Using Linux Clusters for Full-Scale Simulation of Cardiac Electrophysiology

Xing Cai   *Glenn Terje Lines*

[simula . research laboratory]

&

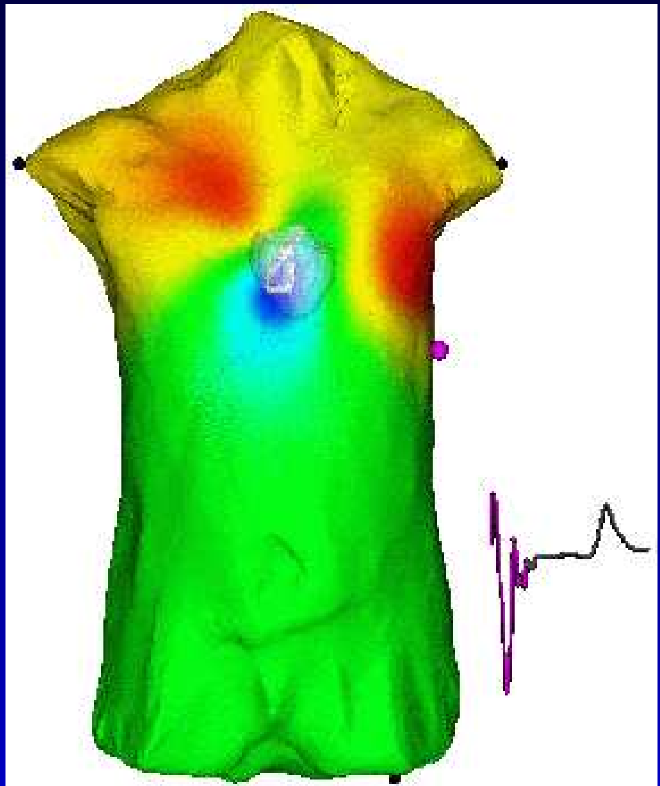
University of Oslo, Norway

LCSC'04, Oct. 19, 2004

Outline

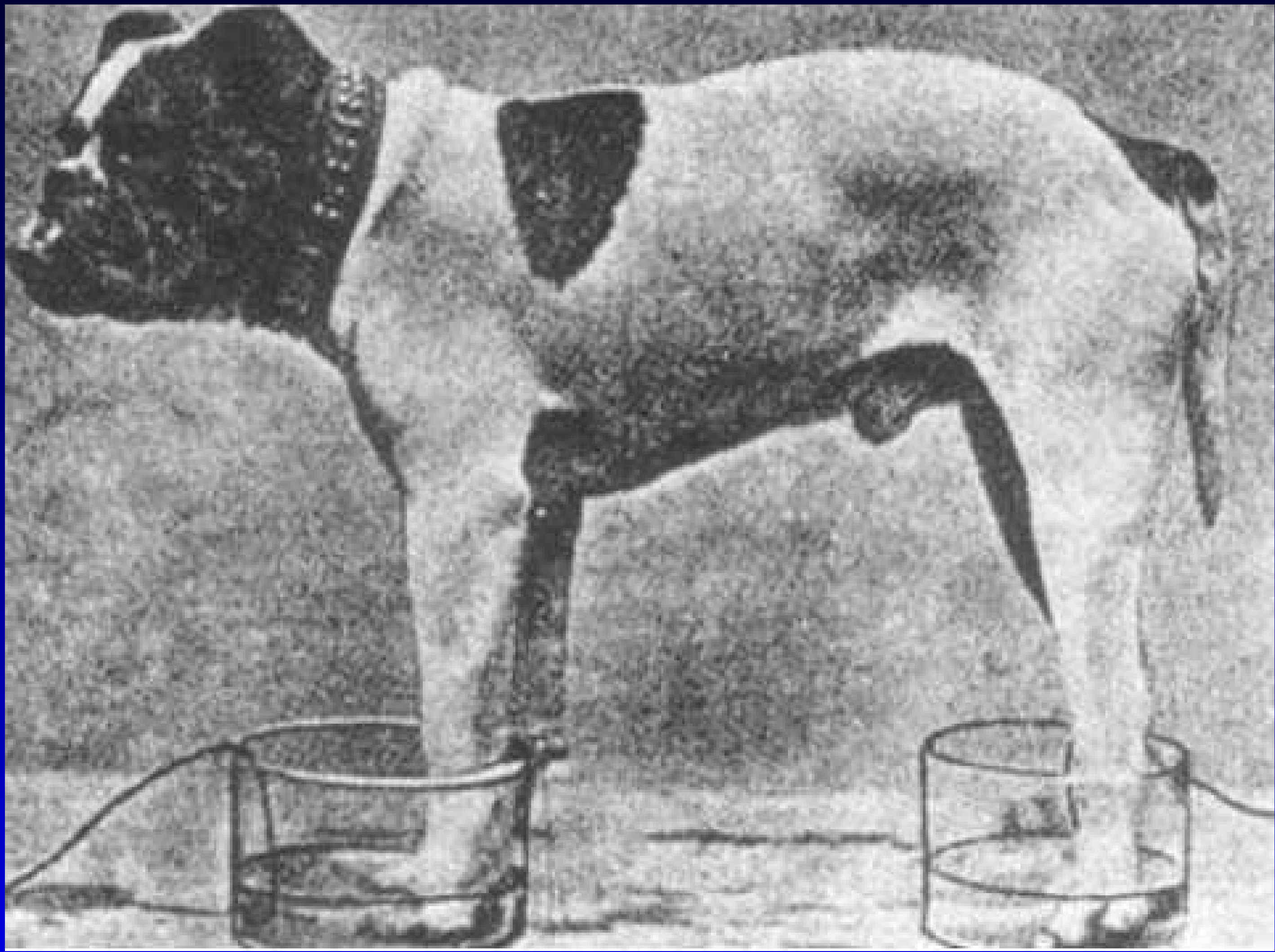
- Cardiac physiology
- Mathematical model
- Computational challenges
- Numerical strategy and parallelization
- Results and future work

Physiology



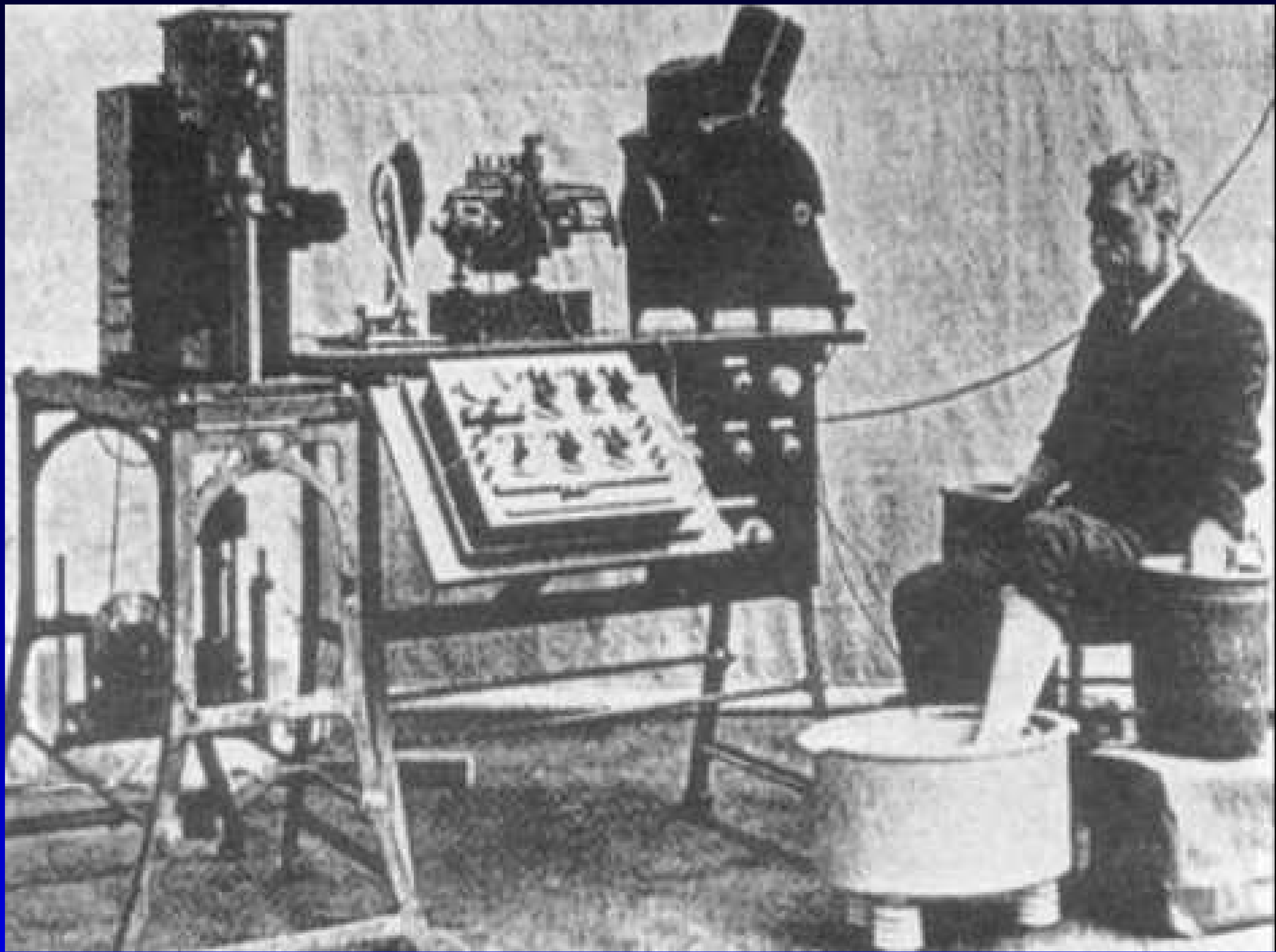
- Ions move in and out of heart cells
- Ionic current $\Rightarrow$ heart muscle contraction $\Rightarrow$ pumping heart
- Electrical activity inside heart is measurable by electrocardiogram (ECG)
- Numerical simulation - diagnostic tool in future(?)

Electrocardiogram (ECG)



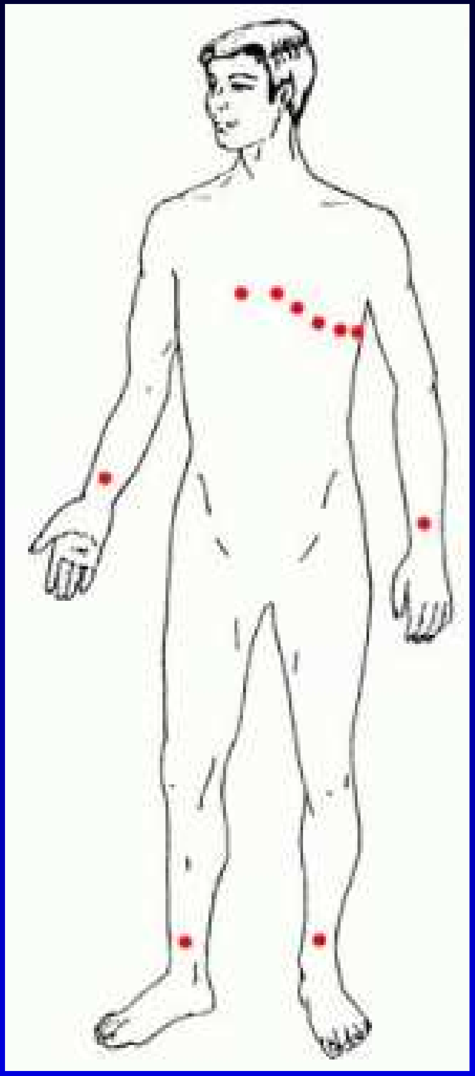
The first ECG was recorded on a dog in 1887 in London.

Electrocardiogram (ECG)



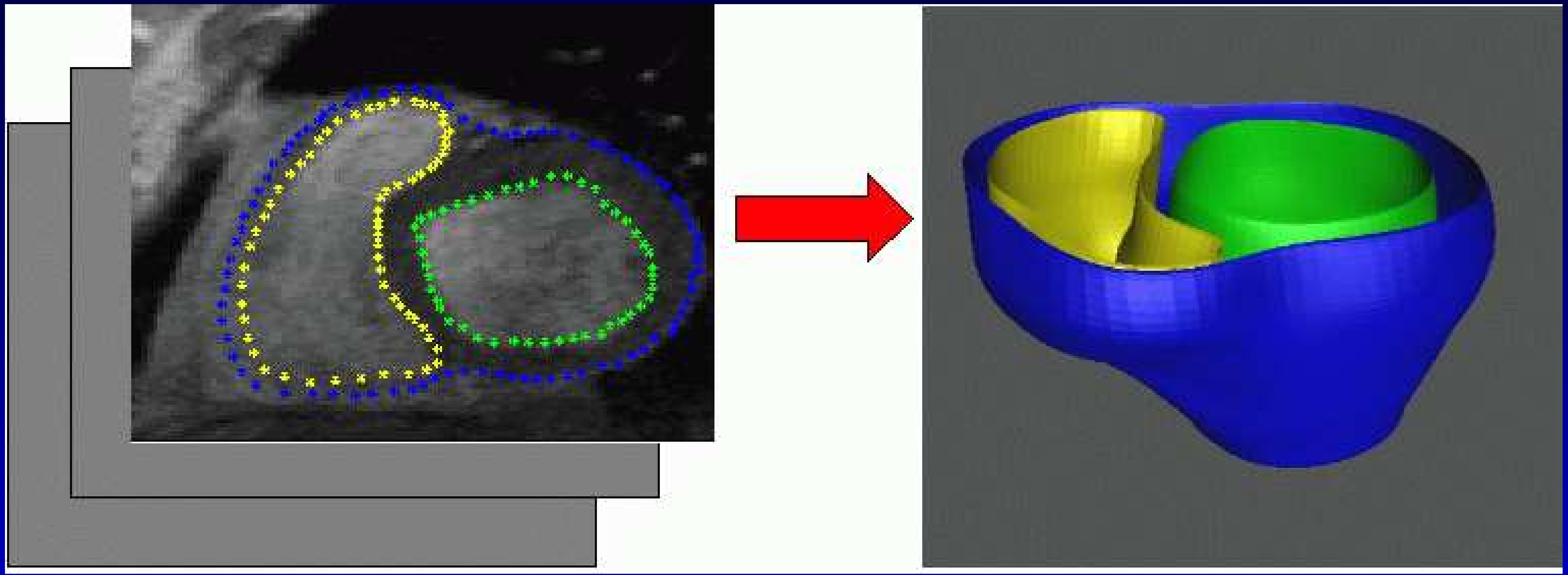
The first commercial ECG machine was built in 1911.

Electrocardiogram (ECG)



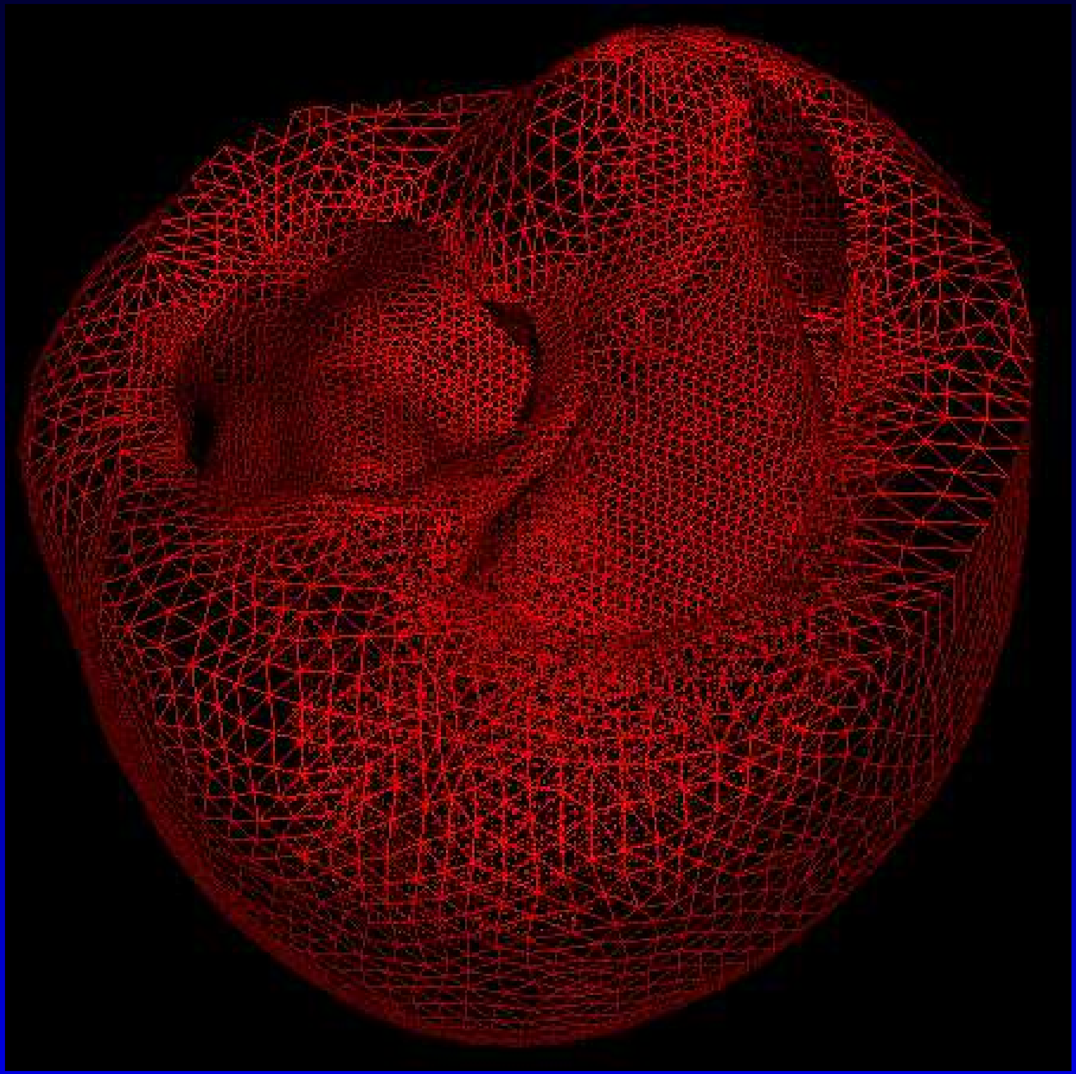
The lead positions were standardized in 1943.

Geometric modeling



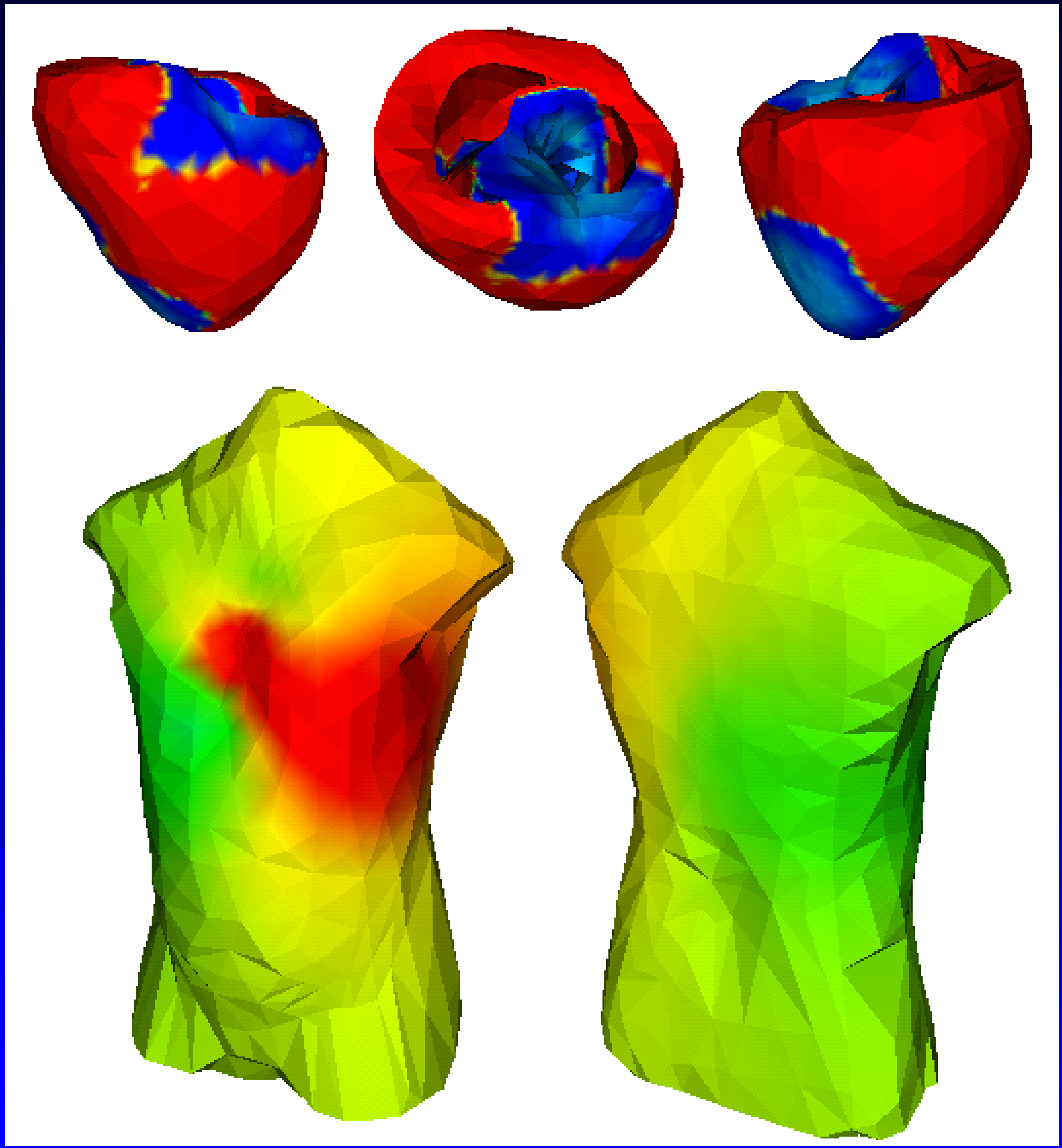
MRI slides $\Rightarrow$ continuous 3D model ($\Rightarrow$ computational mesh)

Computational mesh

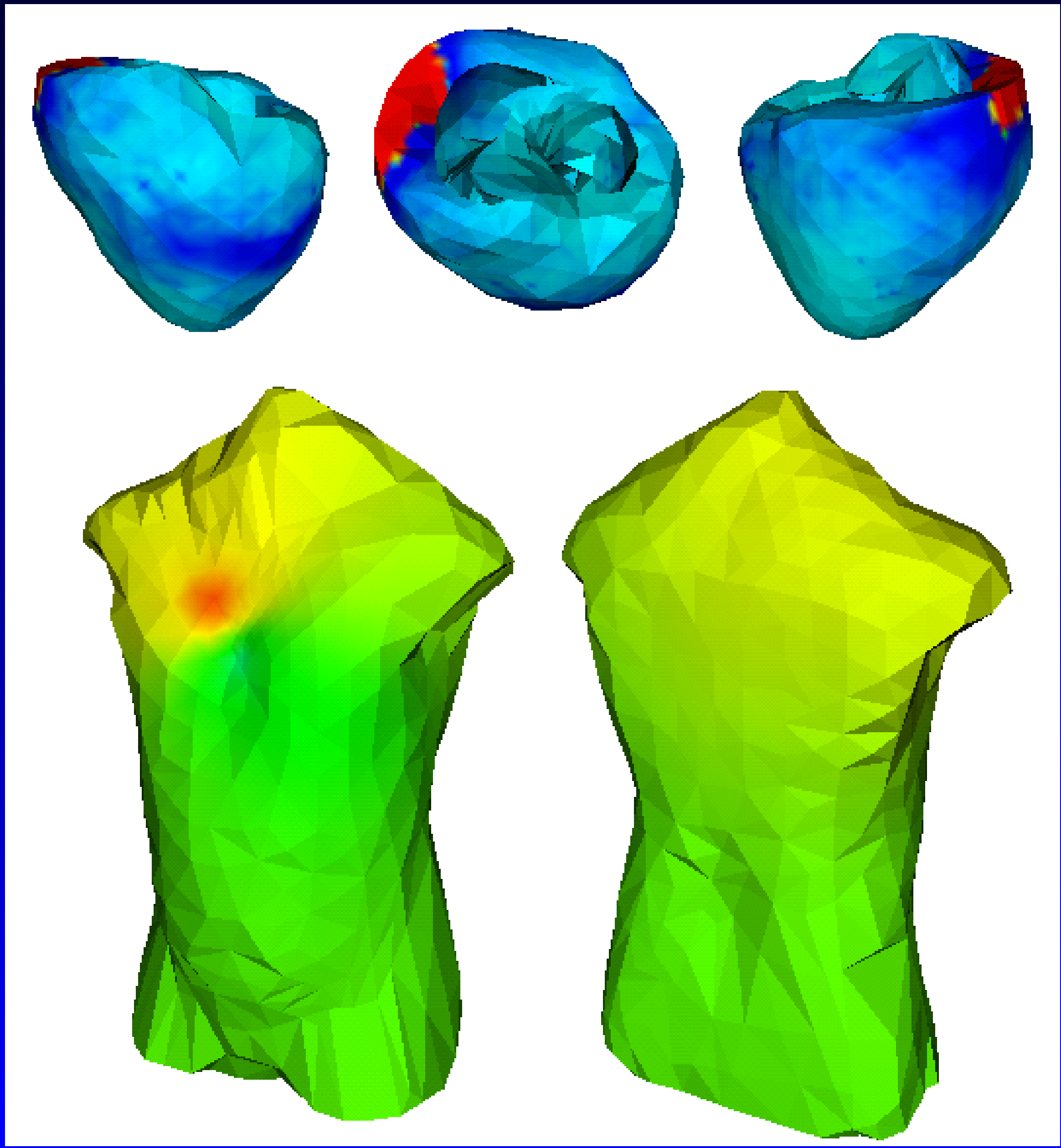


An example tetrahedral finite element mesh for the heart.

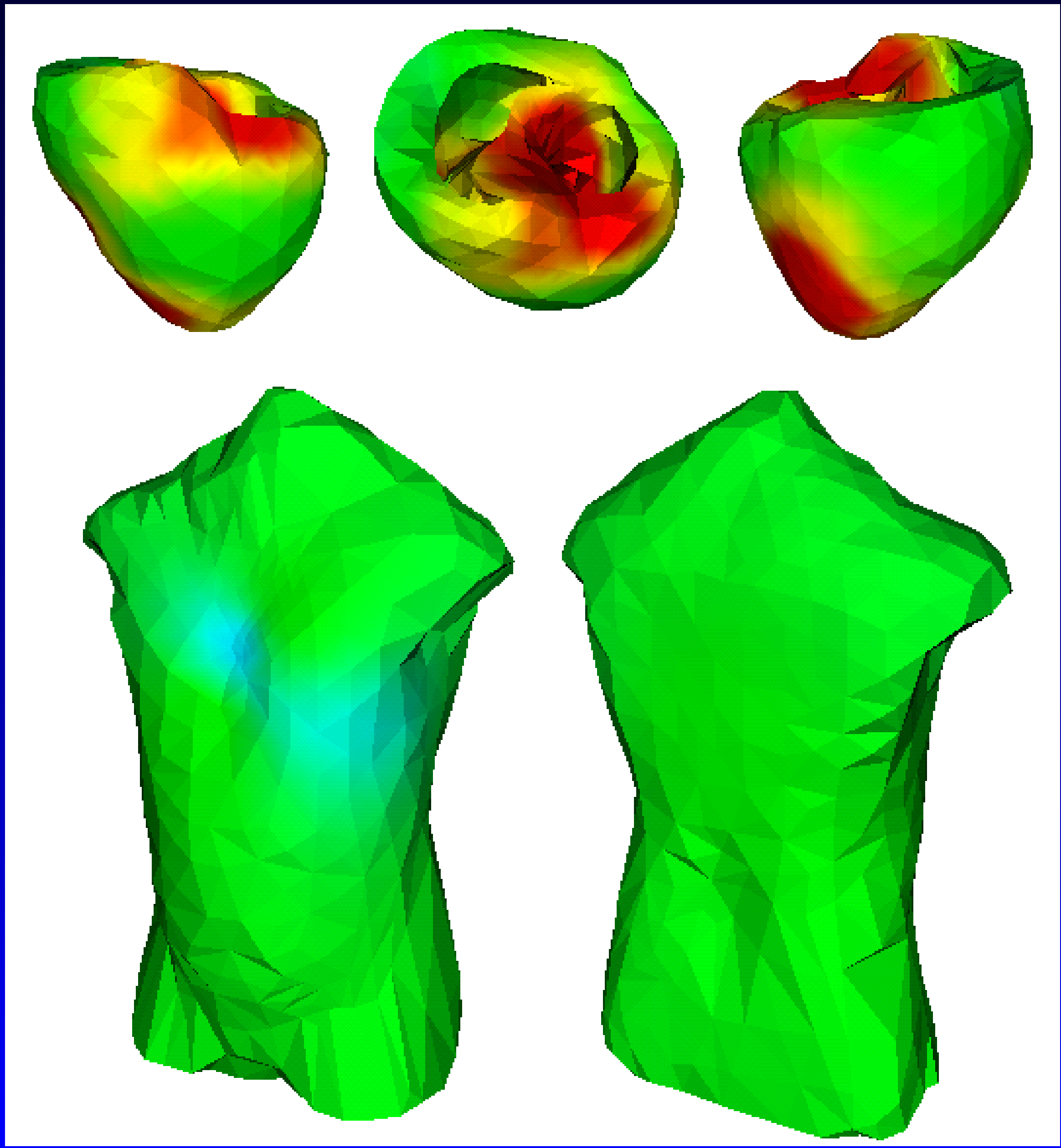
3D Snapshot #1



3D Snapshot #2



3D Snapshot #3



Mathematical model

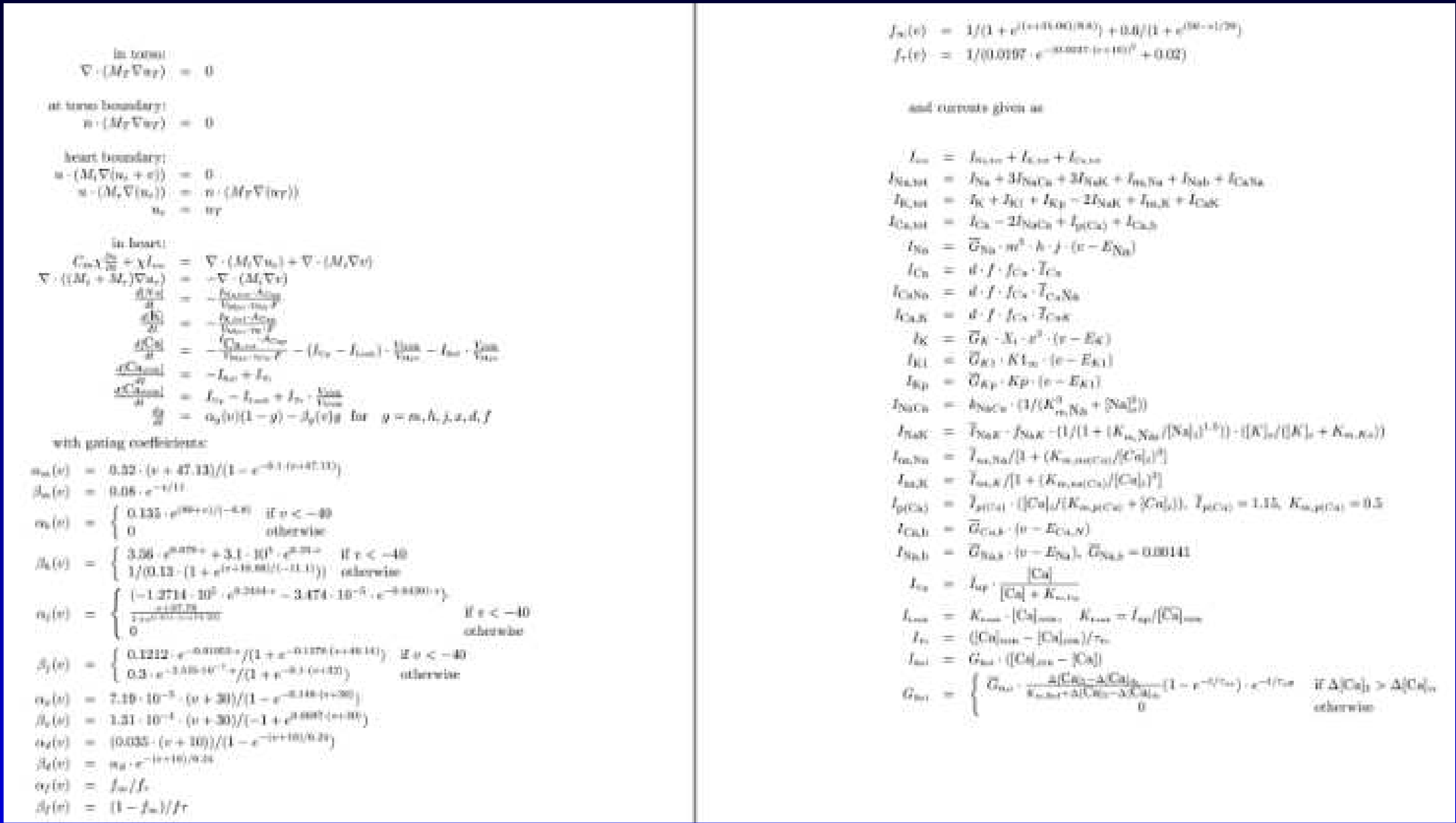
The BiDomain Model in the heart—intracellular space and extracellular space

$$
\begin{aligned}
\chi C_{\mathrm{m}} \frac{\partial v}{\partial t} + \chi I_{\mathrm{ion}} &= \nabla \cdot (M_i \nabla v) + \nabla \cdot (M_i \nabla u_e) \\
0 &= \nabla \cdot (M_i \nabla v) + \nabla \cdot ((M_i + M_e) \nabla u_e)
\end{aligned}
$$

- Transmembrane potential: $v = u_i - u_e$
- Extracellular potential: u_e
- Ionic current $I_{\mathrm{ion}}(v, \mathbf{s})$ relies on ODEs

$$
\frac{\mathrm{d}\mathbf{s}}{\mathrm{d}t} = \mathbf{F}(v, \mathbf{s})
$$

The ODE system



in torso:

$$\nabla \cdot (M_T \nabla u_T) = 0$$

at torso boundary:

$$n \cdot (M_T \nabla u_T) = 0$$

heart boundary:

$$\begin{aligned}
n \cdot (M_i \nabla (u_e + v)) &= 0 \\
n \cdot (M_e \nabla (u_e)) &= n \cdot (M_T \nabla (u_T)) \\
u_e &= u_T
\end{aligned}$$

in heart:

$$\begin{aligned}
C_m \chi \frac{\partial v}{\partial t} + \chi I_{ion} &= \nabla \cdot (M_i \nabla u_e) + \nabla \cdot (M_i \nabla v) \\
\nabla \cdot ((M_i + M_e) \nabla u_e) &= -\nabla \cdot (M_i \nabla v) \\
\frac{d[\mathrm{Na}]}{dt} &= -\frac{I_{Na,tot} \cdot A_{cap}}{V_{myo} \cdot z_{Na} \cdot F} \\
\frac{d[\mathrm{K}]}{dt} &= -\frac{I_{K,tot} \cdot A_{cap}}{V_{myo} \cdot z_K \cdot F} \\
\frac{d[\mathrm{Ca}]}{dt} &= -\frac{I_{Ca,tot} \cdot A_{cap}}{V_{myo} \cdot z_{Ca} \cdot F} - (I_{up} - I_{leak}) \cdot \frac{V_{NSR}}{V_{myo}} - I_{rel} \cdot \frac{V_{JSR}}{V_{myo}} \\
\frac{d[\mathrm{Ca}]_{JSR}}{dt} &= -I_{tr} + I_{rel} \\
\frac{d[\mathrm{Ca}]_{NSR}}{dt} &= I_{up} - I_{leak} + I_{tr} \cdot \frac{V_{JSR}}{V_{NSR}} \\
\frac{dg}{dt} &= \alpha_g(v)(1-g) - \beta_g(v) g \quad \text{for} \quad g = m, h, j, x, d, f
\end{aligned}$$

with gating coefficients:

$$\begin{aligned}
\alpha_m(v) &= 0.32 \cdot (v + 47.13)/(1 - e^{-0.1 \cdot (v+47.13)}) \\
\beta_m(v) &= 0.08 \cdot e^{-v/11} \\
\alpha_h(v) &= \begin{cases} 0.135 \cdot e^{(80+v)/(-6.8)} & \text{if } v < -40 \\ 0 & \text{otherwise} \end{cases} \\
\beta_h(v) &= \begin{cases} 3.56 \cdot e^{0.079 v} + 3.1 \cdot 10^5 \cdot e^{0.35 v} & \text{if } v < -40 \\ 1/(0.13 \cdot (1 + e^{(v+10.66)/(-11.1)})) & \text{otherwise} \end{cases} \\
\alpha_j(v) &= \begin{cases} (-1.2714 \cdot 10^5 \cdot e^{0.2444 v} - 3.474 \cdot 10^{-5} \cdot e^{-0.04391 v}) \cdot \frac{v + 37.78}{1 + e^{0.311 \cdot (v+79.23)}} & \text{if } v < -40 \\ 0 & \text{otherwise} \end{cases} \\
\beta_j(v) &= \begin{cases} 0.1212 \cdot e^{-0.01052 v}/(1 + e^{-0.1378 \cdot (v+40.14)}) & \text{if } v < -40 \\ 0.3 \cdot e^{-2.535 \cdot 10^{-7} v}/(1 + e^{-0.1 \cdot (v+32)}) & \text{otherwise} \end{cases} \\
\alpha_x(v) &= 7.19 \cdot 10^{-5} \cdot (v + 30)/(1 - e^{-0.148 \cdot (v+30)}) \\
\beta_x(v) &= 1.31 \cdot 10^{-4} \cdot (v + 30)/(-1 + e^{0.0687 \cdot (v+30)}) \\
\alpha_d(v) &= (0.035 \cdot (v + 10))/(1 - e^{-(v+10)/6.24}) \\
\beta_d(v) &= \alpha_d \cdot e^{-(v+10)/6.24} \\
\alpha_f(v) &= f_\infty / f_\tau \\
\beta_f(v) &= (1 - f_\infty)/f_\tau
\end{aligned}$$

$$\begin{aligned}
f_\infty(v) &= 1/(1 + e^{((v+12.5)/5.0)}) + 0.6/(1 + e^{(50-v)/20}) \\
f_\tau(v) &= 1/(0.0197 \cdot e^{-(0.0337 \cdot (v+10))^2} + 0.02)
\end{aligned}$$

and currents given as

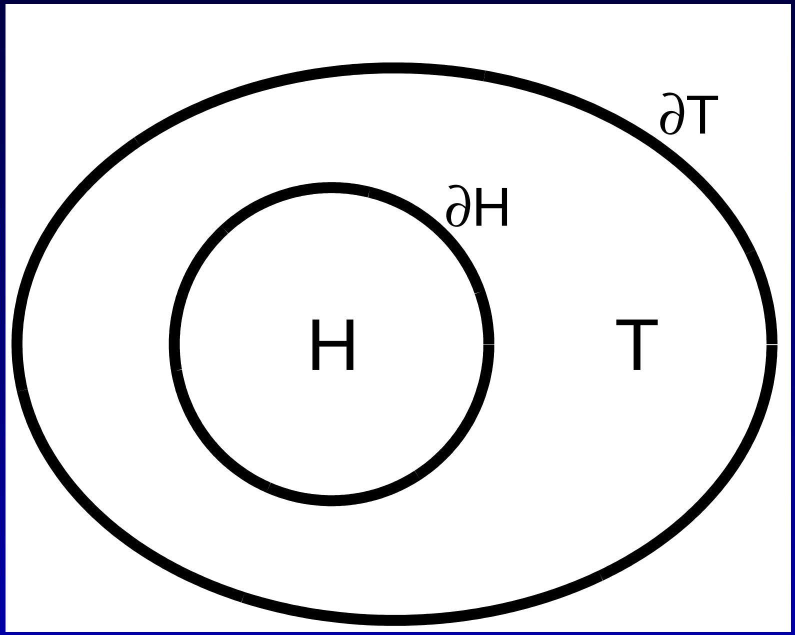
$$\begin{aligned}
I_{ion} &= I_{Na,tot} + I_{K,tot} + I_{Ca,tot} \\
I_{Na,tot} &= I_{Na} + 3 I_{NaCa} + 3 I_{NaK} + I_{ns,Na} + I_{Nab} + I_{CaNa} \\
I_{K,tot} &= I_K + I_{K1} + I_{Kp} - 2 I_{NaK} + I_{ns,K} + I_{CaK} \\
I_{Ca,tot} &= I_{Ca} - 2 I_{NaCa} + I_{p(Ca)} + I_{Ca,b} \\
I_{Na} &= \overline{G}_{Na} \cdot m^3 \cdot h \cdot j \cdot (v - E_{Na}) \\
I_{Ca} &= d \cdot f \cdot f_{Ca} \cdot \overline{I}_{Ca} \\
I_{CaNa} &= d \cdot f \cdot f_{Ca} \cdot \overline{I}_{CaNa} \\
I_{CaK} &= d \cdot f \cdot f_{Ca} \cdot \overline{I}_{CaK} \\
I_K &= \overline{G}_K \cdot X_i \cdot x^2 \cdot (v - E_K) \\
I_{K1} &= \overline{G}_{K1} \cdot K1_\infty \cdot (v - E_{K1}) \\
I_{Kp} &= \overline{G}_{Kp} \cdot Kp \cdot (v - E_{K1}) \\
I_{NaCa} &= k_{NaCa} \cdot (1/(K^3_{m,Na} + [\mathrm{Na}]^3_o)) \\
I_{NaK} &= \overline{I}_{NaK} \cdot f_{NaK} \cdot (1/(1 + (K_{m,Nai}/[\mathrm{Na}]_i)^{1.5})) \cdot ([K]_o/([K]_o + K_{m,Ko})) \\
I_{ns,Na} &= \overline{I}_{ns,Na}/(1 + (K_{m,ns(Ca)}/[Ca]_i)^3) \\
I_{ns,K} &= \overline{I}_{ns,K}/(1 + (K_{m,ns(Ca)}/[Ca]_i)^3) \\
I_{p(Ca)} &= \overline{I}_{p(Ca)} \cdot ([Ca]_i/(K_{m,p(Ca)} + [Ca]_i)), \ \overline{I}_{p(Ca)} = 1.15, \ K_{m,p(Ca)} = 0.5 \\
I_{Ca,b} &= \overline{G}_{Ca,b} \cdot (v - E_{Ca,N}) \\
I_{Na,b} &= \overline{G}_{Na,b} \cdot (v - E_{Na}), \ \overline{G}_{Na,b} = 0.00141 \\
I_{up} &= I_{up} \cdot \frac{[\mathrm{Ca}]}{[\mathrm{Ca}] + K_{m,up}} \\
I_{leak} &= K_{leak} \cdot [\mathrm{Ca}]_{NSR}, \quad K_{leak} = I_{up}/[\overline{\mathrm{Ca}}]_{NSR} \\
I_{tr} &= ([\mathrm{Ca}]_{NSR} - [\mathrm{Ca}]_{JSR})/\tau_{tr} \\
I_{rel} &= G_{rel} \cdot ([\mathrm{Ca}]_{JSR} - [\mathrm{Ca}]) \\
G_{rel} &= \begin{cases} \overline{G}_{rel} \cdot \frac{\Delta[\mathrm{Ca}]_i - \Delta[\mathrm{Ca}]_{th}}{K_{m,rel} + \Delta[\mathrm{Ca}]_i - \Delta[\mathrm{Ca}]_{th}} (1 - e^{-t/\tau_{on}}) \cdot e^{-t/\tau_{off}} & \text{if } \Delta[\mathrm{Ca}]_i > \Delta[\mathrm{Ca}]_{th} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

An example: the Winslow cell model.

Mathematical model (cont'd)

In the torso ($T = \Omega \backslash H$):

$$-\nabla \cdot (M_o \nabla u_o) \;=\; 0.$$



Boundary conditions:

$$\text{on } \partial H: \; M_i \frac{\partial}{\partial n}(v + u_e) = 0, \quad u_e = u_o, \quad M_e \frac{\partial u_e}{\partial n} - M_o \frac{\partial u_o}{\partial n} = 0.$$

$$\text{on } \partial T: \; M_o \frac{\partial u_o}{\partial n} = 0.$$

Computational challenges

- Advanced mathematical model
- Realistic 3D geometries (heart and torso)
- Anisotropic and inhomogeneous conductivities
- Need high resolution in space and time
  - Desired spatial resolution: $0.2\text{mm} \sim 50 \times 10^6$ mesh points
  - Desired temporal resolution: $0.1\text{ms} \sim 10000$ time steps
  - Number of degrees of freedom: e.g., $(2+31) \times 50 \times 10^6$ per time step!

Numerical strategy

During each time step:

- Solve the ODE system $\mathrm{d}\mathbf{s}/\mathrm{d}t = \mathbf{F}(v^{l-1}, \mathbf{s})$
- Solve the two PDEs simultaneously

$$
\begin{aligned}
\chi C_{\mathrm{m}} \frac{v^l - \tilde{v}^{l-1}}{\Delta t} + \chi I_{\mathrm{ion}} &= \nabla \cdot (M_i \nabla v^l) + \nabla \cdot (M_i \nabla u_e^l) \\
0 &= \nabla \cdot (M_i \nabla v^l) + \nabla \cdot ((M_i + M_e) \nabla u_e^l)
\end{aligned}
$$

- 2×2 block linear system:

$$
\begin{bmatrix} \mathbf{I} + \theta \Delta t \mathbf{A}_{\mathrm{v}} & \theta \Delta t \mathbf{A}_{\mathrm{v}} \\ \theta \Delta t \mathbf{A}_{\mathrm{v}} & \theta \Delta t \mathbf{A}_{\mathrm{u}} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}
$$

Observations

- An ODE system (30∼50 degrees of freedom) needs to be solved at *each* mesh point in the heart
- The two PDEs are solved simultaneously by a 2×2 block linear system
- The “scalability bottleneck” is the PDE part
- Objective: find a scalable preconditioner for the block system solver!

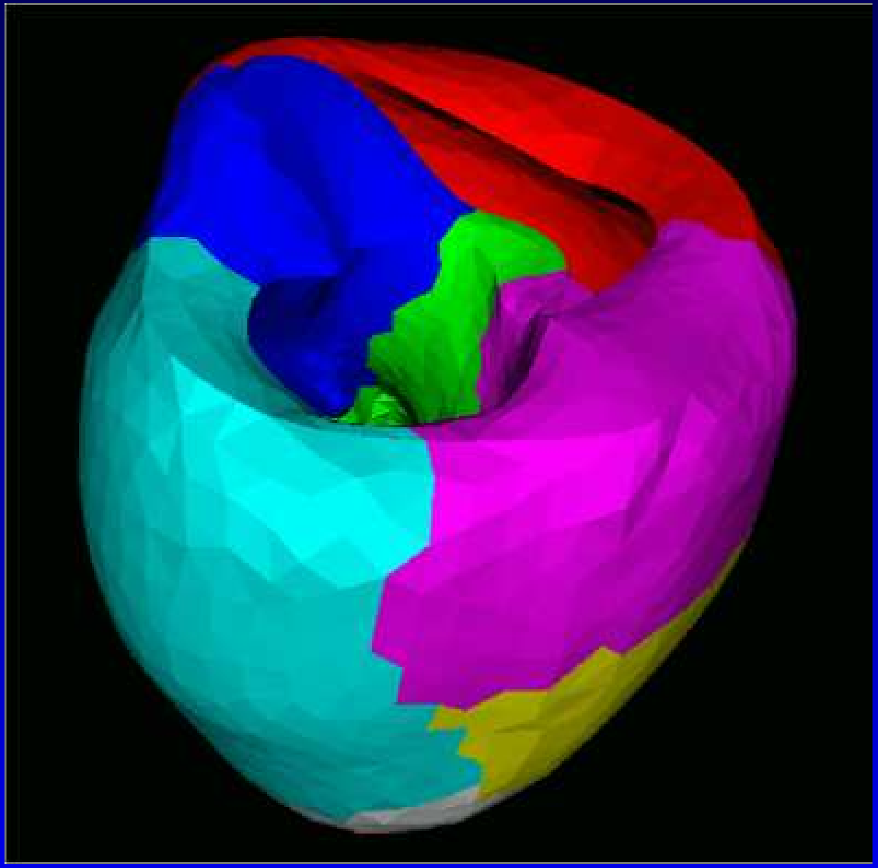
Serial preconditioning

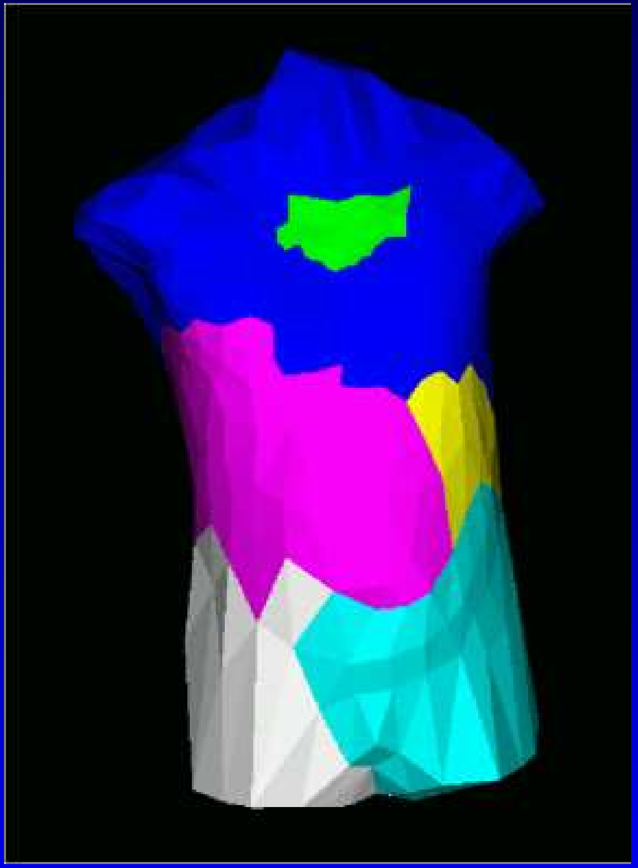
- $\begin{bmatrix} \mathbf{I} + \theta \Delta t \mathbf{A}_{\mathrm{v}} & \mathbf{0} \\ \mathbf{0} & \theta \Delta t \mathbf{A}_{\mathrm{u}} \end{bmatrix}$ as preconditioner for the 2×2 block system
- multigrid V-cycle for solving both $(1,1)$ block and $(2,2)$ block
- new theory; scalable performance

	No preconditioning	With preconditioning
# unknowns	# CG iters	# CG iters
302,166	1999	15
1,552,283	4087	16

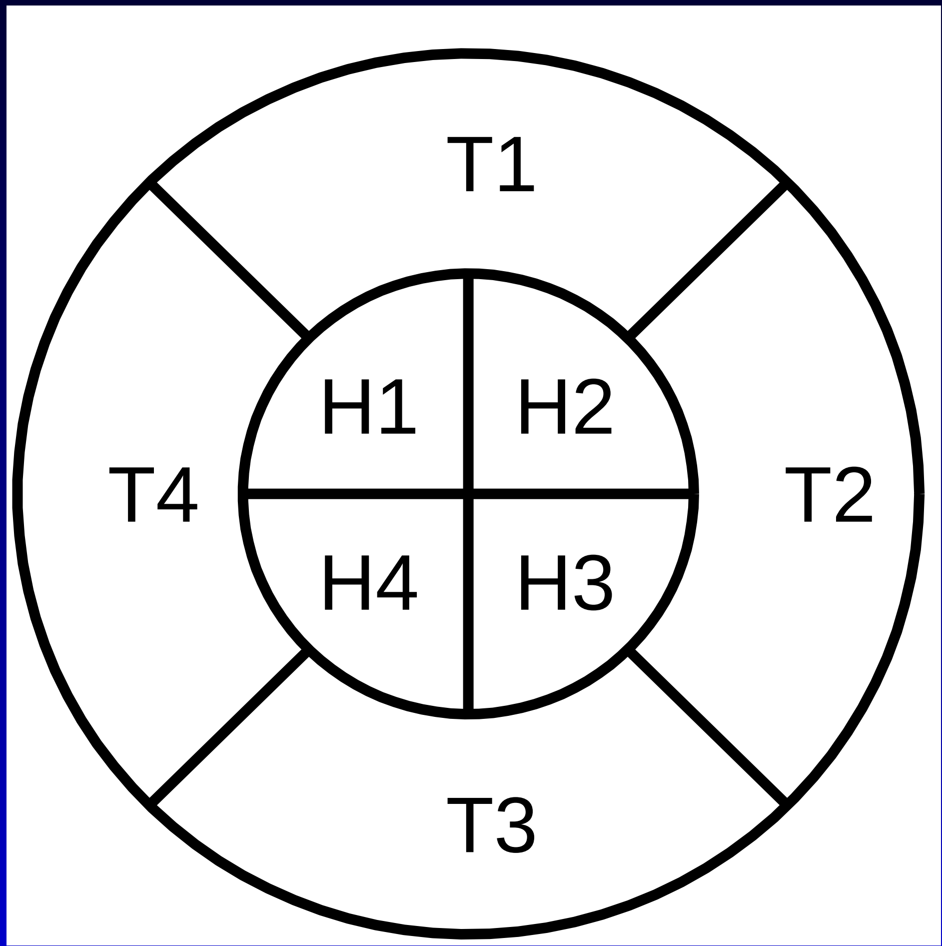
Parallelization

Partitioning H and Ω into subdomains; each processor is responsible for a piece of H and a piece of Ω





Parallelization (cont'd)



Two subdomains per processor: H_i and T_i.

Parallelization (cont'd)

- Each processor only builds local matrices/vectors needed by a subdomain
- No need for physical storage of global matrices or vectors
- Solution of ODEs: embarrassingly parallel
- Solution of PDEs: parallel linear algebra operations + parallel preconditioning
- MPI for communication, portable to any parallel platform

Parallel preconditioner

- $\begin{bmatrix} \mathbf{I} + \theta \Delta t \mathbf{A}_{\mathrm{v}} & \mathbf{0} \\ \mathbf{0} & \theta \Delta t \mathbf{A}_{\mathrm{u}} \end{bmatrix}$ as preconditioner
- Schwarz iteration (overlapping DD) for solving $(1,1)$ block
- Schwarz iteration (overlapping DD) for solving $(2,2)$ block
- Multigrid V-cycle as subdomain solver
- Coarse grid correction in Schwarz iteration

Parallel software

- Need flexibility in the software
  - Plug-and-play of different cell models
  - Plug-and-play of different components in the linear solver
  - Different types of element basis functions
- Programming inside Diffpack — http://www.diffpack.com
  - Object-oriented programming on high levels
  - Fortran-style programming on computation-intensive levels

Numerical scalability

Solving the 2×2 block system:

# unknowns	$P=2$	$P=4$	$P=8$	$P=16$	$P=32$	$P=64$
302,166	9	9	10	10	12	
1,552,283	9	9	10	10	11	11
10,705,353				13	13	14
81,151,611				14	15	15

CG iterations needed (residual reduction factor=10^4)

Unstructured 3D mesh: has up to 252,143,960 tetrahedra

Parallel scalability (I)

Measurements on a switch-based Linux cluster using 1.3 GHz Itanium 2 processors, inter-connected through a 1Gbit/s ethernet, 4GB memory per node

# unknowns	$P = 1$	$P = 2$	$P = 4$	$P = 8$	$P = 16$
302,166	153.19	93.44	53.64	30.55	23.14
1,552,283	982.11	585.88	327.59	195.79	127.79
10,705,353	N/A	N/A	2511.18	1534.65	952.51

Wall-clock time measurements of one time step

Parallel scalability (II)

Measurements on Origin 3800, R14000 600MHz processors

# unknowns	$P = 8$	$P = 16$	$P = 32$
302,166	55.30	33.25	21.70
1,552,283	385.08	242.93	142.51
10,705,353	3406.51	2206.67	1197.09

Wall-clock time measurements of one time step

Remarks

- Improvements can be achieved in the following aspects:
  - Better load balance
  - More efficient serial computations
  - Overhead reduction (latency hiding)
- Storing full-scale simulation results will require a lot of disk space:
  - # mesh points: 5×10^7 (# values 10^8)
  - # time steps: 2500
  - Total disk space: 2×10^{12} bytes (at least)
  - Should use a distributed storage approach

Summary

- Large-scale parallel electro-cardiac simulations require a scalable numerical and parallelization strategy
- Sophisticated numerical techniques are fundamental:
  - Block preconditioning
  - Schwarz iteration (overlapping DD)
  - Multigrid subdomain solver
  - Global coarse grid correction
- Linux-clusters can work as a suitable parallel platform
- More work is needed for further improvement