Parallel, adaptive multigrid methods for parabolic PDEs and applications

Feng Wei Yang

Department of Mathematics
University of Sussex

F.W.Yang@sussex.ac.uk

25 January 2016
Objectives

To solve complex non-linear parabolic systems by applying:

- $2^{nd}$ order central Finite Difference Method (FDM)
- $2^{nd}$ order Backward Differentiation Formula (BDF2)
- Nonlinear multigrid method with Full Approximation Scheme (FAS)
- Adaptive Mesh Refinement (AMR)
- Adaptive time-stepping
- Parallel techniques
Outline

- Multigrid methods
  - Linear multigrid
  - Nonlinear multigrid
- Thin film models from Gaskell et al. and Validation
- Tumour modelling and a model from Wise et al.
- Optimal control for whole cell tracking
Jacobi/Gauss-Seidel iterative methods

- Well-known methods
- Require diagonally-dominant matrices
- Typically have complexity of $O(n^2)$ for general sparse matrices
- ...

Smoothing property

Low frequency of error  
High frequency of error

Convergence of a typical Jacobi iterative method

Rapid Convergence (high-frequency error: short wave length)

Slow Convergence (low-frequency error: long wave length)

source: nkl.cc.u-tokyo.ac.jp
Multigrid V-cycle

Finest grid

Grid level 4

Grid level 3

Grid level 2

Coarsest grid

Grid level 1

Grid level 2

Grid level 3

Grid level 4
A linear problem:

\[ Au = b, \]  \hspace{1cm} (1)

exact error can be obtained as

\[ E = u - v, \]  \hspace{1cm} (2)

residual can be calculated as:

\[ r = b - Av. \]  \hspace{1cm} (3)

Error equation:

\[
AE = A(u - v) \\
= Au - Av \\
= b - Av \\
= r.
\]  \hspace{1cm} (4)
Linear multigrid

**Fine grid**

\[
A^f u^f = b^f \\
\quad r^f = b^f - A^f v^f
\]

**Restriction**

\[
l_f^c r
\]

**Interpolation**

\[
l_c^f e
\]

**Coarse grid**

\[
A^c e^c = l_f^c r
\]

\[
A^f v^f = b^f \\
\quad v = e^f + v^f
\]
The Error Equation (4) does not exist in a nonlinear case

Full Approximate Scheme (FAS)

For problem on coarser grids, a modified RHS is included
Nonlinear multigrid

\[ A^f(v^f) = b^f \]
\[ r^f = b^f - A^f(v^f) \]
\[ v = e^f + v^f \]

Restrictions: \[ l^c_f v, l^c_f r \]
Interpolation: \[ l^c_c e \]

Coarse grid

\[ A^c(v^c) = l^c_f r + l^c_f v \]
\[ e^c = v^c - l^c_f v \]
A nonlinear point-wise smoother

Let’s consider our nonlinear problem:

\[ A(v) = f. \]

It can be rewritten as:

\[ \mathcal{F}(v) = 0. \]

Then the Newton-like nonlinear point-wise smoother at a particular grid point \((i, j) \in \Omega\) can be the following:

\[
V_{i,j}^{\ell+1,t+1} = V_{i,j}^{\ell,t+1} - \frac{\mathcal{F}(v)}{\mathcal{F}'(V_{i,j}^{\ell,t+1})}.
\]
Domain decomposition and guard cells
Droplet spreading model

\[
\frac{\partial h}{\partial t} = \frac{\partial}{\partial x} \left[ \frac{h^3}{3} \left( \frac{\partial p}{\partial x} - \frac{B_0}{\epsilon} \sin \alpha \right) \right] + \frac{\partial}{\partial y} \left[ \frac{h^3}{3} \left( \frac{\partial p}{\partial y} \right) \right]
\]

\[p = -\triangle (h) - \Pi (h) + B_0 h \cos \alpha\]

with Neumann boundary conditions:

\[\partial_n h = 0 \quad \partial_n p = 0 \quad \text{on} \quad \partial \Omega\]
Our solver

- Cell-centred 2nd order finite difference method
- PARAMESH library for mesh generation and AMR
- Fully implicit BDF2 method with adaptive time-stepping
- MLAT variation of FAS multigrid at each time-step
- Newton-block 2 × 2 Red-Black (weighted) Gauss-Seidel smoother
- Full weighting restriction and bilinear interpolation
- Parallelism achieved using MPI
Newton-block smoother

Update at a grid point \((i, j)\):

\[
\begin{pmatrix}
  h_{\ell+1,t+1} \\
  p_{\ell+1,t+1}
\end{pmatrix}_{i,j} = \begin{pmatrix}
  h^{\ell,t+1} \\
  p^{\ell,t+1}
\end{pmatrix}_{i,j} - \begin{pmatrix}
  \frac{\partial F_h}{\partial h_{i,j}} & \frac{\partial F_h}{\partial p_{i,j}} \\
  \frac{\partial F_p}{\partial h_{i,j}} & \frac{\partial F_p}{\partial p_{i,j}}
\end{pmatrix}^{-1} \begin{pmatrix}
  F_h_{i,j}(h, p) \\
  F_p_{i,j}(h, p)
\end{pmatrix}
\]
Results from Gaskell et al. on the left and our results on the right.
A log-log plot demonstrating the linear complexity of multigrid.
Multigrid performance

Results from Gaskell et al. on the left and our results on the right.
AMR with initial condition on the left and final solution on the right.
Evolution of $\delta t$ during $T = [0, 1 \times 10^{-5}]$. 

Adaptive time-stepping
Adaptive multigrid solver

<table>
<thead>
<tr>
<th>Cases</th>
<th>No. leaf nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform $1024^2$</td>
<td>1,048,576</td>
</tr>
<tr>
<td>AMR</td>
<td>168,480</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cases</th>
<th>No. time steps</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed $\delta t$</td>
<td>1000</td>
<td>16721.3</td>
</tr>
<tr>
<td>ATS</td>
<td>45</td>
<td>574.4</td>
</tr>
</tbody>
</table>
Parallel efficiency vs. Actual computational time

- AMR 32x32x32 – 256x256x256
- AMR 64x64x64 – 256x256x256

Number of cores vs. Computational time (seconds)

Parallel efficiency

Number of cores

"AMR 32x32x32 – 256x256x256"
"AMR 64x64x64 – 256x256x256"
Tumour modelling - avascular tumour growth

- Starts with a small cluster of cells
- Nutrient supply through diffusion
- Internal adhesion force
- Three layers of cells:
  - Proliferative cells
  - Dormant cells
  - Dead cells (necrosis)
- Volume loss in necrotic core

source: www.bioinfo.de
Tumour modelling - vascular tumour growth

- TAF chemical factor
- Inducing blood vessel (angiogenesis)
- Exponential growth rate
- Develop secondaries through metastasis

source: www.maths.dundee.ac.uk
Tumour model from Wise et al.

\[
\begin{align*}
\partial_t \phi_T &= M \nabla \cdot (\phi_T \nabla \mu) + S_T(\phi_T, \phi_D, n) - \nabla \cdot (u_S \phi_T) \\
\mu &= f'(\phi_T) - \epsilon^2 \Delta \phi_T \\
\partial_t \phi_D &= M \nabla \cdot (\phi_D \nabla \mu) + S_D(\phi_T, \phi_D, n) - \nabla \cdot (u_S \phi_D) \\
u_S &= -\left(\nabla p - \frac{\gamma}{\epsilon} \mu \nabla \phi_T\right) \\
\nabla \cdot u_S &= S_T(\phi_T, \phi_D, n) \\
0 &= \Delta n + T_C(\phi_T, n) - n(\phi_T - \phi_D)
\end{align*}
\]

with mixed boundary conditions:

\[
\mu = p = 0 \quad n = 1 \quad \partial_n \phi_T = \partial_n \phi_D = 0 \quad \text{on} \quad \partial \Omega,
\]

S. M. Wise, J. S. Lowengrub, V. Cristini,

Tumour model from Wise et al.

\[
\begin{align*}
\partial_t \phi_T &= M \nabla \cdot (\phi_T \nabla \mu) + S_T(\phi_T, \phi_D, n) - \nabla \cdot (u_S \phi_T) \\
\mu &= f'(\phi_T) - \epsilon^2 \Delta \phi_T \\
\partial_t \phi_D &= M \nabla \cdot (\phi_D \nabla \mu) + S_D(\phi_T, \phi_D, n) - \nabla \cdot (u_S \phi_D) \\
[u_S &= -(\nabla p - \frac{\gamma}{\epsilon} \mu \nabla \phi_T)] \\
-\Delta p &= S_T(\phi_T, \phi_D, n) - \nabla \cdot (\frac{\gamma}{\epsilon} \mu \nabla \phi_T) \\
0 &= \Delta n + T_C(\phi_T, n) - n(\phi_T - \phi_D)
\end{align*}
\]

with mixed boundary conditions:

\[
\begin{align*}
\mu = p = 0 &\quad n = 1 &\quad \partial_n \phi_T = \partial_n \phi_D = 0 &\quad \text{on } \partial \Omega,
\end{align*}
\]

S. M. Wise, J. S. Lowengrub, V. Cristini,

Validation between results of Wise et al. and ours.
$2^{nd}$ order convergence rate

<table>
<thead>
<tr>
<th>Levels</th>
<th>Time steps</th>
<th>Infinity norm</th>
<th>Ratio</th>
<th>Two norm</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>5(128$^2$)</td>
<td>1250</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6(256$^2$)</td>
<td>2500</td>
<td>$9.118 \times 10^{-2}$</td>
<td>-</td>
<td>$7.836 \times 10^{-3}$</td>
<td>-</td>
</tr>
<tr>
<td>7(512$^2$)</td>
<td>5000</td>
<td>$1.322 \times 10^{-2}$</td>
<td>6.90</td>
<td>$1.131 \times 10^{-3}$</td>
<td>6.93</td>
</tr>
<tr>
<td>8(1024$^2$)</td>
<td>10000</td>
<td>$2.579 \times 10^{-3}$</td>
<td>5.13</td>
<td>$2.367 \times 10^{-4}$</td>
<td>4.78</td>
</tr>
<tr>
<td>9(2048$^2$)</td>
<td>20000</td>
<td>$6.415 \times 10^{-4}$</td>
<td>4.02</td>
<td>$5.833 \times 10^{-5}$</td>
<td>4.06</td>
</tr>
</tbody>
</table>

$2^{nd}$ order convergence rate seen from the model of tumour growth.
3-D results

$t=50$

$t=100$

$t=150$

$t=200$
3-D results

$t=200$

Cutting through x plane

Cutting through y plane

Cutting through z plane
Model objectives

To track the morphology of cells and reconstruct their movements:

V. Peschetola et al. *Cytoskeleton*, 2013
What is our signature

Particle tracking:
- The morphology of cells are not considered
- Manually tracking is slow
- Automatic tracking algorithms are often flawed
  - Segmentation is suboptimal for real data
  - Tracking through pattern recognition is challenging

Pure geometric math models:
- Resolution of the data matters
- Typically no cell-setting is considered
- It is a complicated procedure to obtain the results
- Computational power and advanced numerical methods have to be included for 3-D real-life cell tracking
Our optimal control model

The volume constrained mean curvature flow with forcing:

\[
\begin{cases}
\mathbf{V}(\mathbf{x}, t) = (-\sigma H(\mathbf{x}, t) + \eta(\mathbf{x}, t) + \lambda V(t)) \mathbf{v}(\mathbf{x}, t) \quad \text{on } \Gamma(t), \ t \in (0, T], \\
\Gamma(0) = \Gamma^0.
\end{cases}
\]

The phase-field approximation of the above equation - Allen-Cahn:

\[
\begin{cases}
\partial_t \phi(\mathbf{x}, t) = \triangle \phi(\mathbf{x}, t) - \frac{1}{\epsilon^2} G'(\phi(\mathbf{x}, t)) - \frac{1}{\epsilon}(\eta(\mathbf{x}, t) - \lambda(t)) \quad \text{in } \Omega \times (0, T], \\
\nabla \phi \cdot \mathbf{v}_\Omega = 0 \quad \text{on } \partial \Omega \times (0, T], \\
\phi(\cdot, 0) = \phi^0 \quad \text{in } \Omega.
\end{cases}
\]
Our optimal control model cont.

The objective functional:

\[ J(\phi, \eta) = \frac{1}{2} \int_{\Omega} (\phi(x, T) - \phi_{\text{obs}}(x))^2 \, dx + \frac{\theta}{2} \int_0^T \int_{\Omega} \eta(x, t)^2 \, dx \, dt, \]

and now we solve the minimisation problem:

\[ \min_{\eta} J(\phi, \eta), \quad \text{with } J \text{ given above.} \]
Our optimal control model cont.

The adjoint equation to help computing the derivative of the objective functional:

\[
\begin{aligned}
\partial_t p(x, t) &= -\nabla p(x, t) + \varepsilon^{-2} G''(\phi(x, t)) p(x, t) & \text{in } \Omega \times [0, T), \\
p(x, T) &= \phi(x, T) - \phi_{obs}(x) & \text{in } \Omega,
\end{aligned}
\]

and we update the control as

\[
\eta^{l+1} = \eta^l - \alpha \left( \theta \eta^l + \frac{1}{\varepsilon} p^l \right).
\]
Numerical challenges

- Number of time steps
- Memory requirement (let’s consider double precision and 100 time steps)
  - 2-D: $512^2$ requires 0.4 gigabytes
  - 3-D: $512^3$ requires 215 gigabytes
Two-grid solution strategy

One complete solve for the Allen-Cahn equation from $t=(0,T)$

One time step

Intermediate grid(s)

Restrict the converged solution of $\phi$

Fine grid for the Allen-Cahn equation

Coarse grid for the adjoint equation

One complete solve for the adjoint equation from $t=[T,0)$

One time step

Interpolate the computed $\eta$

Start the next $\eta$ iteration......
Real world example (1)

$t=0$

$t=T$

Feng Wei Yang
Seminar at Warwick
25 January 2016
Real world example (1)
Euler number for topological changes

We compute this Euler number for these time steps with an "optimized" control $\eta$:

$$
\chi = \frac{1}{2\pi(a - b)} \int_{\Omega(a,b)} \left(-\Delta \phi + \frac{\nabla|\nabla \phi|^2 \cdot \nabla \phi}{2|\nabla \phi|^2}\right) dx.
$$

Real world example (2)
A 3-D example
F.W. Yang, C.E. Goodyer, M.E. Hubbard and P.K. Jimack

“An Optimally Efficient Technique for the Solution of Systems of Nonlinear Parabolic Partial Differential Equations"

*AiES* in review, 2015

F. Yang, C. Venkataraman, V. Styles and A. Madzvamuse

“A Robust and Efficient Adaptive Multigrid Solver for the Optimal Control of Phase Field Formulations of Geometric Evolution Laws"

*CiCP* in review, 2015

F. Yang, C. Venkataraman, V. Styles, V.Kuttenberger, E. Horn, Z.von Guttenberg and A. Madzvamuse

“A Computational Framework for Particle and Whole Cell Tracking Applied to a Real Biological Dataset"

*JBM* in review, 2015