School of Computing

UNIVERSITY OF LEEDS

"On the Development of Implicit Solvers for Time-Dependent Systems"

Peter Jimack

School of Computing, University of Leeds

In collaboration with: P.H. Gaskell, C.E. Goodyer, J.R. Green, Y.-Y. Koh, A.M. Mullis, J. Rosam, M. Sellier, H.M. Thompson & M.A. Walkley.

AMMW03, October 2012

Introduction

1. Introduction
2. Time-dependent partial differential equations/systems
3. Explicit versus implicit temporal discretisation
4. Multigrid solvers
5. Thin-film models for the spreading of viscous drops
6. Phase-field models for simulating solidification of a binary alloy
7. Parallel scalability
8. Summary and discussion

Time-dependent PDEs and systems

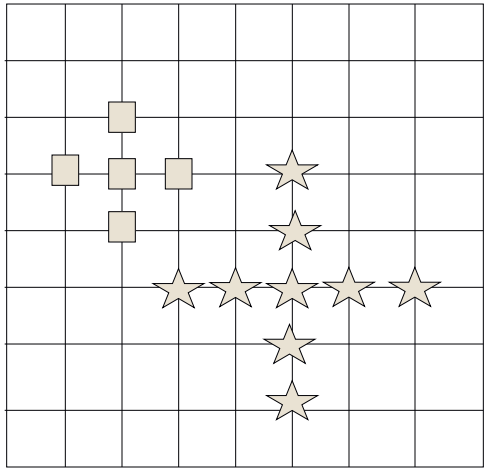
- Models of this type occur in a very wide range of applications including weather and climate modelling
- My experience is primarily with viscous flow and phase-change problems and so I use examples from these applications here (sorry!)
- We will follow the approach of discretising separately in space and time:
  - Spatial discretization leads to a large system of initial value ODEs (often referred to as *method of lines*)
  - The examples used here are finite difference but we have similar experiences with finite element/volume schemes in space
- I hope that I am able to present some ideas that can be of value to the discussions at this meeting!

Explicit versus implicit temporal discretisations

- Explicit schemes are simplest to implement but always come with conditional stability restrictions on the time step size
  - As the spatial resolution grows these can be prohibitive
  - This is especially true if fast transients are present and we do not wish to resolve them
- Unconditionally stable implicit schemes can allow the time step size to be selected based upon accuracy criteria alone – but this comes at the expense of having to solve a very large algebraic system at each step!
- Semi-implicit schemes are can give best of both worlds – e.g. treat nonlinear terms explicitly and linear terms implicitly to increase maximum stable step size
- Can also treat nonlinear terms implicitly however...

Multigrid solvers

• A quick overview



- Consider model equation $-\Delta u = f$
- Discretization leads to an algebraic system
- N unknowns – but number of non-zeros per row is independent of N
- So iterative scheme costs O(N) per sweep

- BUT the error reduces slowly with the number of sweeps...
  - It typically takes a long time for the lowest frequency components of the error to be damped however...
  - *the highest frequency components of the error that can be represented on this grid are damped almost immediately...*

Multigrid solvers – linear example

• Using the differential equation's structure

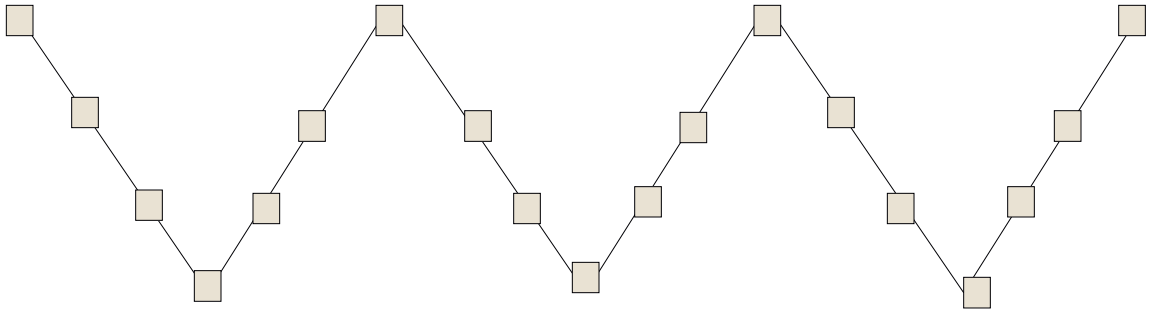
- Discretization on a fine grid gives: $\mathbf{A}^f \mathbf{u}^f = \mathbf{f}^f$
- A few sweeps of Jacobi gives: $\hat{\mathbf{u}}^f$
- The algebraic error $\mathbf{e}^f = \mathbf{u}^f - \hat{\mathbf{u}}^f$ satisfies: $\mathbf{A}^f \mathbf{e}^f = \mathbf{r}^f = \mathbf{f}^f - \mathbf{A}^f \hat{\mathbf{u}}^f$
- Approximate this error on a coarse grid: $\mathbf{A}^c \mathbf{e}^c = \mathbf{r}^c = \mathbf{I}_f^c \mathbf{r}^f$
- Interpolate this coarse grid correction back to the fine grid: $\hat{\mathbf{u}}^f = \hat{\mathbf{u}}^f + \mathbf{I}_c^f \mathbf{e}^c$
- Take a few more sweeps of Jacobi: $\tilde{\mathbf{u}}^f$

Note that it is possible to solve the coarse grid equation using this same algorithm recursively – this is multigrid...

Multigrid solvers

• Theory

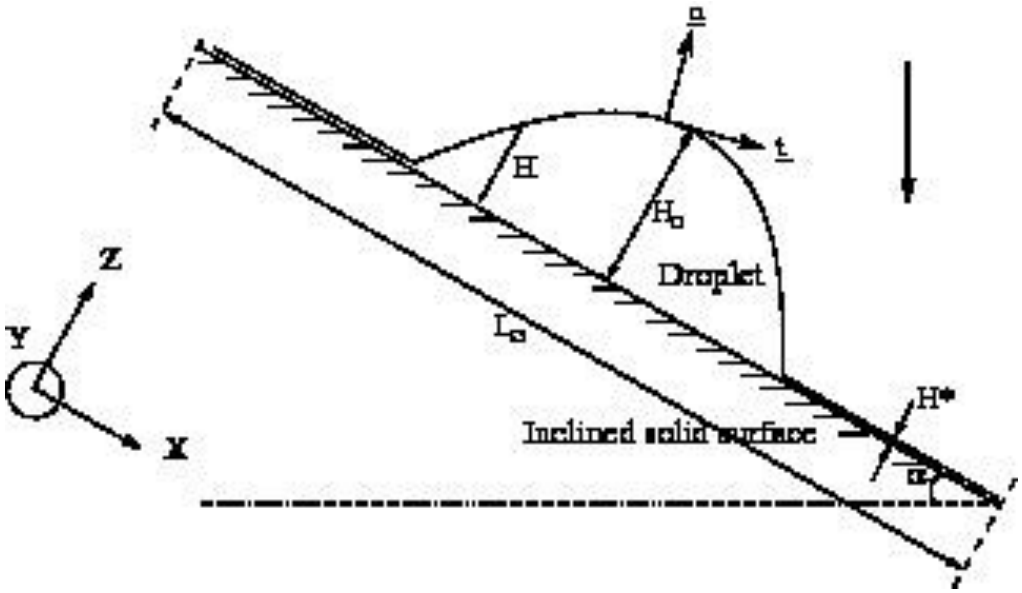
- This gives an MG cycle:



- There exists a rigorous convergence theory, for this and similar cycles, showing mesh-independent convergence rates (e.g. Hackbusch, 1985):
  - requires a “smoothing property” on the iteration
  - requires an “approximation property” on $(\mathbf{I}_f^c, \mathbf{I}_c^f)$
- The cost of each V-cycle is $O(N)$
- Hence the total cost is also $O(N)$

Example I: A thin film model for viscous droplets

- We consider a thin-film model for a viscous droplet flowing down an inclined plane.
- Lubrication approximation based upon $H_0 \langle\langle L_0$



A thin film model for viscous droplets

- This leads to the Reynolds equation for the non-dimensional film thickness h and an associated pressure equation for p:

$$\frac{\partial h}{\partial t} = \frac{\partial}{\partial x}\left[\frac{h^3}{3}\left(\frac{\partial p}{\partial x} - \frac{Bo}{\varepsilon}\sin\alpha\right)\right] + \frac{\partial}{\partial y}\left[\frac{h^3}{3}\left(\frac{\partial p}{\partial y}\right)\right]$$

$$p = -\nabla^2(h+s) - \Pi(h) + Bo(h+s)\cos\alpha$$

- This is a nonlinear time-dependent system in h(x,y,t)
- A very fine spatial mesh is required to represent a thin *precursor film*
- Implicit time-stepping overcomes *excessive* conditional stability conditions – allowing step selection based upon error control...

Nonlinear multigrid solver

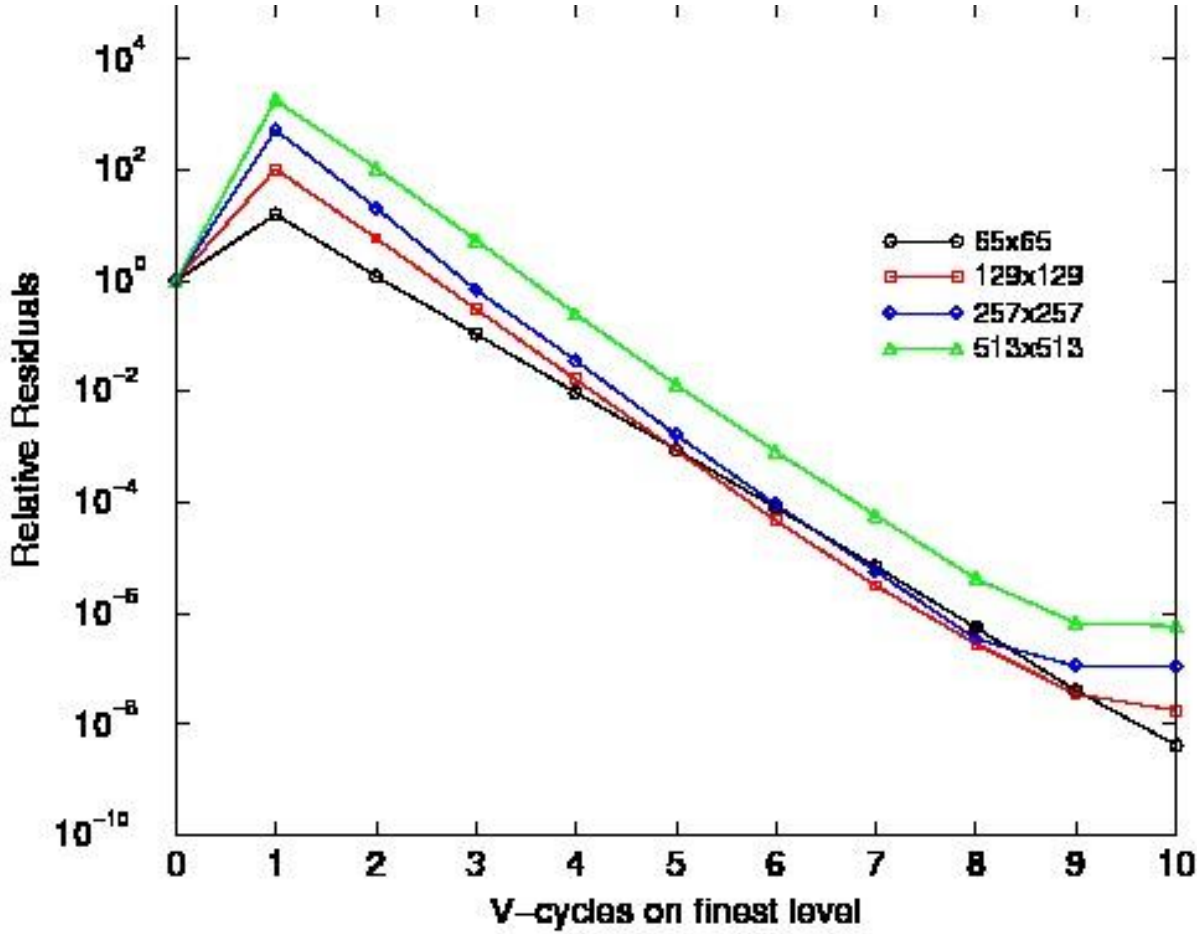
At each time step a large nonlinear algebraic system of equations must be solved for the new values: h_{ij}^{n+1} and p_{ij}^{n+1}.

Unless this can be done efficiently the method is worthless…

- A f*ully coupled* nonlinear Multigrid solver is used to achieve this:
  - based upon the FAS (full approximation scheme) approach to resolve the non-linearity
  - this is a slight variation on the linear multigrid approach outlined above.
  - a pointwise weighted nonlinear Jacobi iterative scheme is seen to be an adequate smoother.
- Excellent, *h*-independent, convergence results are obtained (see below).

Multigrid performance for implicit time stepping

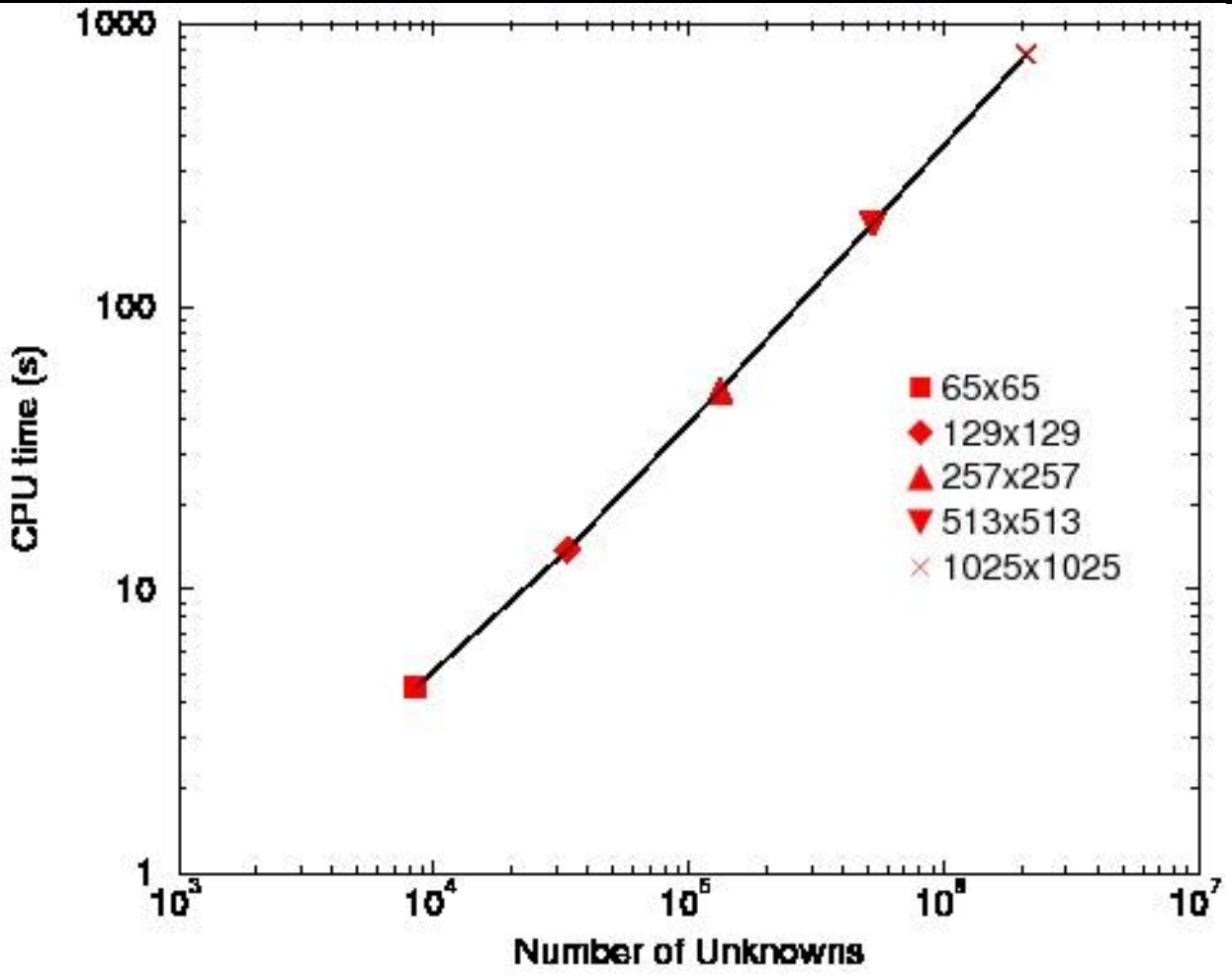
- Convergence rate is mesh independent...



Jimack et al. – AMMW03

Multigrid performance for implicit time stepping

- Cost of solution is optimal – O(N)...



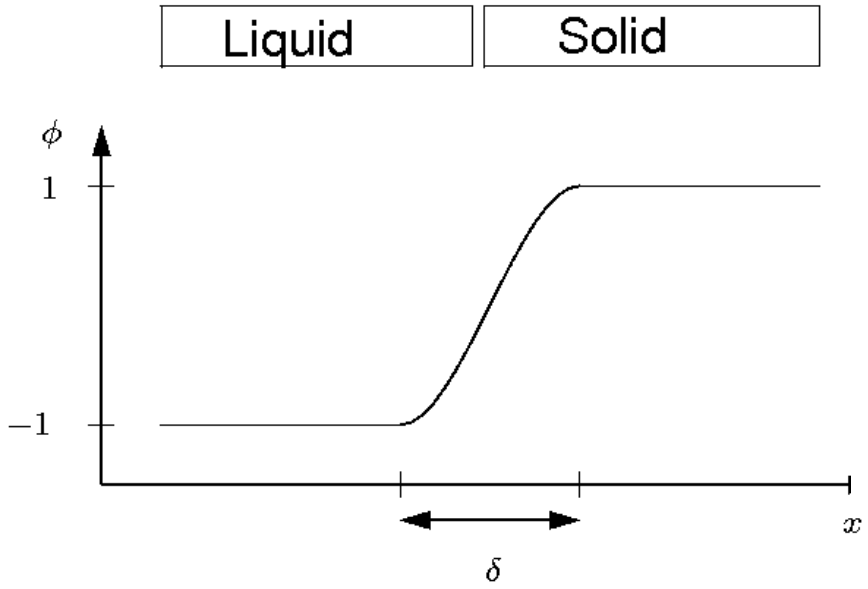
Jimack et al. – AMMW03

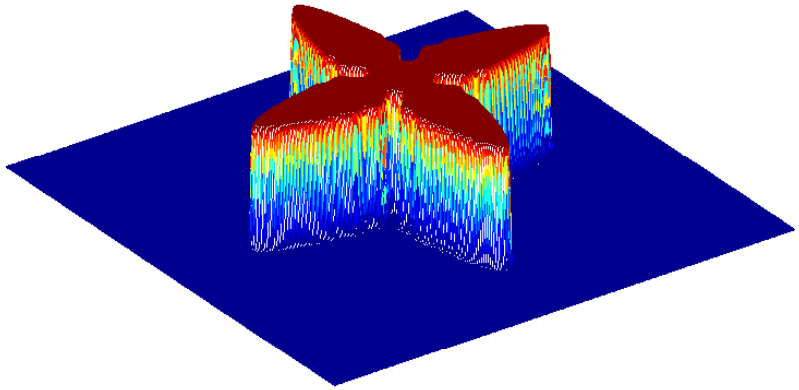
Example II: Phase-field model for alloy solidification

• Basic Idea of Phase-Field

•Introduce an artificial phase-field variable to describes the state (Φ = -1 for liquid and φ = +1 for solid).

•At the interface φ varies smoothly between these bulk values – the motion of this interface is determined from the geometry and a concentration field U.





Jimack et al. – AMMW03

Binary alloy phase-field model (2-d for simplicity)

• Karma's Phase-field model

$$A(\psi)^2 \frac{\partial \phi}{\partial t} = A(\psi)^2 \nabla^2 \phi + 2A(\psi)A'(\psi)\left[\frac{\partial \psi}{\partial x}\frac{\partial \phi}{\partial x} + \frac{\partial \psi}{\partial y}\frac{\partial \phi}{\partial y}\right]$$

$$-\frac{\partial}{\partial x}\left(A(\psi)A'(\psi)\frac{\partial \phi}{\partial y}\right) + \frac{\partial}{\partial y}\left(A(\psi)A'(\psi)\frac{\partial \phi}{\partial x}\right) +$$

$$\phi - \phi^3 - \lambda(1-\phi^2)^2(\theta_{fix} + Mc_\infty U)$$

Phase Equation

Properties:

- highly nonlinear
- noise introduced by anisotropy function A(Ψ)
- where $\psi = \arctan(\phi_y/\phi_x)$

Jimack et al. – AMMW03

UNIVERSITY OF LEEDS

Binary alloy phase-field model (2-d for simplicity)

• Karma's Phase-field Model

$$\left(\frac{1+k}{2}-\frac{1-k}{2}\phi\right)\frac{\partial U}{\partial t}=D\left(-\frac{1}{2}\left[\frac{\partial\phi}{\partial x}\frac{\partial U}{\partial x}+\frac{\partial\phi}{\partial y}\frac{\partial U}{\partial y}\right]+\frac{1-\phi}{2}\nabla^2 U\right)+$$

$$\frac{1}{2\sqrt{2}}\left(\{1+(1-k)U\}\left(\frac{\partial}{\partial x}\left(\frac{\partial\phi}{\partial t}\frac{\phi_x}{|\nabla\phi|}\right)+\frac{\partial}{\partial y}\left(\frac{\partial\phi}{\partial t}\frac{\phi_y}{|\nabla\phi|}\right)\right)\right.$$

$$\left.+(1-k)\left(\frac{\partial U}{\partial x}\left(\frac{\partial\phi}{\partial t}\frac{\phi_x}{|\nabla\phi|}\right)+\frac{\partial U}{\partial y}\left(\frac{\partial\phi}{\partial t}\frac{\phi_y}{|\nabla\phi|}\right)\right)\right)$$

$$+\frac{1}{2}\left((1+(1-k)U)\frac{\partial\phi}{\partial t}\right)$$

Concentration Equation

- Also highly nonlinear – and coupled to the phase equation

Jimack et al. – AMMW03

UNIVERSITY OF LEEDS

Evolution of a typical solution in 3-d

• For a model of an isothermal alloy (concentration and phase fields)

• An example of the growth of a dendritic structure – this animation plots the $\varphi = 0$ isosurface at different time intervals.

•Begins with a small solid seed.

•Here we impose preferred growth along the axis through our choice of anisotropy function A...

Adaptive spatial discretization (3-d)

• Adaptive mesh refinement

•From the solution profiles there is an obvious need for adaptive mesh refinement.

•Here this is based upon hierarchical hexahedral meshes – with local refinement and coarsening – using PARAMESH.

•Have implemented a number of different Finite Difference stencils – all 2^{nd} order (based upon 7, 19 and 27 points).

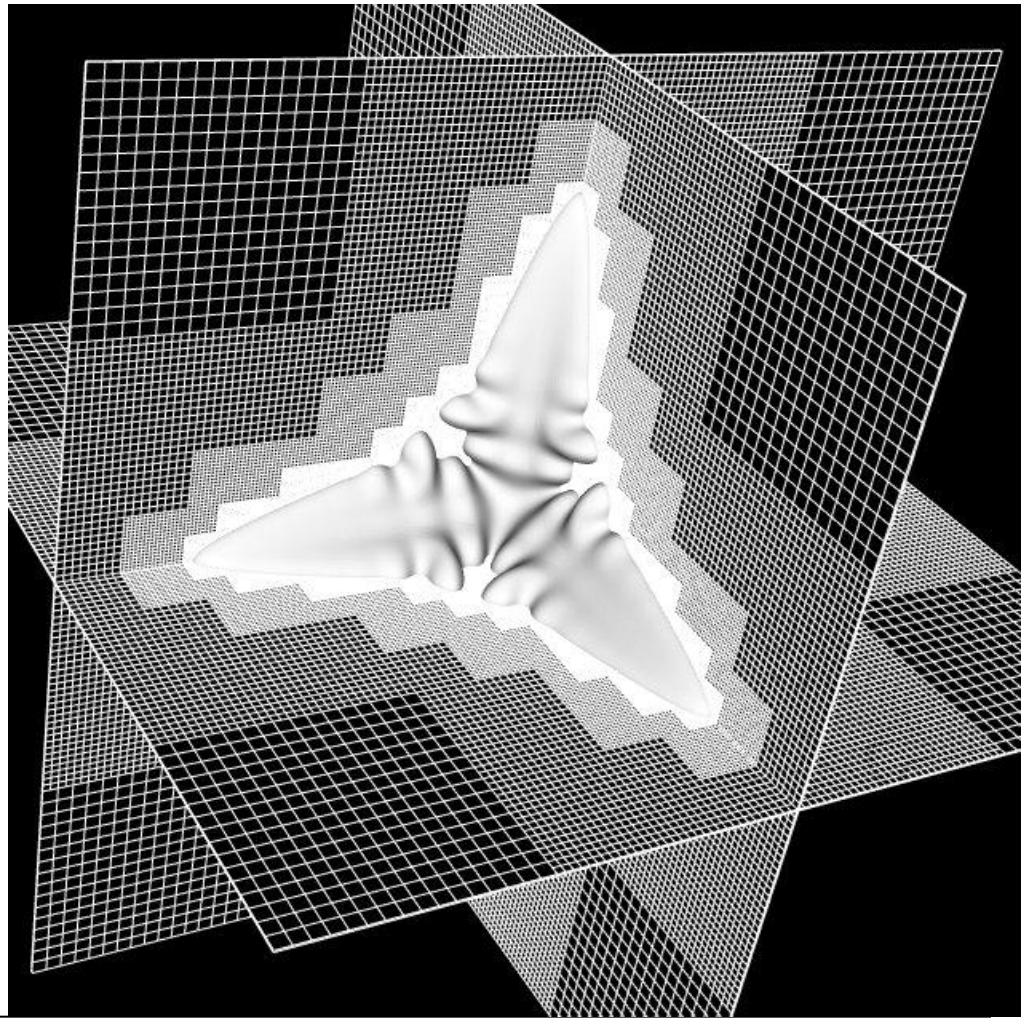
•Adaptive remeshing is controlled by a simple gradient criterion.

•The following diagram illustrates a typical mesh – with refinement concentrated in the regions where the phase variable and the concentration variable have the highest gradients – at a particular snapshot in time...

Adaptive spatial discretization (3-d)

• Adaptive mesh refinement

•Here we see local mesh refinement around the $\varphi = 0$ isosurface....



Jimack et al. – AMMW03

Implicit temporal discretization

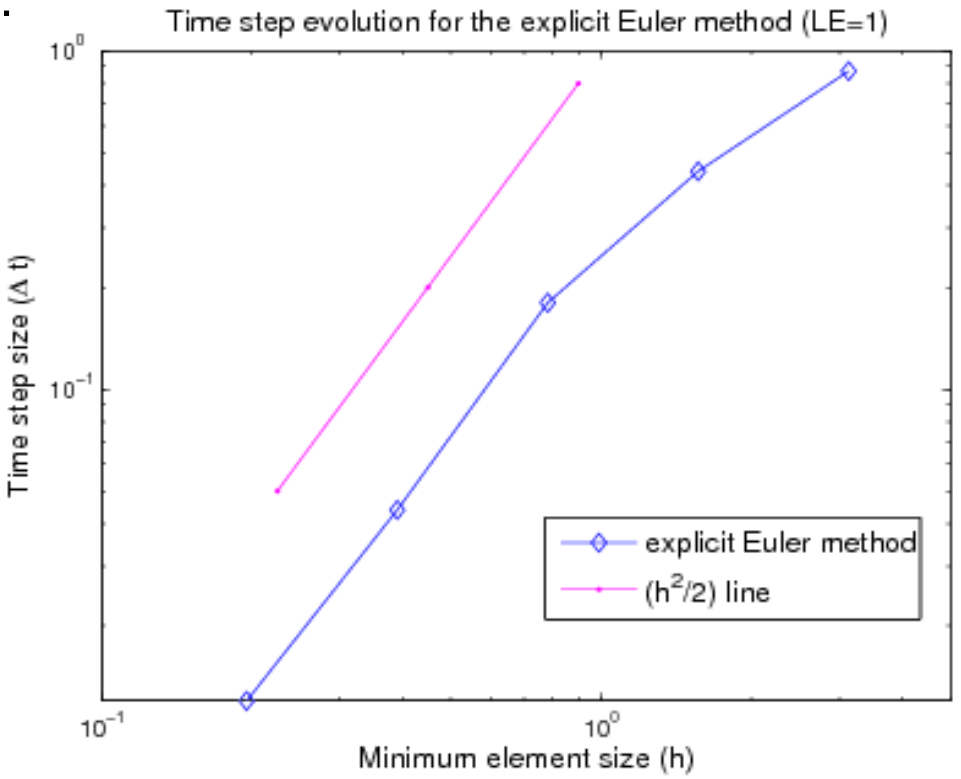
• Explicit time integration methods

- Explicit methods are "easy" to apply but impose a time step restriction:

$$\Delta t = C\frac{h^2}{2}$$

- Very fine mesh resolution is needed,so the time steps become e**xcessively** small (not viable).

- This plot shows the maximum stable time step for an isothermal model...

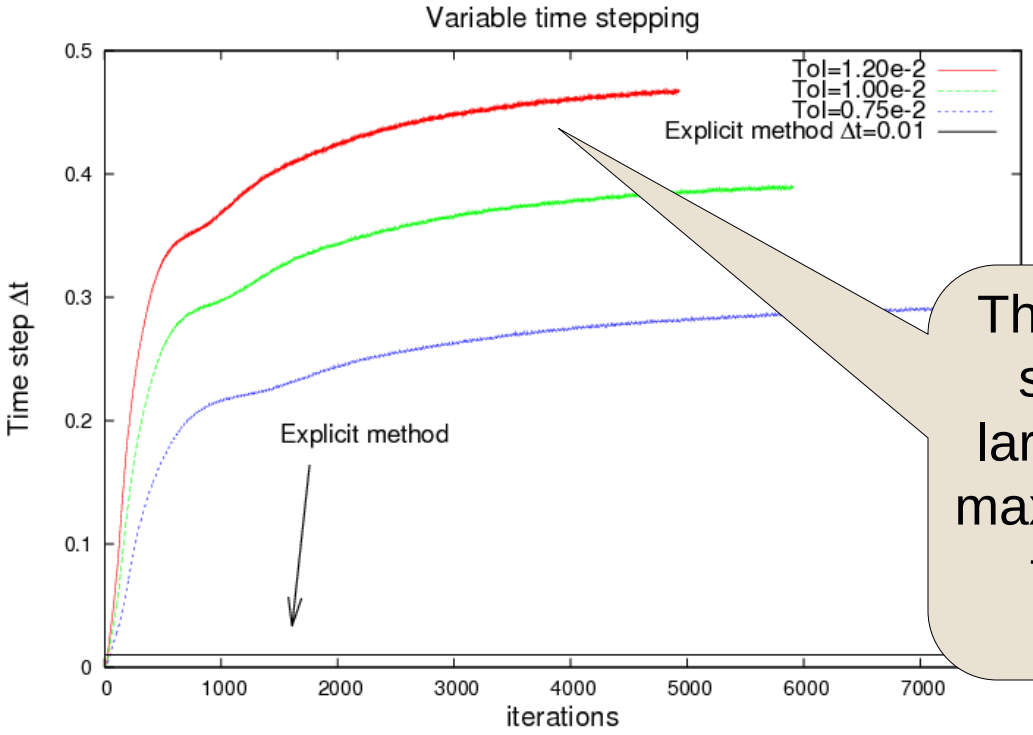


Jimack et al. – AMMW03

Implicit temporal discretisation

• Adaptive time step control

• **Fully implicit** BDF2 method, combined with ***variable time stepping*** *(*based upon a local error estimator), is used to overcome time-step restrictions...



The adaption of the time step leads to a much larger time step than the maximum stable time step for the explicit Euler method !!

Jimack et al. – AMMW03

Nonlinear multigrid solver

• Nonlinear multigrid solver for adaptive meshes

At each time step a large nonlinear algebraic system of equations must be solved for the new values: ϕ_{ijk}^{n+1} and U_{ijk}^{n+1}.

Unless this can be done efficiently the method is worthless…

• A f*ully coupled* nonlinear Multigrid solver is used to achieve this:

  ➢ based upon the FAS (full approximation scheme) approach to resolve the non-linearity

  ➢ and the MultiLevel AdapTive (MLAT) scheme of Brandt to handle the adaptivity

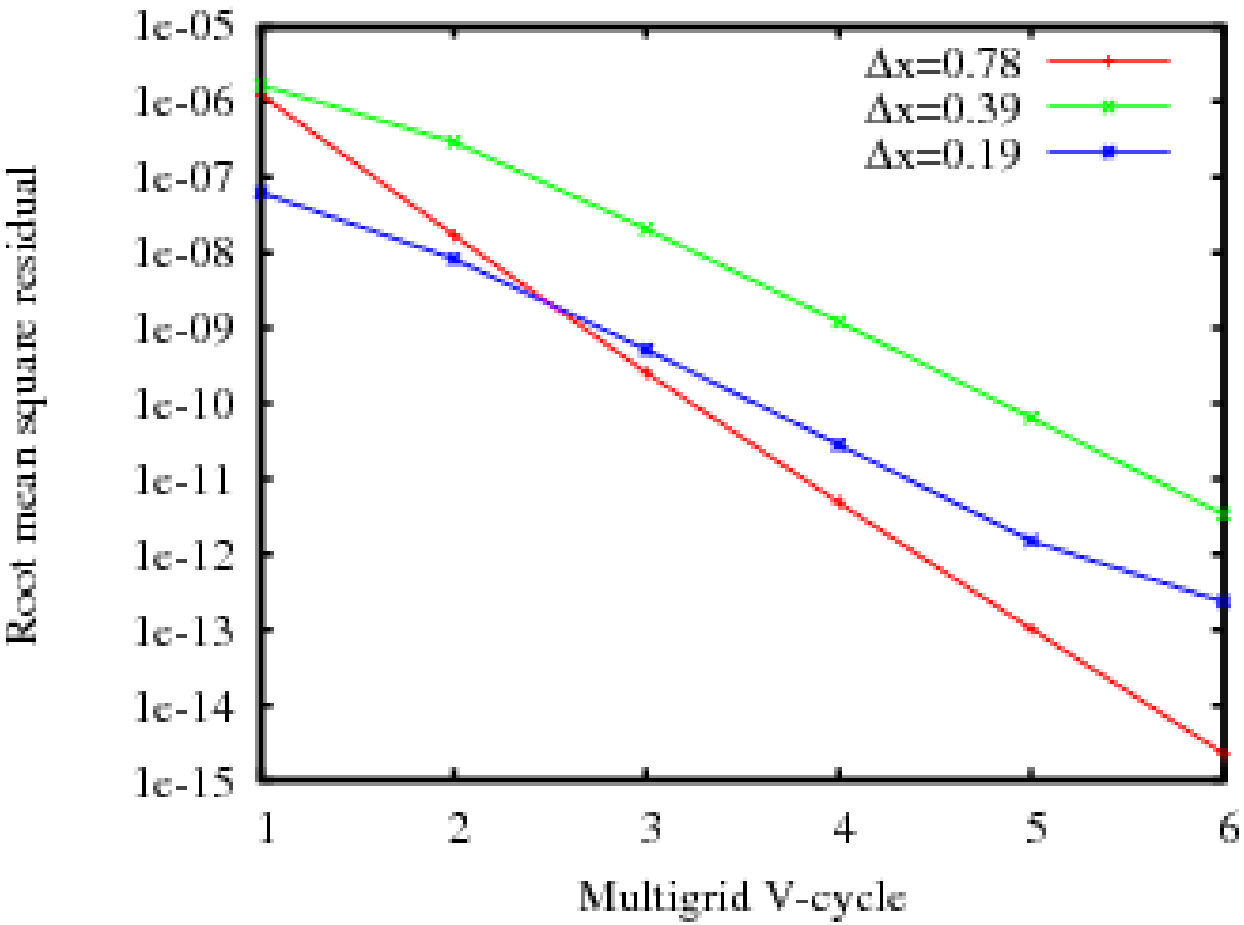
  ➢ a pointwise weighted nonlinear Jacobi iterative scheme is seen to be an adequate smoother.

• Excellent, *h*-independent, convergence results are obtained (see below).

Nonlinear multigrid solver

• Nonlinear multigrid solver for adaptive meshes

• Here we see close to an optimal convergence rate for the nonlinear multigrid solver applied to the isothermal problem in 3-d...

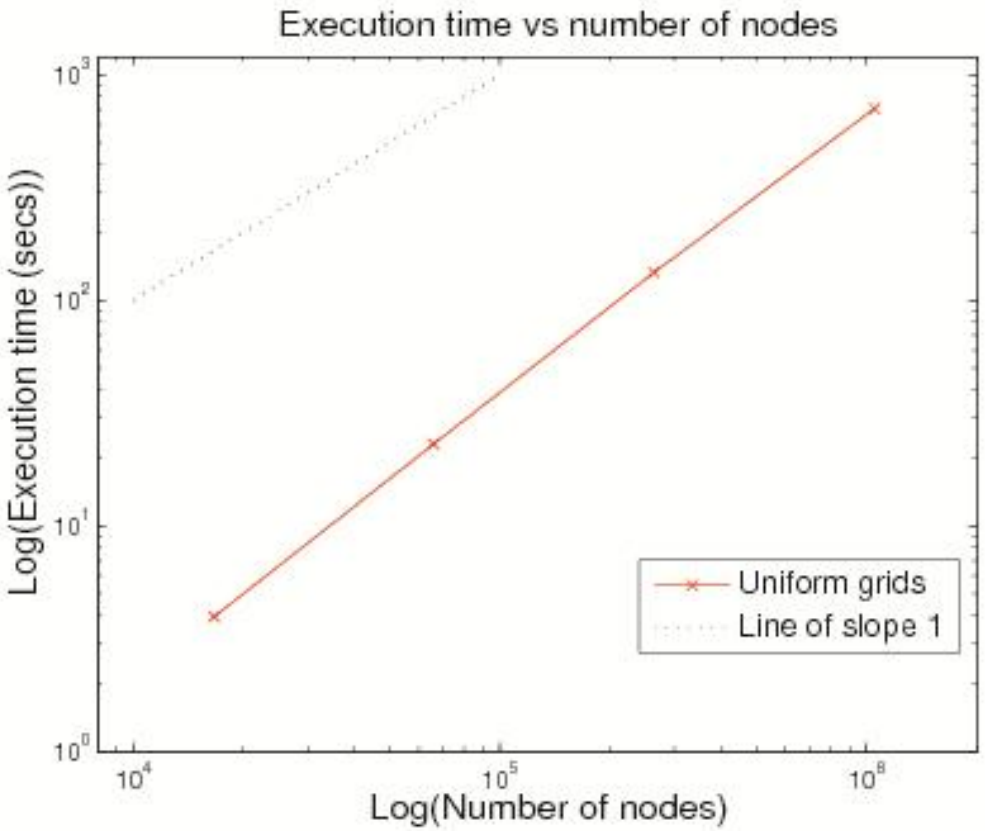


Jimack et al. – AMMW03

Nonlinear multigrid solver

• Nonlinear multigrid solver for adaptive meshes

• Here we see that the solver displays almost linear run time:

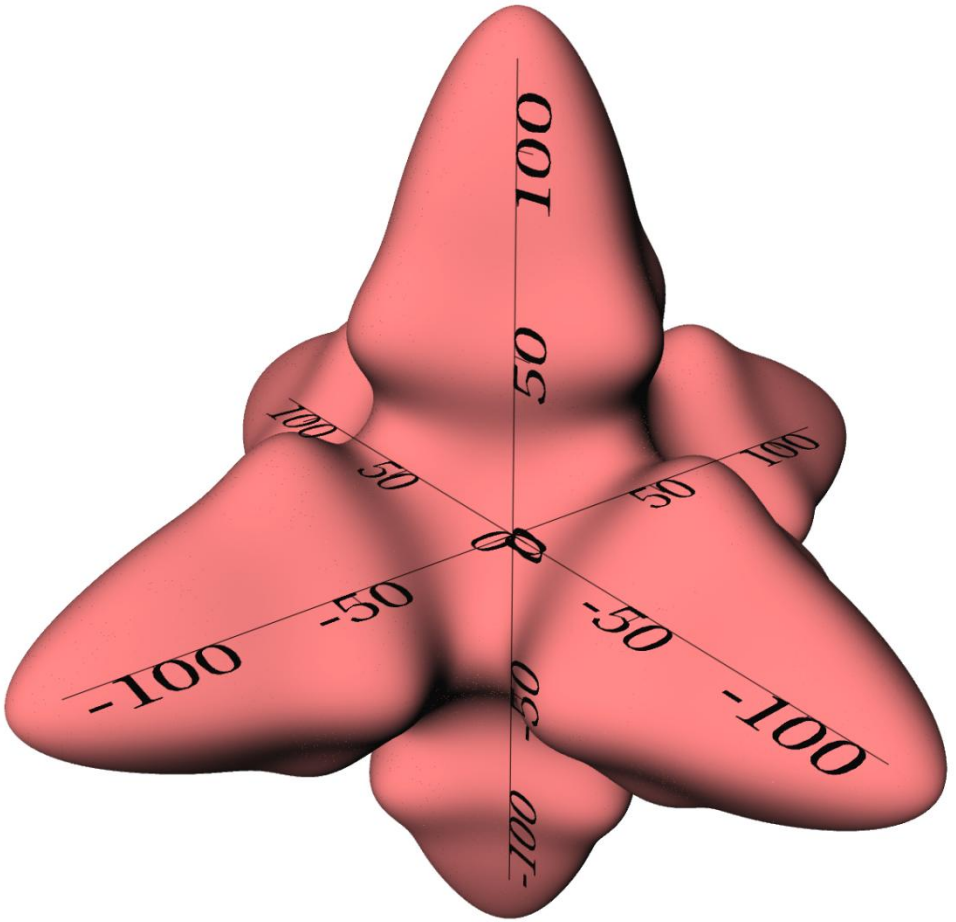


Jimack et al. – AMMW03

Results in 3-d

- For a model of an isothermal alloy

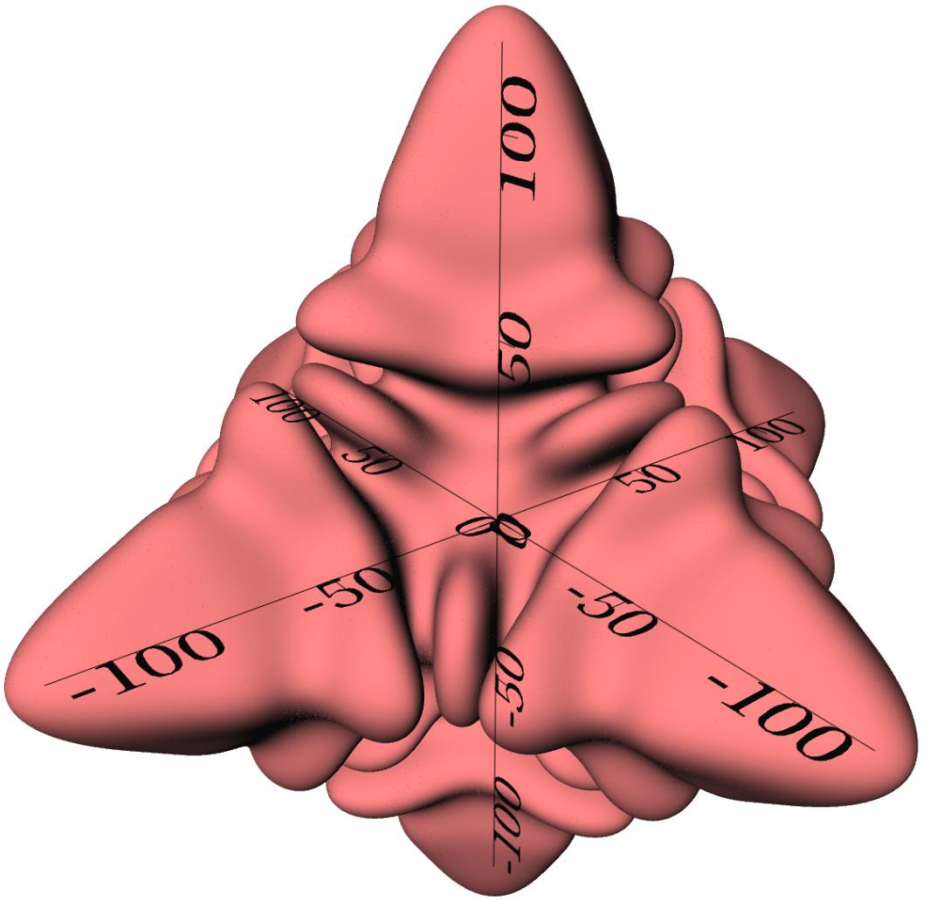
- Altering the run-time parameters gives rise to different dendrite morphologies – in these two images a different under-cooling has been used, with all other parameters held constant...



Results in 3-d

• For a model of an isothermal alloy

• Altering the run-time parameters gives rise to different dendrite morphologies – in these two images a different under-cooling has been used, with all other parameters held constant...



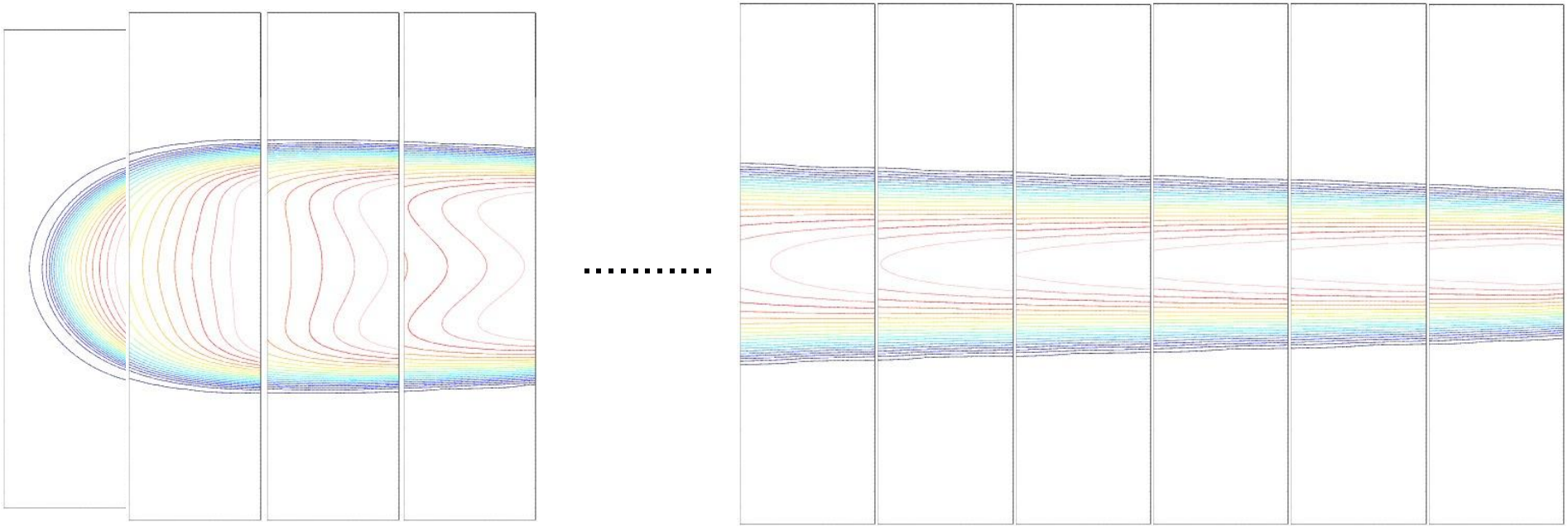
Parallel performance of multigrid

- For very large time-dependent models efficient parallel implementations are essential...
- The bottlenecks for multigrid come from:
  - the need to update on each grid in a *sequential* manner;
  - the limited amount of computational work on the coarsest grids.
- A geometric partition can help with the load balancing on each grid (though this is considerably more complex when local adaptivity is used)
- The issues of how to deal with the coarsest grids are very much open questions:
  - How coarse? How accurate? Are all processors needed at this level?

Parallel performance – thin film example

• Illustration of domain decomposition approach

- Here we illustrate the geometric decomposition of the problem
- Only neighbour to neighbour communication is required



Jimack et al. – AMMW03

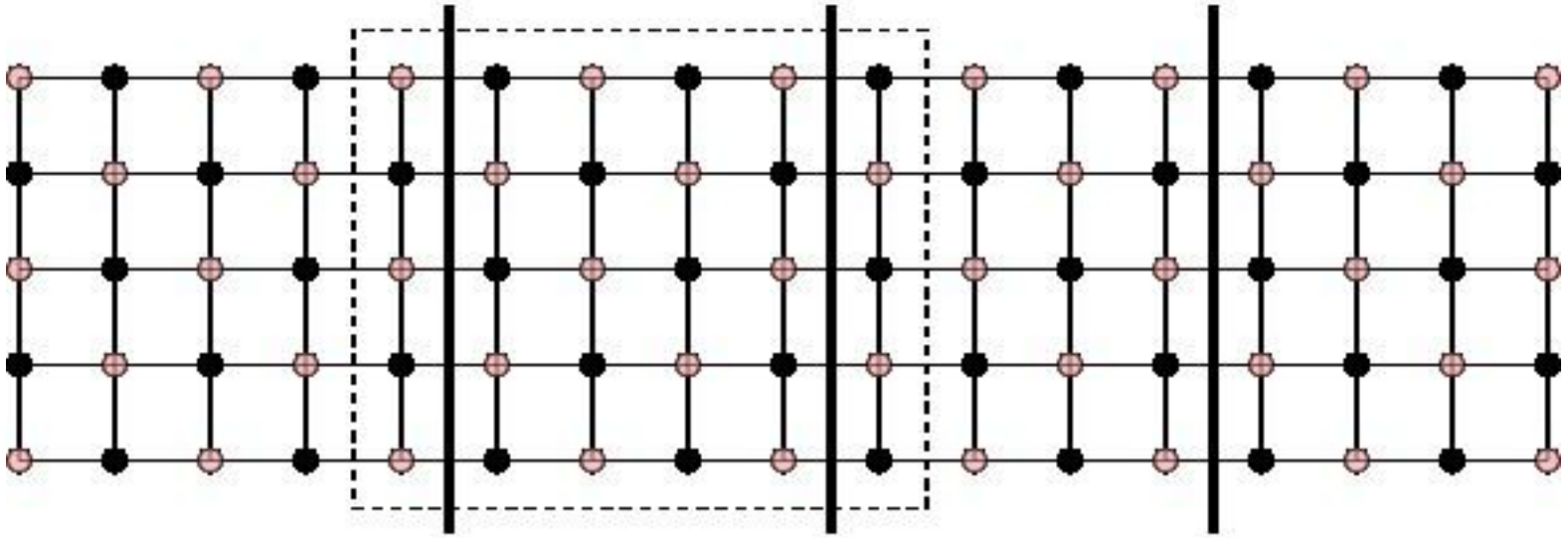
Parallel performance – thin film example

• Illustration of domain decomposition approach

• Here we illustrate the geometric decomposition of the problem on the coarsest grid

• Finer grid levels obey the same geometric partition

• Only neighbour to neighbour communication is required



Jimack et al. – AMMW03

Parallel performance – thin film example

• Illustration of additional capability due to parallel implementation

• This table illustrates the weak scalability of the nonlinear multigrid…

Cores	Size of Grid	Solver Time	Multigrid levels
2	1024 x 1024	56.9s	4
8	2048 x 2048	67.3s	5
32	4096 x 4096	69.9s	5
128	8192 x 8192	73.6s	5

• This is for a fixed number of time steps – coarse grid solve is the bottleneck.

Parallel performance – phase field example

• Illustration of additional capability due to parallel implementation

• Here we demonstrate the *capability* of combining parallelism *and* adaptivity…

Cores	Uniform Grid Level (Cells)	Adapted Grid Level (Equiv Cells)
1	7 (262k)	8 (2.1M)
2		9 (16.8M)
8	8 (2 097k)	10 (134.2M)
20		11 (1073.7M)
64	9 (16.8M)	
128		12 (8.59B)

• But what about scalability?

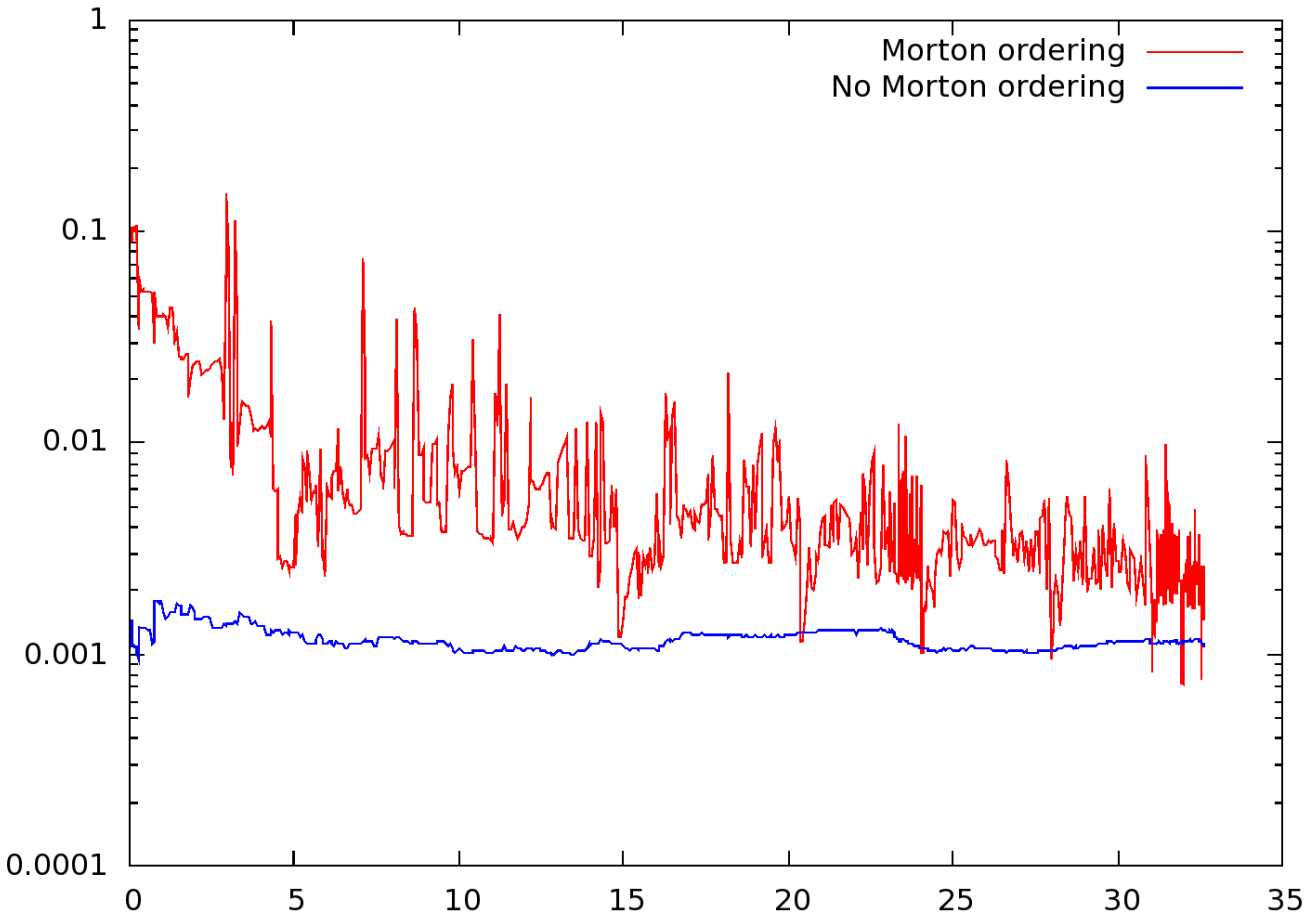
Jimack et al. – AMMW03

Parallel performance – phase field example

• Partitioning strategy is key for good parallel performance

• Here we demonstrate the importance of ensuring a good load balance at *each mesh level* in the MG hierarchy

•This plot shows the wallclock time per 1000 DoF for a parallel adaptive run with two load-balancing strategies...



Jimack et al. – AMMW03

Parallel performance – algebraic multigrid

• Illustration of parallel performance of AGMGv2.3 with MUMPS on coarsest level

• Here we demonstrate the parallel performance of *algebraic MG*

Cores	DoF	Iterations	Setup	Solve
32	31M	17	5.9s	40.4s
64	63M	17	6.2s	40.8s
128	125M	17	6.9s	41.5s
256	251M	17	9.5s	41.8s
512	501M	17	14.8s	42.5s
1024	1003M	17	27.3s	44.0s
2048	2007M	17	69.0s	48.2s
4096	4014M	17	383.0s	59.4s

Discussion

1. (Semi-)Implicit time-stepping is only practical if we have an efficient algebraic equation solver (nonlinear or linear)
2. We make use of multigrid (nonlinear/linear and geometric/algebraic)
3. Can combine with *adaptivity* and with *parallel implementation*
4. Scales well to thousands of cores:
   - Coarse grid solver becomes an issue eventually
   - May need to sacrifice exact coarse grid solve (therefore optimal convergence) for efficient parallel implementation
   - Adaptivity adds to the challenges of scalability
5. Current research is focusing on scaling our (optimal complexity) implicit solvers to tens of thousands of cores...