Parallel Computing on Distributed-Memory Clusters for Long-Time Simulations of Calcium Waves

Matthias K. Gobbert

Department of Mathematics and Statistics
University of Maryland, Baltimore County (UMBC)
gobbert@umbc.edu, www.math.umbc.edu/~gobbert
Related initiatives: HPCF, CIRC, REU Site on HPC, UBM; see links on my webpage

Current collaborators: Xuan Huang^G, Jonas Schäfer^G, Andreas Meister, Yu Wang^G (HPCF RA), Marc Olano, Zana Coulibaly^G, Bradford E. Peercy, Thomas I. Seidman, Andrew Raim^G (HPCF RA), Michael Muscedere^G

Former collaborators: Kyle Stern^G (CIRC RA), Shiming Yang^G (CIRC RA), Kevin Allen^{UG} (URA), Alexander Hanhart^G, Leighton T. Izu

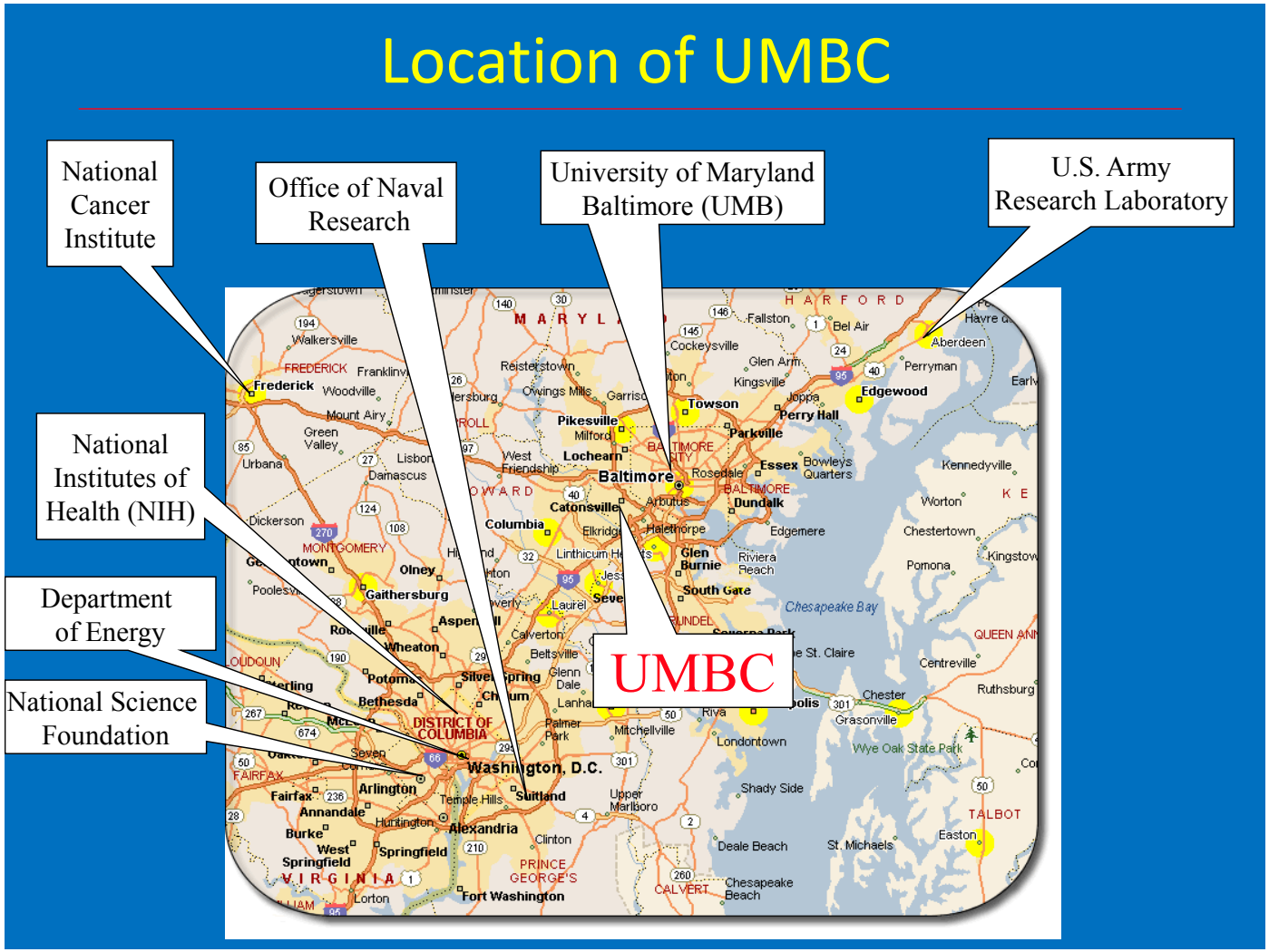
Acknowledgments: NSF, UMBC, HPCF, CIRC

UMBC: An Honors University in Maryland

- Located in the Mid-Atlantic region — 15 minutes South of Baltimore, 30 minutes North of Washington
- Founded in 1966 as third Ph.D.-granting research university in the University System of Maryland
- Science and Technology focus (Applied Math. oldest Ph.D. program, Statistics B.S. only in Maryland), also strengths in Visual Arts, Performing Arts, and Humanities
- 13,000 students (10,000 undergraduate, 3,000 graduate students); ≈ 480 research faculty in ≈ 33 departments — *smallest public research university in U.S.!*
- Ranked #1 "up and coming" university by *U.S. News & World Report* in 2009, 2010 — and again in 2011!

Location of UMBC

UMBC in the Baltimore-Washington corridor of science and technology:



Brief Biography

Matthias K. Gobbert, `www.math.umbc.edu/~gobbert`

1988	Friedrichsgymnasium Kassel	Abitur
1990	Technische Hochschule Darmstadt	Vordiplom (B.Sc.)
1993	Arizona State University	M.N.S.
1996	Arizona State University	Ph.D.
1996–1997	University of Minnesota, Institute for Mathematics and its Applications (IMA)	Post-doc
1997–2003	UMBC, Mathematics and Statistics	Assistant Professor
2003–2010	UMBC, Mathematics and Statistics	Associate Professor
2005–	Center for Interdisciplinary Research and Consulting (CIRC)	`www.umbc.edu/circ`
2008–	High Performance Computing Facility	`www.umbc.edu/hpcf`
2010–	REU Site: Interdisciplinary Program in High Performance Computing	`www.umbc.edu/hpcreu`
2010–	UMBC, Mathematics and Statistics	Professor
2011–2012	Universität Kassel, FB Mathematik	Gastdozent

Current Research Interests

From webpage `www.math.umbc.edu/~gobbert`:

- Scientific computing and parallel algorithms for computing clusters and other architectures
- Applications in the life sciences, computational biology, engineering, statistics, and other areas
- Multiscale modeling and numerical simulation of chemically reactive flows; more information: `www.math.umbc.edu/~gobbert/calcium`
- Numerical methods for stationary and time-dependent partial differential equations

Collaborations — *mostly faculty listed; not all listed!*:

- Math biology, physiology, calcium flow, and related: Xuan Huang^{G}, Samuel Khuvis^{G}, Zana Coulibaly^{G}, Dr. Bradford E. Peercy, Thomas I. Seidman, Math and Stat (UMBC)
- Numerical linear algebra and computer science: Jonas Schäfer^{G}, Dr. Andreas Meister, Uni Kassel, Germany; Yu Wang^{G} (HPCF RA), Marc Olano, CSEE (UMBC)
- Computational statistics: Jonathan McHenry^{G} (CIRC RA), Andrew Raim^{G} (HPCF RA), Dr. Nagaraj K. Neerchal, Math and Stat (UMBC)
- UBM (bioinformatics, information theory): Robert Forder^{UG}, Matthew Brewster^{UG}, Kathryn Cronise^{UG}, Dr. Ivan Erill, Dr. Kevin Omland, Biology (UMBC)
- REU Site (NSA): Randal Mckissack^{UG}, Richard Adjogah^{UG}, CSEE (UMBC)

$\longrightarrow$ Links and more information on my webpage!

How Research Works — Example of Calcium Wave Simulations:

Project history (^{G} for graduate and ^{UG} for undergraduate co-author):

1998	Initial contact by Leighton T. Izu, U. of Maryland, Baltimore
2000–2002	Convergence of FEM and coarse-grained parallel solution: [Hanhart^{G} M.S. thesis; Hanhart^{G}, Gobbert, Izu *JCAM* 2004]
2002–2008	Development of C code with fine-grained MPI parallelism: [Allen^{UG} and Gobbert, *ICCSA* 2003; Allen^{UG}, senior thesis 2003 and *UMBC Review* 2004; Gobbert, *SISC* 2008; Gobbert, HPCF–2008–1; Muscedere^{G}, Gobbert, HPCF–2008–2; Raim^{G} and Gobbert, HPCF–2010–2; Muscedere^{G}, Raim^{G}, Gobbert, HPCF–2010–4; Trott^{G} and Gobbert, HPCF–2010–11]
2007–2008	Convergence of stationary FEM in COMSOL Multiphysics: [Yang^{G} and Gobbert, HPCF–2008–4 and *COMSOL* 2008]
2004–2011	Rigorous convergence of time-dependent FEM: [Seidman, Gobbert, Trott^{G}, Kružík, *Numer. Math.*, submitted; Trott^{G} and Gobbert, *COMSOL* 2011]
2009–2011	Application studies for wave with recovery and spiral wave: [Coulibaly^{G}, Muscedere^{G}, Gobbert, Peercy, HPCF–2009–6; Coulibaly^{G}, Peercy, Gobbert, in preparation (2 papers)]

Outline

- Introduction
  - Biological Background and Motivation
  - Mathematical Model and Numerical Challenges
- Numerical Methods Used
- Numerical Results
  - Convergence Studies
  - Application Problem Studies
  - Performance Studies of Parallel Code
- Conclusions and Future Work

A Model for Calcium Flow in a Heart Cell

Motivations

Leading Causes of Death (Data are for the U.S.)
Number of deaths for leading causes of death

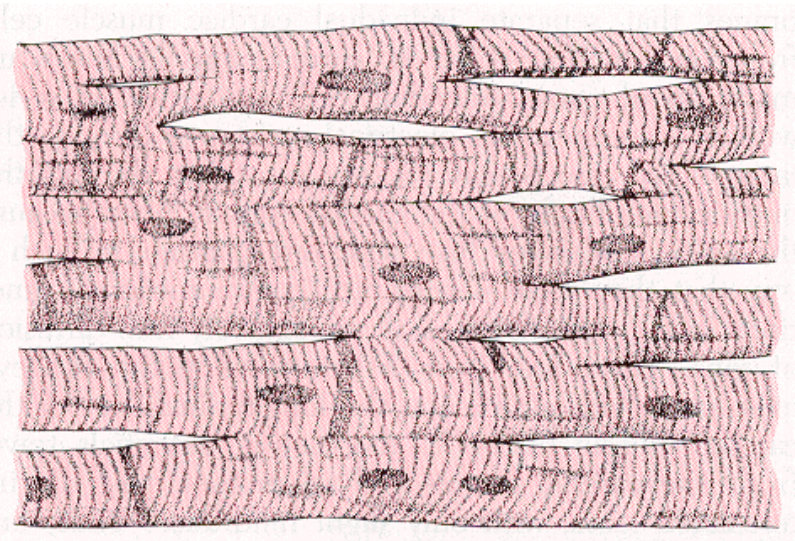
- Heart disease: 616,067
- Cancer: 562,875
- Stroke (cerebrovascular diseases): 135,952
- Chronic lower respiratory diseases: 127,924

(Source: CDC – Centers for Disease Control and Prevention. 2007 Statistics Published in May 2010.)

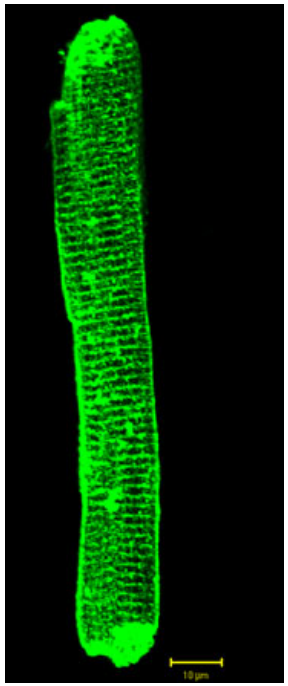
An estimated 81,100,000 American adults (more than 1 in 3) have 1 or more types of Cardiovascular Disease.
(Source: American Heart Association Heart Disease and Stroke Statistics – 2010 Update.)

Heart:

- Consists of numerous cell types: some form heart connective tissue, other cells grow into heart valves, and cardiac muscle cells (heart cells) give the heart its ability to beat and pump blood throughout the body.



Cardiac cell structure (Guyton and Hall, 1996, p. 108)

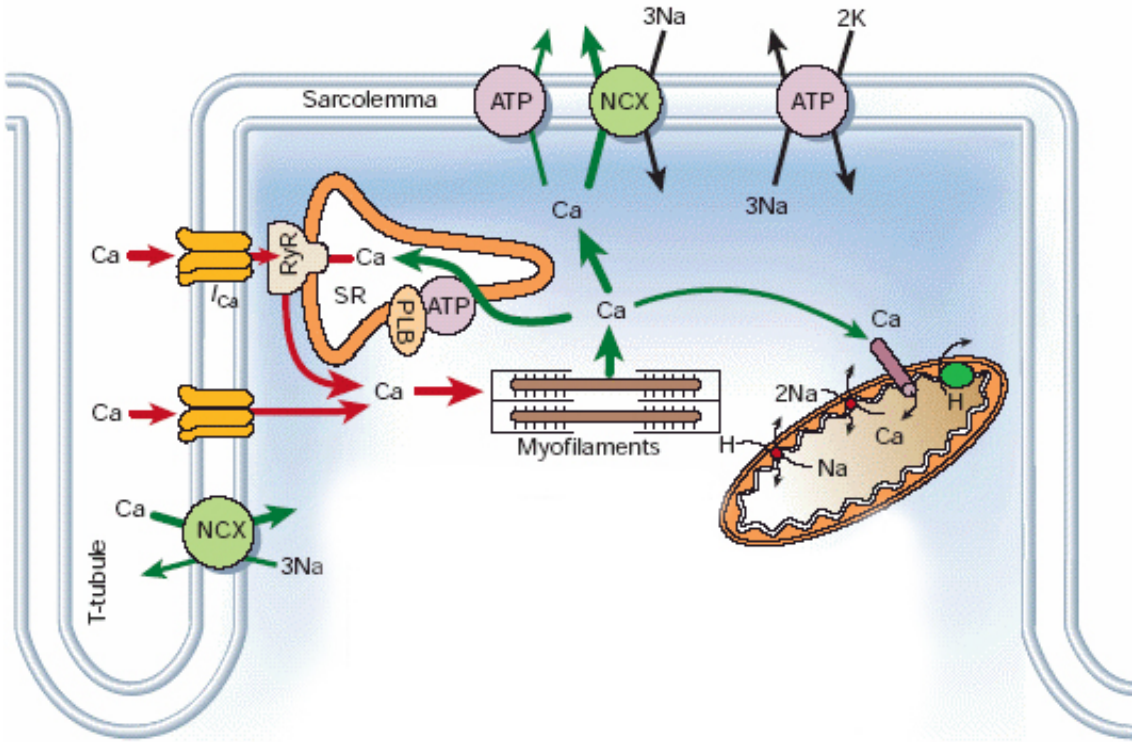


Single heart cell

⇒ **Multiscale Modeling Problem!** We consider one cell.

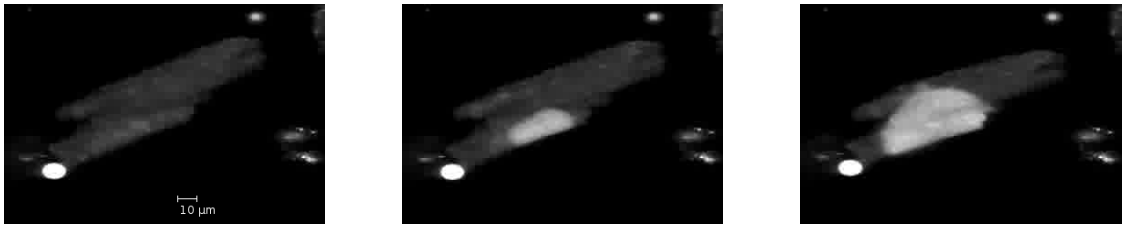
Calcium:

- Plays crucial role in the proper functioning of the heart cell
- Electrical stimulation of the heart cell triggers thousands of calcium sparks that sum to increase intracellular calcium levels. The increase in calcium subsequently results in the activation of calcium-sensitive proteins that are responsible for cell-shortening, or contraction.

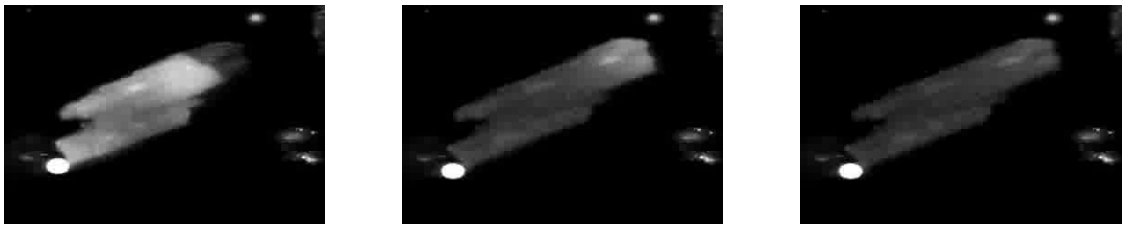


Calcium (cont.):

- Spontaneous calcium sparks can occur without external electrical stimulation!
- Experiments



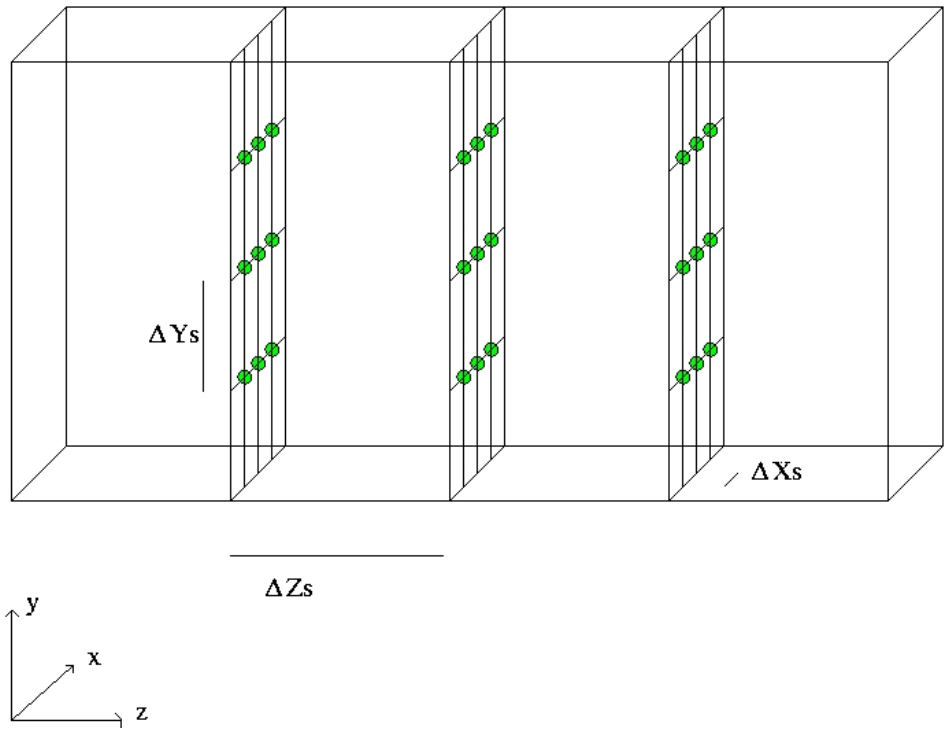
(1.a) t=0ms (1.b) t=399ms (1.c) t= 726ms



(1.d) t=1287ms (1.e) t=2079ms (1.f) t=2409ms

Figures (1.a–1.f) show an experimental calcium wave propagation in a rabbit ventricular myocyte with confocal imaging used to record fluo-4 fluorescence (courtesy of Dr. Kenneth Spitzer, CVRTI, University of Utah). The experiment uses a rabbit ventricular myocyte bathed in solution containing 10mM Ca at 26°C, pH 7.4. The pacing was at a cycle length of 400 milliseconds for a few beats to overload the SR. The frame size is 168μm×168μm.

Domain and Calcium Release Units



Model cell: $\Omega = (-6.4, 6.4) \times (-6.4, 6.4) \times (-32.0, 32.0)$, units of μm,

CRU spacings $\Delta z_s = 2.0$ μm between z-planes and

$\Delta x_s = \Delta y_s = 0.8$ μm on z-plane

$\Rightarrow$ large domain with $15 \times 15 \times 31 = 6{,}975$ calcium release units (CRUs)!

Model for Calcium Release through the CRUs

- Calcium ions (Ca^{2+}) are released in a cell at discrete points called Calcium Release Units (CRUs).
  $15 \times 15 \times 31 = 6{,}975$ CRU lattice
  $\Rightarrow$ **need high-resolution** $128 \times 128 \times 512$ **mesh**
  $\Rightarrow$ 25.6 million DOF for 3-species model
- A high concentration of Ca^{2+} at a CRU will increase the probability that it will release ions ('fires').
- Organized waves of increasing calcium drive the heart beat, but waves can also self-organize spontaneously
- CRUs may open every 1 ms; if opened, stay open for 5 ms; afterwards stay closed for at least 100 ms.
  $\Rightarrow$ **need large final time to simulate several waves**
  on laboratory time scales, for instance, $t_{\text{fin}} = 1{,}000$ ms $= 1$ s.

Propagating Waves of Calcium Release in a Heart Cell

The release of calcium ions from the sarcoplasmic reticulum (SR) occurs through clusters of ryanodine receptors (RyRs)
[Izu et al. 1998, 2001a, 2001b, 2006; Hanhart et al. 2004].
Species in model: calcium ions $\mathrm{C} = u^{(0)}$, fluorescent calcium indicator $\mathrm{F} = u^{(1)}$, endogenous calcium buffer $\mathrm{B} = u^{(2)}$, and F and B bound to calcium as $\mathrm{G} = u^{(3)}$ and $\mathrm{H} = u^{(4)}$.
Reaction model and reaction rates (species generation rates):

$$\begin{aligned} \mathrm{C} + \mathrm{F} \rightleftharpoons \mathrm{G} \quad &\Leftrightarrow \quad R^{(1)} = -k_1^+ u^{(0)} u^{(1)} + k_1^- u^{(3)}, \\ \mathrm{C} + \mathrm{B} \rightleftharpoons \mathrm{H} \quad &\Leftrightarrow \quad R^{(2)} = -k_2^+ u^{(0)} u^{(2)} + k_2^- u^{(4)}, \end{aligned}$$

The total of bound and unbound indicator and buffer species is conserved, that is, $u^{(1)} + u^{(3)} = \overline{u}_1$ and $u^{(2)} + u^{(4)} = \overline{u}_2$ used to eliminate $u^{(3)}$ and $u^{(4)}$ from the reaction rates.

Model for Calcium Release through the CRUs

Model for the calcium release units (CRUs) as point sources:

$$J_{\mathrm{SR}}(u^{(0)}, \mathbf{x}, t) = \sum_{\hat{\mathbf{x}} \in \Omega_s} \sigma \, S_{\hat{\mathbf{x}}}(u^{(0)}, t) \, \delta(\mathbf{x} - \hat{\mathbf{x}})$$

$S_{\hat{\mathbf{x}}}(u^{(0)}, t) = 1$ for $t_{\mathrm{open}} = 5$ ms with probability

$$J_{\mathrm{prob}}(u^{(0)}) = \frac{P_{\max} \, (u^{(0)})^{n_{\mathrm{prob}}}}{(K_{\mathrm{prob}})^{n_{\mathrm{prob}}} + (u^{(0)})^{n_{\mathrm{prob}}}}$$

then closed for $t_{\mathrm{closed}} = 100$ ms.
Model point source mathematically as Dirac delta distribution $\delta(\mathbf{x} - \hat{\mathbf{x}})$ defined by $\delta(\mathbf{x} - \hat{\mathbf{x}}) = 0$ for all $\mathbf{x} \neq \hat{\mathbf{x}}$ and $\int \delta(\mathbf{x} - \hat{\mathbf{x}}) \, \varphi(\mathbf{x}) \, d\mathbf{x} = \varphi(\hat{\mathbf{x}})$ for any continuous function $\varphi(\mathbf{x})$. $\Rightarrow$ **need finite element method (FEM)** ***and*** **classical theory cannot guarantee convergence** $\Rightarrow$ research topic!

Calcium Flow in a Heart Cell

Problem: System of time-dependent reaction-diffusion equations

$$\frac{\partial u^{(i)}}{\partial t} - \nabla \cdot \left(D^{(i)} \nabla u^{(i)} \right) + a_i \, u^{(i)} = f^{(i)} + r^{(i)} + \left(J_{\text{pl}} + J_{\text{SR}} \right) \delta_{i0}$$

for the concentrations $u^{(i)}(\mathbf{x}, t)$, $0 \leq i < n_s$, of the n_s reactive species for all $\mathbf{x} \in \Omega \subset \mathbb{R}^3$ $(d = 3)$ and time $0 \leq t \leq t_{\text{fin}}$.
Calcium flow: $n_s = 3$, species $i = 0$ denotes calcium ions (Ca^{2+}).

- Diffusivity matrices $D^{(i)} \in \mathbb{R}^{3\times 3}$ diagonal positive definite;
- constant $a_i \geq 0$;
- linear forcing term $f^{(i)}(\mathbf{x}, t)$;
- non-linear reaction terms $r^{(i)}(u^{(0)}, \ldots, u^{(n_s-1)})$;
- additional terms for Ca^{2+} species: $J_{\text{pl}} = -J_{\text{pump}} + J_{\text{leak}}$ and J_{SR}.

(Kronecker delta defined as $\delta_{ij} = 0$ for all $i \neq j$ and $\delta_{ij} = 1$ for $i = j$)

Calcium Flow in a Heart Cell

Reaction terms:

$$r^{(i)}(u^{(0)},\ldots,u^{(n_s-1)}) := \begin{cases} \sum_{j=1}^{n_s-1} R^{(j)}(u^{(0)},u^{(j)}), & \text{for } i=0, \\ R^{(i)}(u^{(0)},u^{(i)}), & \text{for } i=1,\ldots,n_s-1. \end{cases}$$

with reaction rates

$$R^{(i)} = -k_i^+ u^{(0)} u^{(i)} + k_i^- \left(\overline{u}_i - u^{(i)}\right) \qquad \text{for } i=1,\ldots,n_s-1.$$

Pump and leak terms $J_{\text{pl}} = -J_{\text{pump}} + J_{\text{leak}}$:

$$J_{\text{pump}}(u^{(0)}) = \frac{V_{\text{pump}}\,(u^{(0)})^{n_{\text{pump}}}}{(K_{\text{pump}})^{n_{\text{pump}}} + (u^{(0)})^{n_{\text{pump}}}}.$$

and $J_{\text{leak}} = J_{\text{pump}}(u^{(0)})$ for $u^{(0)} = 0.1\ \mu\text{M}$ at rest.

Summary of Numerical Methods Used

Method of Lines using Finite Elements

- Take advantage of the regular shape of the domain Ω and use a uniform mesh of 3-D brick elements of size $\Delta x\ \Delta y\ \Delta z$
- Use tri-linear nodal basis functions and expand in global basis functions $u^{(i)}(\mathbf{x},t) \approx u_h^{(i)}(\mathbf{x},t) = \sum_k \mathbf{u}_k^{(i)}(t)\,\varphi_k(\mathbf{x})$ with $\mathbf{u}_k^{(i)}(t) \approx u^{(i)}(\mathbf{x}_k,t)$
- Take advantage of the structure and constant coefficients to precompute the stiffness matrix, mass matrix, etc. analytically.

Then need to solve the ODE problem

$$\hat{M}\,\frac{d\mathbf{u}^{(i)}}{dt} = -(K^{(i)} + M_a^{(i)})\,\mathbf{u}^{(i)} + \mathbf{F}^{(i)} + \mathbf{r}^{(i)} + (\mathbf{j}_{\rm pl} + \Sigma)\,\delta_{i0}.$$

with lumped mass matrix $\hat{M}$ (for all species) [Thomée 2006] and

$$\Sigma_k := \int_\Omega J_{\rm SR}\,\varphi_k\,d\mathbf{x} = \sigma \sum_{\hat{\mathbf{x}}\in\Omega_s} S_{\hat{\mathbf{x}}} \int_\Omega \delta(\mathbf{x}-\hat{\mathbf{x}})\,\varphi_k(\mathbf{x})\,d\mathbf{x} = \sigma \sum_{\hat{\mathbf{x}}\in\Omega_s} S_{\hat{\mathbf{x}}}\,\varphi_k(\hat{\mathbf{x}}).$$

Summary of Other Numerical Method Components

- ODE method: fully implicit NDFk, $1 \leq k \leq 5$, with automatic step size and order selection [Shampine and Reichelt, *SISC* 1997]
- non-linear solver: Newton method with analytical Jacobian [Gobbert, *SISC* 2008]
- linear solver: iterative QMR method with matrix-free products for all system matrices and their transposes [Demmel, *SIAM* 1997]
- matrix-free implementation of all matrix-vector products $\Rightarrow$ Jacobian-Free Newton-Krylov (JFNK) method
- since matrix-free, all matrices are always up to date, hence convergence of Newton is best possible

Convergence of the Finite Element Method for Highly Non-Smooth Source Terms

Scalar Test Problem with Smooth Source Term

Scalar test problem with $n_s = 1$, $D = 1$, $a = 0$, $f \equiv 0$, and all application-related functions set to 0.

Error on Ω against reference solution (estimated convergence order $q^{(\text{est})}$)			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	3.9999e–02	5.6112e–02	6.9959e–02
$32 \times 32 \times 128$	9.9770e–03 (2.0033)	1.3984e–02 (2.0046)	1.7424e–02 (2.0054)
$64 \times 64 \times 256$	2.3849e–03 (2.0647)	3.3408e–03 (2.0655)	4.1608e–03 (2.0661)
$128 \times 128 \times 512$	4.7750e–04 (2.3204)	6.6881e–04 (2.3205)	8.3285e–04 (2.3207)

- Classical theory for $u_t - \nabla \cdot (D \nabla u) = f$ with $f \in L^2(\Omega)$ [e.g., Thomeé 2006]:
  $\|u - u_h\|_{L^2(\Omega)} \leq C\, h^q$ with $q = 2$ on $\Omega \subset \mathbb{R}^d$ for all $d = 1, 2, 3$

Scalar Test Problem with Non-Smooth Source Term

Scalar test problem with $n_s = 1$, $D = 1$, $a = 0$, $f \equiv 0$, and all application-related functions set to 0, except 1 CRU at cell center.

Error on Ω against reference solution (estimated convergence order $q^{(\text{est})}$)			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.8651e+03	1.8503e+03	1.8415e+03
$32 \times 32 \times 128$	1.7120e+03 (0.124)	1.6974e+03 (0.124)	1.6951e+03 (0.120)
$64 \times 64 \times 256$	1.4537e+03 (0.236)	1.4531e+03 (0.224)	1.4529e+03 (0.222)
$128 \times 128 \times 512$	9.6843e+02 (0.586)	9.6832e+02 (0.586)	9.6829e+02 (0.585)

- Heuristic arguments and computational evidence indicate $\|u - u_h\|_{L^2(\Omega)} \leq C\, h^q$ with $q = 2 - d/2$ for $d = 1, 2, 3$ on $\Omega \subset \mathbb{R}^d$, that is, $\|u - u_h\|_{L^2(\Omega)} \leq C\, h^{1/2}$ on $\Omega \subset \mathbb{R}^3$
  [Hanhart, Gobbert, and Izu, *JCAM* 2004; Gobbert, *SISC* 2008]
- **Rigorous proof available now!**
  [Seidman, Gobbert, Trott, and Kružík, *Numer. Math.*, submitted]

Scalar Test Problem with Non-Smooth Source Term

Test in Ω excluding the neighborhood Ω_0 of the CRU:

Error on $\Omega\backslash\Omega_0$ against reference solution (estimated convergence order $q^{(\text{est})}$)			
	$t=2$	$t=3$	$t=4$
$16\times16\times64$	2.6478e–01	9.8039e–01	2.2494e+00
$32\times32\times128$	1.2526e–01 (1.080)	3.7741e–01 (1.377)	6.6924e–01 (1.749)
$64\times64\times256$	3.7324e–02 (1.747)	1.1385e–01 (1.729)	1.8863e–01 (1.827)
$128\times128\times512$	8.0971e–03 (2.205)	2.3743e–02 (2.262)	3.8368e–02 (2.298)
Error in species mass (estimated convergence order $q^{(\text{est})}$)			
	$t=2$	$t=3$	$t=4$
$16\times16\times64$	1.4332e+00	3.6992e+00	4.4979e+00
$32\times32\times128$	1.2033e+00 (0.252)	1.2032e+00 (1.620)	1.2032e+00 (1.902)
$64\times64\times256$	3.1122e–01 (1.951)	3.1111e–01 (1.951)	3.1110e–01 (1.951)
$128\times128\times512$	7.8784e–02 (1.982)	7.8737e–02 (1.982)	7.8707e–02 (1.983)

- Convergence second-order in $\Omega\backslash\Omega_0$
- Mass is conserved throughout Ω

[Gobbert, *SISC* 2008]

Long-Time Simulation of Calcium Flow on High-Resolution Meshes

Long-Time Simulation of Calcium Flow: Results

Following original work [Izu et al. 1998], consider two cases for the amount of calcium injected per CRU:

$$\sigma = \begin{cases} 51.8213655\ \mu\text{M}\ \mu\text{m}^3\ /\ \text{ms} & \text{for } I_{\text{SR}} = 10\ \text{pA} \\ 103.6430533\ \mu\text{M}\ \mu\text{m}^3\ /\ \text{ms} & \text{for } I_{\text{SR}} = 20\ \text{pA} \end{cases}$$

in the model for the calcium release units (CRUs) as point sources

$$J_{\text{SR}}(u^{(0)}, \mathbf{x}, t) = \sum_{\hat{\mathbf{x}} \in \Omega_s} \sigma\, S_{\hat{\mathbf{x}}}(u^{(0)}, t)\, \delta(\mathbf{x} - \hat{\mathbf{x}})$$

$S_{\hat{\mathbf{x}}}(u^{(0)}, t) = 1$ for $t_{\text{open}} = 5$ ms with probability

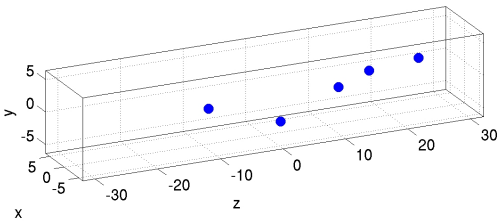
$$J_{\text{prob}}(u^{(0)}) = \frac{P_{\max}\,(u^{(0)})^{n_{\text{prob}}}}{(K_{\text{prob}})^{n_{\text{prob}}} + (u^{(0)})^{n_{\text{prob}}}}$$

then closed for $t_{\text{closed}} = 100$ ms.

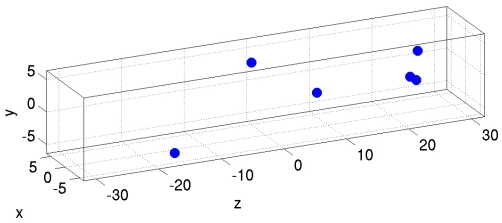
Movies available at `http://www.math.umbc.edu/~gobbert/calcium`

[Gobbert, *SISC* 2008]

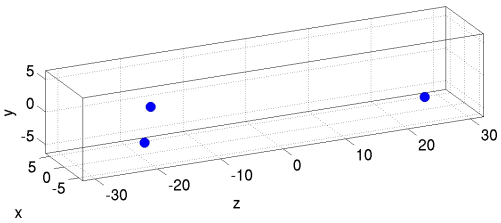
Calcium Flow: Open calcium release units for $\sigma = 51.82$



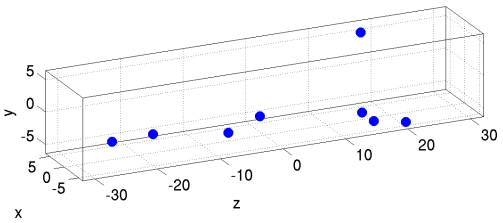
$t = 100$



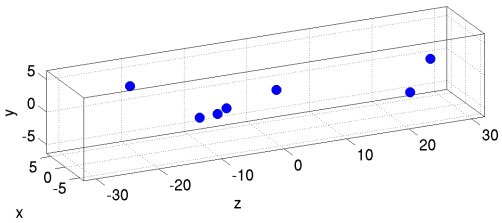
$t = 200$



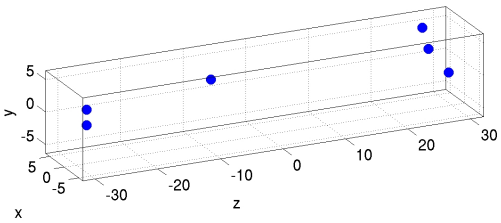
$t = 300$



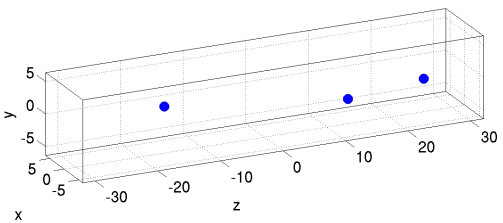
$t = 400$



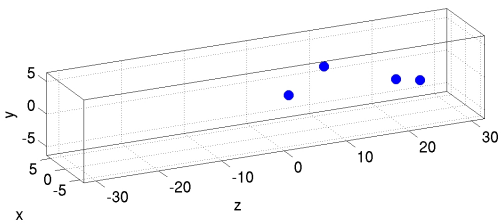
$t = 500$



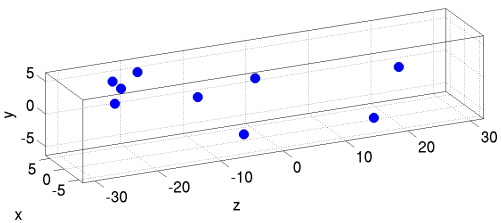
$t = 600$



$t = 700$

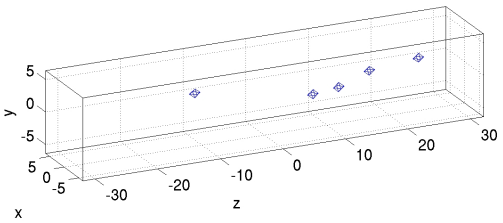


$t = 800$

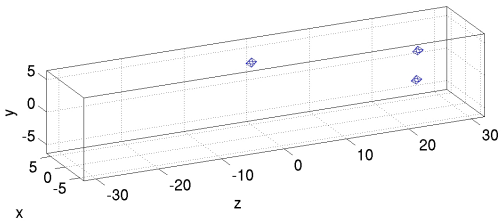


$t = 900$

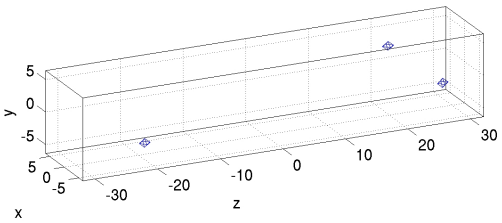
Calcium Flow: Calcium concentration for $\sigma = 51.82$



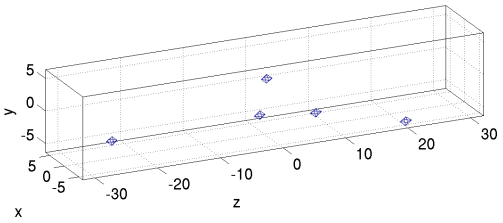
$t = 100$



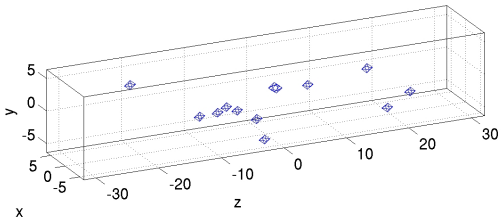
$t = 200$



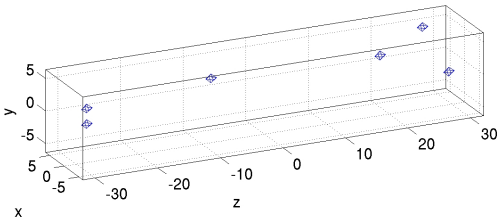
$t = 300$



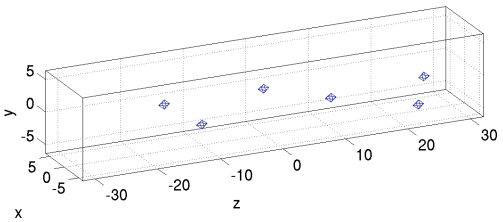
$t = 400$



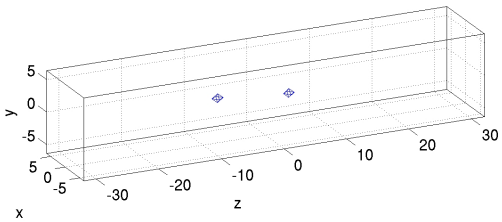
$t = 500$



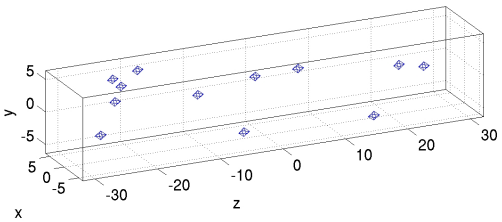
$t = 600$



$t = 700$

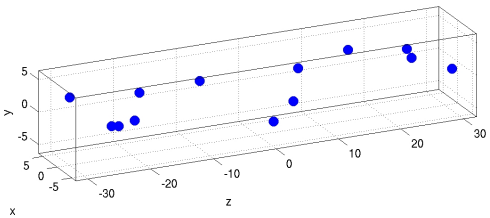


$t = 800$

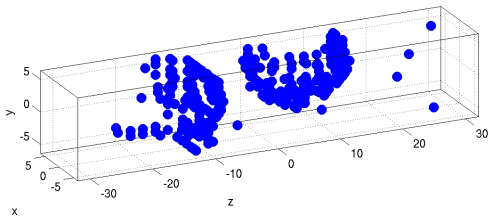


$t = 900$

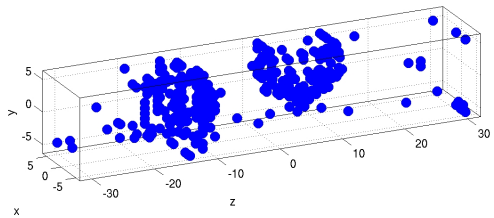
Calcium Flow: Open calcium release units for $\sigma = 103.64$



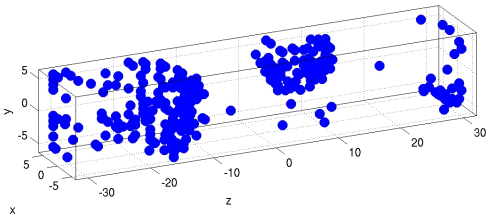
$t = 100$



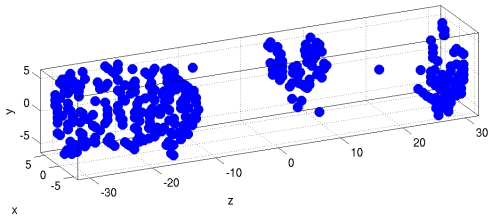
$t = 200$



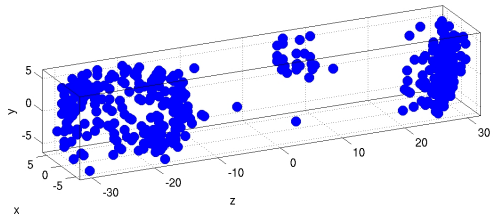
$t = 300$



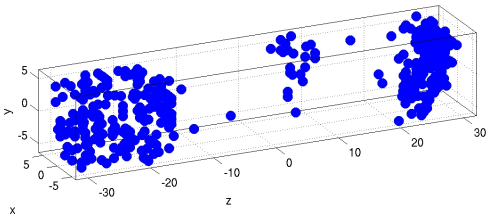
$t = 400$



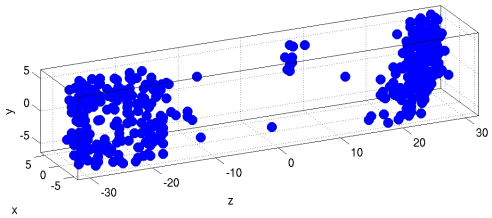
$t = 500$



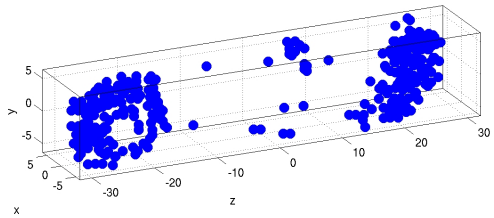
$t = 600$



$t = 700$

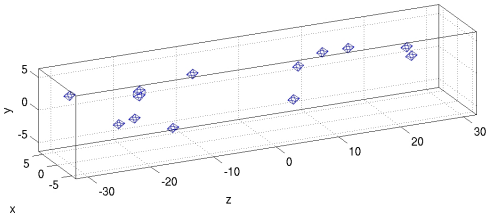


$t = 800$

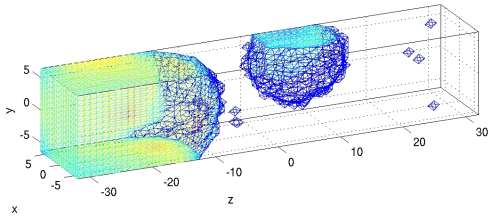


$t = 900$

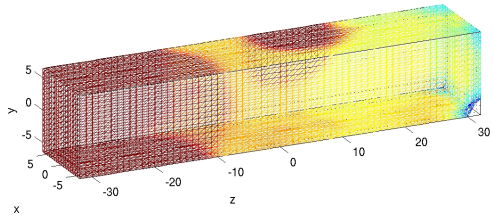
Calcium Flow: Calcium concentration for $\sigma = 103.64$



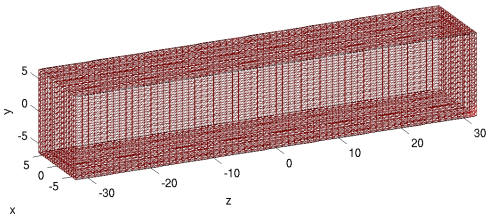
$t = 100$



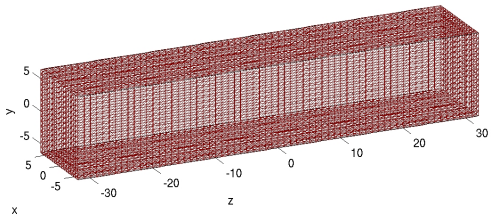
$t = 200$



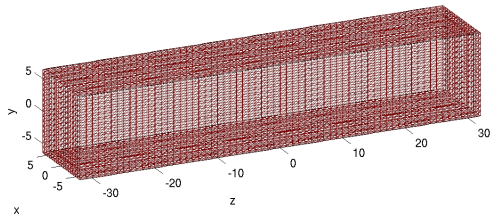
$t = 300$



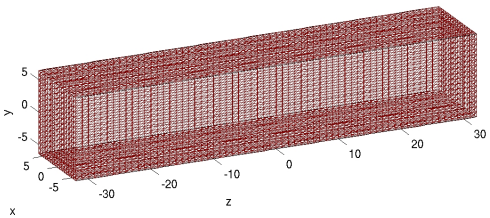
$t = 400$



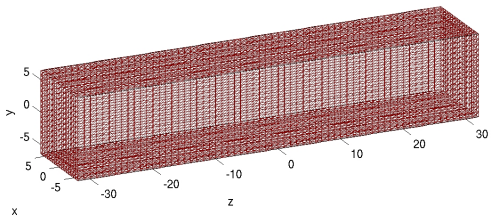
$t = 500$



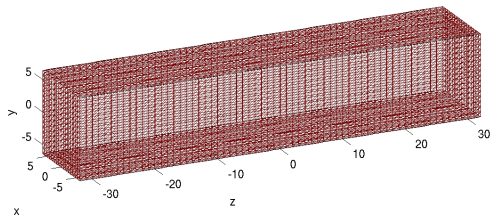
$t = 600$



$t = 700$



$t = 800$

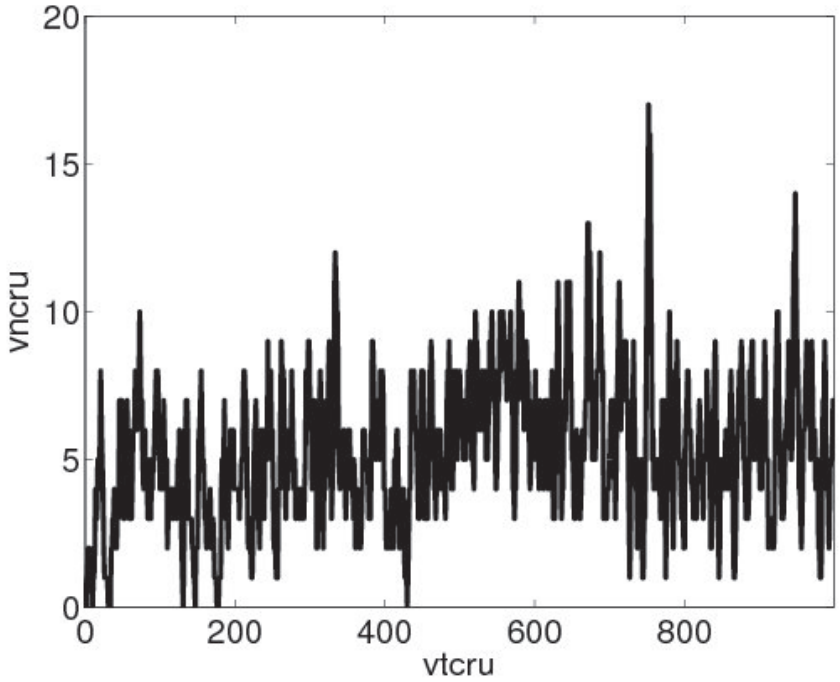


$t = 900$

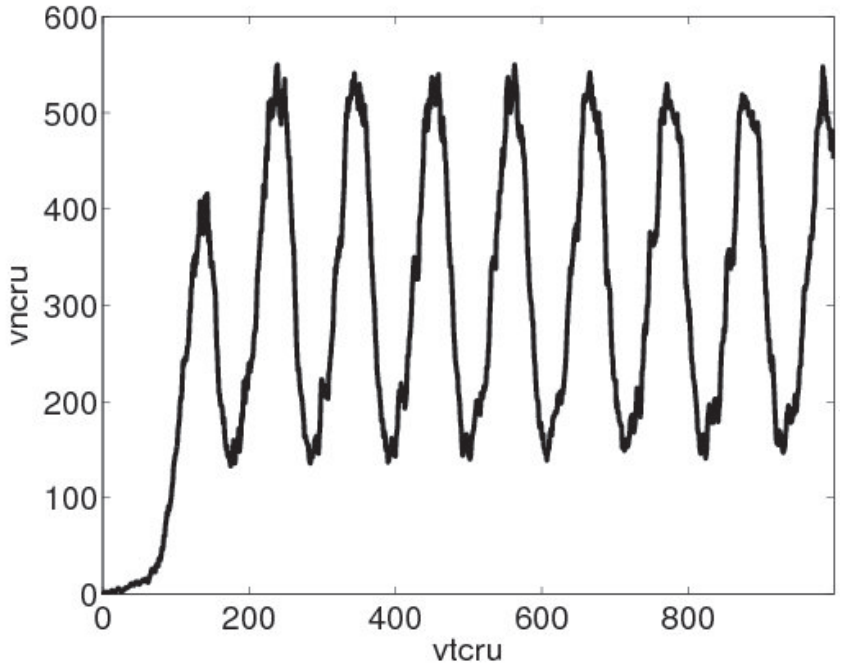
Contrast General-Purpose / Special-Purpose Code

MPSalsa, Sandia NL	This code SPARK
linear finite elements	linear finite elements
unstructured mesh	structured mesh
fully implicit time-stepping	fully implicit time-stepping
trapezoidal rule (BDF2)	NDFk, $1 \leq k \leq 5$,
with fixed step size	with variable step size
with fixed order $k = 2$	with mean$(k) \approx 2.9$
Newton	Newton with anal. Jac. and ≥ 0
(1 to 2 iterations)	(1 or 2 iterations)
GMRES with ILU pre-cond.	QMR without pre-cond.
(< 10 iter.)	(≈ 4 iter.)
cell cross-section 2 z-planes	entire cell (31 z-planes)
about 200,000 DOF	about 3 million DOF (*matrix-free*)
$t_{\text{fin}} = 100$ ms, 10,000 steps	$t_{\text{fin}} = 1{,}000$ ms, $\approx 58{,}000$ steps
"12 to 36 hours on	40 hours on
4 to 16" dual-processor nodes	16 dual-processor nodes
[Izu et al., *Biophys. J.* 2006]	[Gobbert, *SISC* 2008]

Number of Open CRUs vs. time



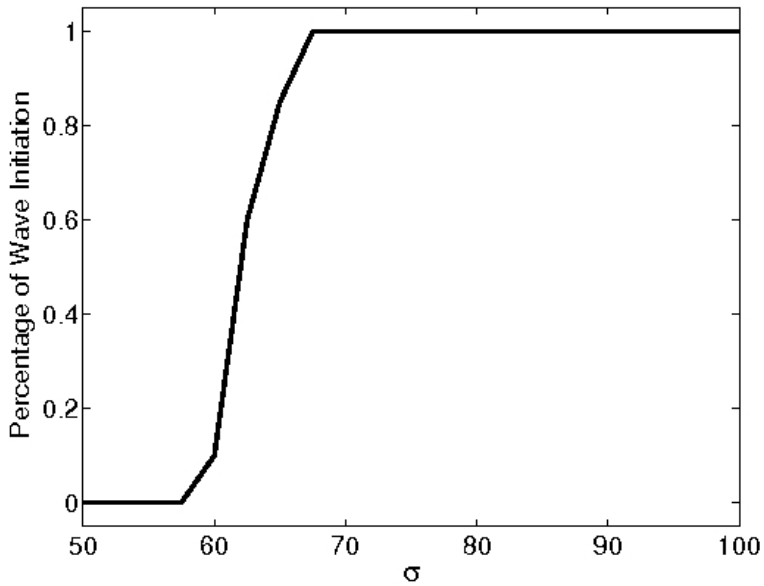
(a) $\sigma = 50$



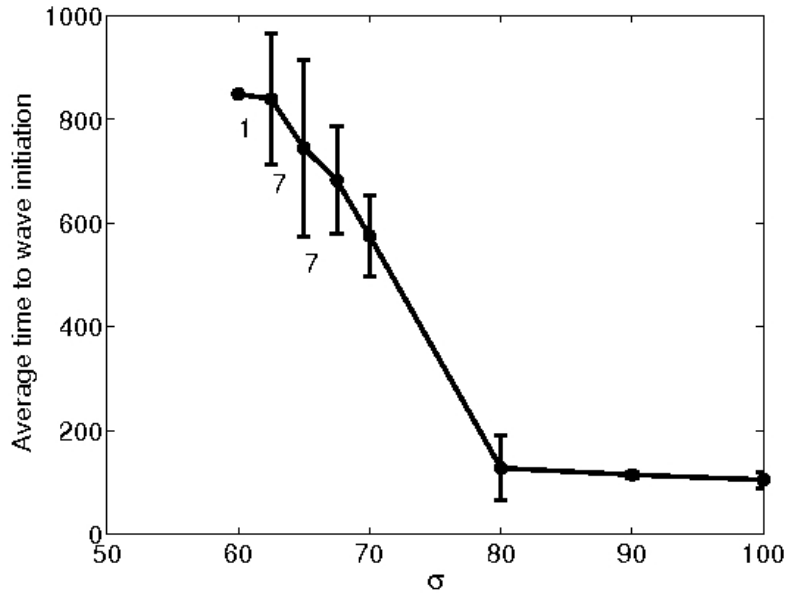
(b) $\sigma = 100$

- Total number of open CRUs throughout cell vs. time indicates whether wave self-initiates.
- Studies with values including $\sigma = 50, 60, 70, 80, 90, 100$ of the flux density narrow the range of its critical value.
- Coulibaly, Muscedere, Gobbert, and Peercy, HPCF–2009–6, `www.umbc.edu/hpcf`.

Wave Self-Initiation as Function of Flux Density



(a)



(b)

- (a) Fraction of simulation runs resulting in wave self-initiation for each flux density σ with 20 different seeds to the pseudo-random number generator.
- (b) Average time for wave self-initiation and its standard deviation as error bar for each σ.
- Coulibaly, Muscedere, Gobbert, and Peercy, HPCF–2009–6, 2009, `www.umbc.edu/hpcf`.

Summary

Previous work summary:

- Model allows waves to self-organize!
- wave self-organization is sensitive to σ and V_{pump},
- when wave self organize, calcium concentration increases without bound,
- model numerics are correct!

Conclusion:

- need to find the set of parameters generating biophysically acceptable behaviors.

Deterministic Firing Case

By defining $P_{\max} = 1$ and let $n_{\text{prob}} \to \infty$ and obtain

$$J_{\text{prob}}(c) := \begin{cases} 0 & \text{if } c < K_{\text{prob}}, \\ 1/2 & \text{if } c = K_{\text{prob}}, \\ 1 & \text{if } c > K_{\text{prob}}, \end{cases}$$

Model of Calcium Release:

$$J_{\text{SR}}(c, \mathbf{x}, t) := \begin{cases} 0 & \text{if } c \leq K_{\text{prob}}, \\ \sigma\delta(\mathbf{x} - \hat{\mathbf{x}}) & \text{if } c > K_{\text{prob}}. \end{cases}$$

Combining both expressions gives:

$$J_{\text{SR}}(c, \mathbf{x}, t) := \sigma H(c - K_{\text{prob}}) \Big[H(t - T^m) - H(t - T^m - \Delta_{\text{open}}) \Big] \delta(\mathbf{x} - \hat{\mathbf{x}}),$$

Δ_{open} = how long CRU stays opened. T^m are release times defined by:

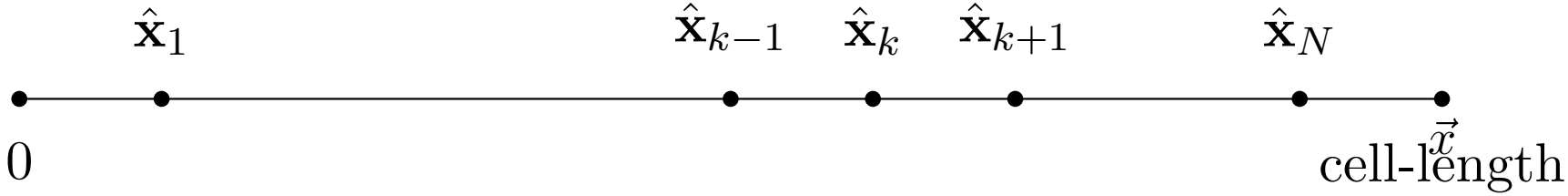
$$T^m = \{\inf t \mid c > K_{\text{prob}}, \ \frac{\partial c}{\partial t} > 0; \ t \geq T^{m-1}\}, \ m = 0, \ 1, \ \ldots,$$

[Coulibaly, Peercy, and Gobbert, Insight Into Spontaneous Recurrent Calcium Waves in a 3-D Cardiac Cell Based on Analysis of a 1-D Deterministic Model, in preparation]

We consider the 1D equation involving only one species:

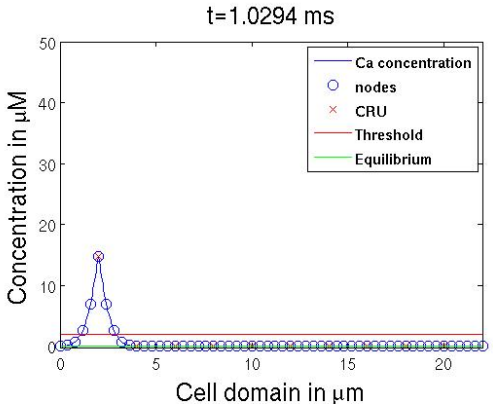
$$c_t = \overbrace{D_c c_{xx}}^{\text{diffusion}} - \overbrace{J_{\text{pump}}}^{\text{loss}} + \overbrace{J_{\text{leak}} + J_{\text{CRU}}}^{\text{creation}},$$

$t \in [0, t_{\text{fin}}]$, $x \in [0, \text{cell-length}]$, $c_x(0) = c_x(\text{cell-length}) = 0$.

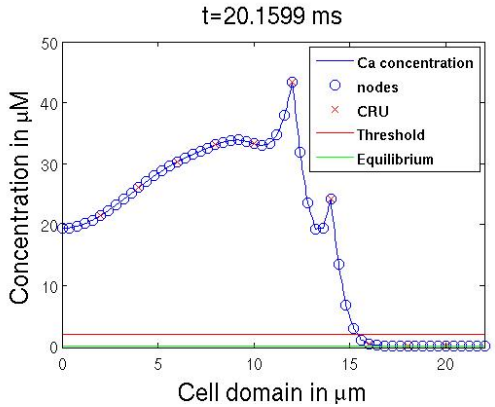


1D simulation details & advantages:

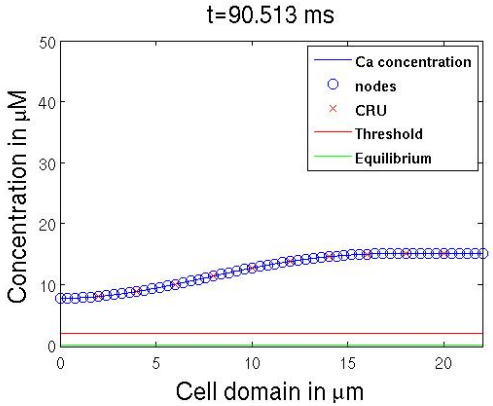
- uses FEM with linear basis functions (motivated by presence of Dirac Delta function),
- no stochasticity (only one run per parameter set is needed),
- characterization of wave phenomena in the system can be automated.



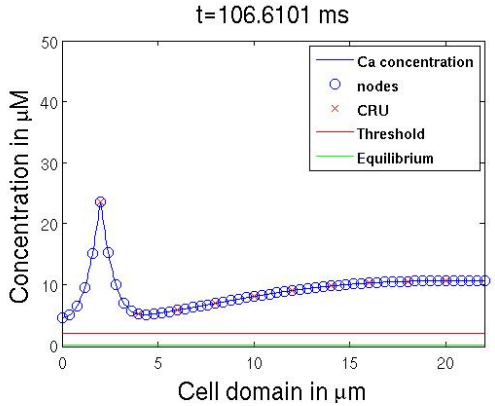
(a)



(b)

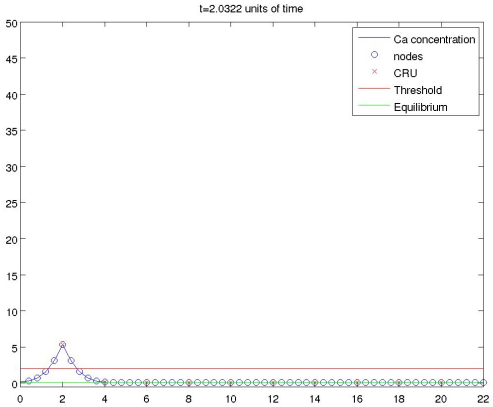


(c)

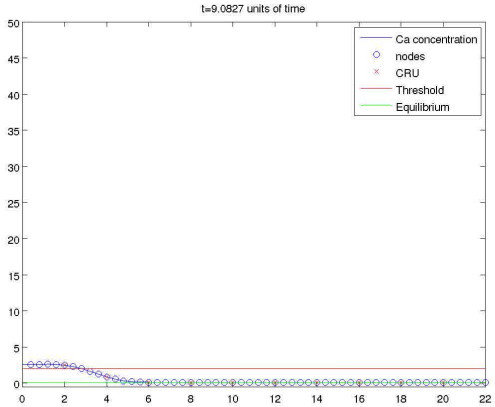


(d)

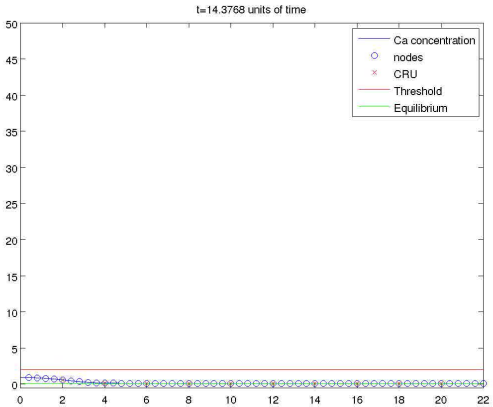
Figures (a–d) show instances of a simulation run resulting in a wave followed by an increase of calcium concentration without bound (similar to 3D case). $V_{\text{pump}} = 0.3, \sigma = 15$.



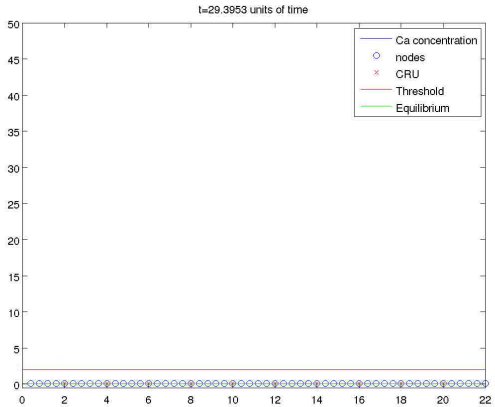
(4.a)



(4.b)

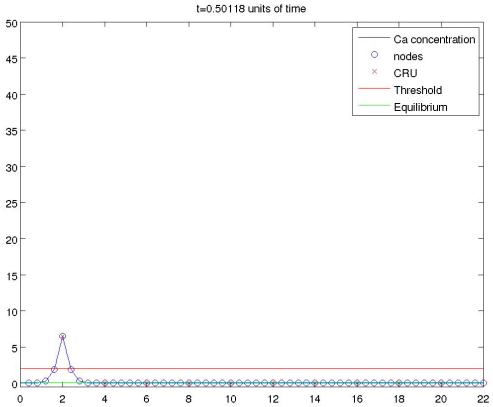


(4.c)

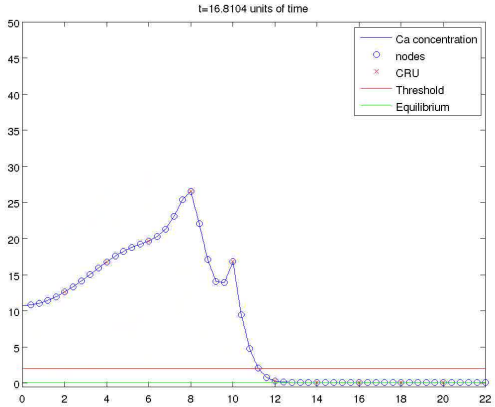


(4.d)

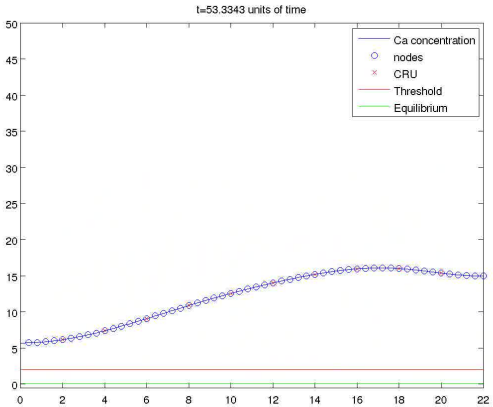
Figures (4.a–4.d) show instances of a simulation run resulting in no wave (similar to 3D case).$V_{\text{pump}} = 0.3, \sigma = 2$.



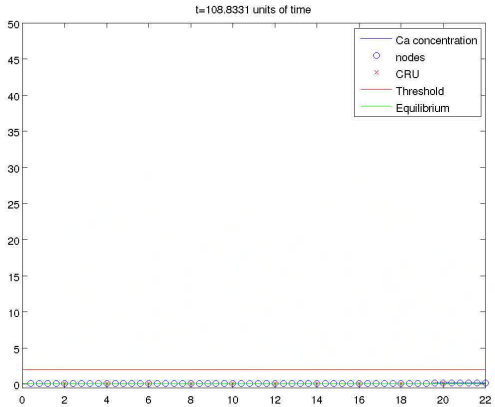
(a)



(b)

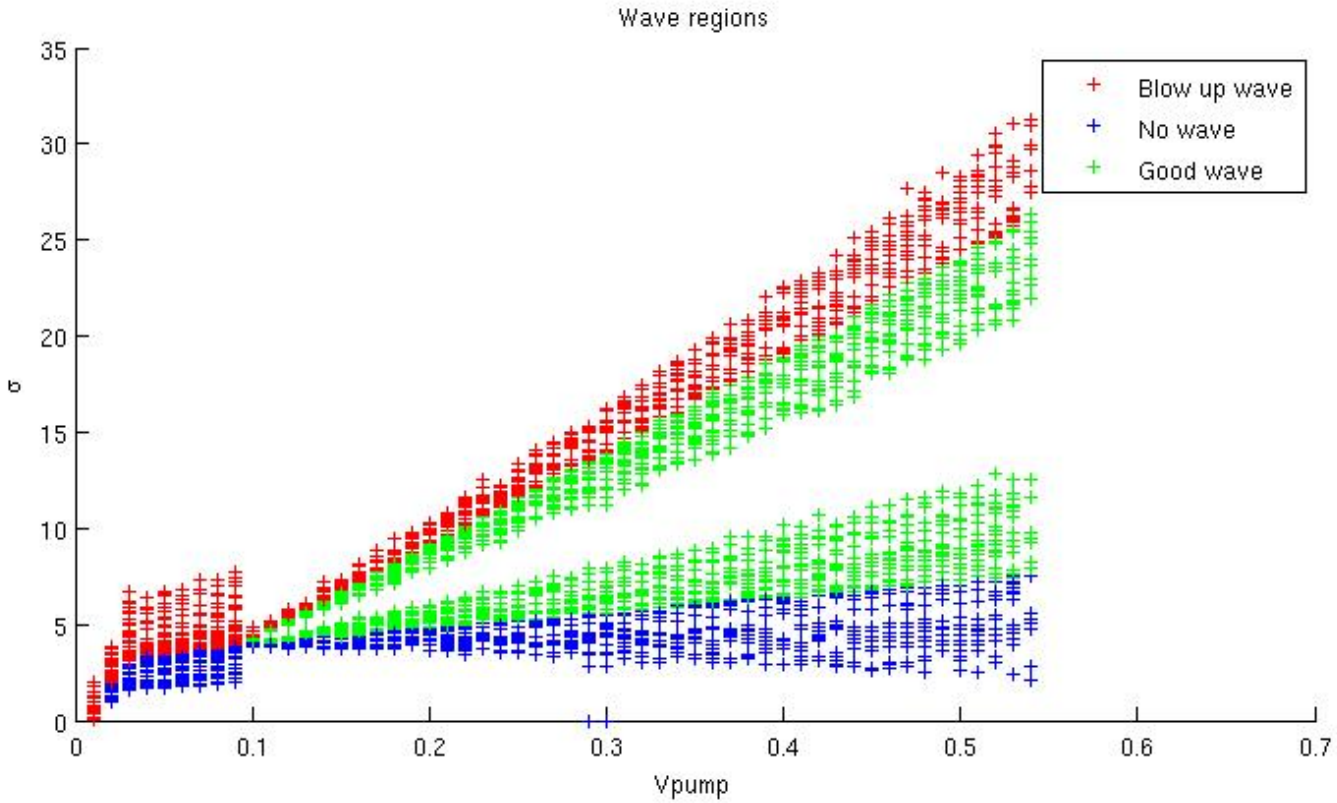


(c)

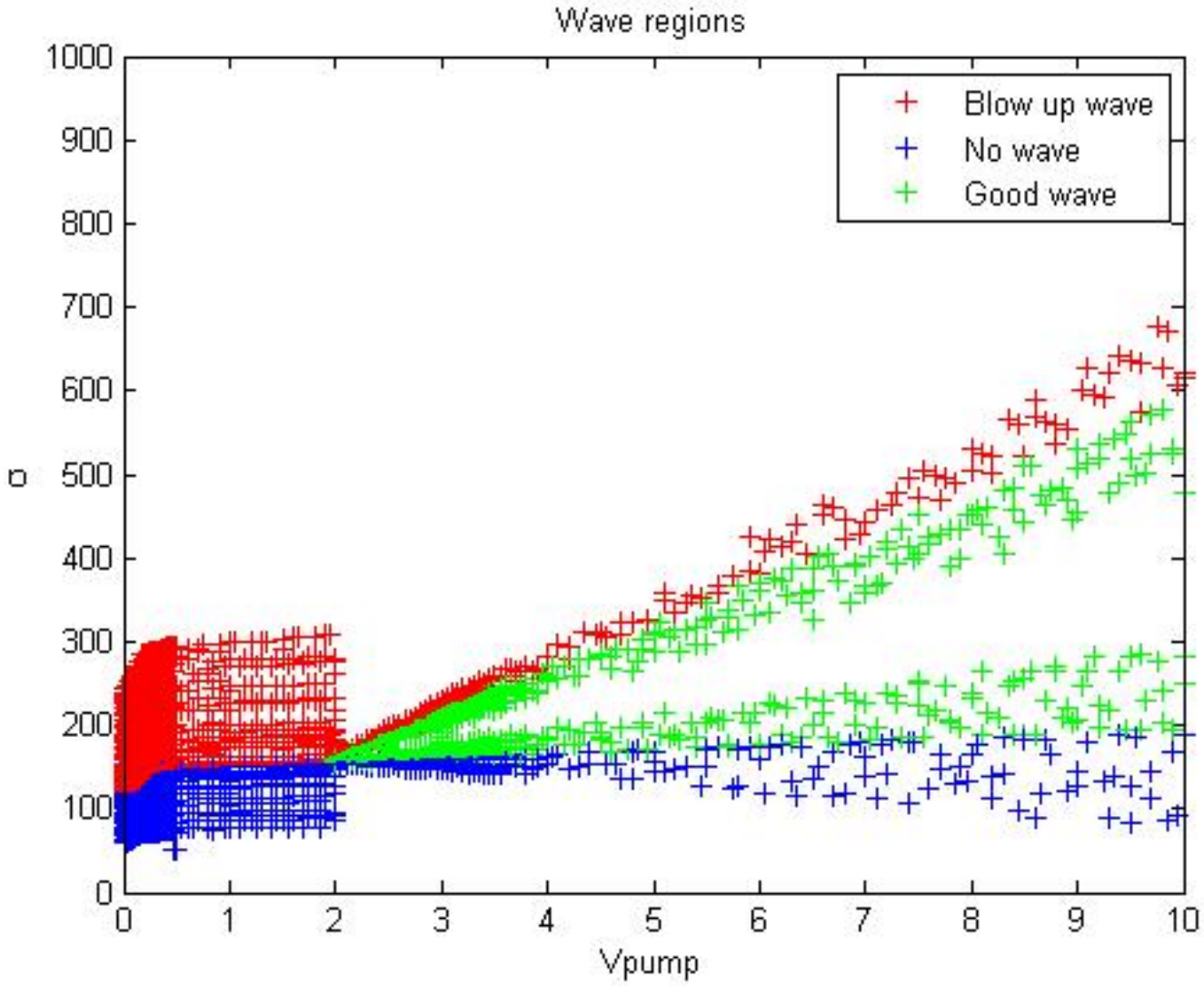


(d)

Figures (a–d) show instances of a simulation run resulting in a wave propagation followed by a recovery (no calcium increase without bound). $V_{\text{pump}} = 0.3, \sigma = 10$.



Wave region obtained from parameter study in the one dimensional-single-species version of the model.



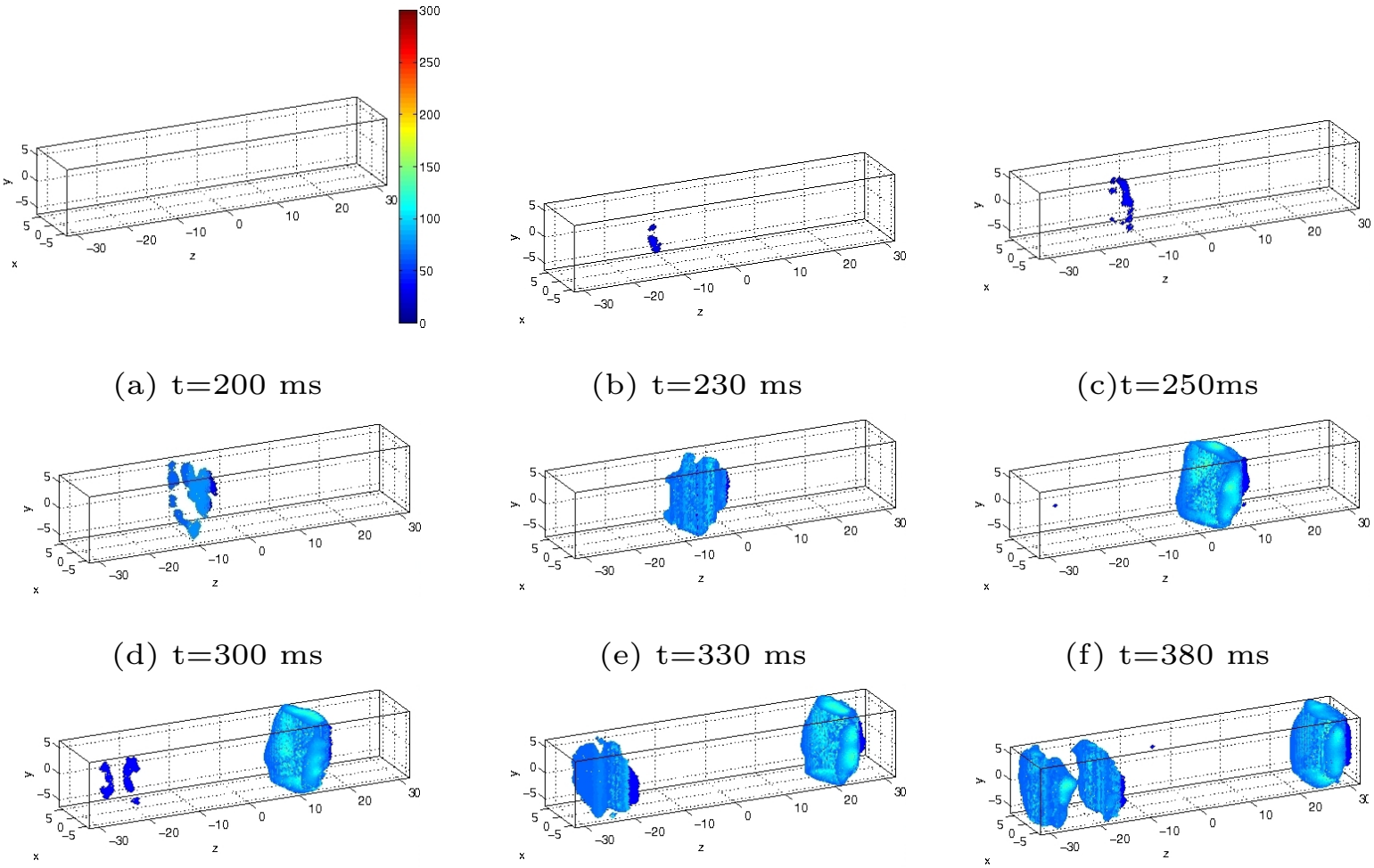
Wave region obtained from parameter study in the one-dimensional version of the model with three species. Notice the shift of different regions to the right.

1-D Deterministic Model

Previous Parameter search in 3-D:

- $V_{\text{pump}} \in \{0.2, 1.6, 3.2, 6.4\}$,
- $\sigma = 70 \pm \epsilon$.
- resulted in no wave propagation or increase of calcium concentration.

1D results suggest an increase in both V_{pump} and σ.
Example: $V_{\text{pump}} = 8$, $\sigma = 200$.
[Coulibaly, Peercy, and Gobbert, Insight Into Spontaneous Recurrent Calcium Waves in a 3-D Cardiac Cell Based on Analysis of a 1-D Deterministic Model, in preparation]

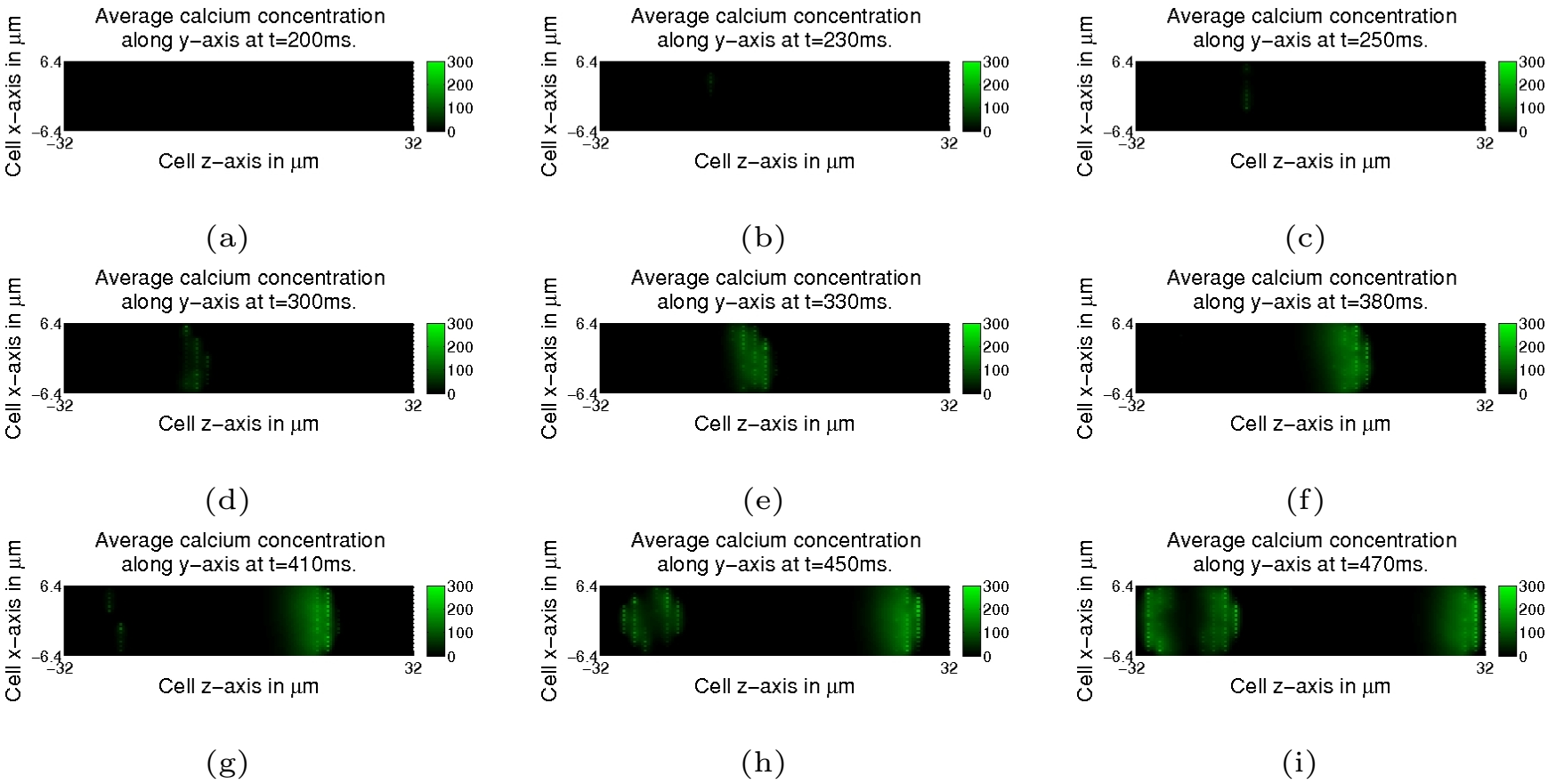


(a) t=200 ms (b) t=230 ms (c)t=250ms

(d) t=300 ms (e) t=330 ms (f) t=380 ms

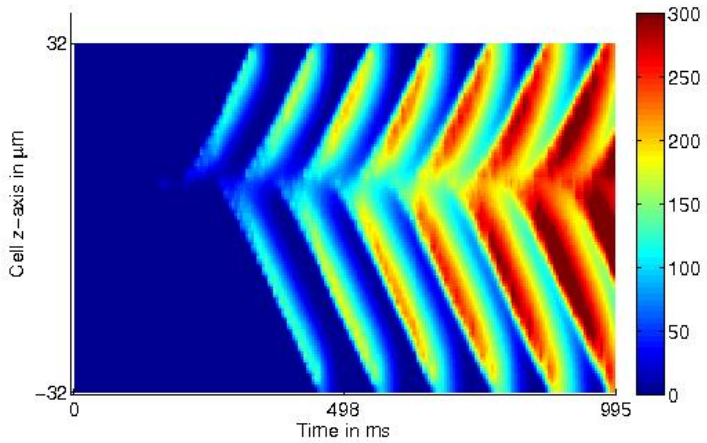
(g) t=410 ms (h) t=450 ms (i) t=470 ms

Isosurface plots of the calcium concentration throughout the 3 dimensional cell domain; critical isolevel of 65 μM. $V_{\text{pump}} = 8$ μM/ms, $\sigma = 200$ μM μm^3/ms.

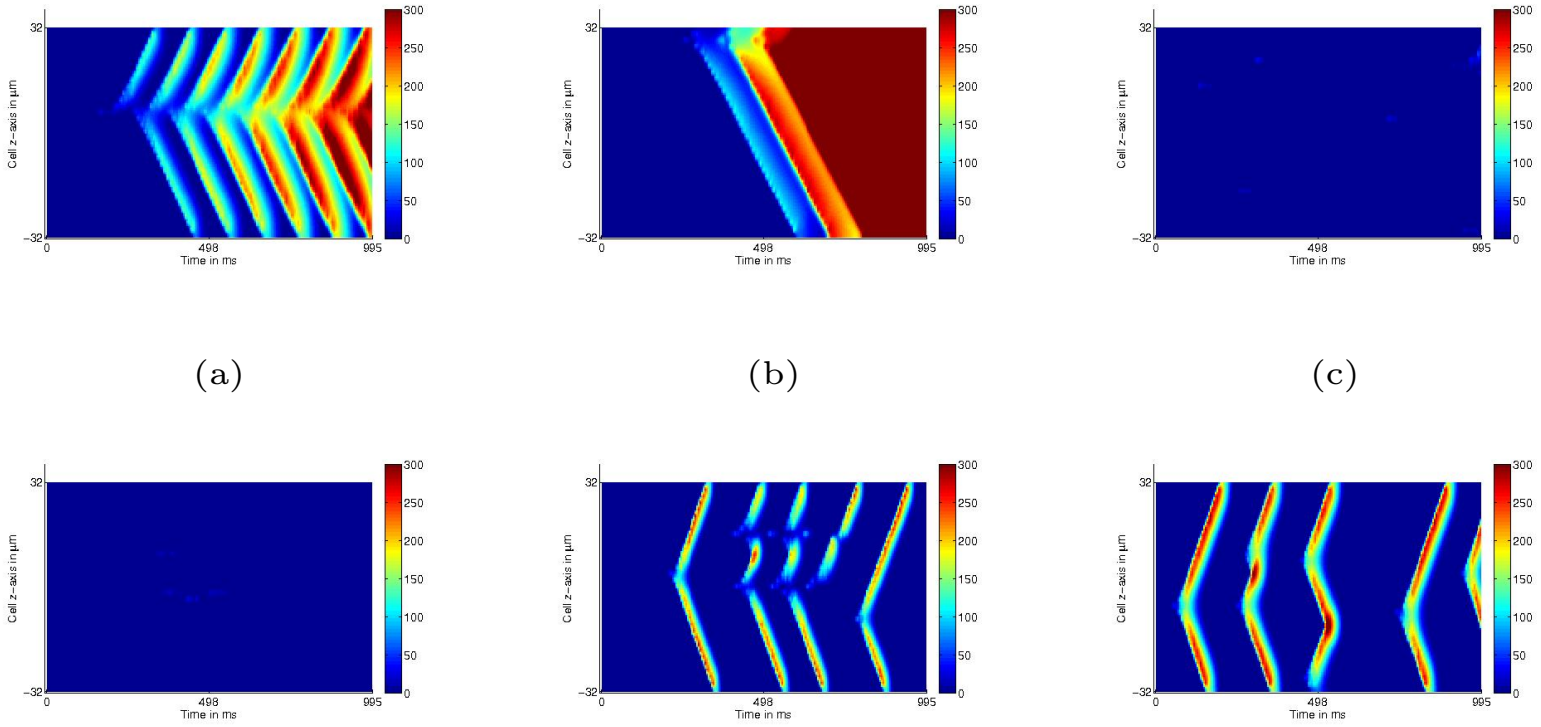


Confocal images (post-processing producing output similar to experiments) from run resulting in a wave propagation with recovery. The run is the same as the one in the previous slide.

Line-scans summarize (in a single figure) the overall behavior of the simulation; the horizontal axis corresponds to time (from 0 ms to 1000 ms), while the vertical axis corresponds to the longitudinal axis taken from the cell domain. Each line-scan plots calcium concentration (using a color-bar) at various times of the simulation along points of the longitudinal axis.



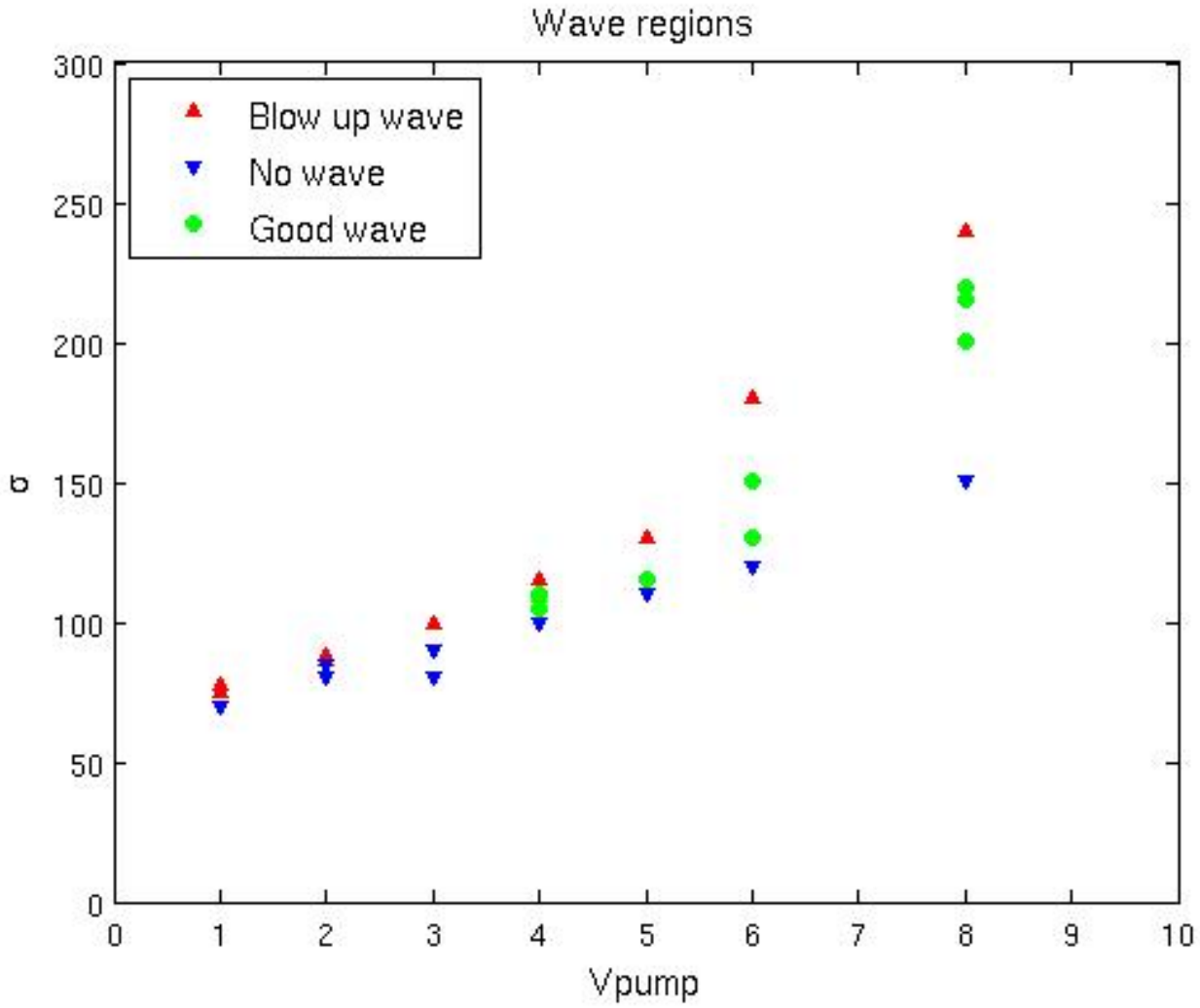
Sample line-scan of a simulation resulting in an increase of calcium concentration without bond.



(a) (b) (c)

(d) (e) (f)

Figures (a) and (b) line-scans from a blow up simulation. Figures (c) and (d) line-scans from a simulation resulting in no wave. Figures (e) and (f) line-scans from a simulation resulting in a wave propagation with recovery.



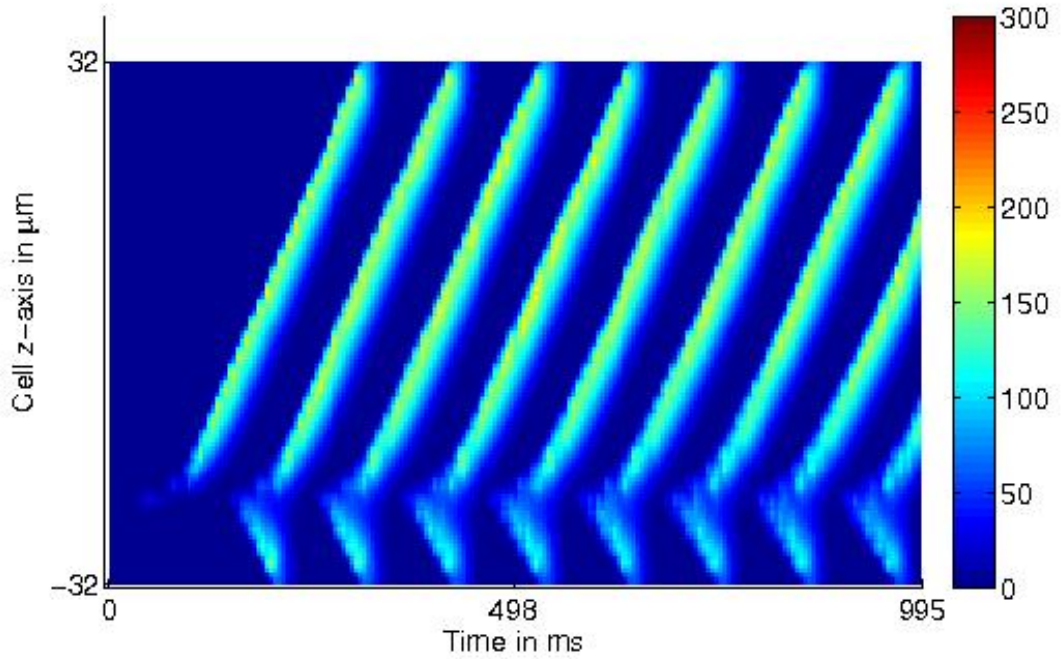
Wave region obtained from parameter study in the 3 dimensional stochastic model with buffers. V_{pump} is measured in μM/ms and σ is measured in μM μm^3/ms.

- 1D Deterministic parameter search allowed us to have a better *feel* about 3D parameter sets,
- 3D model, with the correct set of a parameter, generates spontaneous recurrent calcium waves with recovery!

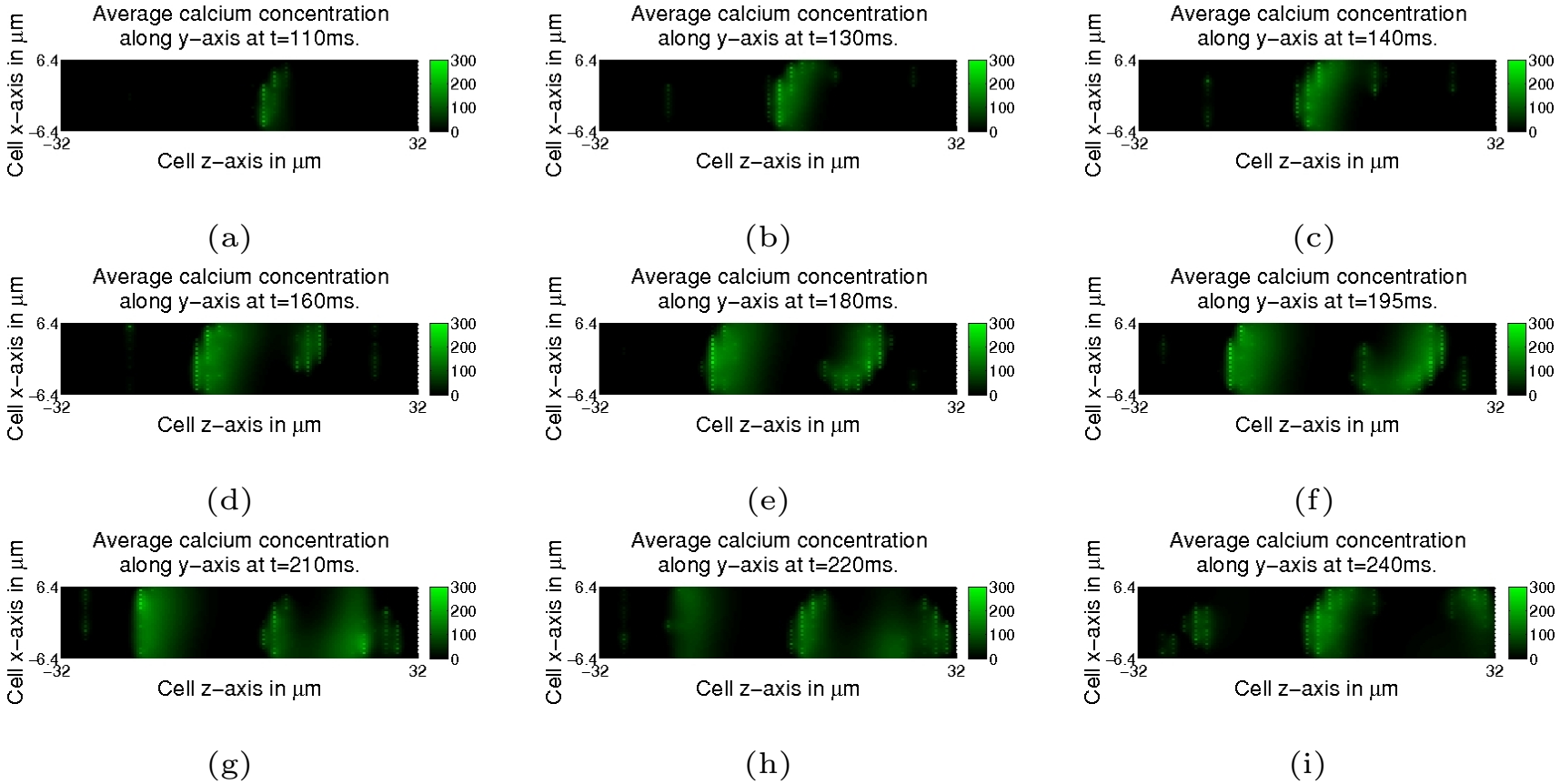
[Coulibaly, Peercy, and Gobbert, Insight Into Spontaneous Recurrent Calcium Waves in a 3-D Cardiac Cell Based on Analysis of a 1-D Deterministic Model, in preparation]

Spontaneous Spiral Waves

Spiral waves are phenomenon that can be seen experimentally. From simulation run observation, it is possible with the presented model to produce sustainable spiral wave by picking V_{pump} and σ in the "no wave"/"wave" region. Example: $V_{\text{pump}} = 4$ and $\sigma = 110$.

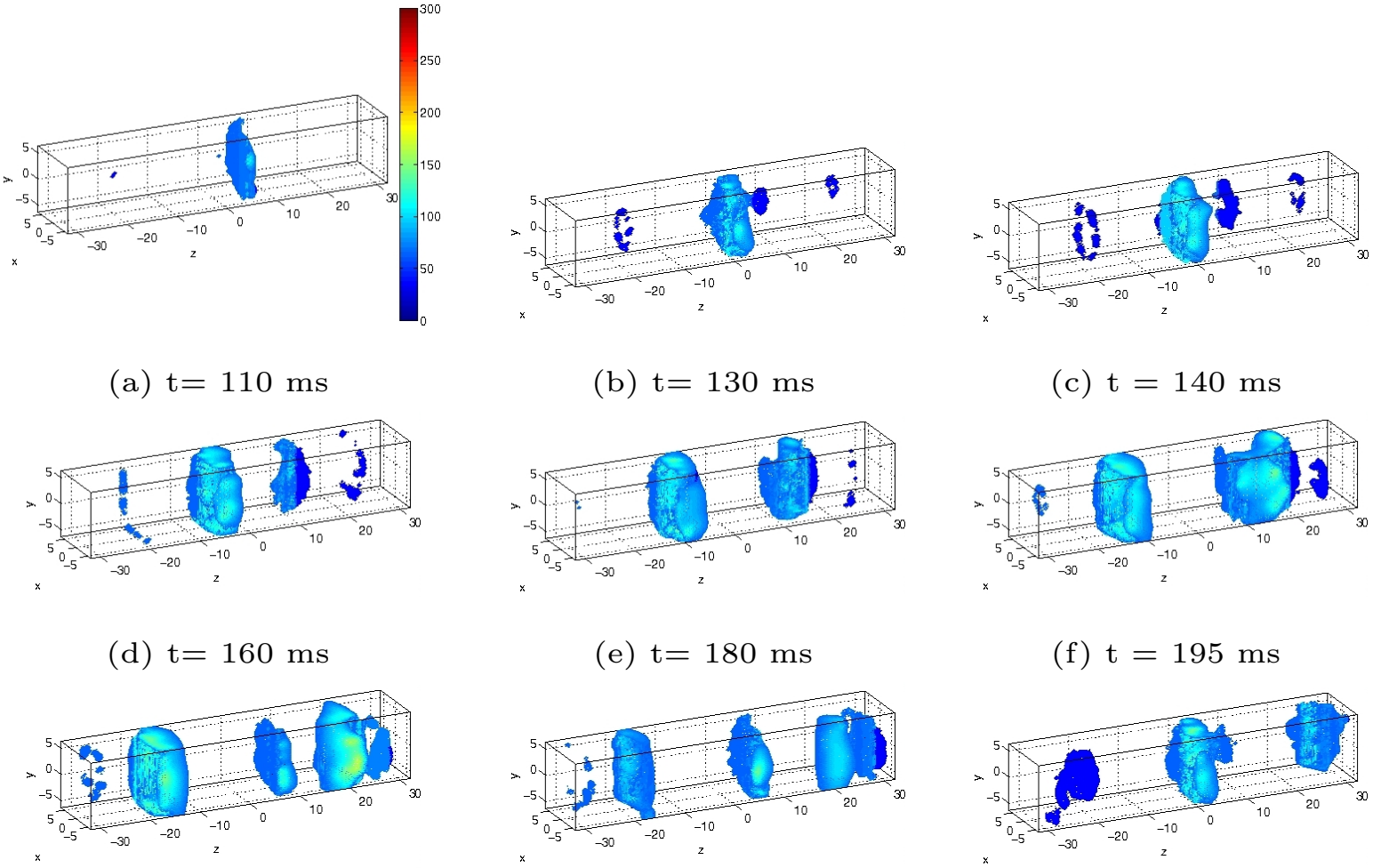


Line-scan from a simulation resulting in a spiral wave



Confocal images (slice of a cell) from a simulation run resulting in a spiral. The repeating spiraling behavior of the wave occurs throughout the duration of the simulation.

[Coulibaly, Peercy, and Gobbert, Spontaneous Spiral Wave Initiation in a 3-D Cardiac Cell, in preparation]



(a) t= 110 ms (b) t= 130 ms (c) t = 140 ms

(d) t= 160 ms (e) t= 180 ms (f) t = 195 ms

(g) t= 210 ms (h) t = 220 ms (i) t = 240 ms

Isosurface plots of the calcium concentration throughout the 3 dimensional cell domain; critical isolevel of 65 μM. Eye of the spiral is clearly visible.

[Coulibaly, Peercy, and Gobbert, Spontaneous Spiral Wave Initiation in a 3-D Cardiac Cell,

The Distributed-Memory Cluster tara and Parallel Performance

UMBC High Performance Computing Facility (HPCF)

The community-based, interdisciplinary core facility for scientific computing and research on parallel algorithms at UMBC

- Initiated in 2008 with participation of over 20 faculty from more than 10 departments and research centers from all three colleges at UMBC
- Summer 2008: 35-node cluster (two dual-core AMD Opteron) with InfiniBand (dual data rate DDR) and 14 TB central storage
- Replacement in Summer 2009: 86-node cluster (two quad-core Intel Nehalem) with new InfiniBand (quad data rate QDR) and 160 TB central storage
- Original 32 nodes purchased with seed funding from UMBC and contributions from faculty; extension partially funded by NSF grants (MRI program $200,000, SCREMS program for Math & Stat $40,000) and contributions from faculty and UMBC
- System administration by Division of Information Technology; user support by HPCF RAs in collaboration with CIRC
- Governed by HPCF Governance Committee; point of contact: Matthias K. Gobbert, gobbert@umbc.edu, 410–455–2404
- All details, list of projects and publications, user info at `www.umbc.edu/hpcf`.

History of the Clusters in HPCF

- 2003 — kali: $150,000
  33-node cluster (two single-core Intel Xeon)
  with Myrinet and 0.5 TB central storage
  funding: $75k NSF SCREMS, $75k UMBC cost-sharing, discount from IBM
- 2008 — hpc: $270,000
  35-node cluster (two dual-core AMD Opteron)
  with InfiniBand (dual data rate DDR) and 14 TB central storage
  funding: $100k seed money UMBC, $70k faculty, $100,000 DoIT
- 2009 — tara: $600,000
  86-node cluster (two quad-core Intel Nehalem)
  with new InfiniBand (quad data rate QDR) and 160 TB central storage
  funding: $360k faculty, $240k NSF MRI and SCREMS
- 2012 — extension of tara intended: $500,000 estimated
  42 hybrid nodes (two eight-core Intel Nehalem, two NVIDIA GPGPUs)
  with InfiniBand (quad data rate QDR)
  funding: to be determined, suggestions welcome!

$\longrightarrow$ all purchased / to be purchased from IBM!

User and Governance Committees

- User Committee:
  - Matthias Gobbert (Math/Stat),
  - Lynn Sparling (Physics),
  - Ian Thorpe (Chemistry);
  - UMBC DoIT system administration staff ex-officio
- HPCF Governance Committee:
  - Matthias Gobbert (Math/Stat),
  - Lynn Sparling (Physics),
  - Ian Thorpe (Chemistry),
  - Ray Hoff (Physics),
  - Curtis Menyuk (CSEE),
  - Larrabee Strow (Physics),
  - Marc Olano (CSEE)
- They represent over 120 long-term research users from over 35 research groups and additionally over 40 short-term users (REU Site, students in a class, etc.).

Detailed Hardware Specifications of Cluster tara

86-node distributed-memory cluster, IBM Server x iDataPlex

- 82 compute and 2 development nodes, 1 user node, and 1 management node:
  - two quad-core Intel Nehalem X5550 (2.6 GHz, 8 MB cache per core)
  - 24 GB memory
- Networks connecting all components:
  - high performance (low latency, wide bandwidth) quad-data rate (QDR) InfiniBand (IB) interconnect for parallel communications
  - ethernet for management and login
- 160 TB central storage with partition for each research group, connected through InfiniBand

For more information, see the webpage `www.umbc.edu/hpcf` of the UMBC High Performance Computing Facility (HPCF).

Tara: Overall System (Both Racks and Central Storage)

Tara: iDataPlex Rack and Detail

Tara: Front and Back of the InfiniBand Switch

How to Program a Distributed-Memory Cluster

- Memory is distributed across nodes and only accessible by local CPUs
- Total memory in compute nodes: 82×24 GB $= 1{,}968$ GB
- The processors have their own cache, but share the memory of a node, therefore:
  - Should one use both processors per node or only one? (A traditional observation is that best relative performance is achieved by using only one! [PETSc])
  - Should one use all cores per processor?
- Algorithm design: Divide problem into pieces with as little dependence on each other as possible, then program communication explicitly using MPI (Message Passing Interface) $\Rightarrow$ fully portable code.
- Typical PDE problems:
  - domain split $\Rightarrow$ communication of solution values on interfaces (lower-dimensional region)
  - communication at every time-step / in every iteration

Contrast to Other Types of Parallel Architectures

- Shared-memory machines: data is in (apparently) one piece of memory and is accessible from all (often, special-purpose) processors ⇒ issues of consistency (of memory and cache)
- Often, specialized vendor-directives in code needed. Often, more suited to data-parallelism.
- For suitable problems (with regular structure), shared-memory machines should in principle be faster. But: substantially more expensive.
- Beowulf clusters built from commodity hardware are an affordable alternative ⇒ better ROI (more bang for the buck).
- But inherently harder to work with (algorithm design, MPI coding, setup of hardware and software, choice of components)
  ⇒ **Challenging research topic!**

Performance Measures for Parallel Computing

- Speedup $S_p(N)$: How much faster are p processors over 1 processor (for problem of fixed size N)? Optimal value: $S_p(N) = p$.
- Efficiency $E_p(N)$: How close to optimal is the speedup?
  Optimal value: $E_p(N) = 1 = 100\%$.

$$S_p(N) = \frac{T_1(N)}{T_p(N)}, \quad E_p(N) = \frac{S_p(N)}{p}$$

  $T_p(N) =$ wall clock time for problem of size N on p processors.
- Speedup and efficiency *for fixed problem size* are tough measures of parallel performance, because inevitably communication will eventually dominate (for truly parallel code).
- *Scaled* speedup and efficiency: Increase problem size N along with using more processors. Example: On p processors, solve a problem of size pN.
  $\Rightarrow$ Less tough measures of performance, realistic measure of the usefulness of parallel computing for memory-intensive applications.

Issue: What is $T_p(N)$?

- Parallel program spends time in calculations $T_{calc}(N)$ and in communications $T_{comm}(N)$; communication time is affected by latency (initialization of communication) and bandwidth (throughput capability). $\Rightarrow$ Coefficients in model are difficult to determine reliably.
- Fundamental problem of parallel computing: Communications hurt but are unavoidable (for truly parallel algorithm), hence we must include them in our timings. $\Rightarrow$ Wall clock time is used (not: CPU time).
- What wall clock time? Additional issues: OS delays, MPI/network startup, file access for input (1 file read by all processors) and output (all processors write a file, to where? central or local), etc.
- What is $T_1(N)$? Parallel code with 1 processor or serial code with same algorithm or serial code with different ("best known") algorithm. Example: Jacobi vs. Gauss-Seidel (or SOR) for linear solve.
- In summary, two ways to get good speedup: fast parallel code or slow serial timing (slow due to any reason); clearly only first method acceptable.

Parallel Performance Study for Elliptic Test Problem

Elliptic Problem

- Classical elliptic test problem of the Poisson equation with homogeneous Dirichlet boundary conditions
- Find the solution $u(x, y)$ on the unit square in two spatial dimensions $\Omega = (0, 1) \times (0, 1) \subset \mathbb{R}^2$, given

$$-\triangle u = f \quad \text{in } \Omega, \qquad \triangle \text{ is the Laplace operator}$$
$$u = 0 \quad \text{on } \partial\Omega \qquad f \text{ is a known function}$$

- Approximation by finite difference method results in a large, sparse, highly structured system of linear equations

$$-\triangle u(x_{k_1}, x_{k_2}) \approx \frac{u_{k_1-1,k_2} - 2u_{k_1,k_2} + u_{k_1+1,k_2}}{h^2} + \frac{u_{k_1,k_2-1} - 2u_{k_1,k_2} + u_{k_1,k_2+1}}{h^2}$$

- Concretely, must solve $Au = b$ where b are discretized values of f and

$$A = \begin{bmatrix} S & T & & & \\ T & S & T & & \\ & \ddots & \ddots & \ddots & \\ & & T & S & T \\ & & & T & S \end{bmatrix} \in \mathbb{R}^{N^2 \times N^2}$$

$$S = \text{tridiag}(-1, 4, -1) \in \mathbb{R}^{N \times N},$$
$$T = -I \in \mathbb{R}^{N \times N}$$

Elliptic Problem

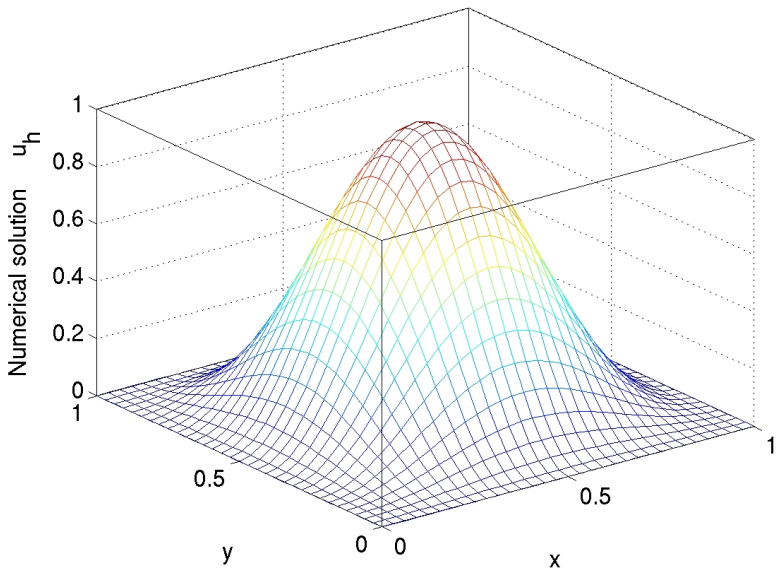
- Featured test problem has right-hand side function

$$f(x_1, x_2) = (-2\pi^2)\left(\cos(2\pi x_1)\,\sin^2(\pi x_2) + \sin^2(\pi x_1)\,\cos(2\pi x_2)\right),$$

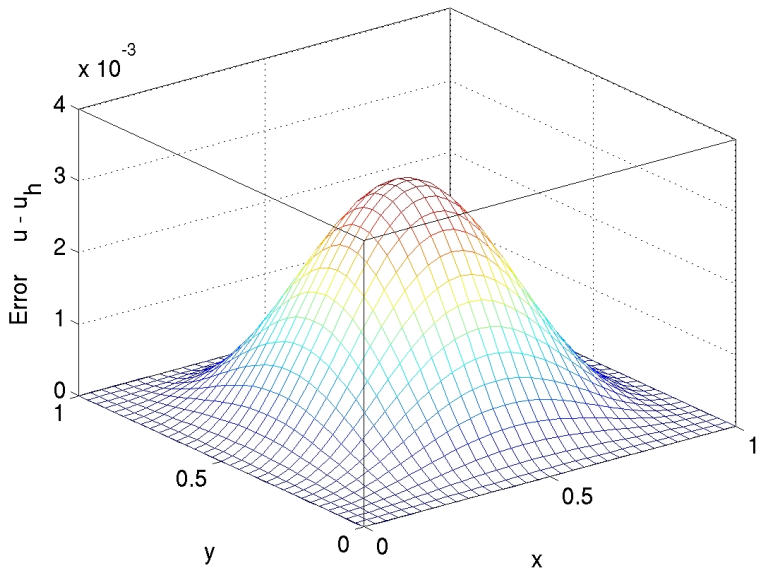
- This has a known solution

$$u(x_1, x_2) = \sin^2(\pi x_1)\,\sin^2(\pi x_2)$$

- On a mesh with 33×33 points and mesh spacing $h = 1/32 = 0.03125$, the numerical solution $u_h(x_1, x_2)$ can be plotted



(a) Numerical solution u_h



(b) Error $u - u_h$

Elliptic Application

- Finite difference mesh on unit square domain with $(N+2) \times (N+2)$ elements has N^2 degrees of freedom (DOF)
  $\Rightarrow 4N^2$ double-precision numbers are stored

N	DOF	$\|u - u_h\|_{L^\infty(\Omega)}$	#iter	memory usage (MB) predicted	memory usage (MB) observed
32	1,024	3.2189e–3	47	< 1	12
64	4,096	8.0356e–4	95	< 1	12
128	16,384	2.0081e–4	191	< 1	12
256	65,536	5.0191e–5	385	2	13
512	262,144	1.2543e–5	781	8	19
1,024	1,048,576	3.1327e–6	1,579	32	44
2,048	4,194,304	7.8097e–7	3,191	128	140
4,096	16,777,216	1.9356e–7	6,447	512	524
8,192	67,108,864	4.6817e–8	13,028	2,048	2,061
16,384	268,435,456	8.0469e–9	26,321	8,192	8,207
32,768	1,073,741,824	2.9562e–9	53,136	32,768	33,923

- Notice that serial run *not* possible for $N = 32{,}768$ on tara with 24 GB per node
  $\Rightarrow$ **Parallel computing enables the solution of a larger problem!**
- We implement the conjugate gradient method in parallel, to iteratively solve the given linear system
- This necessarily involves communications between processes every iteration
- Raim and Gobbert, HPCF–2010–2, 2010, `www.umbc.edu/hpcf`

tara: Elliptic Application Problem: Performance by Number of Nodes

(a) Mesh resolution $N \times N = 8{,}192 \times 8{,}192$, system dimension 67,108,864

	1 node	2 nodes	4 nodes	8 nodes	16 nodes	32 nodes	64 nodes
1 ppn	04:36:37	02:20:54	01:14:14	00:38:47	00:19:31	00:09:20	00:04:36
2 ppn	02:07:34	01:08:57	00:34:11	00:18:09	00:08:25	00:04:14	00:02:08
4 ppn	01:15:55	00:40:23	00:20:55	00:09:38	00:05:52	00:03:00	00:01:30
6 ppn	00:56:59	00:29:28	00:14:43	00:07:32	00:04:26	00:02:29	00:01:23
8 ppn	00:53:55	00:26:26	00:12:54	00:06:30	00:03:20	00:01:43	00:00:50

(b) Mesh resolution $N \times N = 16{,}384 \times 16{,}384$, system dimension 268,435,456

	1 node	2 nodes	4 nodes	8 nodes	16 nodes	32 nodes	64 nodes
1 ppn	33:57:41	19:44:19	09:54:27	05:01:30	02:34:15	01:17:16	00:37:23
2 ppn	16:21:30	08:31:15	04:31:04	02:23:42	01:09:54	00:34:00	00:17:01
4 ppn	10:03:34	05:01:41	02:41:11	01:24:53	00:47:29	00:22:45	00:11:43
6 ppn	08:20:03	04:04:07	02:02:50	01:02:55	00:32:32	00:17:35	00:08:59
8 ppn	07:07:54	03:39:54	01:57:19	00:56:47	00:26:50	00:13:44	00:07:04

(c) Mesh resolution $N \times N = 32{,}768 \times 32{,}768$, system dimension 1,073,741,824

	1 node	2 nodes	4 nodes	8 nodes	16 nodes	32 nodes	64 nodes
1 ppn	N/A	N/A	N/A	N/A	N/A	N/A	05:21:27
2 ppn	N/A	N/A	N/A	N/A	N/A	N/A	02:15:16
4 ppn	N/A	N/A	N/A	N/A	N/A	N/A	01:27:53
6 ppn	N/A	N/A	N/A	N/A	N/A	N/A	01:03:53
8 ppn	N/A	N/A	N/A	N/A	N/A	N/A	00:55:07

- Raim and Gobbert, HPCF–2010–2, 2010, `www.umbc.edu/hpcf`
- Optimal halving doesn't occur from ppn 2 to 4, or ppn 4 to 8, but improvement is still substantial
- $\Longrightarrow$ Use all cores per node to maximize overall cluster productivity!

Summary: Elliptic Test Problem

Poisson equation, finite difference discretization, conjugate gradient method, highly efficient matrix-free implementation, mesh resolution $N \times N = 4{,}096 \times 4{,}096$, system dimension 16,777,216.

Cluster (year)	serial (1 core) time	1 node all cores time (speedup)	32 node 1 core per node time (speedup)	32 node all cores time (speedup)
kali (2003)	02:00:49		00:04:05 (29.59)	00:04:49 (25.08)
hpc (2008)	01:51:29	00:32:37 (3.42)	00:03:23 (32.95)	00:01:28 (76.01)
tara (2009)	00:31:16	00:06:39 (4.70)	00:01:05 (28.86)	00:00:09 (208.44)

- kali (2003): two (single-core) Intel Xeon (2.0 GHz) per node
  $\Rightarrow$ 2 cores per node; Myrinet interconnect
- hpc (2008): two dual-core AMD Opteron (2.67 GHz) per node
  $\Rightarrow$ 4 cores per node; InfiniBand (DDR) interconnect
- tara (2009): two quad-core Intel Nehalem X5550 (2.67 GHz) per node
  $\Rightarrow$ 8 cores per node; InfiniBand (QDR) interconnect

Parallel Performance Study for Parabolic Test Problem

Parabolic Problem

Problem: Time-dependent, scalar, linear reaction-diffusion equation in three space dimensions that is a simplification of a multi-species model of calcium flow in heart cells

Find the concentration of the single species $u(x, y, z, t)$ for all $(x, y, z) \in \Omega$ and $0 \leq t \leq T$ such that

$$\begin{aligned} \frac{\partial u}{\partial t} - \nabla \cdot (D \nabla u) &= 0 \quad && \text{in } \Omega \text{ for } 0 < t \leq T, \\ \mathbf{n} \cdot (D \nabla u) &= 0 \quad && \text{on } \partial\Omega \text{ for } 0 < t \leq T, \\ u &= u_{\text{ini}}(x, y, z) \quad && \text{in } \Omega \text{ at } t = 0, \end{aligned} \tag{1}$$

- $\Omega = (-6.4, 6.4) \times (-6.4, 6.4) \times (-32.0, 32.0) \subset \mathbb{R}^3$ micrometers
- $\mathbf{n} = \mathbf{n}(x, y, z)$ denotes the unit outward normal vector at the surface point (x, y, z) of $\partial\Omega$.
- Diffusion coefficients $D = \text{diag}(D_x, D_y, D_z)$ where $D_x = D_y = 0.15$ and $D_z = 0.30$ in micrometers squared per millisecond
- Muscedere, Raim, and Gobbert, HPCF–2010–4, 2010, `www.umbc.edu/hpcf`

Parabolic Problem

- The initial distribution is chosen to be:

$$u_{\text{ini}}(x,y,z) = \cos^2\left(\frac{\lambda_x x}{2}\right)\cos^2\left(\frac{\lambda_y y}{2}\right)\cos^2\left(\frac{\lambda_z z}{2}\right), \tag{2}$$

  where $\lambda_x = \pi/X$, $\lambda_y = \pi/Y$ and $\lambda_z = \pi/Z$.
- The true solution is computed using separation of variables and Fourier analysis to be

$$u(x,y,x,t) = \frac{1+\cos(\lambda_x x)e^{-D_x\lambda_x^2 t}}{2}\,\frac{1+\cos(\lambda_y y)e^{-D_y\lambda_y^2 t}}{2}\,\frac{1+\cos(\lambda_z z)e^{-D_z\lambda_z^2 t}}{2}. \tag{3}$$

- System evolves from the non-uniform initial distribution $u_{\text{ini}}(x,y,z)$ to the constant steady state solution $u_{SS} \equiv 1/8$
- Steady state is not reached with our final simulation time of $T = 100$ ms.
- Muscedere, Raim, and Gobbert, HPCF–2010–4, 2010, `www.umbc.edu/hpcf`

- Standard FEM theory predicts $\|u_h(\cdot,t) - u(\cdot,t)\|_{L^2(\Omega)} = \mathcal{O}(h^q)$ where $q = 2$
- We estimate q by

$$q^{(est)} = \log_2 \left(\frac{\|u_{2h}(\cdot,t) - u(\cdot,t)\|_{L^2_\Omega}}{\|u_h(\cdot,t) - u(\cdot,t)\|_{L^2_\Omega}} \right). \tag{4}$$

$N_x \times N_y \times N_z$	$t = 30$ ms		$t = 40$ ms		$t = 50$ ms	
$32 \times 32 \times 128$	2.6954e–02		2.1525e–02		1.7019e–02	
$64 \times 64 \times 256$	6.7459e–03	(1.9984)	5.3837e–03	(1.9994)	4.2556e–03	(1.9997)
$128 \times 128 \times 512$	1.6843e–03	(2.0019)	1.3451e–03	(2.0009)	1.0636e–03	(2.0004)
$256 \times 256 \times 1024$	4.1852e–04	(2.0088)	3.3541e–04	(2.0037)	2.6569e–04	(2.0012)
$512 \times 512 \times 2048$	1.0213e–04	(2.0349)	8.3029e–05	(2.0143)	6.6226e–05	(2.0043)
$1024 \times 1024 \times 4096$	2.3358e–05	(2.1284)	2.0048e–05	(2.0501)	1.6401e–05	(2.0136)

- $L^2(\Omega)$-norm of the finite element error decreases by a factor of about 4
- Estimated convergence order $q^{(est)} \approx 2$ (Second order convergence)
- $\Longrightarrow$ Tolerance of the iterative linear solver is tight enough to ensure a sufficiently accurate solution of the linear system
- $\Longrightarrow$ Tolerance on ODE error is small enough to ensure that the time error does not dominate
- Muscedere, Raim, and Gobbert, HPCF–2010–4, 2010, `www.umbc.edu/hpcf`

Parabolic Application

$N_x \times N_y \times N_z$	DOF	memory usage (MB) predicted	observed
$32 \times 32 \times 128$	140,481	23	33
$64 \times 64 \times 256$	1,085,825	174	177
$128 \times 128 \times 512$	8,536,833	1,368	1,379
$256 \times 256 \times 1024$	67,700,225	10,846	10,859
$512 \times 512 \times 2048$	539,233,281	86,394	86,544

- Notice that serial run *not* possible for $512 \times 512 \times 2048$ with 24 GB
  $\Rightarrow$ **Parallel computing enables the solution of a larger problem!**
- Number of time steps taken by the ODE solver is 208.

tara: Parabolic Application Problem: Performance by Number of Nodes

(a) Mesh resolution $N_x \times N_y \times N_z = 128 \times 128 \times 512$, DOF = 8,536,833							
	1 node	2 nodes	4 nodes	8 nodes	16 nodes	32 nodes	64 nodes
1 ppn	00:46:14	00:23:44	00:12:09	00:06:37	00:03:23	00:01:45	00:00:52
2 ppn	00:23:38	00:11:52	00:06:40	00:03:25	00:01:49	00:00:58	00:00:30
4 ppn	00:12:54	00:06:20	00:03:47	00:02:02	00:00:59	00:00:33	00:00:18
8 ppn	00:07:07	00:03:53	00:02:00	00:00:59	00:00:32	00:00:20	00:00:12
(b) Mesh resolution $N_x \times N_y \times N_z = 256 \times 256 \times 1024$, DOF = 67,700,225							
	1 node	2 nodes	4 nodes	8 nodes	16 nodes	32 nodes	64 nodes
1 ppn	10:58:00	04:54:49	02:45:21	01:22:05	00:41:33	00:20:56	00:10:36
2 ppn	04:48:32	02:26:29	01:24:19	00:40:52	00:21:10	00:10:53	00:05:40
4 ppn	03:04:01	01:22:23	00:44:27	00:23:53	00:12:33	00:06:17	00:03:04
8 ppn	01:27:12	00:43:35	00:22:03	00:11:52	00:06:04	00:03:33	00:01:57
(c) Mesh resolution $N_x \times N_y \times N_z = 512 \times 512 \times 2048$, DOF = 539,233,281							
	1 node	2 nodes	4 nodes	8 nodes	16 nodes	32 nodes	64 nodes
1 ppn	N/A	N/A	N/A	18:58:45	09:36:22	05:01:33	02:27:50
2 ppn	N/A	N/A	N/A	09:26:10	04:46:39	02:24:10	01:12:10
4 ppn	N/A	N/A	N/A	05:25:31	02:49:06	01:23:08	00:43:20
8 ppn	N/A	N/A	N/A	02:43:45	01:21:29	00:40:56	00:22:11

- 04:50:18 to run $1024 \times 1024 \times 4096$ problem with DOF = 4,304,410,625 on 512 processes (8 per node)
- Muscedere, Raim, and Gobbert, HPCF–2010–4, 2010, `www.umbc.edu/hpcf`

Parallel Performance Study for 3-Species Application Problem

Three-Species Application

Memory requirements: Finite element mesh with $N_x \times N_y \times N_z$ elements has $N = (N_x + 1)(N_y + 1)(N_z + 1)$ nodes and $n_{eq} = n_s N = 3N$ degrees of freedom (DOF) for the application problem with $n_s = 3$ species.

$N_x \times N_y \times N_z$	N	n_{eq} = DOF	Total memory
$16 \times 16 \times 64$	18,785	56,355	9 MB
$32 \times 32 \times 128$	140,481	421,443	68 MB
$64 \times 64 \times 256$	1,085,825	3,257,475	522 MB
$128 \times 128 \times 512$	8,536,833	25,610,499	4,103 MB

- Studies use $N_x \times N_y \times N_z$ mesh with $N_x = N_y$ and $N_z = 4N_x$ in accordance with cell shape.
- Final time for all studies $t_{\text{fin}} = 1{,}000$ ms, as required for application problem. Number of time steps by the ODE solver is over 56,000.
- Notice that serial run *is* possible for $128 \times 128 \times 512$ with 24 GB, but would be unacceptably slow, as it will turn out!
  $\Rightarrow$ **Parallel computing enables the solution in acceptable amount of time!**

tara: 3 Species Application Problem: Performance by Number of Nodes

(a) Mesh resolution $N_x \times N_y \times N_z = 16 \times 16 \times 64$, DOF = 56,355

	1 node	2 nodes	4 nodes	8 nodes	16 nodes	32 nodes	64 nodes
1 process per node	00:30:09	00:15:29	00:08:19	00:04:25	00:02:36	00:01:56	00:01:51
2 processes per node	00:15:41	00:08:01	00:04:25	00:02:31	00:01:45	00:01:31	N/A
4 processes per node	00:08:04	00:04:19	00:02:31	00:01:39	00:01:25	N/A	N/A
8 processes per node	00:04:27	00:02:34	00:01:36	00:01:10	N/A	N/A	N/A

(b) Mesh resolution $N_x \times N_y \times N_z = 32 \times 32 \times 128$, DOF = 421,443

	1 node	2 nodes	4 nodes	8 nodes	16 nodes	32 nodes	64 nodes
1 process per node	05:16:13	02:40:35	01:21:26	00:42:09	00:22:29	00:13:13	00:07:45
2 processes per node	02:41:35	01:20:33	00:41:27	00:21:46	00:12:56	00:07:44	00:05:24
4 processes per node	01:24:06	00:43:23	00:22:00	00:12:06	00:07:38	00:05:14	N/A
8 processes per node	00:45:17	00:23:13	00:12:39	00:07:46	00:05:09	N/A	N/A

(c) Mesh resolution $N_x \times N_y \times N_z = 64 \times 64 \times 256$, DOF = 3,257,475

	1 node	2 nodes	4 nodes	8 nodes	16 nodes	32 nodes	64 nodes
1 process per node	57:14:46	29:32:28	14:34:19	08:15:46	03:51:04	01:59:36	01:05:30
2 processes per node	29:08:07	14:39:54	07:21:31	03:46:57	02:00:56	01:04:30	00:43:10
4 processes per node	14:39:19	08:05:01	04:03:45	02:05:39	01:08:09	00:38:44	00:25:21
8 processes per node	08:19:29	04:14:48	02:10:41	01:10:29	00:41:40	00:27:47	N/A

(d) Mesh resolution $N_x \times N_y \times N_z = 128 \times 128 \times 512$, DOF = 25,610,499

	1 node	2 nodes	4 nodes	8 nodes	16 nodes	32 nodes	64 nodes
1 process per node						21:44:14	11:18:17
2 processes per node					23:45:20	11:17:57	06:43:25
4 processes per node				26:42:05	12:12:05	06:35:41	03:53:24
8 processes per node			23:44:55	12:30:28	06:53:45	03:57:07	02:33:53

- Near-optimal speedup from 1 to 2, from 2 to 4 as well as from 4 to 8 processes per node. $\Longrightarrow$ Use all cores per node!
- Trott and Gobbert, HPCF–2010–11, 2010, www.umbc.edu/hpcf

Summary: Parabolic Test and Application Problems

Parabolic scalar (1 species) test problem and 3-species application problem, mesh resolution $N_x \times N_y \times N_z = 64 \times 64 \times 256$

Cluster	problem	serial 1 core time	(1 node) all cores time (speedup)	32 node 1 core per node time (speedup)	32 node all cores time (speedup)
kali (2003)	scalar test	02:19:23		00:05:07 (27.51)	00:04:09 (33.56)
hpc (2008)	scalar test	00:09:43	00:02:42 (3.60)	00:00:22 (26.50)	00:00:10 (58.30)
tara (2009)	scalar test	00:04:33	00:00:42 (6.50)	00:00:12 (22.75)	00:00:04 (68.25)
tara (2009)	3-species	57:14:46	08:19:29 (6.88)	01:59:36 (28.72)	00:27:47 (123.63)

- Scalar test problem: system dimension 1,085,825;
  3-species application problem: system dimension 3,257,475
- kali: ODE solver implicit Euler (BDF1), linear solver CG;
  hpc and tara: ODE solver NDFk, $1 \leq k \leq 5$, linear solver QMR

Conclusions and Future Work

Conclusions:

- Code is valid simulation tool for the process,
  based on numerical analysis and on computational convergence studies
- Model captures wave initiation and allows waves to self-organize,
  most recently examples of wave with recovery and of spiral wave!
- Parallel computing enables studies of desired resolution for full length of laboratory time scale in reasonable wall clock time
- Performance study guides effective use of cluster:
  Use cores per node, i.e., 8 processes per node for tara

Future work = available research projects:

- Parallel Implementation of Matrix-Free Preconditioning in C with MPI
- Parallel Implementation of the Linear Solve in CUDA for GPGPUs
- Mathematical and Numerical Multi-Scale Algorithms for Calcium Flow in a Heart Cell

Research Project 1: Matrix-Free Preconditioning

Available Research Project 1:
Parallel Implementation of Matrix-Free Preconditioning in C with MPI

- Problem: Krylov subspace methods (CG, QMR, etc.) should be preconditioned. As rough guide, expect a preconditioned method to use 10 times fewer iterations.
- Issue 1: Since the current implementation is matrix-free for memory efficiency, also the implementation of the preconditioning needs to be matrix-free. An algorithm for this idea has been implemented in Matlab that needs to be translated into C.
- Issue 2: If we do not communicate between parallel processes at all, then the preconditioning will not be very effective. But communicating between *all* neighboring processes is likely too costly. Since memory hierarchy has several levels today, some communication may be acceptable and improve iteration count. A flexible implementation in MPI is needed to test various strategies.

Research Project 2: GPU Programming

Available Research Project 2:
Parallel Implementation of the Linear Solve in CUDA for GPGPUs

- GPGPU = general purpose GPU, GPU = graphics processing unit.
- CUDA = C-style programming language for GPGPUs from NVIDIA
- GPGPUs hold great potential for parallel computing, due to their extremely good cost-per-processor. Rough example: 8-core CPU is same cost in dollars as a GPGPU with 128 processors!
- Issues: GPGPUs are SIMD = single-instruction multiple-data and shared-memory, that is, very different from currently dominating MPI (Message Passing Interface) approach. For programming, we need to learn CUDA, an extension of C.
- GPGPUs have complicated memory hierarchy, whose use we need to control explicitly (different from caches in CPUs). Therefore, the algorithm needs to be carefully implemented and tested.

Research Project 3: Multi-Scale Algorithm for Calcium Flow

Available Research Project 3: Mathematical and Numerical Multi-Scale Algorithms for Calcium Flow in a Heart Cell

- Models for calcium flow in a cell are actually multi-scale: (i) calcium channel (under μm), (ii) calcium release unit (μm), (iii) one cell (tens of μm), (iv) whole heart (cm).
- Currently, calcium release units (CRUs) are modeled as point sources with constant-in-time parameters and independent of actual species concentrations at the location.
- Mathematical multi-scale model: instead of one formula for a CRU, call another model for each CRU to obtain more precise data
  $\Rightarrow$ makes the one-cell simulator more accurate
- Numerical multi-scale model: instead of one formula for a CRU, call another simulator to obtain data for each open CRU, thus allowing less resolution in areas without open CRUs
  $\Rightarrow$ makes the one-cell simulator more efficient
- Issue: Interface one-cell simulator with CRU simulators.

Center for Interdisciplinary Research and Consulting (CIRC)

CIRC is a consulting service for mathematics and statistics provided by the Department of Mathematics and Statistics at UMBC.

- Established in 2003, CIRC is dedicated to support interdisciplinary research for the UMBC campus community and for outside industry researchers.
- CIRC provides a full range of consulting services on mathematics and statistics from free initial consulting to long term support for research programs.
- Mathematics and statistics students gain hands on interdisciplinary experience vital for industry and academia jobs.
- Graduate students are involved as RAs and via Math/Stat 750 Interdisciplinary Consulting.
- In collaboration with the Division of Information Technology (DoIT), CIRC provides hands-on workshops on mathematical and statistical software packages including: MATLAB, COMSOL Multiphysics, Mathematica, Microsoft Access, SAS, S-Plus, and SPSS.

All information: `www.umbc.edu/circ`

REU Site: Interdisciplinary Program in High Performance Computing

- NSF-funded summer research program for undergraduates: 8 weeks, $3,600 stipend plus room and board, travel, books, tuition, fees covered
- Fully transferrable course Math 447 on scientific, parallel, and statistical computing in Phase I (Weeks 1 through 3) leveraging HPCF
- Collaborative research projects with interdisciplinary clients (AT&T Labs, NIH, Census Bureau, Oak Ridge NL, NSA, and other departments in 2010 and 2011) in Phase II (Weeks 3 through 8) leveraging CIRC
- Deliverables: (i) result to client, (ii) presentation at university-wide poster session, (iii) publications (HPCF tech. rep., SIURO, etc.)
- All aspects of program in teams of participants, continuously supported by graduate TAs/RAs (UMBC funded) and faculty
- Participants diverse: gender, race/ethnicity, univ. vs. liberal arts colleges
- Holistic approach: GRE course, field trips, professional development workshops, lectures on LaTeX, CV, paper publication process, academic integrity, advice for graduate school, etc.
- Contact: hpcreu@umbc.edu, `www.umbc.edu/hpcreu`;
  Matthias K. Gobbert and Nagaraj K. Neerchal, {gobbert,nagaraj}@umbc.edu