Multigrid techniques applied in an optimization context

M. Mouffe
CERFACS, Toulouse

joint work with

S. Gratton       Ph. L. Toint
CERFACS, Toulouse FUNDP, Namur

...and more and more others

29 February 2008
Outline

Multigrid for linear systems

Introduction
Multigrid techniques

Recursive multilevel trust-region methods (RMTR)

Trust-region methods
Multigrid ideas in RMTR
Specificities of RMTR

Numerical results

Conclusion
Outline

▶ Multigrid for linear systems
  - Introduction
  - Multigrid techniques

Recursive multilevel trust-region methods (RMTR)
  - Trust-region methods
  - Multigrid ideas in RMTR
  - Specificities of RMTR

Numerical results

Conclusion
Outline

Multigrid for linear systems

▶ Introduction
Multigrid techniques

Recursive multilevel trust-region methods (RMTR)

Trust-region methods
Multigrid ideas in RMTR
Specificities of RMTR

Numerical results

Conclusion
Does it have a sense?

YES!

Solve a linear system $\iff$ Minimize a quadratic function

$$
\min f(x) = \frac{x^T A x}{2} - x^T b + c \iff \nabla_x f(x) = Ax - b = 0 \iff Ax = b
$$
Introduction

Why multigrid

- Solution based on discretization:
  - High accuracy ⇒ computational cost

- Use of coarse grids:
  1. find a good starting point
  2. solve a subproblem (e.g. the TR subproblem)

- Well-known for solving SPD linear systems resulting of the discretization of a continuous problem
  [W. Briggs, V.E. Henson and S. McCormick, 2000]

- Nonlinear systems
  [W. Hackbucz and A. Reusken, 1989]
Linear systems

Solve $Ax = b$

→ Choice of using an iterative method
Smoothing/Relaxation methods (Gauss-Seidel)

- Cheap ($O(n)$)
- Quick in reducing oscillatory components of the error
- Slow in reducing smooth components of the error
Solution:

Use coarser representations
Outline

Multigrid for linear systems

- Introduction
  - Multigrid techniques

Recursive multilevel trust-region methods (RMTR)

- Trust-region methods
- Multigrid ideas in RMTR
- Specificities of RMTR

Numerical results

Conclusion
Transfert operators: Geometric case

Restriction

\[ R_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_i-1} \]

Prolongation

\[ P_i : \mathbb{R}^{n_i-1} \rightarrow \mathbb{R}^{n_i} \]

\[ R_i = \sigma P_i^T \]
Multigrid techniques

Coarse problem definition:

Use the transfer operators

If \( A_i x_i = b_i \) is the linear system at level \( i \)
Then we define at level \( i - 1 \)

\[
A_{i-1} = R_i A_i P_i \\
b_{i-1} = R_i b_i
\]

(Or you already have a coarse definition of \( A \) and \( b \))
Multigrid for linear systems

Mesh refinement: Find a good starting point

- Solve the problem on the coarsest level
  ⇒ Good starting point for the next fine level

- Do the same on each level
  ⇒ Good starting point for the finest level

- Finally solve the problem on the finest level

M. Mouffe

Multigrid techniques applied in an optimization context
2-levels scheme

Residual equation at level $i$:

$$A_i e_{i,k} = r_{i,k}$$

where $e_{i,k} = \text{error}$ and $r_{i,k} = \text{residual at iteration } k$

Solving a linear system $\Rightarrow$ too expensive
2-levels scheme

Approximate the error using coarse grids

In practice:

1. Compute $A_{i-1}$ and $r_{i-1,k}$
2. Find $e_{i-1,*}$ the solution of the residual equation at level $i - 1$
3. Prolongate $e_{i-1,*}$ to define an approximation of the error at level $i$:
   \[
   e_{i,k} = P_i e_{i-1,*}
   \]
4. Correct your current iterate: $x_{i,k} = x_{i,k} - e_{i,k}$
Combining smoothing and 2-levels scheme

Error behaviour during the multigrid process:

1. Smoothing Reduces oscillatory error
2. Restriction of the problem Smooth error appears more oscillatory
3. Smoothing on the coarse grid Reduces coarse oscillatory error
   ⇒ Reduces smooth fine error
4. Prolongate the solution Oscillatory error reappears
5. Smoothing again Reduces this oscillatory error
Recursive use $\Rightarrow$ V-cycle
Multigrid techniques applied in an optimization context
Outline

Multigrid for linear systems
  Introduction
  Multigrid techniques

▸ Recursive multilevel trust-region methods (RMTR)
  Trust-region methods
  Multigrid ideas in RMTR
  Specificities of RMTR

Numerical results

Conclusion
Outline

Multigrid for linear systems

Introduction
Multigrid techniques

Recursive multilevel trust-region methods (RMTR)

▶ Trust-region methods
Multigrid ideas in RMTR
Specificities of RMTR

Numerical results

Conclusion
Why trust-region methods

▶ Newton method: local quadratic convergence
▶ Trust-region methods: Convergence for all starting point (Global convergence)
▶ Reduces to the Newton method when close enough to the solution ⇒ Quadratic convergence
▶ Overview of convergence results and algorithms [A. Conn, N. Gould and Ph. Toint, 2000]
Trust-region methods

Trust-region mechanism (Bound-constrained optimization)

- Define a model $m_k$ of the objective function $f$
- Define a trust region where the model is supposed to represent well the objective function
- **Compute a step** (TR subproblem)
  - inside the TR
  - that sufficiently reduces $m_k$
  - such that $x_k + s_k \in \{x : l \leq x \leq u\}$
- Step acceptance and TR radius $\Delta$ update related to the ratio

\[
\frac{f(x_{k+1}) - f(x_k)}{m_k(x_{k+1}) - m_k(x_k)}
\]

- Refuse the step and **shrink** the TR when the ratio is smaller than a constant
- Accept the step and **possibly enlarge** the TR when the ratio is large enough
Trust-region methods

Trust-region mechanism
Criticality measure and sufficient decrease condition

- **Criticality measure**: \( \chi_k = \min_{x_k + d \in C, ||d|| \leq 1} g(x_k)^Td \)
  - Unconstrained ⇒ Reduces to gradient norm
  - \( \chi_k = 0 \) at the exact solution

- **Stopping criterion**: \( \chi_k < tol \)

- **Sufficient decrease condition on the model**:
  \[
  m(x_{k+1}) - m(x_k) \geq \kappa \chi_k \min \left[ \frac{\chi_k}{1 + ||\nabla^2 f(x_k)||}, \Delta_k, 1 \right]
  \]

- Globally convergent algorithm
Outline

Multigrid for linear systems

Introduction
Multigrid techniques

Recursive multilevel trust-region methods (RMTR)

Trust-region methods
▶ Multigrid ideas in RMTR
Specificities of RMTR

Numerical results

Conclusion
Multigrid ideas in RMTR

- Suppose you have a set of discretizations of the objective function $f$:

  $$\{f_i\}_{i=0}^r$$

  with $f_r = f$.

- Transfer operators:

  $$R_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_i-1} \quad \text{Restriction}$$
  $$P_i : \mathbb{R}^{n_i-1} \rightarrow \mathbb{R}^{n_i} \quad \text{Prolongation}$$

- Coarse model: Need to be first order coherent

  $\Leftrightarrow$

  “tau correction” in multigrid
Multigrid ideas in RMTR

Smoothing

In multigrid: Smoothing = Solving the equations of the linear system one by one

In optimization:

- Smoothing = Solving the minimization problem along the coordinate axes $j$.
- Bound-constrained unidirectional problem (Trust-region and possibly original bounds constraints)
- The final step is defined by $s = \sum_j s_j$
Multigrid ideas in RMTR

**Smoothing** ⇒ **Sufficient decrease**

For unconstrained optimization

If the minimization begins in the direction \( \arg\max_j |g_j| \)

Convergence theory available

[S. Gratton, A. Sartenaer and Ph. Toint, 2005]

For bound-constrained optimization

If the minimization begins in the direction \( \arg\max_j g_j^T d_j \)

where \( d \) is defined by

\[
\arg\min_{x_k + d \in C} g(x_k)^T d \quad \text{with} \quad ||d|| \leq 1
\]
Coarse step

Step computation on a coarse level:

- Define $x_{i-1,0} = R_i x_{i,k}$
- Find a coarse step $s_{i-1}$ (e.g. using smoothing)
  - inside a coarse version of the TR
  - inside a coarse version of the bounds, and
  - that reduces sufficiently a coarse model $f_{i-1}$ of the objective function $f_i$
- Use $P_i$ to obtain a fine step $s_i$ by $s_i = P_i s_{i-1}$
- Coarse sufficient decrease $\Rightarrow$ fine sufficient decrease
Multigrid ideas in RMTR

Full multigrid (FMG)

As in multigrid methods:

- Mesh refinement
- V-cycles
Outline

Multigrid for linear systems

- Introduction
- Multigrid techniques

Recursive multilevel trust-region methods (RMTR)

- Trust-region methods
- Multigrid ideas in RMTR
  - Specificities of RMTR

Numerical results

Conclusion
Free cycles

Solve at a coarse level $i$ until $\chi_i < \varepsilon_i$
Descent condition

Use coarse levels only if

\[ \chi_{i-1,0} \geq \kappa_{\chi} \chi_{i,k} \]

not worth working on coarser level
if the problem is already solved there
Specificities of RMTR

Coarse problem definition

- Coarse TR: Restriction of the fine TR using $R_i$
- Coarse bounds: Gelman & Mandel’s definition

\[
[l_{r-1}]_j = [R_r x_{r,k}]_j + \max_{t = 1, \ldots, n_r} [l - x_{r,k}]_t
\]

\[
[u_{r-1}]_j = [R_r x_{r,k}]_j + \min_{t = 1, \ldots, n_r} [u - x_{r,k}]_t
\]
Nonlinear multigrid?

Multigrid: Solving a 1st order Taylor approximation
(e.g. a linear system)

⇒

RMTR is equivalent if a 2nd order Taylor model is used in the trust-region method (e.g. quadratic minimization problem)
Outline

Multigrid for linear systems
  Introduction
  Multigrid techniques

Recursive multilevel trust-region methods (RMTR)
  Trust-region methods
  Multigrid ideas in RMTR
  Specificities of RMTR

Numerical results

Conclusion
Some treated problems

- Quadratic minimization:
  \[
  \min_u u^T Au - 2u^T b \iff Au = b
  \]

- Minimal surface:
  \[
  \min_u \int \int \sqrt{1 + u_x^2 + u_y^2} \, dx \, dy
  \]
Quadratic minimization

<table>
<thead>
<tr>
<th>Mesh Refinement</th>
<th>RMTR$_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nb iter</td>
<td>365</td>
</tr>
<tr>
<td>Nb eval f</td>
<td>2</td>
</tr>
<tr>
<td>Nb eval g</td>
<td>2</td>
</tr>
<tr>
<td>Nb eval H</td>
<td>1</td>
</tr>
</tbody>
</table>
Minimal surface with obstacle

<table>
<thead>
<tr>
<th>Mesh Refinement</th>
<th>RMTR(_\infty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nb iter</td>
<td>1410</td>
</tr>
<tr>
<td>Nb eval f</td>
<td>95</td>
</tr>
<tr>
<td>Nb eval g</td>
<td>85</td>
</tr>
<tr>
<td>Nb eval H</td>
<td>58</td>
</tr>
</tbody>
</table>
Outline

Multigrid for linear systems

Introduction
Multigrid techniques

Recursive multilevel trust-region methods (RMTR)

Trust-region methods
Multigrid ideas in RMTR
Specificities of RMTR

Numerical results

► Conclusion
Conclusion

RMTR:

- Optimization algorithm inspired by multigrid ideas
- Adapts multigrid techniques
- Proved globally convergent
- Adapted to bound-constrained problems
- Very efficient on discretized problems
Thank you!