Motivation

Development and analysis of efficient and robust numerical tools to speed up computations in physical modeling.

Hour 265.5
Motivation

Development and analysis
of efficient and robust numerical tools
to speed up computations in physical modeling
Multigrid-based solvers for mixed partial differential equations

Samuel Melchior, Vincent Legat and Paul Van Dooren

joint work with Andy Wathen

http://www.climate.be/SLIM

Center for Systems Engineering and Applied Mechanics (CESAME)

Université catholique de Louvain (UCL), Belgium

October 12, 2010
Numerical simulation vs real-life experiments

Investigations in laboratory can be expensive and not practical
Numerical simulation vs real-life experiments

Need of a mathematical model of the physics

and efficient tools for solving the equations
Iterative solver for 3D ocean models

Goal: time step solely determined by the physics

Solution: Implicit time integration with a good iterative solver

At each iteration of the nonlinear solver, a linearized system $Ax = f$ must be solved using, e.g., a Krylov solver

GMRES with a block-preconditioner that takes into account the physics and the structural properties of the linear system

Step-by-step implementation (in GMSH) and analysis

1D diffusion equation
2D shallow water equations
3D steady-state incompressible Navier–Stokes equations
Coming up next...

Structure and discretization of the model

Preconditioned Krylov solvers

Geometric multigrid for the velocity block

Application on oceanic flows
Model for incompressible flows of Newtonian fluids

The steady-state incompressible Navier–Stokes equations:

\[
\rho (u \cdot \nabla) u = -\nabla p + \mu \nabla^2 u
\]

\[
\nabla \cdot u = 0
\]
Model for incompressible flows of Newtonian fluids

The steady-state incompressible Navier–Stokes equations:

\[ \rho(u \cdot \nabla)u = -\nabla p + \mu \nabla^2 u \]
\[ \nabla \cdot u = 0 \]
Model for incompressible flows of Newtonian fluids

The steady-state incompressible Navier–Stokes equations:

\[ \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = - \nabla p + \mu \nabla^2 \mathbf{u} \]
\[ \nabla \cdot \mathbf{u} = 0 \]
Model for incompressible flows of Newtonian fluids

The steady-state incompressible Navier–Stokes equations:

\[ \rho (u \cdot \nabla) u = -\nabla p + \mu \nabla^2 u \]
\[ \nabla \cdot u = 0 \]

\[ Re = \frac{\rho UL}{\mu} \]

Convection \quad Diffusion

Pressure gradient \quad Incompressibility constraint
Model for incompressible flows of Newtonian fluids

The steady-state incompressible Navier–Stokes equations:

\[ \rho (u \cdot \nabla) u = -\nabla p + \mu \nabla^2 u \]

\[ \nabla \cdot u = 0 \]

Incompressibility constraint

\[ Re = \frac{\rho UL}{\mu} \]
Model for incompressible flows of Newtonian fluids

The steady-state incompressible Navier–Stokes equations:

\[ \rho (u \cdot \nabla) u = -\nabla p + \mu \nabla^2 u \]
\[ \nabla \cdot u = 0 \]

Convection \quad Pressure\ gradient \quad Diffusion

\[ \rho (u \cdot \nabla) u \]
\[ -\nabla p + \mu \nabla^2 u \]

Incompressibility constraint

\[ Re = \frac{\text{Convection}}{\text{Diffusion}} = \frac{\rho UL}{\mu} \]

A particular case: the Stokes equations when \( Re \rightarrow 0 \)
The advection–diffusion equation \( \frac{\partial T}{\partial t} + w \cdot \nabla T = \alpha \nabla^2 T \) in \( \Omega \),

with \( T = \sum_j \phi_j T_j \), leads to the discrete Galerkin system

\[
\sum_{j=1}^n \left[ \int_\Omega \phi_i \phi_j d\Omega \frac{dT_j}{dt} + \left( \int_\Omega \phi_i (w \cdot \nabla \phi_j) d\Omega + \alpha \int_\Omega \nabla \phi_i \cdot \nabla \phi_j d\Omega \right) T_j \right] = 0 \quad \forall i
\]

or

\[
Q \frac{dT}{dt} + \left( N + \alpha A \right) T = f
\]
Discretization of the Navier–Stokes equations

Substituting \( u = \sum_{j=1}^{n_u} \phi_j u_j \) and \( p = \sum_{j=1}^{n_p} \psi_j p_j \)
into the weak form of

\[
\rho(u \cdot \nabla)u - \mu \nabla^2 u + \nabla p = 0
\]

\[-\nabla \cdot u = 0\]

yields

\[
Ax = \begin{bmatrix} F(u) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}
\]
Why is it called a saddle-point problem?

The mixed solutions of the system

\[
\begin{bmatrix}
F & B^T \\
B & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{u} \\
\mathbf{p}
\end{bmatrix} =
\begin{bmatrix}
f \\
0
\end{bmatrix}
\]

satisfies

\[ \mathcal{L}(\mathbf{u}, \mathbf{q}) \leq \mathcal{L}(\mathbf{u}, \mathbf{p}) \leq \mathcal{L}(\mathbf{v}, \mathbf{p}), \quad \forall \mathbf{v}, \mathbf{q} \]

where

\[ \mathcal{L}(\mathbf{v}, \mathbf{q}) = \frac{1}{2} \mathbf{v}^T F \mathbf{v} - \mathbf{f}^T \mathbf{v} + (B\mathbf{v})^T \mathbf{q} \]

\[ \text{Functional to minimize} \]

\[ \text{Constraint on the minimization} \]

\[ \text{Lagrange multiplier of the constraint} \]

It is well-posed if the LBB condition is satisfied.
Meshes dedicated to stratified oceanic flows

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Number of wedges</th>
<th>Number of unknowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>192</td>
<td>3216</td>
</tr>
<tr>
<td>M2</td>
<td>588</td>
<td>9053</td>
</tr>
<tr>
<td>M3</td>
<td>1998</td>
<td>28717</td>
</tr>
<tr>
<td>M4</td>
<td>4602</td>
<td>64497</td>
</tr>
</tbody>
</table>
Goal: solving PDEs with a linear complexity

Example: the 3D Stokes equations
Coming up next...

Structure and discretization of the model

Preconditioned Krylov solvers

Geometric multigrid for the velocity block

Application on oceanic flows
Krylov methods: a class of iterative solvers

**Principle:** minimization of residual in shifted Krylov subspace

\[ \mathbf{x}(k) \in \mathbf{x}(0) + \text{span}\{\mathbf{r}(0), A\mathbf{r}(0), A^2\mathbf{r}(0), \ldots, A^{k-1}\mathbf{r}(0)\} \]

\[ \mathcal{K}_k(A, \mathbf{r}(0)) \]

where \( \mathbf{r}(0) = \mathbf{f} - A\mathbf{x}(0) \)

The choice of the method and its rate of convergence ...

depend on the matrix spectrum (clustered, real, positive, ...) 

\[ \frac{\|\mathbf{r}(k)\|}{\|\mathbf{r}(0)\|} \leq \kappa(\mathcal{V}) \min_{p_k \in \Pi_k, p_k(0) = 1} \max_j |p_k(\Lambda_{jj})|, \text{ where } A = \mathcal{V} \Lambda \mathcal{V}^{-1} \]
The spectrum is the key property

Stagnation of MINRES without preconditioner

Example: the Stokes equations

Preconditioning = building an equivalent system that can be solved at a faster rate of convergence

\[ Ax = f \Rightarrow M_L^{-1} A M_R^{-1} (M_R x) = M_L^{-1} f \]
Taking the block structure into account

With the coefficient matrix written as \( A = \begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \)

and its Schur complement \( S = -BF^{-1}B^T \)

the ideal block preconditioner \( M_* = \begin{bmatrix} F & B^T \\ 0 & S \end{bmatrix} \)

satisfies \( AM_*^{-1} = \begin{bmatrix} I & 0 \\ BF^{-1} & I \end{bmatrix} \)

\( \Rightarrow \) The whole spectrum of the matrix becomes 1 by preconditioning
This ideal preconditioner is not practical

With $\mathcal{M}^{-1}_* = \begin{bmatrix} F^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & S^{-1} \end{bmatrix}$

it would converge in two expensive iterations

With $\mathcal{M}^{-1} = \begin{bmatrix} M_F^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -\Delta t B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & M_S^{-1} \end{bmatrix}$

it converges in a few cheap iterations if $M_F$ and $M_S$ are:

- easy to apply their inverse
- good approximations of $F$ and $S$
How to quantify the quality of approximations?

**Spectral equivalence**

∃ $c_1, c_2 > 0$ independent of the mesh size $h$, such that $\forall u \neq 0$:

$$c_1 \leq \frac{\langle Au, u \rangle}{\langle MAu, u \rangle} \leq c_2$$

where $A$ and $MA$ are symmetric and positive definite.

If the Richardson iteration $u_{(k)} = u_{(k-1)} + MA^{-1}(f_u - Au_{(k-1)})$ converges at a rate $\rho < 1$, $c_1$ (resp. $c_2$) equals $1 - \rho$ (resp. $1 + \rho$)

This can be achieved with **multigrid** schemes.
How to quantify the quality of approximations?

**Spectral equivalence**

\[ \exists \ c_1, c_2 > 0 \text{ independent of the mesh size } h, \text{ such that } \forall u \neq 0 : \]

\[ c_1 \leq \frac{\langle Au, u \rangle}{\langle MAu, u \rangle} \leq c_2 \]

where \( A \) and \( MA \) are symmetric and positive definite

For the steady-state Stokes equations, \( F = A \) implies that the Schur complement \( S = -BA^{-1}B^T \).

It is spectrally equivalent to the pressure mass matrix \( Q_p \) where \( c_1 \) (resp. \( c_2 \)) is the squared LBB (resp. boundedness) constant.

17
The 3D steady-state Stokes equations

The spectrum of the preconditioned matrix can be bounded

![Graphs showing the spectrum of the preconditioned matrix.](image)
The 3D steady-state Stokes equations

Irregular grid are more stable than regular ones
The 3D steady-state Stokes equations

A practical, scalable and efficient iterative solver is obtained

$\Lambda_{jj} \in S = [-a, -b] \cup [c, d]$

$\|\hat{r}_{(2k)}\| \leq 2 \left( \frac{\sqrt{ad} - \sqrt{bc}}{\sqrt{ad} + \sqrt{bc}} \right)^k \|\hat{r}_{(0)}\|$
Nonsymmetric iterations for the Stokes equations

Choice of the preconditioning side and sign of $M_S$
Preconditioned Krylov solvers

Generalization to the Navier–Stokes equations

Slower convergence for higher Reynolds numbers

\[ M_F^{-1} = F^{-1} \]
Generalization to the Navier–Stokes equations

The pressure convection–diffusion preconditioner:

The Schur complement $S$ is approximated by $M_S^{-1} = Q_p^{-1} F_p A_p^{-1}$
where $Q_p$ and $A_p$ are the pressure mass and stiffness matrix and $F_p = \mu A_p + \rho N_p$ is the nonsymmetric part of the correction

- for the Stokes problem, $F_p A_p^{-1} = \mu I$
- diagonally preconditioned Chebyshev iterations for $Q_p^{-1}$
- inverse of singular $A_p$ approximated by algebraic multigrid

Algebraic multigrid should be replaced by geometric multigrid
for stability of coarser operators (e.g. streamline diffusion)
The nonsymmetric correction improves convergence

\[ M_F^{-1} = F^{-1} \]
The nonsymmetric correction improves convergence

\[ M_S = Q \]

\[ M_S = QF_p^{-1}A_p \]

Fit: \( \Delta n = 0.825Re^{0.595} \)
Coming up next...

- Structure and discretization of the model
- Preconditioned Krylov solvers
- Geometric multigrid for the velocity block
- Application on oceanic flows
How can we take advantage of several meshes?

Numerically, the propagation is node by node

- The finer the mesh, the slower the propagation
- Only high frequency error is easy to reduce: smoothing
How can we take advantage of several meshes?

Low frequencies are high frequencies on coarser meshes
- Coarse meshes propagate faster the physical process
- Coarse grid corrections introduce high frequency error
How can we take advantage of several meshes?

Combination of meshes to reduce all frequency component

- Convergence is independent of the discretization size
- This is the textbook multigrid efficiency
Convergence in the frequency domain

Smoothing of the highest frequency components
Convergence in the frequency domain

First coarse grid correction
Convergence in the frequency domain

Smoothing of the intermediate high frequency components
Convergence in the frequency domain

Second coarse grid correction
Convergence in the frequency domain

Smoothing of the lowest high frequency components
Convergence in the frequency domain

Third coarse grid correction

Frequency number
Two main ingredients to build the algorithm

- Approximate $L_2$ projection between non-nested meshes
- Multi-directional Gauss-Seidel smoother to tackle advection

\[
\begin{align*}
\mathbf{u}(0) &\leftarrow 0, \quad \mathbf{r}(0) \leftarrow \mathbf{b}, \quad k \leftarrow 0 \\
\text{while}(\|\mathbf{r}(k)\| > \varepsilon) & \\
&\quad \hat{\mathbf{u}}(k), \hat{\mathbf{r}}(k) \leftarrow \text{few iterations of smoothing on } \mathbf{u}(k) \\
&\quad \text{solve } P^T A P e^{2h}(k) = P^T \hat{\mathbf{r}}(k) \\
&\quad \mathbf{u}(k+1) \leftarrow \hat{\mathbf{u}}(k) + P e^{2h}(k), \quad \mathbf{r}(k+1) \leftarrow \hat{\mathbf{r}}(k) - A P e^{2h}(k), \quad k \leftarrow k + 1
\end{align*}
\]
Coming up next...

Structure and discretization of the model

Preconditioned Krylov solvers

Geometric multigrid for the velocity block

Application on oceanic flows
3D baroclinic free surface SLIM model

Discontinuous Galerkin discretization on an unstructured mesh

◊ layers of prisms vertically aligned for efficient computations
◊ discontinuous interpolations for consistency and stability

Integration of the discretized horizontal momentum and continuity equations would yield a stable discrete formulation of the 2D SWE

\[
\frac{\partial u}{\partial t} + u \cdot \nabla u + f e_z \times u = -g \nabla \eta + \frac{1}{H} \nabla \cdot (H \nu_t \nabla u) + \frac{\tau}{\rho H}
\]

\[
\frac{\partial \eta}{\partial t} + \nabla \cdot (H \mathbf{u}) = 0
\]

Pseudo-time stepping used as an nonlinear iterative solver
Model problem: the 2D shallow-water equations

Assumptions: no nonlinearities

\[
\frac{\partial u}{\partial t} - \nabla \cdot (\nu_t \nabla u) + fe_z \times u + g \nabla \eta = \frac{\tau}{\rho h}
\]

\[
- \frac{\partial \eta}{\partial t} - \nabla \cdot (Hu) = 0
\]

After DG space- and Backward Euler time-discretizations

\[
Ax = \begin{bmatrix} Q_u + \Delta t(A + C') & \Delta tB^T \\ \Delta tB & -\frac{1}{gH}Q\eta \end{bmatrix} \begin{bmatrix} u \\ g\eta \end{bmatrix} = \begin{bmatrix} f_u \\ f\eta \end{bmatrix} = f
\]
Explicit multigrid for the Stommel test case

Investigation with RK33 smoothers

1 smoothing iteration

5 smoothing iterations

10 smoothing iterations

Number of iterations x 10^4

Runge Kutta 33

Multigrid V cycles
Coming up next...

Structure and discretization of the model

Preconditioned Krylov solvers

Geometric multigrid for the velocity block

Application on oceanic flows
Nested structure of the iterative solvers

Example: the steady-state Navier–Stokes equations

With efficient solvers, implicit would just be accelerated explicit
Conclusions

The linearized Navier-Stokes equations can be solved efficiently with a preconditioned GMRES method.

The convergence of the Krylov methods depends on the spectrum of the preconditioned matrix.

The preconditioner is obtained from the approximation of an ideal block-preconditioner.

In the Stokes case, bounds on the convergence of the preconditioned MINRES can be computed.

In further analysis, it should be compared to standard methods.
Practical considerations and perspectives

Matrix free solver
- No need to assemble the linear system
- Only its residual involved in order to compute the solution

Algorithm partly coded in Lua
- High flexibility on parameters and type of cycle
- Costly operations still performed in C++

Additional issues to investigate
- Effect of discontinuous discretization on convergence
- Preconditioner taking into account the Coriolis term
- Streamline diffusion on too coarse grid
and I hope...
and I hope...

Thank you for your attention!
Finite Element Method for discretization

The weak form of the heat equation \( \frac{\partial T}{\partial t} + \mathbf{w} \cdot \nabla T = \alpha \nabla^2 T \) in \( \Omega \)

is \( \frac{\partial}{\partial t} \left< T \hat{T} \right> + \left< (\mathbf{w} \cdot \nabla)T \hat{T} \right> + \alpha \left< \nabla T \cdot \nabla \hat{T} \right> = 0 \quad \forall \hat{T} \in \mathcal{T}_0 \)

where \( \left< \cdot \right> = \int_{\Omega} \cdot \ d\Omega \) and \( \mathcal{T}_0 \) is the space of test functions \( \hat{T} \)

If \( T = \sum_{j=1}^{n} \phi_j T_j \) and \( \hat{T} \in \mathcal{T}_0^h = \text{span}\{\phi_i\} \), it follows that

\[
\sum_{j=1}^{n} \left[ \left< \phi_i \phi_j \right> \frac{d}{dt} + \left< \phi_i (\mathbf{w} \cdot \nabla \phi_j) \right> + \alpha \left< \nabla \phi_i \cdot \nabla \phi_j \right> \right] T_j = 0, \ \forall i \in \{1, \ldots, n\}
\]

or equivalently

\[
Q \frac{dT}{dt} + (N + \alpha A) T = f
\]

\[
F
\]