Scientific Computing and Computational Science

What's the difference?

Gustaf Söderlind

Numerical Analysis, Lund University

LUND
UNIVERSITY

What is Computational Science?

Computational *X*

X = *Physics / Biology / Fluid Dynamics / Electromagnetics / Science & Engineering / Mechanics / Chemistry / Physiology..*

Characteristics

- Application driven
- Bridges gap between theory and experiment
- Sometimes replaces experiment
- Of ever increasing importance
- "Standard" software and computational methodology

What is Scientific Computing?

Independent research area combining *mathematics, numerical analysis, computer science, software engineering*

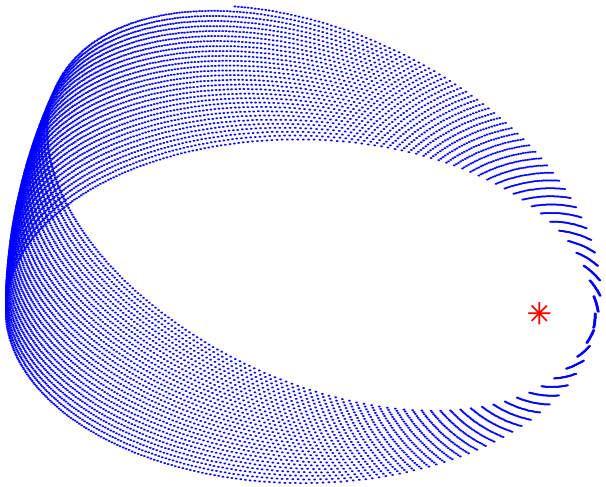
Provides basic methodology behind Computational *X*

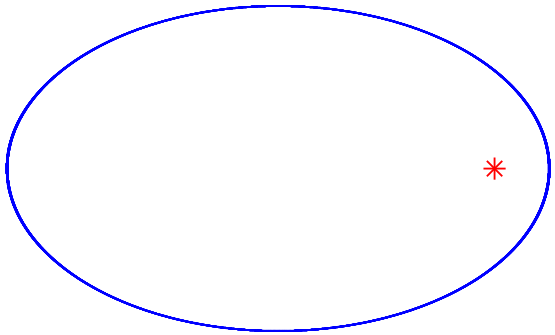
Focal points

- Analysis and construction of computational methods
- Stability, accuracy, efficiency, theorem & proof
- Algorithm and software design
- Advanced computers and computing
- Special needs in special applications

Scientific computing *Kepler problem*

Constant step size vs adaptive symplectic integration method

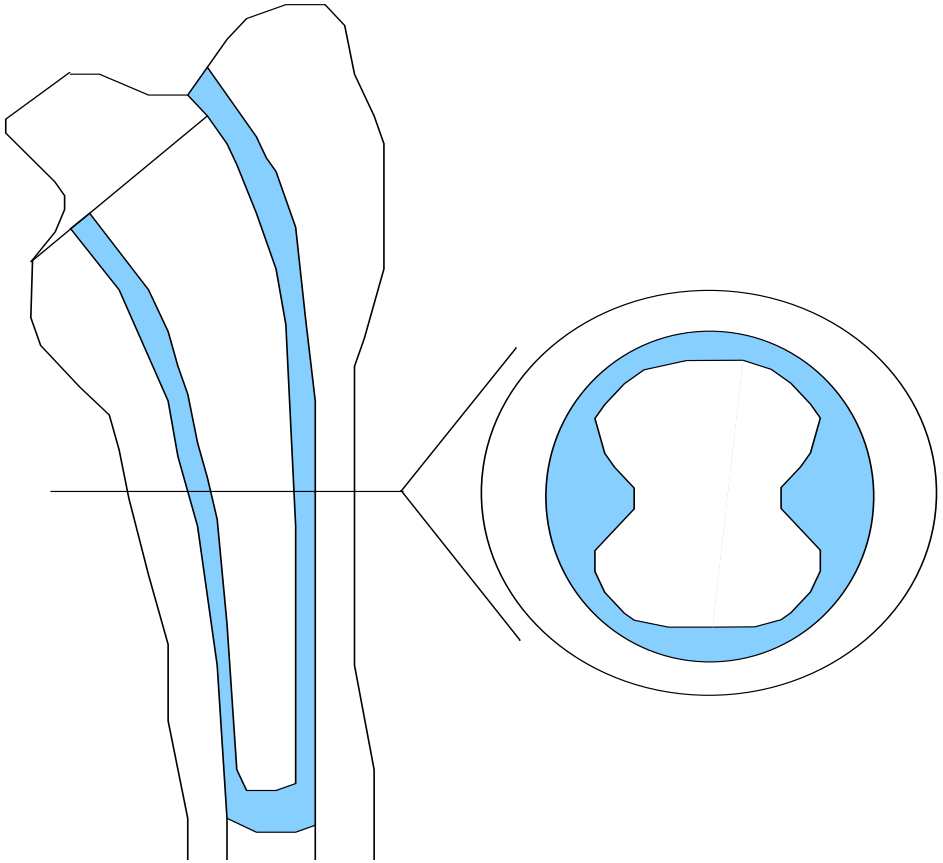




Special construction of Hamiltonian step size control system

Same average work (10,000 steps) but 30 times better accuracy and numerical precession completely suppressed

Example of interaction *Hip joint prosthesis model*



Example of interaction *Hip joint prosthesis model*

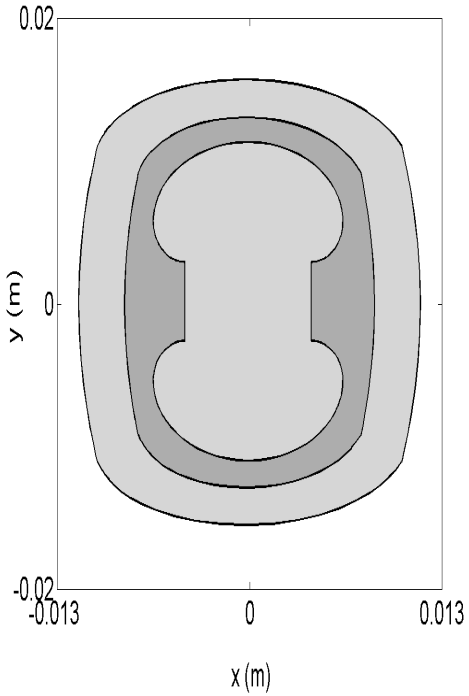
Reaction–diffusion model

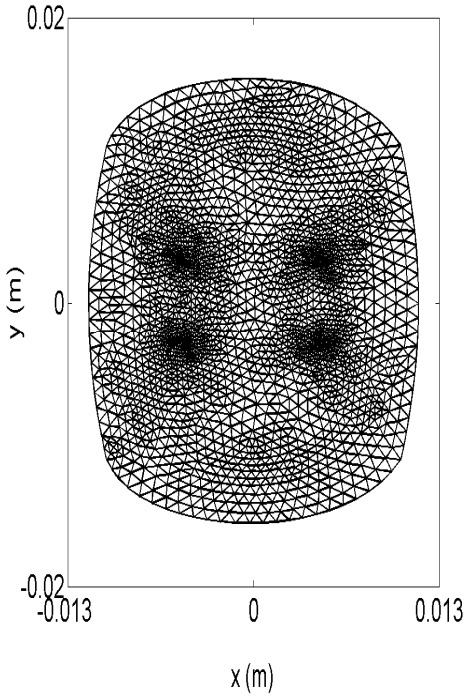
$$\frac{\partial}{\partial t}\begin{bmatrix} T \\ C \end{bmatrix} = \begin{bmatrix} \nabla \cdot (D \nabla T) + Q(C,T) \\ G(C,T) \end{bmatrix}$$

- Heat transfer/polymerization
- Coupled nonlinear PDE/ODE model

Example of interaction *Hip joint prosthesis model*

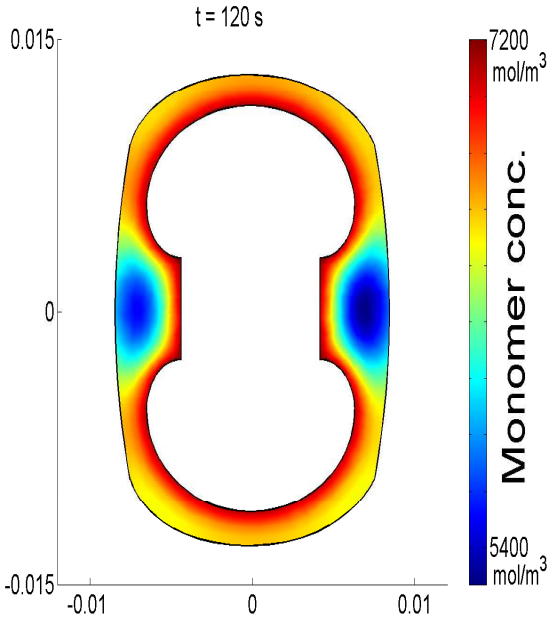
Finite element discretization

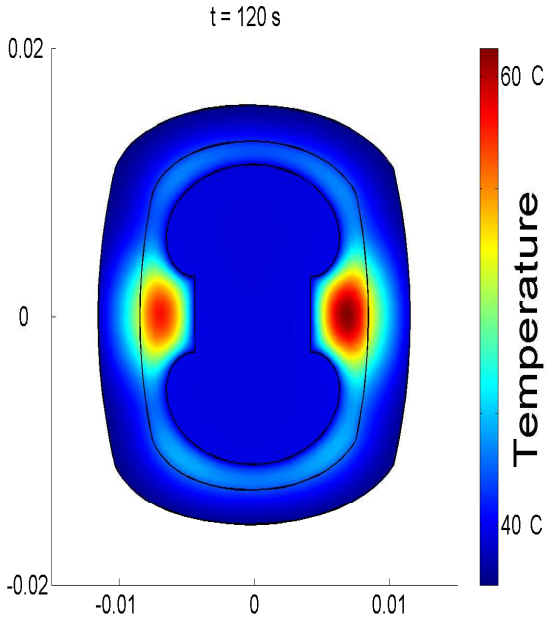




Example of interaction *Hip joint prosthesis model*

Computational results





A simplified test problem *Diffusion,* $u_t = u_{xx}$

Finite difference discretization $u(t_n, x_j) \approx u_j^n$

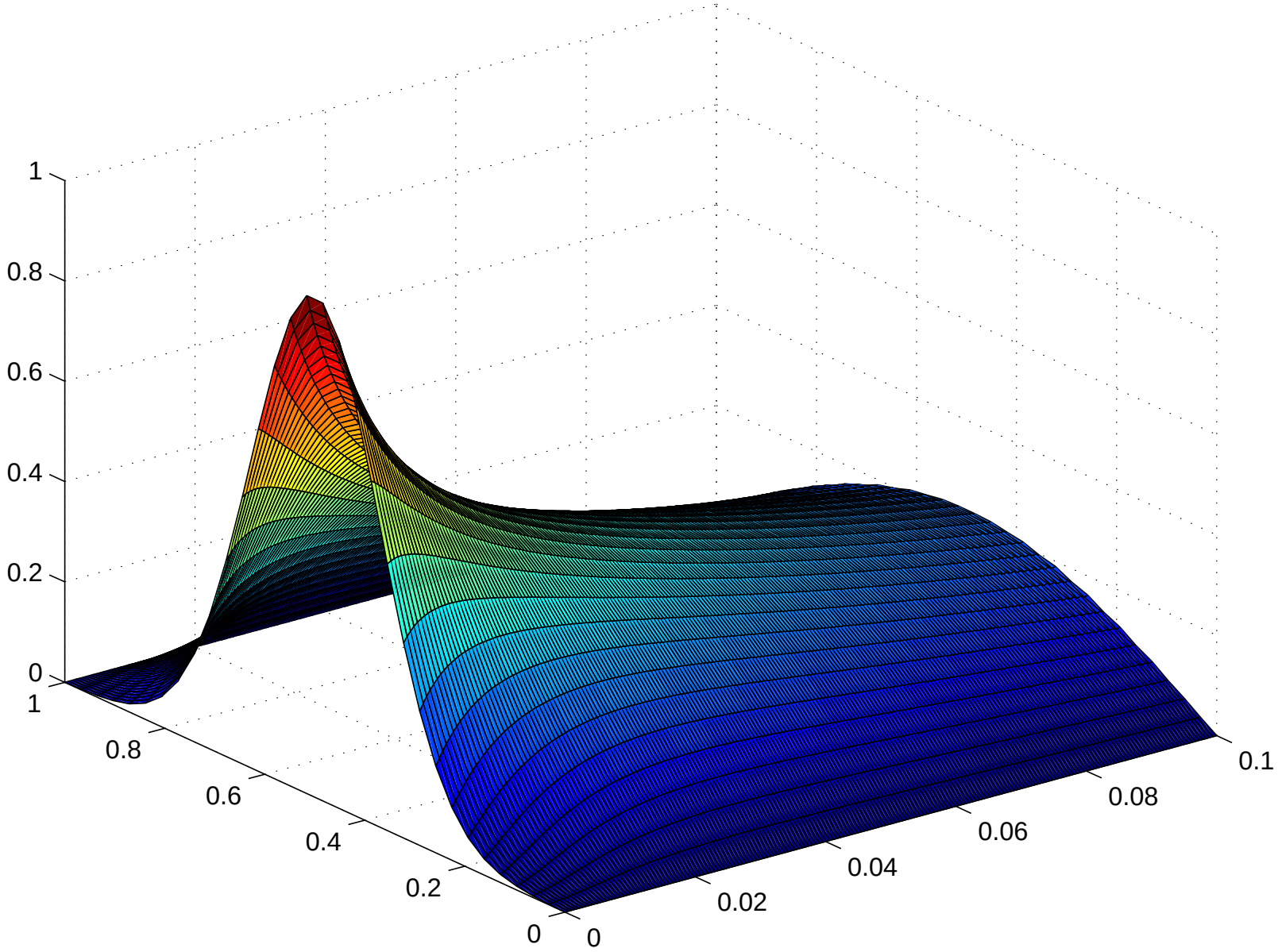
$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta x^2}$$

Putting $\mu = \Delta t / \Delta x^2$ we obtain recursion

$$u_j^{n+1} = u_j^n + \mu \cdot (u_{j-1}^n - 2u_j^n + u_{j+1}^n)$$

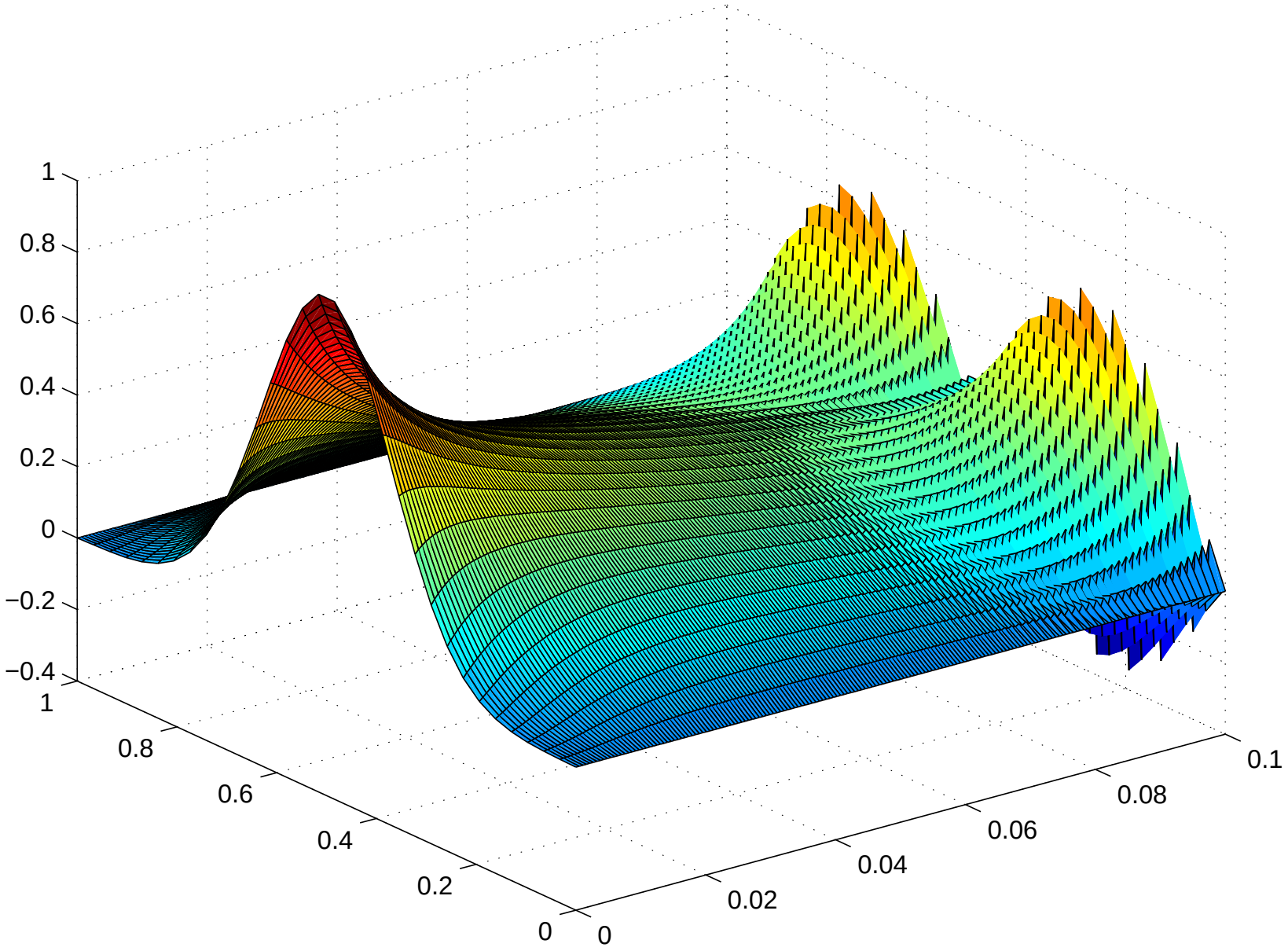
$N = 30$ points in $[0, 1]$ with $M = 187$ time steps on $[0, 0.1]$

Stable solution at $\mu = .514$



$N = 30$ points in $[0, 1]$ with $M = 184$ time steps on $[0, 0.1]$

Unstable solution at $\mu = .522$



Why does that happen?

Answer – discretization methods produce approximations that do not exactly replicate mathematical behavior

Necessary to understand basic principles of scientific computing to be successful in computational science

The bottom line

Volvo knows how to construct a good car...

The bottom line

Volvo knows how to construct a good car...

...but that doesn't make *you* the perfect driver!

The bottom line

Volvo knows how to construct a good car...

...but that doesn't make you the perfect driver!

Consider obtaining a driver's license!

Take-home message

In scientific computing you often encounter Murphy's Law

Don't expect to be successful in computational science with black-box codes – you'll need to understand how and why they work, and what to do when they don't

Consider getting a basic training in scientific computing – you'll need it!