on mephisto and hector [II/2012]



Strong scaling results on mephisto in sec. [II/2012]

6.9 Mill. DoFs; potential problem

Mephisto cluster: 5×2 Xeon X5650 CPUs, 60 cores at 2.67GHz;
5×4 NVidia Tesla 2070

N	CARP-PETSc	CARP-PT-GPU	Eff. PETSc	Eff. PT	Speedup
6	126,897	8,266	1	1	15.3
8	95,057	6,519	1	0.95	14.6
10	76,060	5,667	1	0.87	13.4
12	66,134	5,586	0.96	0.73	11.8
16	49,842	4,479	0.95	0.69	11.1
20	40,523	3,807	0.93	0.65	10.6

A first summary

- GPU/CPU-MG scales very well for small/medium size parallel computers
- Usage of GPU pays off, even for unstructured data
- Redesign of communication also for hierarchical communication
- extension to elasticity solver?
- non-linear problems?
- Use of `#pragma`-driven parallelization for general manycore environments?.