

# Educating in Software Engineering Methods With Computational Biology



Joe Pitt-Francis

Oxford Supercomputing Centre

Integrative Biology Workshop 29 & 30 September 2005



# Short term goals

---

- Dave Gavaghan needed 4 week course
- Educate “doctoral training” students in coding and simulation
- Find out for ourselves about building large simulation codes
- Find out what we’re all doing
- “Have a laff”



# Long term goals

---

Have a simulation code that we all understand (~25 of us)

- Easy to switch numerical techniques
- Easy to model new things (soft tissue mechanics, heterogeneities etc.)
- Easy to maintain when original author disappears



# Master plan (Gavaghan et al.)

---

- Do some teaching and exercises first
- Use C++ with PETSc (for linear algebra and parallelisation)
- Extreme/agile programming techniques
- Simple models of cardiac, tumour, soft tissue mechanics...
- Later to become “CHASTE”



# Extreme programming

---

- Iterative and incremental development
- Test-driven development
- Emergent design & refactoring
- Pair programming
- Collective code ownership
- Coding standards
- Retrospectives

# Teaching phase (1.5 weeks)

---



Lee Momtahan: Software Engineering



Joe Pitt-Francis: Parallel computing



Blanca Rodriguez: Cardiac modelling



Dan Stokeley: Software standards



Jon Whiteley: Finite element method



## “XP phase” (2.5 weeks)

---

- ~25 programmers (students and research officers) working in two rooms
- Working day 2pm-5pm and 9am-2pm
- Stand-up meetings at 2pm
- Distribution of user stories and assignment of tasks to groups/pairs
- Three major iterations

# “Maintenance phase” (Summer)

---

- Up to 6 programmers in one room
- One working day per week
- Two main iterations:
  - New infrastructure and code tidy
  - Parallelisation, memory leaks, simulation
- User stories, iterations, automated documentation and Wiki on Trac website

# Trac website

[720] (changeset) - Chaste - Trac

logged in as jmpf@comlab.ox.ac.uk | [Logout](#) | [Settings](#) | [Help/Guide](#) | [About Trac](#)

[Wki](#) | [Time line](#) | [Roadmap](#) | [Browse Source](#) | [View Tickets](#) | [New Ticket](#) | [Search](#)

[Previous Changeset](#) | [Next Changeset](#)

## Changeset 720

**Timestamp:** Wed Sep 21 16:46:35 2005  
**Author:** jmpf@comlab.ox.ac.uk

**Message:** Fixed [TestSimpleLinearSolver?](#) to run in parallel.

This revision ought to close [ticket:31](#).

**Files:** ■ [trunk/pde/test/TestSimpleLinearSolver.hpp \(diff\)](#)

Unmodified  Added  Removed  Modified

**trunk/pde/test/TestSimpleLinearSolver.hpp**

r637	r720	
51	51	PetscScalar *lhs_elements;
52	52	VecGetArray(lhs_vector, &lhs_elements);
53		TS_ASSERT_DELTA(lhs_elements[0], 1.0, 0.000001);
54		TS_ASSERT_DELTA(lhs_elements[1], 1.0, 0.000001);

Done chaste.ediamond.ox.ac.uk



# Current status

---

We have:

- Small suite of ODE solvers
- Several cardiac cell model classes
- Mesh and data I/O in various formats
- FE assemblers (Discrete Galerkin etc.)
- Elliptic, Parabolic and coupled PDE solvers
- Parallel MPI code “everywhere”
- One monodomain cardiac simulation



# Current status

---

We also wanted:

- Moving meshes
- Simple tumour growth simulation
- Parallel data writers
- ...and more things in the future



# Conclusion

---

- Everybody learned a lot
- We have some good, usable code...
- ...and some bad, unusable code
- We are beginning to effect other research projects in terms of standards, code repositories, continuous testing etc.