Recursive trust-region methods for multilevel nonlinear optimization

Philippe Toint\textsuperscript{1}  Annick Sartenaer\textsuperscript{1}  Serge Gratton\textsuperscript{2}

\textsuperscript{1}Department of Mathematics, University of Namur, Belgium

( philippe.toint@fundp.ac.be )

\textsuperscript{2}CERFACS, Toulouse, France

Beijing, September 25-28, 2006
Outline for Annick’s talk and mine

1. A recursive multilevel trust-region algorithm
2. First-order convergence results
3. A practical recursive algorithm
4. Second-order convergence
5. Numerical experience
6. Ongoing work
Outline

1. A recursive multilevel trust-region algorithm
2. First-order convergence results
3. A practical recursive algorithm
4. Second-order convergence
5. Numerical experience
6. Ongoing work
The problem

\[ \min_{x \in \mathbb{R}^n} f(x) \]  

- \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) nonlinear, twice-continuously differentiable and bounded below
- No convexity assumption
- (1) results from the discretization of some infinite-dimensional problem on a relatively fine grid for instance (\( n \) large)

\[ \rightarrow \text{Iterative search of a first-order critical point } x_* \quad (\text{s.t. } \nabla f(x_*) = 0) \]
Basic trust-region algorithm

At iteration \( k \) (at \( x_k \)):

1. Define a local model \( m_k(x_k + s) \) of \( f \) around \( x_k \) (Taylor’s model).
2. Compute a candidate step \( s_k \) that (approximately) solves

\[
\begin{align*}
\text{minimize } & \quad m_k(x_k + s) \\
\text{subject to } & \quad \|s\| \leq \Delta_k
\end{align*}
\]

3. Compute \( f(x_k + s_k) \) and \( \rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} \).
4. Update the iterate \( x_k \) and the trust-region radius \( \Delta_k \).

\[
x_{k+1} = \begin{cases} 
  x_k + s_k & \text{if } \rho_k \geq \eta_1 \\
  x_k & \text{if } \rho_k < \eta_1
\end{cases}
\]

\[
\Delta_{k+1} = \begin{cases} 
  \max(\alpha_2\|s_k\|, \Delta_k) & \text{if } \rho_k \geq \eta_2 \\
  \Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2) \\
  \alpha_1\|s_k\| & \text{if } \rho_k < \eta_1
\end{cases}
\]

where \( 0 < \eta_1 \leq \eta_2 < 1 \) and \( 0 < \alpha_1 < 1 < \alpha_2 \).
Dominating cost per iteration

- Computation of $f(x_k + s_k)$ and its derivatives
- Numerical solution of the subproblem

$$\begin{array}{ll}
\minimize_{s \in \mathbb{R}^n} & m_k(x_k + s) \\
\text{subject to} & \|s\| \leq \Delta_k
\end{array}$$

Assume now that

A set of alternative simplified models of $f$ is known

$\longrightarrow$ how can we exploit this knowledge to reduce the cost of solving the trust-region subproblem?
Dominating cost per iteration

- Computation of $f(x_k + s_k)$ and its derivatives
- Numerical solution of the subproblem

$$\begin{align*}
\text{minimize}_{s \in \mathbb{R}^n} \quad & m_k(x_k + s) \\
\text{subject to} \quad & ||s|| \leq \Delta_k
\end{align*}$$

Assume now that

A set of alternative simplified models of $f$ is known

how can we exploit this knowledge to reduce the cost of solving the trust-region subproblem?
Example: hierarchy of problem descriptions

Finest problem description

Restriction ↓ \( R \) \( P \uparrow \) Prolongation

Fine problem description

Restriction ↓ \( R \) \( P \uparrow \) Prolongation

... 

Restriction ↓ \( R \) \( P \uparrow \) Prolongation

Coarse problem description

Restriction ↓ \( R \) \( P \uparrow \) Prolongation

Coarsest problem description
Sources for such problems

- Parameter estimation in
  - discretized ODEs
  - discretized PDEs
- Optimal control problems
- Variational problems (minimum surface problem)
- Surface design (shape optimization)
- Data assimilation in weather forecast (different levels of physics in the models)
The minimum surface problem

\[ \min_v \int_0^1 \int_0^1 \left( 1 + (\partial_x v)^2 + (\partial_y v)^2 \right)^{\frac{1}{2}} \, dx \, dy \]

with the boundary conditions:

\[ \begin{cases} 
  f(x), & y = 0, \quad 0 \leq x \leq 1 \\
  0, & x = 0, \quad 0 \leq y \leq 1 \\
  f(x), & y = 1, \quad 0 \leq x \leq 1 \\
  0, & x = 1, \quad 0 \leq y \leq 1 
\end{cases} \]

where

\[ f(x) = x \ast (1 - x) \]
The solution at different levels

$n = 3^2 = 9$

$n = 7^2 = 49$

$n = 15^2 = 225$

$n = 31^2 = 961$

$n = 63^2 = 3969$

$n = 127^2 = 16129$
The framework

Assume that

- we know a collection of functions \( \{f_i\}_{i=0}^r \) s.t. \( f_i : \mathbb{R}^{n_i} \to \mathbb{R} \in C^2 \) and \( n_i \geq n_{i-1} \)
- \( n_r = n \) and \( f_r(x) = f(x) \) for all \( x \in \mathbb{R}^n \)

such that, for each \( i = 1, \ldots, r \)

- \( f_i \) is “more costly” to minimize than \( f_{i-1} \)
- there exist full-rank linear operators:

\[
\begin{align*}
R_i &: \mathbb{R}^{n_i} \to \mathbb{R}^{n_{i-1}} \quad \text{(the restriction)} \\
P_i &: \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i} \quad \text{(the prolongation)}
\end{align*}
\]

such that \( \sigma_i P_i = R_i^T \quad (\sigma_i = \|P_i\|^{-1}) \)

Terminology

- a particular \( i \) is referred to as a level
- a subscript \( i \) is used to denote a quantity corresponding to the \( i \)-th level
The idea

\[
\min_{x \in \mathbb{R}^n} f_R(x) = f(x) \quad \rightarrow \quad \text{at } x_k: \begin{cases} 
\min_{s \in \mathbb{R}^n} \quad m_k(x_k + s) = f_R(x_k) + \nabla f_R(x_k)^T s + \frac{1}{2} s^T H_k s \\
\text{s.t. } \|s\| \leq \Delta_k
\end{cases}
\]

\[
\text{or (whenever suitable)}
\]

\[
\begin{align*}
\text{at } x_k: \\
\text{Compute } \nabla f_R(x_k) \text{ (possibly } H_k) & \quad \text{Candidate step } s_k \\
\text{Restriction } \downarrow R & \quad P \uparrow \text{Prolongation}
\end{align*}
\]

\[
\text{use } f_{R-1} \text{ to construct a coarse local model of } f_R \text{ and minimize it within the fine trust region } (\Delta_k)
\]

\[
\rightarrow \quad \text{If more than two levels are available } (r > 1), \text{ do this recursively}
\]
Example of recursion with 5 levels \((r = 4)\)

**Notation:** double subscript

\[ k : \text{index of the current iteration within level} \ i \]

\[ i : \text{level index} \ (0 \leq i \leq r) \]
Additional ingredients

- Construction of the coarse local models: first-order coherence
- Use of the coarse local models: coarsening condition
- Trust-region constraint preservation
Construction of the coarse models

At a given iteration \((i, k)\) with current iterate \(x_{i,k}\)

- Restrict \(x_{i,k}\) to create the starting iterate \(x_{i-1,0}\) at level \(i - 1\)

\[x_{i-1,0} = R_i x_{i,k}\]

- Define the lower level model \(h_{i-1}\) around \(x_{i-1,0}\)

\[h_{i-1}(x_{i-1,0} + s_{i-1}) \overset{\text{def}}{=} f_{i-1}(x_{i-1,0} + s_{i-1}) + v_{i-1}^T s_{i-1}\]

where

\[v_{i-1} = R_i \nabla h_i(x_{i,k}) - \nabla f_{i-1}(x_{i-1,0})\]

so that

\[\nabla h_{i-1}(x_{i-1,0}) = R_i \nabla h_i(x_{i,k})\]

\[\longrightarrow\] Coherence of first-order information
Use of the coarse models

- When $\| R_i \nabla h_i(x_{i,k}) \| \geq \kappa \| \nabla h_i(x_{i,k}) \|$ where $\kappa \in (0, \min[1, \min_i \| R_i \|])$

  (0.01)

  and

- When $\| R_i \nabla h_i(x_{i,k}) \| > \epsilon_{i-1}$ where $\epsilon_{i-1} \in (0, 1)$ is a measure of the first-order criticality for $h_{i-1}$ judged sufficient at level $i - 1$

  and

- When $i > 0$
Choosing a model

Assume that we enter level $i$ and want to (locally) minimize $h_i$ starting from $x_{i,0}$

At iteration $k$ of this minimization

- Choose a local model of $h_i$ at $x_{i,k}$:
  - Taylor’s model
  - the coarse model

- Compute a candidate step $s_{i,k}$ that generates a decrease on this model within

  $$\mathcal{B}_{i,k} = \{ s_i \mid \|s_i\|_i \leq \Delta_{i,k} \}$$

  where $\Delta_{i,k} > 0$ and $\| \cdot \|_i$ is a level-dependent norm
Using Taylor’s model

The step $s_{i,k}$ is computed such that it approximately solves

$$\begin{align*}
\text{minimize}_{s_i \in \mathbb{R}^{n_i}} & \quad m_{i,k}(x_{i,k} + s_i) = h_i(x_{i,k}) + g_{i,k}^T s_i + \frac{1}{2} s_i^T H_{i,k} s_i \\
\text{subject to} & \quad \|s_i\|_i \leq \Delta_{i,k}
\end{align*}$$

where $g_{i,k} \overset{\text{def}}{=} \nabla h_i(x_{i,k})$ and $H_{i,k} \approx \nabla^2 h_i(x_{i,k})$

The decrease of $m_{i,k}$ is understood in its usual meaning for trust-region methods, i.e., $s_{i,k}$ must satisfy the “sufficient decrease” condition:

$$m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k}) \geq \kappa_{\text{red}} \|g_{i,k}\| \min \left[ \frac{\|g_{i,k}\|}{1 + \|H_{i,k}\|}, \Delta_{i,k} \right]$$

for some $\kappa_{\text{red}} \in (0, 1)$
Using the coarse model

Defining the level-dependent norm \( \| \cdot \|_{i-1} \) by

\[
\| \cdot \|_r = \| \cdot \|_2 \quad \text{and} \quad \| s_{i-1} \|_{i-1} = \| P_i s_{i-1} \|_i \quad \text{for} \quad i = 1, \ldots, r
\]

then the lower level subproblem consists in approximately solving

\[
\begin{align*}
\text{minimize} \quad & s_{i-1} \in \mathbb{R}^{n_{i-1}} \quad h_{i-1}(x_{i-1,0} + s_{i-1}) \\
\text{subject to} \quad & \| s_{i-1} \|_{i-1} \leq \Delta_{i,k}
\end{align*}
\]

yielding a point \( x_{i-1, *} \) such that

\[
h_{i-1}(x_{i-1, *}) < h_{i-1}(x_{i-1,0})
\]

and a corresponding step \( x_{i-1, *} - x_{i-1,0} \) which is brought back to level \( i \)

\[
s_{i,k} = P_i(x_{i-1, *} - x_{i-1,0})
\]
Preserving the trust-region constraint

Trust-region radius update:

\[
\Delta_{i,k+1} = \min\left[ \Delta^+_{i,k}, \Delta_{i+1} - \|x_{i,k+1} - x_{i,0}\|_i \right]
\]

where

\[
\Delta^+_{i,k} = \begin{cases} 
\max[\alpha_2\|s_{i,k}\|, \Delta_{i,k}] & \text{if } \rho_{i,k} \geq \eta_2 \\
\Delta_{i,k} & \text{if } \rho_{i,k} \in [\eta_1, \eta_2) \\
\alpha_1\|s_{i,k}\| & \text{if } \rho_{i,k} < \eta_1 
\end{cases}
\]
Algorithm RMTR($i$, $x_i, 0, g_i, 0, \Delta_{i+1}$)

(First call with arguments $r$, $x_r, 0$, $\nabla f_r(x_r, 0)$ and $\infty$)

Step 0: Initialization

- Compute $v_i$ and $h_i(x_i, 0)$
- Set $\Delta_{i, 0} = \Delta_{i+1}$ (or some $\Delta_s$ if $i = r$) and $k = 0$

Step 1: Model choice

- If $i = 0$ or if $\|R_i g_i, k\| < 0.01 \|g_i, k\|$ or if $\|R_i g_i, k\| \leq \epsilon_{i-1}$, go to Step 3

- Otherwise, choose to go to Step 2 (recursive step) or to Step 3 (Taylor step)
Step 2: Recursive step computation

- Call **Algorithm RMTR**(\(i - 1, R_i x_{i,k}, R_i g_{i,k}, \Delta_{i,k}\)), yielding an approximate solution \(x_{i-1,*}\) of

\[
\begin{align*}
\text{minimize} & \quad s_{i-1} \in \mathbb{R}^{n_{i-1}} \\
\text{subject to} & \quad h_{i-1}(R_i x_{i,k} + s_{i-1}) \\ & \quad \|s_{i-1}\|_{i-1} \leq \Delta_{i,k}
\end{align*}
\]

- Define \(s_{i,k} = P_i(x_{i-1,*} - R_i x_{i,k})\)

- Set \(\delta_{i,k} = h_{i-1}(R_i x_{i,k}) - h_{i-1}(x_{i-1,*})\) and go to Step 4

Step 3: Taylor step computation

- Choose \(H_{i,k}\) and compute \(s_{i,k} \in \mathbb{R}^{n_i}\) that approximately solves

\[
\begin{align*}
\text{minimize} & \quad m_{i,k}(x_{i,k} + s_i) = h_i(x_{i,k}) + g_{i,k}^T s_i + \frac{1}{2} s_i^T H_{i,k} s_i \\
\text{subject to} & \quad \|s_i\|_i \leq \Delta_{i,k}
\end{align*}
\]

- Set \(\delta_{i,k} = m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k})\) and go to Step 4
Step 4: Acceptance of the trial point

- Compute \( h_i(x_{i,k} + s_{i,k}) \) and \( \rho_{i,k} = \frac{h_i(x_{i,k}) - h_i(x_{i,k} + s_{i,k})}{\delta_{i,k}} \)
- Define \( x_{i,k+1} = \begin{cases} x_{i,k} + s_{i,k} & \text{if } \rho_{i,k} \geq \eta_1 \text{ (successful iteration)} \\ x_{i,k} & \text{if } \rho_{i,k} < \eta_1 \end{cases} \)

Step 5: Termination

- Compute \( g_{i,k+1} \)
- If \( \|g_{i,k+1}\|_\infty \leq \epsilon_i \) or \( \|x_{i,k+1} - x_{i,0}\|_i > (1 - \epsilon)\Delta_{i+1} \)

  \[ \text{return with } x_{i,*} = x_{i,k+1} \]

Step 6: Trust-region radius update

- Set
  \[ \Delta_{i,k}^+ = \begin{cases} \max[\alpha_2\|s_{i,k}\|, \Delta_{i,k}] & \text{if } \rho_{i,k} \geq \eta_2 \text{ (very successful iteration)} \\ \Delta_{i,k} & \text{if } \rho_{i,k} \in [\eta_1, \eta_2) \\ \alpha_1\|s_{i,k}\| & \text{if } \rho_{i,k} < \eta_1 \end{cases} \]
- Set \( \Delta_{i,k+1} = \min[\Delta_{i,k}^+, \Delta_{i+1} - \|x_{i,k+1} - x_{i,0}\|_i] \)
- Increment \( k \) by one and go to Step 1
Outline

1. A recursive multilevel trust-region algorithm
2. First-order convergence results
3. A practical recursive algorithm
4. Second-order convergence
5. Numerical experience
6. Ongoing work
Global convergence and complexity

Based on the trust-region technology:

- Uses the **sufficient decrease argument** (imposed in Taylor’s iterations)
- Plus the **coarsening condition** \( \| R_i g_{i,k} \| \geq 0.01 \| g_{i,k} \| \)

**Main results:**

- **Convergence to first-order critical points** at all levels
- **Weak upper bound** \( \mathcal{O}(1/\epsilon^2_r) \) on the number of iterations to achieve a given accuracy
If iteration \((i, k)\) is a recursive iteration:

a minimization sequence at level \(i - 1\) initiated at iteration \((i, k)\)
denotes all successive iterations at level \(i - 1\) until a return is made to level \(i\)
The set $\mathcal{R}(i, k)$

At iteration $(i, k)$ we associate the set:

$$\mathcal{R}(i, k) \overset{\text{def}}{=} \{(j, \ell) \mid \text{iteration } (j, \ell) \text{ occurs within iteration } (i, k)\}$$
Key results

Consider the set

\[ \mathcal{V}(i, k) \overset{\text{def}}{=} \{ (j, \ell) \in \mathcal{R}(i, k) \mid \delta_{j,\ell} \geq c \|g_{i,k}\| \Delta_{j,\ell} \} \quad c \in (0, 1) \]

If \( x_{i,k} \) is non-critical and \( \Delta_{i,k} \) is small enough

then:

- \( \mathcal{V}(i, k) = \mathcal{R}(i, k) \)
- The total number of iterations in \( \mathcal{R}(i, k) \) is finite
- All iterations \( (j, \ell) \in \mathcal{R}(i, k) \) are very successful
- \( \Delta^+_{i,k} \geq \Delta_{i,k} \)

Because we impose nonzero tolerances \( \epsilon_i \) on the gradient norms

then:

- Each minimization sequence contains at least one successful iteration
- All the trust-region radii are bounded away from zero
Furthermore:

- The number of iterations at each level is finite
- Algorithm RMTR needs at most

\[
\left\lfloor \frac{f(x_r,0) - f_{low}}{\theta(\epsilon_{min})} \right\rfloor
\]

successful Taylor iterations at any level to obtain an iterate \( x_{r,k} \) such that

\[\|g_{r,k}\| \leq \epsilon_r\]

where

- \( \epsilon_{min} = \min_{i=0,\ldots,r} \epsilon_i \)
- \( f_{low} \) is a known lower bound on \( f \)
- \( \theta(\epsilon) = O(\epsilon^2) \) for small values of \( \epsilon \) (can be estimated)
This complexity bound in $1/\epsilon^2$ for small $\epsilon$:

- is in terms of iteration numbers, thus only implicitly accounts for the cost of computing a Taylor step
- is only modified by a constant factor if all iterations (successful and unsuccessful) are considered
- thus gives a worst case upper bound on the number of function and gradient evaluations
- is of the same order as the corresponding bound for the pure gradient method (not surprising since based on the “sufficient decrease” condition)

- involves the number of successful Taylor iterations summed up on all levels, meaning that successful such iterations at cheap low levels decrease the number of necessary expensive ones at higher levels
- does not depend on the problem dimension but on the properties of the problem and of the algorithm
Global convergence result

If Algorithm RMTR is called at the uppermost level with $\epsilon_r = 0$, then:

$$\lim_{k \to \infty} \|g_{r,k}\| = 0$$

If the trust region becomes asymptotically inactive at all levels and all $\epsilon_i$ are driven to zero, then each minimization sequence becomes infinite and:

$$\lim_{k \to \infty} \|g_{i,k}\| = 0$$

for every level $i = 0, \ldots, r$
Two comments

“Premature” termination does not affect the convergence results at the upper level provided each minimization sequence contains at least one successful iteration

One can:

- Stop a minimization sequence after a preset number of successful iterations
- Use fixed lower-iterations patterns like the V or W cycles in multigrid methods

We did not use the form of the lower levels functions \( \{f_i\}_{i=0}^{r-1} \)

One can:

- Choose \( f_i = 0 \) for \( i = 0, \ldots, r - 1 \), which implies that the lower level model \( h_{i-1}(x_{i-1,0} + s_{i-1}) \) reduces to the linear model \( (R_ig_{i,k})^T s_{i-1} \)
Outline

1. A recursive multilevel trust-region algorithm
2. First-order convergence results
3. A practical recursive algorithm
4. Second-order convergence
5. Numerical experience
6. Ongoing work
A practical RMTR algorithm

- How to efficiently compute appropriate steps at Taylor iterations?
- How to improve the coarse models to ensure second-order coherence?
- Which structure consider for the recursions?
- How to compute the starting point at the finest level?
- Which choice for the prolongation and restriction operators ($P_i$ and $R_i$)?
Taylor iterations: solving and smoothing

\[
\begin{align*}
\text{minimize}_{s_i \in \mathbb{R}^{n_i}} & \quad m_{i,k}(x_{i,k} + s_i) = h_i(x_{i,k}) + g_{i,k}^T s_i + \frac{1}{2} s_i^T H_{i,k} s_i \\
\text{subject to} & \quad \|s_i\|_i \leq \Delta_{i,k}
\end{align*}
\]

- **At the coarsest level:**
  - **Solve** using the exact Moré-Sorensen method (small dimension)

- **At finer levels:**
  - **Solve** using a Truncated Conjugate-Gradient (TCG) algorithm
  - **Smooth** using a smoothing technique from multigrid methods
    (to reduce the high frequency residual/gradient components)
SCM Smoothing

Adaptation of the Gauss-Seidel smoothing technique to optimization:

- **Sequential Coordinate Minimization (SCM smoothing)**
  
  (≡ successive one-dimensional minimizations of the model along the coordinate axes when positive curvature)

  \[
  \text{From } s_i^0 = 0 \text{ and for } j = 1, \ldots, n_i:\n  s_i^j \leftarrow \min_{\alpha} m_{i,k}(x_{i,k} + s_i^{j-1} + \alpha e_{i,j})
  \]

  where \(e_{i,j}\) is the \(j\)th vector of the canonical basis of \(\mathbb{R}^{n_i}\)

  **Cost:** 1 SCM smoothing cycle \(\approx 1\) matrix-vector product
Three issues

- How to **impose sufficient decrease in the model**?

- How to **impose the trust-region constraint**?

- What to do if a **negative curvature is encountered**?
Start the first SCM smoothing cycle by minimizing along the largest gradient component (enough to ensure sufficient decrease).

While inside the trust region, perform (at most \( p \)) SCM smoothing cycles (reasonable cost).

If the step lies outside the trust region, apply a variant of the dogleg strategy (very rare in practice).

If negative curvature is encountered during a cycle:

- Remember the step to the trust-region boundary which produces the largest model reduction during the cycle (stop the SCM smoothing).
- Select the final step as that giving the maximum reduction.
Second-order and Galerkin models

At level $i - 1$ (model for level $i$):

- **First-order coherence**

  \[ h_{i-1}(x_{i-1,0} + s_{i-1}) = f_{i-1}(x_{i-1,0} + s_{i-1}) + v_{i-1}^T s_{i-1} \]

  with $x_{i-1,0} = R_i x_{i,k}$ and $v_{i-1} = R_i g_{i,k} - \nabla f_{i-1}(x_{i-1,0})$

  \[ \Rightarrow g_{i-1,0} = \nabla h_{i-1}(x_{i-1,0}) = R_i g_{i,k} \]

- **Second-order coherence** (more costly)

  \[ h_{i-1}(x_{i-1,0} + s_{i-1}) = f_{i-1}(x_{i-1,0} + s_{i-1}) + v_{i-1}^T s_{i-1} + \frac{1}{2} s_{i-1}^T W_{i-1} s_{i-1} \]

  with $W_{i-1} = R_i H_{i,k} P_i - \nabla^2 f_{i-1}(x_{i-1,0})$

  \[ \Rightarrow \nabla^2 h_{i-1}(x_{i-1,0}) = R_i H_{i,k} P_i \]
Galerkin model (second-order coherent)

\[ h_{i-1}(x_{i-1,0} + s_{i-1}) = v_{i-1}^T s_{i-1} + \frac{1}{2} s_{i-1}^T W_{i-1} s_{i-1} \]

with

- \( v_{i-1} = R_{i}g_{i,k} \)
- \( W_{i-1} = R_{i}H_{i,k}P_{i} \)

⇒ “Restricted” version of the quadratic model at the upper level
Recursion forms

Example of recursion with 5 levels \((r = 4)\)
Free and fixed form recursions

Because:

- The convergence properties of Algorithm RMTR still hold if the minimization at lower levels \((i = 0, \ldots, r - 1)\) is stopped after the first successful iteration

  \[ \Rightarrow \text{Flexibility that allows different recursion patterns} \]

- Alternance of successful SCM smoothing iterations with recursive or TCG successful iterations (at all levels but the coarsest) is very fruitful

  \[ \Rightarrow \text{This alternance is imposed for each recursion form} \]

- TCG iterations are much more expensive than recursive iterations

  \[ \Rightarrow \text{A recursive iteration is always attempted whenever allowed} \]
  \[ \text{(i.e., when } i > 0 \text{ and } \|R_i g_{i,k}\| \geq 0.01\|g_{i,k}\| \text{ and } \|R_i g_{i,k}\| > \epsilon_{i-1}) \]
Free form recursion

- **The minimization at each level is stopped when the termination condition on the gradient norm or on the step size is satisfied** (Step 5 of Algorithm RMTR)

- **Alternance of successful SCM smoothing iterations with recursive (or TCG) iterations is imposed**
Fixed form recursion (possibly truncated)

- A maximum number of successful iterations at each level is specified

**V-form recursion:**

- One succ. SCM smoothing
- followed by
- One succ. recursive (or TCG) iteration
- followed by
- One succ. SCM smoothing

**W-form recursion:**

- One succ. SCM smoothing
- followed by
- One succ. recursive (or TCG) iteration
- followed by
- One succ. SCM smoothing
- followed by
- One succ. recursive (or TCG) iteration
- followed by
- One succ. SCM smoothing
V-form recursion

\[ f_4(= f) \text{ Level 4} \]
\[ f_3 \text{ Level 3} \]
\[ f_2 \text{ Level 2} \]
\[ f_1 \text{ Level 1} \]
\[ f_0 \text{ Level 0} \]

Smoothing \quad Solving (MS)
W-form recursion

Smoothing

Solving (MS)

Gratton, Sartenaer, Toint

Recursive Multilevel Optimization
Computing the starting point at the finest level

Use a mesh refinement technique to compute $x_{r,0}$:

- Select a random starting point $x_{0,0}$ at level 0

For $i = 0, \ldots, r - 1$

- Apply Algorithm RMTR to solve

$$\min_x f_i(x)$$

(with increasing accuracy)

- Prolongate the solution to level $i + 1$ using cubic interpolation
Prolongations and restrictions

- The prolongation $P_i$ is the linear interpolation operator

- The restriction $R_i$ is $P_i^T$ normalized to ensure that $\|R_i\| = 1$

- $P_i$ and $R_i$ are never assembled
Outline

1. A recursive multilevel trust-region algorithm
2. First-order convergence results
3. A practical recursive algorithm
4. Second-order convergence
5. Numerical experience
6. Ongoing work
Convergence to weak minimizers

- Convergence to **second-order critical points** requires the **eigen-point condition**

  \[
  \text{If } \tau_{i,k} \text{ (the smallest eigenvalue of } H_{i,k} \text{) is negative, then}
  \]

  \[
  m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k}) \geq \kappa_{\text{eip}} |\tau_{i,k}| \min[\tau_{i,k}^2, \Delta_{i,k}^2]
  \]

  where \(\kappa_{\text{eip}} \in (0, \frac{1}{2})\)

  \[\rightarrow\] Too costly to impose a posteriori on recursive iterations

- The **SCM smoothing technique** limits its exploration of the model’s curvature to the coordinate axes and thus only guarantees

  \[
  \text{If } \mu_{i,k} \text{ (the most negative diagonal element of } H_{i,k} \text{) is negative, then}
  \]

  \[
  m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k}) \geq \frac{1}{2} |\mu_{i,k}| \Delta_{i,k}^2
  \]
Asymptotic positive curvature:

- along the coordinate axes at the finest level \( i = r \)
- along the prolongation of the coordinate axes at levels \( i = 1, \ldots, r - 1 \)
- along the prolongation of the coarsest subspace \( i = 0 \)

“Weak” minimizers
Outline

1. A recursive multilevel trust-region algorithm
2. First-order convergence results
3. A practical recursive algorithm
4. Second-order convergence
5. Numerical experience
6. Ongoing work
DN: a Dirichlet-to-Neumann transfer problem

(Lewis and Nash, 2005)

\[
\min_{a : [0, \pi] \rightarrow \mathbb{R}} \int_0^\pi (\partial_y u(x, 0) - \phi(x))^2 \, dx
\]

where \( u \) is the solution of the boundary value problem

\[
\begin{cases}
\Delta u(x, y) = 0 & \text{in } S, \\
u(x, y) = a(x) & \text{on } \Gamma, \\
u(x, y) = 0 & \text{on } \partial S \setminus \Gamma.
\end{cases}
\]

with

- \( S = \{ (x, y), \ 0 \leq x \leq \pi, \ 0 \leq y \leq \pi \} \)
- \( \Gamma = \{ (x, y), \ 0 \leq x \leq \pi, \ y = 0 \} \)

\( \phi(x) = \sum_{i=1}^{15} \sin(i \cdot x) + \sin(40 \cdot x) \)

\( \rightarrow \) The discretized problem is a 1D linear least-squares problem
DN: a Dirichlet-to-Neumann transfer problem

\[
\min_{a : [0,\pi] \rightarrow \mathbb{R}} \int_0^\pi (\partial_y u(x, 0) - \phi(x))^2 \, dx
\]

where \( u \) is the solution of the boundary value problem

\[
\begin{align*}
&\Delta u(x, y) = 0 \quad \text{in} \ S, \\
u(x, y) = a(x) \quad \text{on} \ \Gamma, \\
u(x, y) = 0 \quad \text{on} \ \partial S \setminus \Gamma.
\end{align*}
\]

with

- \( S = \{(x, y), \ 0 \leq x \leq \pi, \ 0 \leq y \leq \pi\} \)
- \( \Gamma = \{(x, y), \ 0 \leq x \leq \pi, \ y = 0\} \)

\[
\phi(x) = \sum_{i=1}^{15} \sin(i \ x) + \sin(40 \ x)
\]

\[\rightarrow\] The discretized problem is a 1D linear least-squares problem
Q2: a simple quadratic example

\[-\Delta u(x, y) = f \text{ in } S_2 \]
\[u(x, y) = 0 \text{ on } \partial S_2\]

where

- \(f\) is such that the analytical solution to the problem is
  \[u(x, y) = 2y(1 - y) + 2x(1 - x)\]
- \(S_2 = \{(x, y), 0 \leq x \leq 1, 0 \leq y \leq 1\}\)

\[\rightarrow\] 5-point finite-difference discretization:

\[A_i x = b_i \quad (A_i \text{ sym pd})\]

at level \(i\)

\[\rightarrow \min_{x \in \mathbb{R}^{n_r}} \frac{1}{2} x^T A_r x - x^T b_r\]
Q2: a simple quadratic example

\[-\Delta u(x, y) = f \text{ in } S_2 \]
\[u(x, y) = 0 \text{ on } \partial S_2\]

where

- $f$ is such that the analytical solution to the problem is
  \[u(x, y) = 2y(1-y) + 2x(1-x)\]
- $S_2 = \{(x, y), 0 \leq x \leq 1, 0 \leq y \leq 1\}$

\[\rightarrow\] 5-point finite-difference discretization:

\[A_i x = b_i \quad (A_i \text{ sym pd})\]

at level $i$

\[\min_{x \in \mathbb{R}^{nr}} \frac{1}{2} x^T A_r x - x^T b_r\]
Q3: a 3D quadratic example

\[-(1 + \sin(3\pi x)^2) \Delta u(x, y, z) = f \text{ in } S_3\]
\[u(x, y, z) = 0 \text{ on } \partial S_3\]

where

- \(f\) is such that the analytical solution to the problem is
  \[u(x, y, z) = x(1-x)y(1-y)z(1-z)\]

- \(S_3 = [0, 1] \times [0, 1] \times [0, 1]\)

\[\rightarrow\] 7-point finite-difference discretization:

\[A_i x = b_i \quad (A_i \text{ sym pd})\]

at level \(i\) (systems made symmetric)

\[\rightarrow \min_{x \in \mathbb{R}^{nr}} \frac{1}{2} x^T A_r x - x^T b_r\]
Q3: a 3D quadratic example

\[-(1 + \sin(3\pi x)^2) \ \Delta u(x, y, z) = f \ \text{in} \ S_3\]
\[u(x, y, z) = 0 \ \text{on} \ \partial S_3\]

where

- \( f \) is such that the analytical solution to the problem is
  \[u(x, y, z) = x(1 - x)y(1 - y)z(1 - z)\]
- \( S_3 = [0, 1] \times [0, 1] \times [0, 1] \)

\[\rightarrow\] 7-point finite-difference discretization:
\[A_i x = b_i \quad (A_i \ \text{sym pd})\]

at level \( i \) (systems made symmetric)
\[\rightarrow \ \min_{x \in \mathbb{R}^{n_r}} \ \frac{1}{2} x^T A_r x - x^T b_r\]
Surf: the minimum surface problem

\[
\min_v \int_0^1 \int_0^1 \left( 1 + (\partial_x v)^2 + (\partial_y v)^2 \right)^{\frac{1}{2}} \, dx \, dy
\]

with the boundary conditions

\[
\begin{aligned}
& f(x), \quad y = 0, \quad 0 \leq x \leq 1 \\
& 0, \quad x = 0, \quad 0 \leq y \leq 1 \\
& f(x), \quad y = 1, \quad 0 \leq x \leq 1 \\
& 0, \quad x = 1, \quad 0 \leq y \leq 1
\end{aligned}
\]

where

\[ f(x) = x \ast (1 - x) \]
Surf: the minimum surface problem

\[
\min_v \int_0^1 \int_0^1 \left( 1 + (\partial_x v)^2 + (\partial_y v)^2 \right)^{\frac{1}{2}} \, dx \, dy
\]

with the boundary conditions

\[
\begin{aligned}
& f(x), \quad y = 0, \quad 0 \leq x \leq 1 \\
& 0, \quad x = 0, \quad 0 \leq y \leq 1 \\
& f(x), \quad y = 1, \quad 0 \leq x \leq 1 \\
& 0, \quad x = 1, \quad 0 \leq y \leq 1
\end{aligned}
\]

where

\[
f(x) = x \ast (1 - x)
\]

\[\rightarrow\] Discretization using a finite element basis

\[\rightarrow\] Nonlinear convex problem
Inv: an inverse problem from image processing

Image deblurring problem (Vogel, 2002)

\[
\min \mathcal{J}(f) \quad \text{where} \quad \mathcal{J}(f) = \frac{1}{2} \| T f - d \|_2^2 + TV(f)
\]

where \( TV(f) \) is the discretization of the total variation function

\[
\int_0^1 \int_0^1 (1 + (\partial_x f)^2 + (\partial_y f)^2)^{\frac{1}{2}} \, dx \, dy
\]

\( \rightarrow \) Same discretization scheme than for Surf

\( \rightarrow \) Nonlinear convex problem
Inv: an inverse problem from image processing

Image deblurring problem

\[
\min J(f) \quad \text{where} \quad J(f) = \frac{1}{2} \|Tf - d\|_2^2 + TV(f)
\]

where \( TV(f) \) is the discretization of the total variation function

\[
\int_0^1 \int_0^1 (1 + (\partial_x f)^2 + (\partial_y f)^2)^{\frac{1}{2}} \, dx \, dy
\]

\( \rightarrow \) Same discretization scheme than for Surf

\( \rightarrow \) Nonlinear convex problem

(Vogel, 2002)
Vogel’s problem data and result
Opt: an optimal control problem

Solid ignition problem

(Borzi and Kunisch, 2006)

\[
\min_f \mathcal{J}(u(f), f) = \int_{S_2} (u - z)^2 + \frac{\beta}{2} \int_{S_2} (e^u - e^z)^2 + \frac{\nu}{2} \int_{S_2} f^2
\]

where

- \( S_2 = \{ (x, y), 0 \leq x \leq 1, 0 \leq y \leq 1 \} \)

\[
\begin{cases}
-\Delta u + \delta e^u &= f \quad \text{in } S_2 \\
u &= 0 \quad \text{on } \partial S_2
\end{cases}
\]

\( \nu = 10^{-5}, \delta = 6.8, \beta = 6.8, z = \frac{1}{\pi^2} \)

\[ \rightarrow \text{Discretization by finite differences in } S_2 \]

\[ \rightarrow \text{Nonlinear convex problem} \]
Opt: an optimal control problem

Solid ignition problem

\[
\min_{f} \mathcal{J}(u(f), f) = \int_{S_2} (u - z)^2 + \frac{\beta}{2} \int_{S_2} (e^u - e^z)^2 + \frac{\nu}{2} \int_{S_2} f^2
\]

where

- \( S_2 = \{(x, y), 0 \leq x \leq 1, 0 \leq y \leq 1\} \)
- \( \nu = 10^{-5}, \delta = 6.8, \beta = 6.8, z = \frac{1}{\pi^2} \)

\[ -\Delta u + \delta e^u = f \quad \text{in} \quad S_2 \]
\[ u = 0 \quad \text{on} \quad \partial S_2 \]

\[ \rightarrow \quad \text{Discretization by finite differences in } S_2 \]

\[ \rightarrow \quad \text{Nonlinear convex problem} \]
NC: a nonconvex example

Penalized version of a constrained optimal control problem

\[
\min_{u, \gamma} J(u, \gamma) = \int_{S_2} (u - u_0)^2 + \int_{S_2} (\gamma - \gamma_0)^2 + \int_{S_2} f^2
\]

where

- \( S_2 = \{(x, y), \ 0 \leq x \leq 1, \ 0 \leq y \leq 1\} \)
- \[
\begin{cases}
-\Delta u + \gamma u - f_0 &= f \quad \text{in} \ S_2 \\
u &= 0 \quad \text{on} \ \partial S_2
\end{cases}
\]
- \( \gamma_0(x, y) = u_0(x, y) = \sin(x(1-x))\sin(y(1-y)) \)
- \( -\Delta u_0 + \gamma_0 u_0 = f_0 \)

\[\rightarrow\] Discretization by finite differences

\[\rightarrow\] Nonconvex least-squares problem
A recursive multilevel trust-region algorithm
First-order convergence results
A practical recursive algorithm
Second-order convergence
Numerical experience
Ongoing work

NC: a nonconvex example

Penalized version of a constrained optimal control problem

\[
\min_{u, \gamma} J(u, \gamma) = \int_{S_2} (u - u_0)^2 + \int_{S_2} (\gamma - \gamma_0)^2 + \int_{S_2} f^2
\]

where

- \( S_2 = \{(x, y), 0 \leq x \leq 1, 0 \leq y \leq 1\} \)
- \[
\begin{aligned}
-\Delta u + \gamma u - f_0 &= f & \text{in } S_2 \\
u &= 0 & \text{on } \partial S_2
\end{aligned}
\]
- \( \gamma_0(x, y) = u_0(x, y) = \sin(x(1-x))\sin(y(1-y)) \)
- \( -\Delta u_0 + \gamma_0 u_0 = f_0 \)

\[\rightarrow\] Discretization by finite differences

\[\rightarrow\] Nonconvex least-squares problem
Comparison of three algorithms

- **AF** ("All on Finest"): standard Newton trust-region algorithm (with TCG as subproblem solver) applied at the finest level

- **MR** ("Mesh Refinement"): discretized problems solved in turn from the coarsest level to the finest one, using the same standard Newton trust-region method

  (Starting point at level $i + 1$ obtained by prolongating the solution at level $i$)

- **FM** ("Full Multilevel"): Algorithm RMTR
The default full multilevel (FM) algorithm

- Newton quadratic model at the finest level
- Galerkin models \( f_i = 0 \) at coarse levels
- W-form recursion performed at each level
- Recursive iteration always attempted when allowed
- A single smoothing cycle allowed at SCM smoothing iterations
Comparison of computational kernels

For the quadratic problems:

- Number of smoothing cycles (FM) vs number of matrix-vector products (AF, MR)

For the non-quadratic problems:

- Number of smoothing cycles (FM) vs number of matrix-vector products (MR)
- Number of $f$, $g$, and $H$ evaluations
Time performance of computational kernels

Quadratic problems
- DN
- Q1
- Q3

Non-quadratic problems
- Surf
- Inv
- Opt
- NC

Gratton, Sartenaer, Toint
Recursive Multilevel Optimization
Performance results on quadratic problems

Dirichlet–to–Neumann (1D) test–case : nb cycles/matvec

Quadratic (2D) test–case : nb cycles/matvec

Quadratic (3D) test–case : nb cycles/matvec
On quadratic problems: the number of smoothing cycles is fairly independent of the mesh size and dimension

→ Similar behaviour as the linear multigrid approach

→ The trust-region machinery introduced in the multigrid setting does not alter the property
Performance results on Surf

Minimum surface (2D) test-case : nb cycles/matvec

Minimum surface (2D) test-case : nb function eval

Minimum surface (2D) test-case : nb gradient eval

Minimum surface (2D) test-case : nb Hessian eval
Performance results on Inv

Inverse problem (2D) test-case: nb cycles/matvec

Inverse problem (2D) test-case: nb function eval

Inverse problem (2D) test-case: nb gradient eval

Inverse problem (2D) test-case: nb Hessian eval

Gratton, Sartenaer, Toint

Recursive Multilevel Optimization
Performance results on Opt

Optimal control (2D) test-case: nb cycles/matvec

Optimal control (2D) test-case: nb function eval

Optimal control (2D) test-case: nb gradient eval

Optimal control (2D) test-case: nb Hessian eval

Gratton, Sartenaer, Toint

Recursive Multilevel Optimization
Operations counts for Opt (at the finest level)

- One eval of $f = 14n_r$ flops
- One eval of $g = 56n_r$ flops
- One cycle/matvec = $10n_r$ flops

$\rightarrow$ **MR** much more expensive than **FM** because the gain in the number of smoothing cycles is much superior to the loss in $f$ and $g$ evaluations

**FM**: 4394 flops

**MR**: 148470 flops
A recursive multilevel trust-region algorithm
First-order convergence results
A practical recursive algorithm
Second-order convergence
Numerical experience
Ongoing work

Performance results on NC

Gratton, Sartenaer, Toint

Recursive Multilevel Optimization
Comparison of algorithmic variants

→ Sensibility investigation of FM

W2: two smoothing cycles per SCM smoothing iteration instead of one
W3: three smoothing cycles per SCM smoothing iteration instead of one
V1: V-form recursions instead of W-form recursions
F1: free form recursions instead of W-form recursions
LMOD: first-order coherent model rather than Galerkin model \((f_i = 0)\)
QMOD: second-order coherent model rather than Galerkin model \((f_i = 0)\)
LINT: linear rather than cubic interpolation in the initialization phase
Current comparative conclusions

→ Encouraging!

- Algorithm **RMTR** (default version **FM**) is more efficient than mesh refinement (**MR**) for large instances

- Pure quadratic recursion (Galerkin model) is very efficient

- **W-form and free form recursions** are most efficient
Outline

1. A recursive multilevel trust-region algorithm
2. First-order convergence results
3. A practical recursive algorithm
4. Second-order convergence
5. Numerical experience
6. Ongoing work
A (more natural) $\ell_\infty$ version

RMTR
- 2-norm criticality measure
- good results, but annoying trust region scaling problem (recursion)

RMTR-$\infty$
- $\infty$-norm (bound constraints)
- new criticality measure
- new possibilities for step length

An $\ell_\infty$ version (with M. Weber)
Multigrid enhancements (with D. Tomanos and M. Weber)
Constrained problems (with M. Mouffe et al.)

A recursive multilevel trust-region algorithm
First-order convergence results
A practical recursive algorithm
Second-order convergence
Numerical experience
Ongoing work
Possibility for asymmetric trust regions (more freedom)
In lower levels, can be represented as a bound constrained subproblem
We will impose that the lower level steps must remain inside the restriction of the upper level trust region: If

$$\mathcal{B}_{up} = \{x \mid l_{up} \leq x \leq u_{up}\}$$

then

$$\mathcal{B}_{low} = R\mathcal{B}_{up} = \{x \mid Rl_{up} \leq x \leq Ru_{up}\}$$

The step $s_{up} = Ps_{low}$ will not necessarily be inside the upper level trust region!
But: If $\Delta_{up} = \text{radius}(\mathcal{B}_{up})$, then

$$\|s_{up}\|_{\infty} \leq \|P\|_{\infty} \|R\|_{\infty} \Delta_{up}. $$
A new Criticality Measure

- Each lower level subproblem is constrained by the restriction of the upper level trust region; we can consider the lower level subproblem as a bound constrained optimization problem.

- Instead of evaluating $g_{\text{low}}$ to check criticality, we will look at

$$\chi(x_{\text{low}}) = \left| \min_{d \in R B_{\text{up}}} \langle g_{\text{low}}, d \rangle \right|.$$  

- We only use recursion if:

$$\chi_{\text{low}} \geq \kappa \chi \chi_{\text{up}}$$

- We have found a solution to the current level $i$ if

$$\chi < \varepsilon^\chi_i.$$
Algebraic multigrid

No need for predefined grids; lower level information is obtained automatically through a preprocessing phase which can be expensive (but usually the resolution phase is faster for a single system).

1. Problem definition
2. Choice of smoothing operator (smooth error detection)
3. Construction of intergrid operators and subgrids.
P-multigrid

Need an initial grid (not necessarily uniform); Construct the finer grids by using higher level polynomials.

- Choice of the basis: shape functions, hierarchical basis, ... with different degrees at each level
- Same number of nodes for each level
- Construction of the intergrid operators that have to interpolate correctly the basis
Bound constrained problems

minimize \( f(x) \)
subject to \( x \geq 0 \),

Issues:

- Restriction of bound constraints \( \neq \) bounds!
- Fast active set identification
- Bound compatible smoothing operator
- Criticality measure
Equality constrained problems

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad c(x) = 0,
\end{align*}
\]

Issues:
- Error in the adjoint equation
- Inexact tangential steps
- Iterative solvers
- (Filters?)
Perspectives

- More numerical experiments
- Hessian approximation schemes
- Combination with non-monotone techniques, filter methods, ...
- Real applications in data assimilation ...
- ... and much more!