Recursive Trust-Region Methods for Multiscale Nonlinear Optimization

Serge Gratton[1] Annick Sartenaer[2] Philippe Toint[2]

[1]CERFACS, Toulouse, France

[2]University of Namur, Belgium

Sparse Days at CERFACS, June 15-16, 2006

Outline

The problem

$$\min_{x \in \mathbb{R}^n} \; f(x) \qquad (1)$$

- $f : \mathbb{R}^n \to \mathbb{R}$ 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)

$\longrightarrow$ Iterative search of a first-order stationary point x_* (s.t. $\nabla f(x_*) = 0$)

Classical trust-region algorithm

At iteration k

1. Define a model $m_k(x_k + s)$ of f around x_k (Taylor's model)
2. Compute s_k that (approximately) solves $\begin{cases} \text{minimize}_{s \in \mathbb{R}^n} & m_k(x_k + s) \\ \text{subject to} & \boxed{\|s\| \leq \Delta_k} \end{cases}$
3. Compute $f(x_k + s_k)$ and $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$
4. Update the iterate and the trust-region radius

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if } \rho_k \geq 0.01 \\ x_k & \text{if } \rho_k < 0.01 \end{cases} \qquad \Delta_{k+1} = \begin{cases} \max[\,3\|s_k\|, \Delta_k] & \text{if } \rho_k \geq 0.95 \\ \Delta_k & \text{if } 0.01 \leq \rho_k < 0.95 \\ 0.5\|s_k\| & \text{if } \rho_k < 0.01 \end{cases}$$

Dominating cost per iteration

- Computation of $f(x_k + s_k)$ and its derivatives
- Numerical solution of the subproblem $\begin{cases} \text{minimize}_{s \in \mathbb{R}^n} & m_k(x_k + s) \\ \text{subject to} & \|s\| \leq \Delta_k \end{cases}$

Assume now that

A set of alternative simplified models of f is known

$\longrightarrow$ how can we exploit this knowledge to reduce the cost?

Dominating cost per iteration

- Computation of $f(x_k + s_k)$ and its derivatives
- Numerical solution of the subproblem $\begin{cases} \text{minimize}_{s \in \mathbb{R}^n} & m_k(x_k + s) \\ \text{subject to} & \|s\| \leq \Delta_k \end{cases}$

Assume now that

A set of alternative simplified models of f is known

$\longrightarrow$ how can we exploit this knowledge to reduce the cost?

Motivation

- 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

$$\min_{v} \int_0^1 \int_0^1 \left(1 + (\partial_x v)^2 + (\partial_y v)^2\right)^{\frac{1}{2}} dx\, dy$$

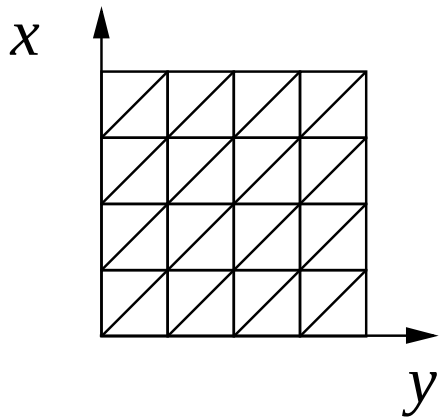
with the boundary conditions

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

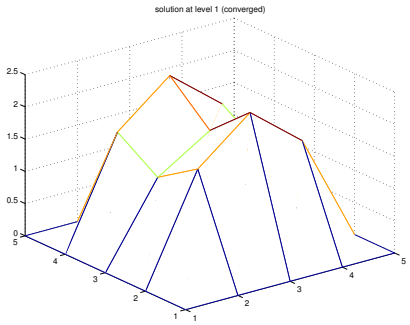
where

$$f(x) = x * (1 - x)$$

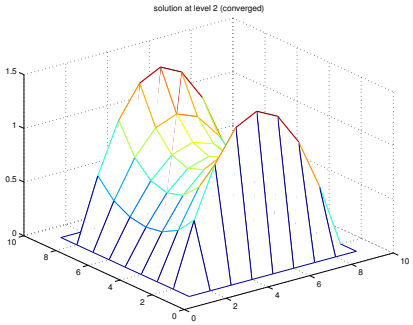
$\longrightarrow$ Discretization using a finite element basis



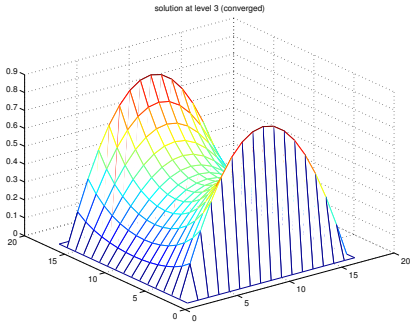
The solution at different levels



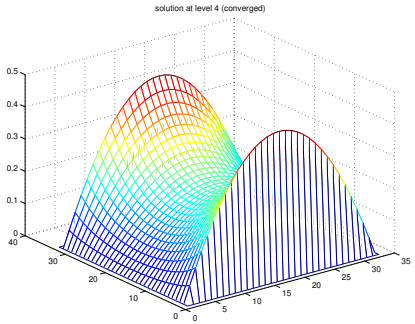
$n = 3^2 = 9$



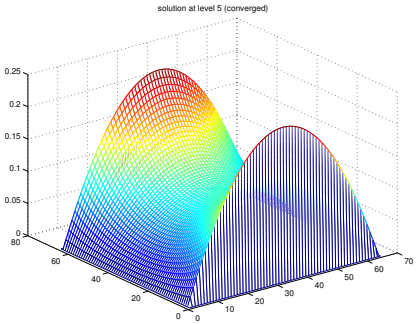
$n = 7^2 = 49$



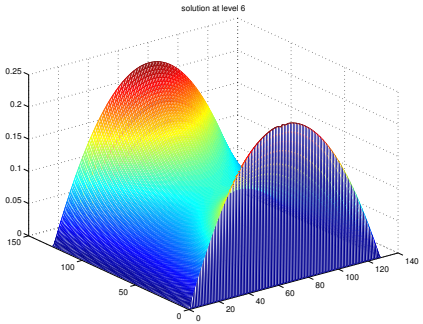
$n = 15^2 = 225$



$n = 31^2 = 961$



$n = 63^2 = 3969$



$n = 127^2 = 16129$

Recursive multiscale trust-region algorithm

Assume that

- we know a collection of functions $\{f_i\}_{i=0}^r$ s.t. $f_i : \mathbb{R}^{n_i} \to \mathbb{R} \ \in \mathcal{C}^2$ 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:

$$\left.\begin{array}{l} R_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_{i-1}} \text{ (the restriction)} \\ P_i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i} \text{ (the prolongation)} \end{array}\right\} \text{ such that } \sigma_i P_i = R_i^T \quad (\sigma_i = \|P_i\|^{-1})$$

Terminology

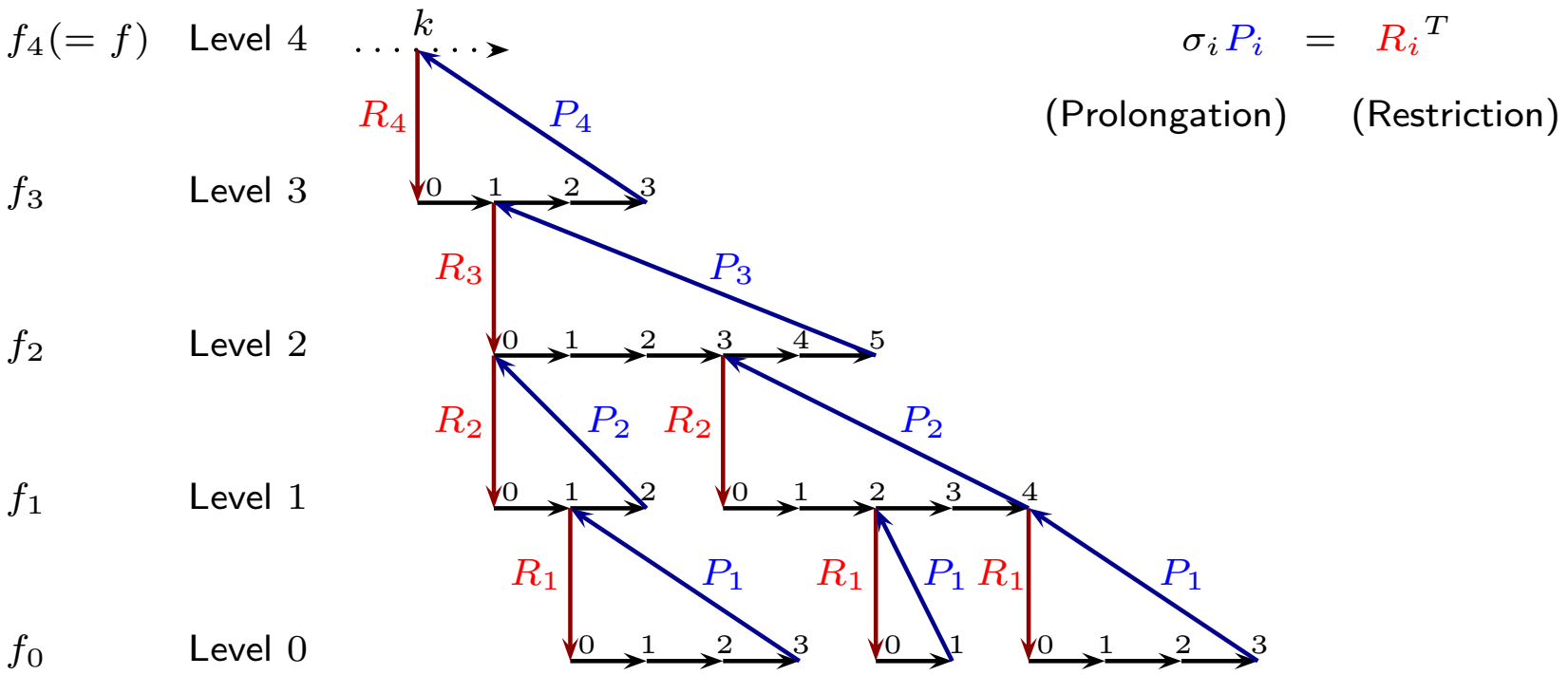
- a particular i is referred to as a level
- a subscript i is used to denote a quantity corresponding to the i-th level

The idea

- Use f_{r-1} to construct an alternative model h_{r-1} for $f_r \ \ (= f)$
  - in the neighbourhood of the current iterate
  - cheaper than a Taylor's model of f_r
- Whenever suitable use h_{r-1} to compute the step in the trust-region algorithm
- If more than two levels are available ($r > 1$), do this recursively

$\longrightarrow$ The approximation process stops in any case at level 0, where a Taylor's model of h_0 is always used

Example of recursion



Example of recursion with 5 levels $(r = 4)$

How do we construct the alternative 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 $x_{i-1,0}$

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

where

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

with $g_{i,k} = \nabla h_i(x_{i,k})$ (by convention $h_r = f_r = f$)

So that

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

$\longrightarrow$ Coherence of first-order information

and, for s_i and s_{i-1} satisfying $s_i = P_i s_{i-1}$ and by $\sigma_i P_i = R_i^T$

$$g_{i,k}^T s_i = g_{i,k}^T P_i s_{i-1} = \frac{1}{\sigma_i} g_{i,k}^T R_i^T s_{i-1} = \frac{1}{\sigma_i} g_{i-1,0}^T s_{i-1}$$

$\longrightarrow$ The first-order behaviours of h_i and h_{i-1} are coherent in a neighbourhood of $x_{i,k}$ and $x_{i-1,0}$ respectively

When can we use the alternative models?

- When $\|g_{i-1,0}\| = \|R_i g_{i,k}\|$ is large enough compared to $\|g_{i,k}\|$, that is

$$\|R_i g_{i,k}\| \geq 0.01 \, \|g_{i,k}\|$$

and

- When

$$\|R_i g_{i,k}\| > \epsilon_{i-1}$$

where $\epsilon_{i-1} \in (0,1)$ is a measure of the first-order criticality for h_{i-1} judged sufficient at level $i-1$

What do we do with the models?

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

At iteration k of this minimization

We first choose a model at iterate $x_{i,k}$ between

- **[Taylor's model]** $m_{i,k}(x_{i,k}+s_i) = h_i(x_{i,k}) + g_{i,k}^T s_i + \frac{1}{2}\, s_i^T H_{i,k} s_i$
  where $H_{i,k} \approx \nabla^2 h_i(x_{i,k})$
- **[Alternative model]** $h_{i-1}(x_{i-1,0}+s_{i-1}) = f_{i-1}(x_{i-1,0}+s_{i-1}) + v_{i-1}^T s_{i-1}$
  where $x_{i-1,0} = R_i x_{i,k}$

We next compute a 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

Taylor's model

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

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

$\rightarrow$ 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" or "Cauchy point" 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 $\kappa_{\text{red}} \in (0,1)$

Alternative model

We define the level-dependent norm $\|\cdot\|_{i-1}$ by

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

$\rightarrow$ The lower level subproblem then consists in approximately solving

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

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

$$s_{i,k} = P_i(x_{i-1,*} - x_{i-1,0})$$

Algorithm RMTR(i, $x_{i,0}$, $g_{i,0}$, Δ_{i+1})

(First call with arguments r, $x_{r,0}$, $\nabla f_r(x_{r,0})$ and ∞)

Step 0: Initialization

- Compute $v_i = g_{i,0} - \nabla f_i(x_{i,0})$ and $h_i(x_{i,0})$
- Set $\Delta_{i,0} = \Delta_{i+1}$ (or some Δ_r^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)

Algorithm RMTR(i, $x_{i,0}$, $g_{i,0}$, Δ_{i+1})

(First call with arguments r, $x_{r,0}$, $\nabla f_r(x_{r,0})$ and ∞)

Step 0: Initialization

- Compute $v_i = g_{i,0} - \nabla f_i(x_{i,0})$ and $h_i(x_{i,0})$
- Set $\Delta_{i,0} = \Delta_{i+1}$ (or some Δ_r^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

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

- 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

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

- 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 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

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

- 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

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

- 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 $x_{i,k+1} = \begin{cases} x_{i,k}+s_{i,k} & \text{if } \rho_{i,k} \geq \eta_1 \\ x_{i,k} & \text{if } \rho_{i,k} < \eta_1 \end{cases}$ (successful iteration)

Step 5: Termination

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

  return with $x_{i,*} = x_{i,k+1}$

Step 6: Trust-region radius update

- Set $\Delta_{i,k}^+ = \begin{cases} \max[\,3\|s_{i,k}\|, \Delta_{i,k}] & \text{if } \rho_{i,k} \geq 0.95 \\ \Delta_{i,k} & \text{if } 0.01 \leq \rho_{i,k} < 0.95 \\ 0.5\|s_{i,k}\| & \text{if } \rho_{i,k} < 0.01 \end{cases}$
- Set $\Delta_{i,k+1} = \min\left[\Delta_{i,k}^+,\ \Delta_{i+1} - \|x_{i,k+1}-x_{i,0}\|_i\right]$
- Increment k by one and go to Step 1

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 $x_{i,k+1} = \begin{cases} x_{i,k}+s_{i,k} & \text{if } \rho_{i,k} \geq \eta_1 \\ x_{i,k} & \text{if } \rho_{i,k} < \eta_1 \end{cases}$ (successful iteration)

Step 5: Termination

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

  return with $x_{i,*} = x_{i,k+1}$

Step 6: Trust-region radius update

- Set $\Delta_{i,k}^+ = \begin{cases} \max[\,3\|s_{i,k}\|, \Delta_{i,k}] & \text{if } \rho_{i,k} \geq 0.95 \\ \Delta_{i,k} & \text{if } 0.01 \leq \rho_{i,k} < 0.95 \\ 0.5\|s_{i,k}\| & \text{if } \rho_{i,k} < 0.01 \end{cases}$
- Set $\Delta_{i,k+1} = \min\left[\Delta_{i,k}^+,\ \Delta_{i+1} - \|x_{i,k+1}-x_{i,0}\|_i\right]$
- Increment k by one and go to Step 1

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 $x_{i,k+1} = \begin{cases} x_{i,k} + s_{i,k} & \text{if } \rho_{i,k} \geq \eta_1 \\ x_{i,k} & \text{if } \rho_{i,k} < \eta_1 \end{cases}$ (successful iteration)

Step 5: Termination

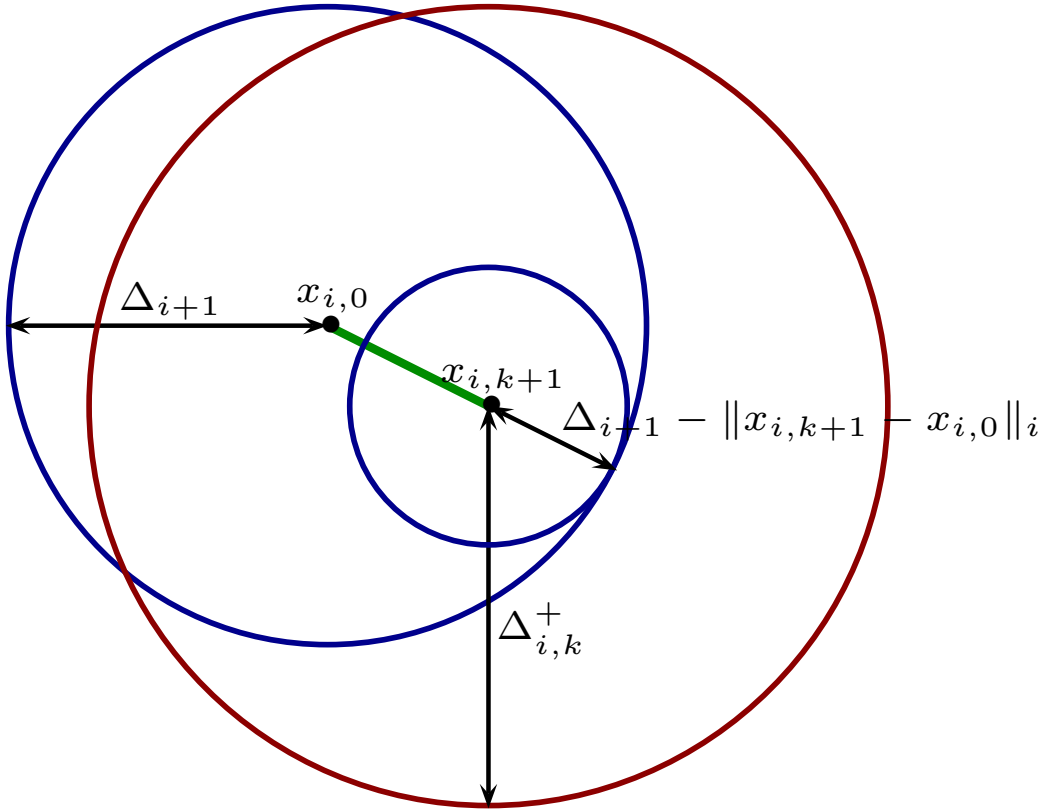
- Compute $g_{i,k+1}$
- If $\|g_{i,k+1}\|_\infty \leq \epsilon_i$ or $\|x_{i,k+1} - x_{i,0}\|_i > (1-\varepsilon)\Delta_{i+1}$

  return with $x_{i,*} = x_{i,k+1}$

Step 6: Trust-region radius update

- Set $\Delta_{i,k}^+ = \begin{cases} \max[\,3\|s_{i,k}\|, \Delta_{i,k}] & \text{if } \rho_{i,k} \geq 0.95 \\ \Delta_{i,k} & \text{if } 0.01 \leq \rho_{i,k} < 0.95 \\ 0.5\|s_{i,k}\| & \text{if } \rho_{i,k} < 0.01 \end{cases}$
- Set $\Delta_{i,k+1} = \min\left[\Delta_{i,k}^+,\ \Delta_{i+1} - \|x_{i,k+1} - x_{i,0}\|_i\right]$
- Increment k by one and go to Step 1

Trust-region constraint preservation



Trust-region radius update: $\Delta_{i,k+1} = \min\left[\Delta^+_{i,k},\ \Delta_{i+1} - \|x_{i,k+1} - x_{i,0}\|_i\right]$

A practical RMTR algorithm

- How to efficiently compute appropriate steps at Taylor iterations?
- How to improve the alternative models to ensure second-order coherence?
- Which structure consider for the recursions?
- How to dynamically manage the accuracy thresholds?
- 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

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

- At the coarsest level:
  - Solve using the exact Moré-Sorensen method (small dimension)
- At finer levels:
  - Solve using a Truncated Conjugate-Gradient (TCG) algorithm

  or
  - Smooth using a smoothing technique from multigrid methods
    (to reduce the high frequency residual/gradient components)

SCM Smoothing

$\longrightarrow$ Adaptation of the Gauss-Seidel smoothing technique to optimization:

- Sequential Coordinate Minimization (SCM smoothing)

  ($\equiv$ successive one-dimensional minimizations of the model along the coordinate axes when positive curvature)

  From $s_i^0 = 0$ and for $j = 1, \ldots, n_i$:

  $$s_i^j \Longleftarrow \min_{\alpha} m_{i,k}(x_{i,k} + s_i^{j-1} + \alpha e_{i,j})$$

  where $e_{i,j}$ is the jth 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 curvatures are encountered during one 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 $v_{i-1} = R_i g_{i,k} - \nabla f_{i-1}(x_{i-1,0})$

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

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

$$\Rightarrow \quad \nabla^2 h_{i-1}(x_{i-1,0}) = R_i H_{i,k} P_i$$

- Galerkin model (second-order coherent)

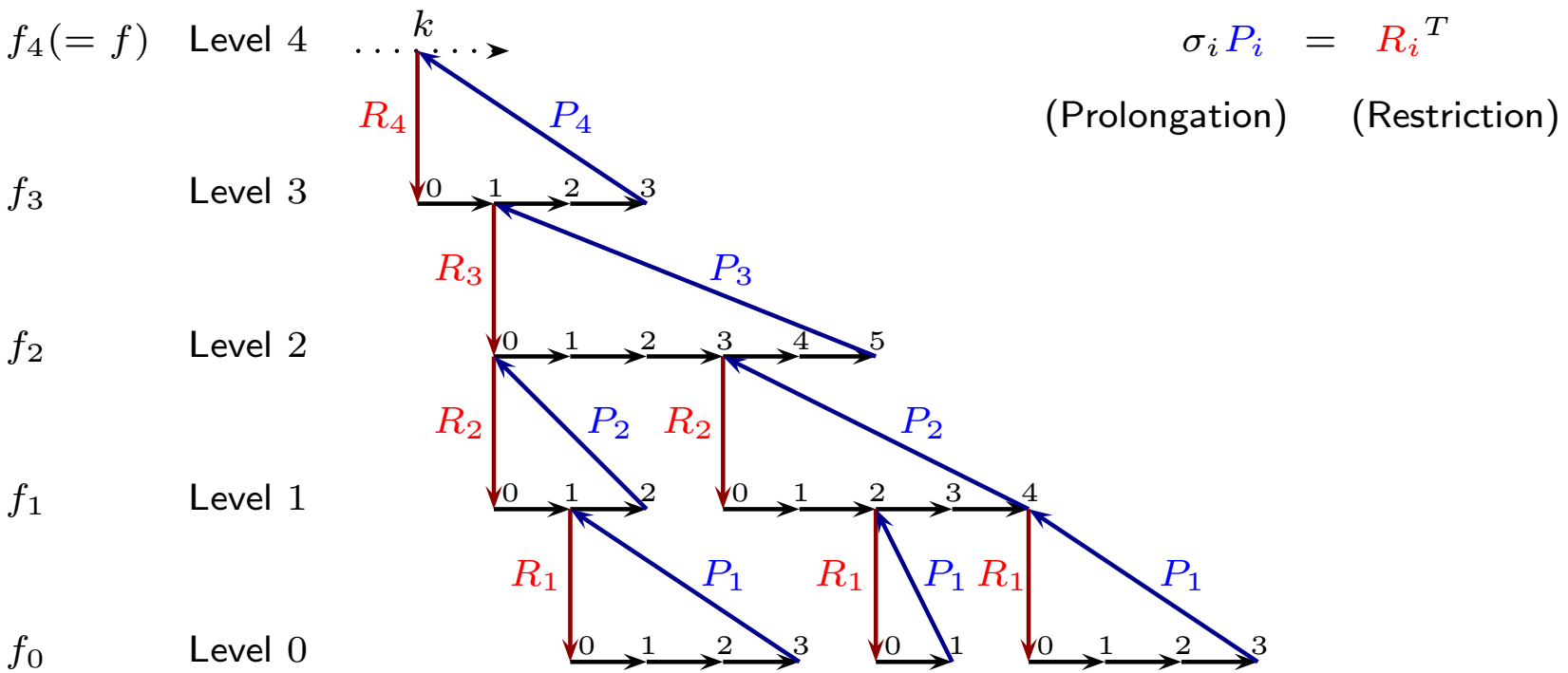
$$h_{i-1}(x_{i-1,0} + s_{i-1}) = v_{i-1}^T s_{i-1} + \frac{1}{2} s_{i-1}^T W_{i-1} s_{i-1}$$

with

- $v_{i-1} = R_i g_{i,k}$
- $W_{i-1} = R_i H_{i,k} P_i$

$\Rightarrow$ "Restricted" version of the quadratic model at the upper level

Recursion forms

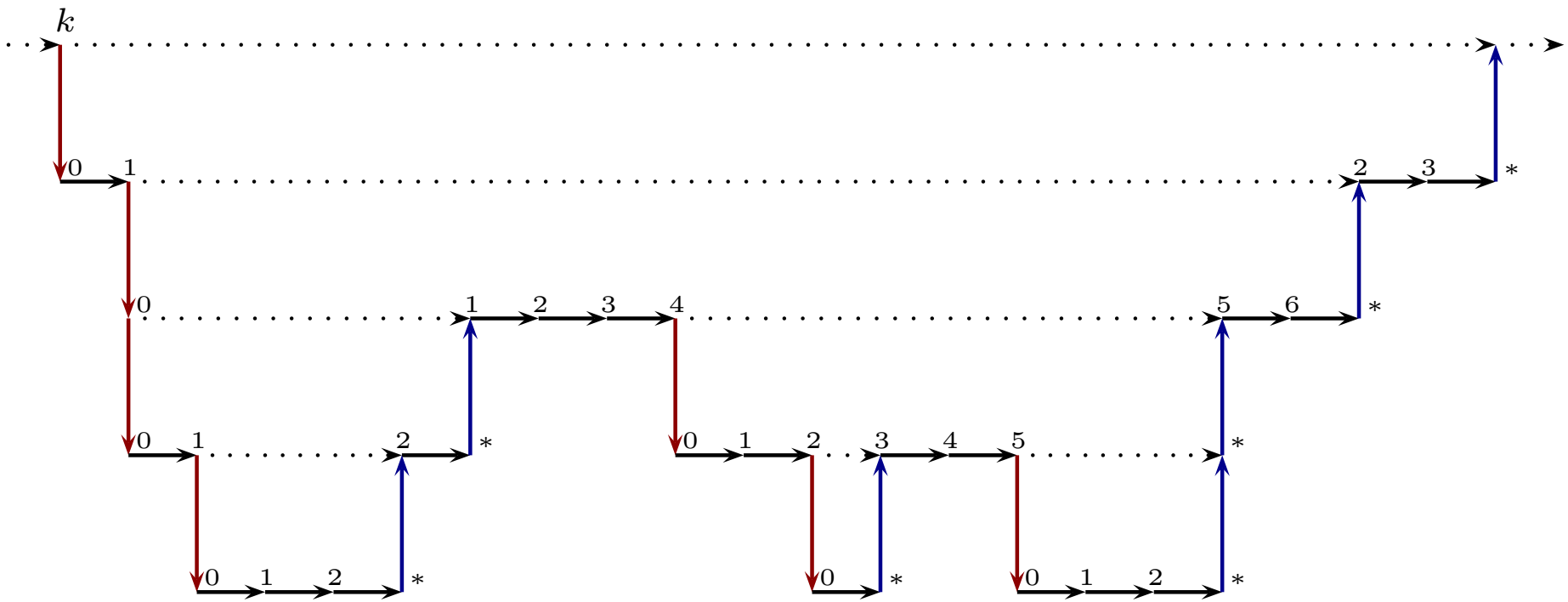


$\sigma_i P_i = {R_i}^T$

(Prolongation) (Restriction)

Example of recursion with 5 levels $(r = 4)$

Another view of the same example



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

$\Rightarrow$ 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

$\Rightarrow$ This alternance is imposed for each recursion form

- TCG iterations are much more expensive than recursive iterations

$\Rightarrow$ A recursive iteration is always attempted whenever allowed
(i.e., when $i > 0$ and $\|R_i g_{i,k}\| \geq 0.01 \|g_{i,k}\|$ 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

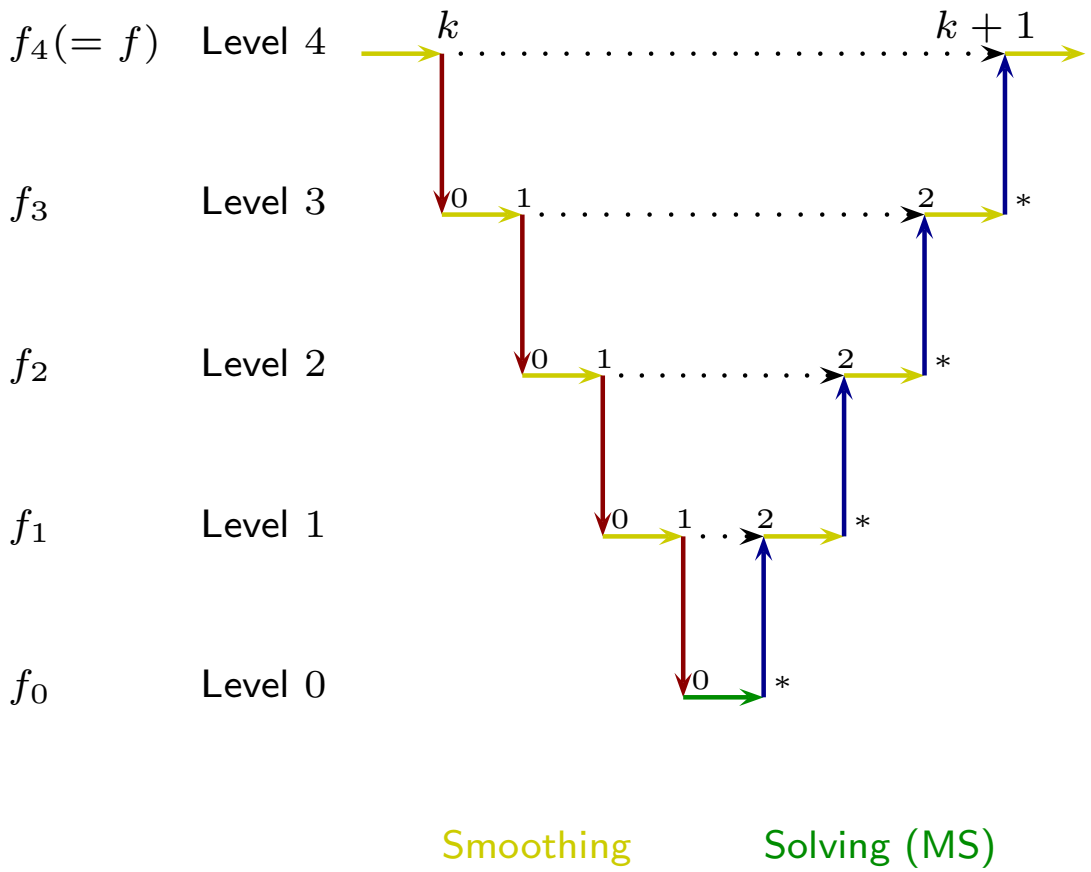
V-form recursion:

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

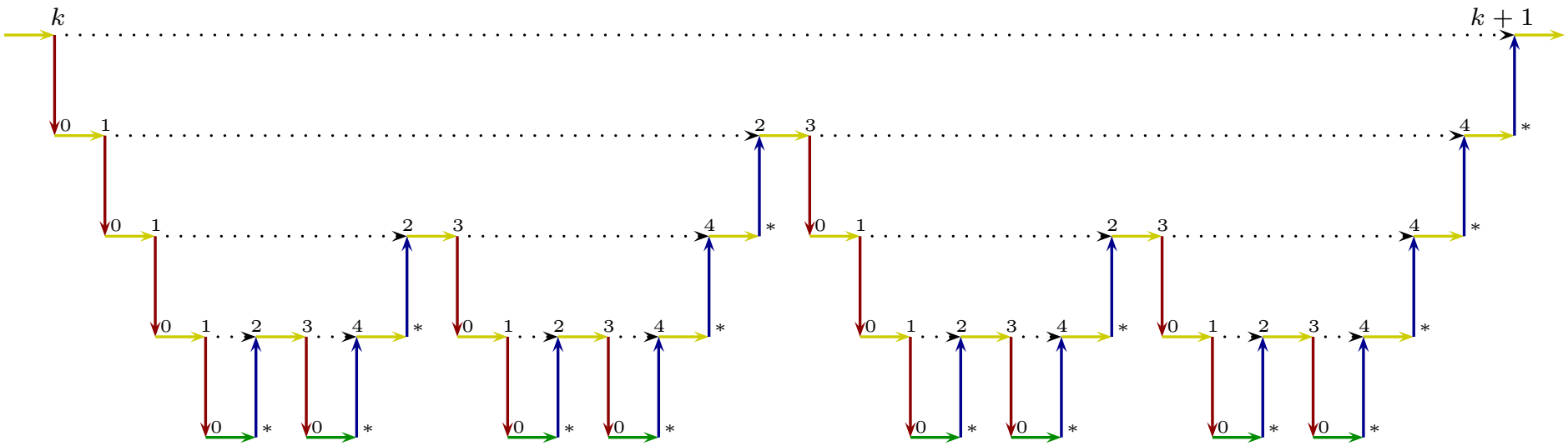
W-form recursion:

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



W-form recursion



Accuracy thresholds on the gradient

At level $i < r$

$$\epsilon_i = \min\left(0.01,\ \frac{\epsilon_{i+1}}{\nu_i^{\psi}}\right)$$

where

- ϵ_r: user-supplied gradient-accuracy requirement for the finest level
- ψ: dimension of the underlying continuous problem
- ν_i: discretization mesh-size

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

Global convergence and complexity

Based on the trust-region technology:

- Uses the sufficient decrease argument (imposed in Taylor's iterations)
- Plus a conditional 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 ($\mathcal{O}(1/\epsilon_r^2)$) on the number of iterations to achieve a given accuracy

Convergence to weak minimizers

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

> If $\tau_{i,k}$ (the smallest eigenvalue of $H_{i,k}$) 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 $\kappa_{\text{eip}} \in (0, \frac{1}{2})$

$\longrightarrow$ 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

> If $\mu_{i,k}$ (the most negative diagonal element of $H_{i,k}$) is negative, then
>
> $$m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k}) \geq \tfrac{1}{2} |\mu_{i,k}| \Delta_{i,k}^2$$

$\longrightarrow$ Asymptotic positive curvature:

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

$\longrightarrow$ "Weak" minimizers

DN: a Dirichlet-to-Neumann transfer problem

(Lewis and Nash, 2005)

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

where u is the solution of the boundary value problem

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

with

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

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

$\longrightarrow$ The discretized problem is a 1D linear least-squares problem

DN: a Dirichlet-to-Neumann transfer problem

(Lewis and Nash, 2005)

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

where u is the solution of the boundary value problem

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

with

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

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

$\longrightarrow$ The discretized problem is a 1D linear least-squares problem

Q2: a simple quadratic example

$$\begin{aligned} -\Delta u(x,y) &= f \text{ in } S_2 \\ u(x,y) &= 0 \text{ on } \partial S_2 \end{aligned}$$

where

- f is such that the analytical solution to the problem is

$$u(x,y) = 2y(1-y) + 2x(1-x)$$

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

$\longrightarrow$ 5-point finite-difference discretization:

$$A_i x = b_i \qquad (A_i \text{ sym pd})$$

at level i

$$\longrightarrow \min_{x \in \mathbb{R}^{n_r}} \frac{1}{2} x^T A_r x - x^T b_r$$

Q2: a simple quadratic example

$$\begin{aligned} -\Delta u(x,y) &= f \text{ in } S_2 \\ u(x,y) &= 0 \text{ on } \partial S_2 \end{aligned}$$

where

- f is such that the analytical solution to the problem is

$$u(x,y) = 2y(1-y) + 2x(1-x)$$

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

$\longrightarrow$ 5-point finite-difference discretization:

$$A_i x = b_i \qquad (A_i \text{ sym pd})$$

at level i

$$\longrightarrow \quad \min_{x \in \mathbb{R}^{n_r}} \frac{1}{2} x^T A_r x - x^T b_r$$

Q3: a 3D quadratic example

$$\begin{aligned} -(1+\sin(3\pi x)^2)\,\Delta u(x,y,z) &= f \text{ in } S_3 \\ u(x,y,z) &= 0 \text{ on } \partial S_3 \end{aligned}$$

where

- f is such that the analytical solution to the problem is

$$u(x,y,z) = x(1-x)y(1-y)z(1-z)$$

- $S_3 = [0,1] \times [0,1] \times [0,1]$

$\longrightarrow$ 7-point finite-difference discretization:

$$A_i x = b_i \qquad (A_i \text{ sym pd})$$

at level i (systems made symmetric)

$\longrightarrow$ $\min_{x \in \mathbb{R}^{n_r}} \frac{1}{2} x^T A_r x - x^T b_r$

Q3: a 3D quadratic example

$$\begin{aligned} -(1+\sin(3\pi x)^2)\,\Delta u(x,y,z) &= f \text{ in } S_3 \\ u(x,y,z) &= 0 \text{ on } \partial S_3 \end{aligned}$$

where

- f is such that the analytical solution to the problem is

$$u(x,y,z) = x(1-x)y(1-y)z(1-z)$$

- $S_3 = [0,1] \times [0,1] \times [0,1]$

$\longrightarrow$ 7-point finite-difference discretization:

$$A_i x = b_i \qquad (A_i \text{ sym pd})$$

at level i (systems made symmetric)

$\longrightarrow$ $\min_{x \in \mathbb{R}^{n_r}} \frac{1}{2} x^T A_r x - x^T b_r$

Surf: the minimum surface problem

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

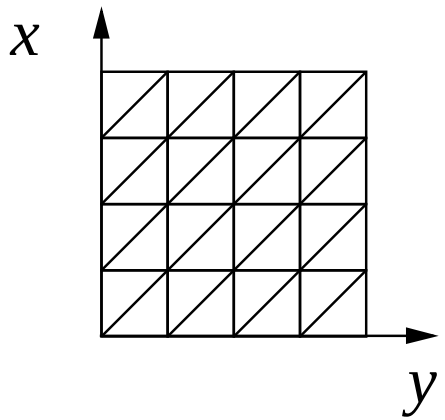
$\longrightarrow$ Discretization using a finite element basis

with the boundary conditions

$$\begin{cases} f(x), & y = 0, \quad 0 \le x \le 1 \\ 0, & x = 0, \quad 0 \le y \le 1 \\ f(x), & y = 1, \quad 0 \le x \le 1 \\ 0, & x = 1, \quad 0 \le y \le 1 \end{cases}$$

where

$$f(x) = x * (1 - x)$$



$\longrightarrow$ Nonlinear convex problem

Surf: the minimum surface problem

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

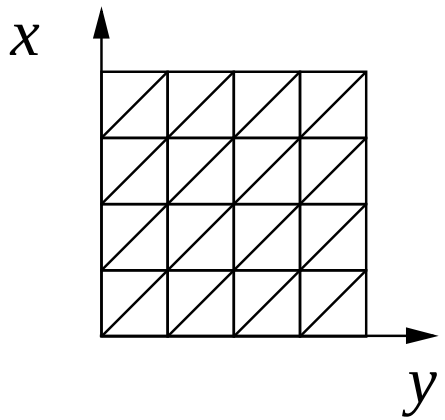
with the boundary conditions

$$\begin{cases} f(x), & y = 0, \quad 0 \le x \le 1 \\ 0, & x = 0, \quad 0 \le y \le 1 \\ f(x), & y = 1, \quad 0 \le x \le 1 \\ 0, & x = 1, \quad 0 \le y \le 1 \end{cases}$$

where

$$f(x) = x * (1 - x)$$

$\longrightarrow$ Discretization using a finite element basis



$\longrightarrow$ Nonlinear convex problem

Inv: an inverse problem from image processing

Image deblurring problem (Vogel, 2002)

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

where $TV(f)$ is the discretization of the total variation function $\longrightarrow$ Same discretization scheme than for Surf

$$\int_0^1 \int_0^1 \left(1 + (\partial_x f)^2 + (\partial_y f)^2\right)^{\frac{1}{2}} dx\, dy \quad \longrightarrow \text{ Nonlinear convex problem}$$

Inv: an inverse problem from image processing

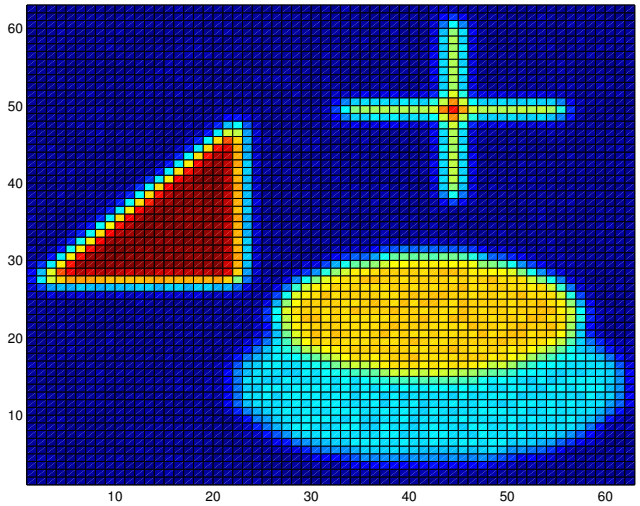
Image deblurring problem (Vogel, 2002)

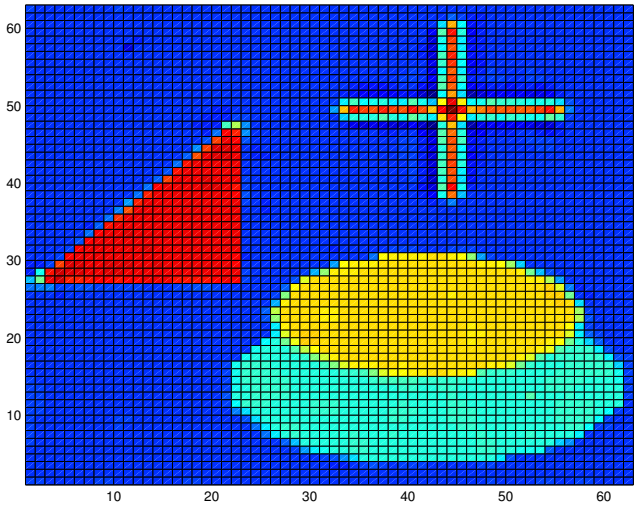
$$\min \mathcal{J}(f) \quad \text{where} \quad \mathcal{J}(f) = \frac{1}{2}\|Tf - d\|_2^2 + TV(f)$$

where $TV(f)$ is the discretization of the total variation function $\longrightarrow$ Same discretization scheme than for Surf

$$\int_0^1 \int_0^1 \left(1 + (\partial_x f)^2 + (\partial_y f)^2\right)^{\frac{1}{2}} dx\, dy \quad \longrightarrow \text{ Nonlinear convex problem}$$

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

- $S_2 = \{(x, y),\ 0 \leq x \leq 1,\ 0 \leq y \leq 1\}$ $\longrightarrow$ Discretization by finite differences in S_2
- $\begin{cases} -\Delta u + \delta e^u = f & \text{in } S_2 \\ u = 0 & \text{on } \partial S_2 \end{cases}$ $\longrightarrow$ Nonlinear convex problem
- $\nu = 10^{-5}$, $\delta = 6.8$, $\beta = 6.8$, $z = \frac{1}{\pi^2}$

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

- $S_2 = \{(x, y),\ 0 \leq x \leq 1,\ 0 \leq y \leq 1\}$ $\longrightarrow$ Discretization by finite differences in S_2
- $\begin{cases} -\Delta u + \delta e^u = f & \text{in } S_2 \\ u = 0 & \text{on } \partial S_2 \end{cases}$ $\longrightarrow$ Nonlinear convex problem
- $\nu = 10^{-5}$, $\delta = 6.8$, $\beta = 6.8$, $z = \frac{1}{\pi^2}$

NC: a nonconvex example

Penalized version of a constrained optimal control problem

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

where

- $S_2 = \{(x,y),\ 0 \le x \le 1,\ 0 \le y \le 1\}$
- $\left\{ \begin{array}{rcll} -\Delta u + \gamma u - f_0 & = & f & \text{in } S_2 \\ u & = & 0 & \text{on } \partial S_2 \end{array} \right.$
- $\begin{array}{rcl} \gamma_0(x,y) & = & u_0(x,y) \\ & = & \sin(x(1-x))\sin(y(1-y)) \end{array}$
- $-\Delta u_0 + \gamma_0 u_0 = f_0$

$\longrightarrow$ Discretization by finite differences

$\longrightarrow$ Nonconvex least-squares problem

NC: a nonconvex example

Penalized version of a constrained optimal control problem

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

where

- $S_2 = \{(x,y),\ 0 \le x \le 1,\ 0 \le y \le 1\}$
- $\left\{ \begin{array}{rcll} -\Delta u + \gamma u - f_0 & = & f & \text{in } S_2 \\ u & = & 0 & \text{on } \partial S_2 \end{array} \right.$ $\longrightarrow$ Discretization by finite differences
- $\begin{array}{rcl} \gamma_0(x,y) & = & u_0(x,y) \\ & = & \sin(x(1-x))\sin(y(1-y)) \end{array}$ $\longrightarrow$ Nonconvex least-squares problem
- $-\Delta u_0 + \gamma_0 u_0 = f_0$

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

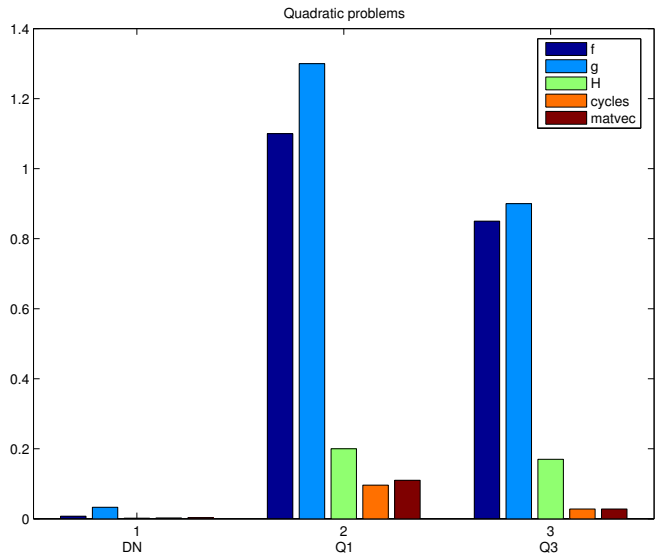
For the quadratic problems:

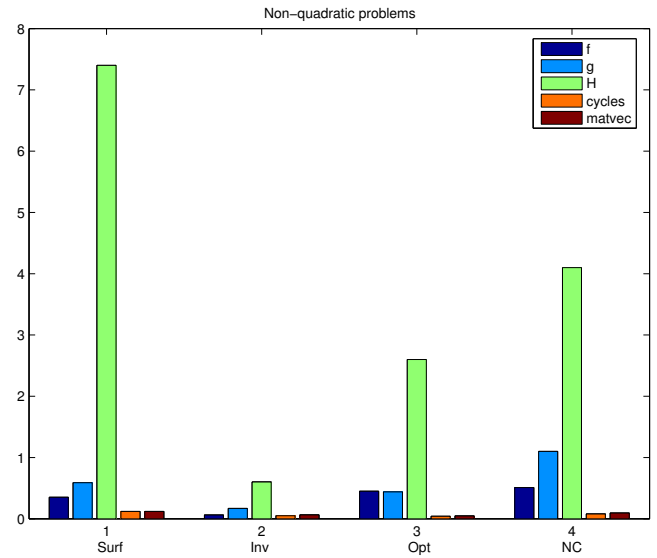
- Number of smoothing cycles (**FM**) vs number of matrix-vector products (**AF**, **MR**)

For the non-quadratic problems:

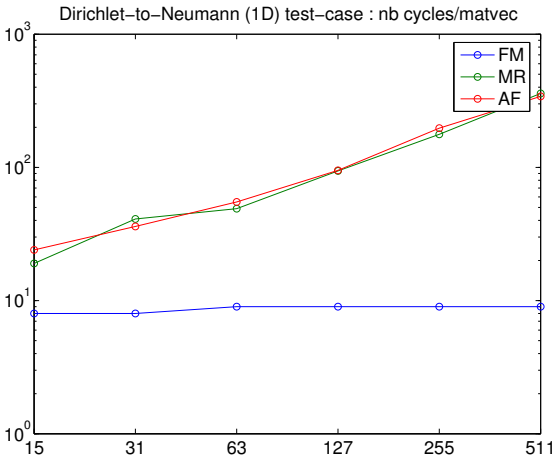
- Number of smoothing cycles (**FM**) vs number of matrix-vector products (**MR**)
- Number of f, g, and H evaluations

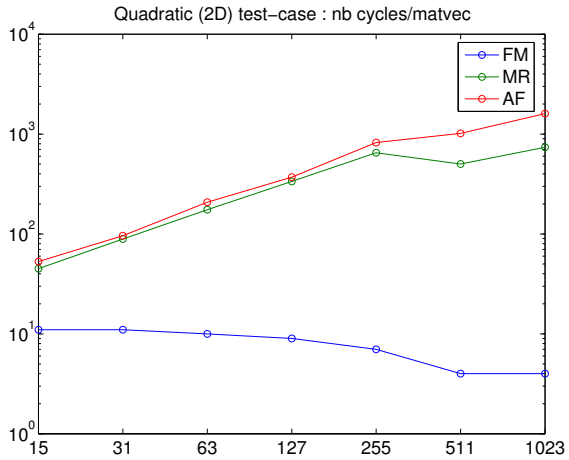
Time performance of computational kernels

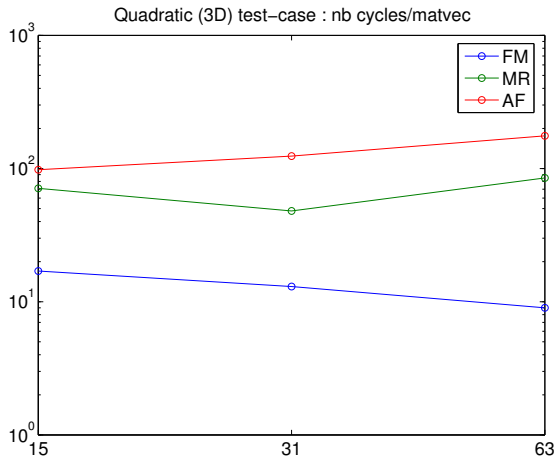




Performance results on quadratic problems





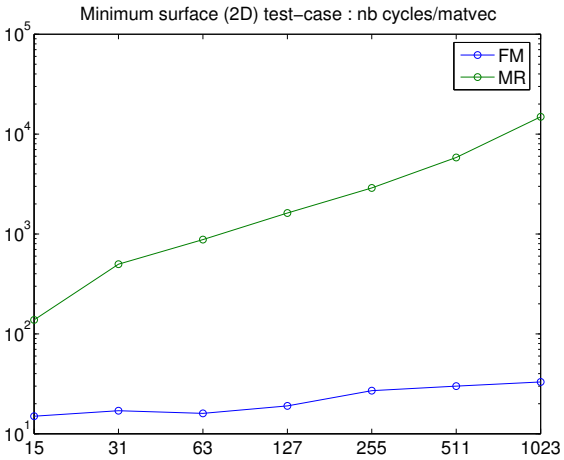


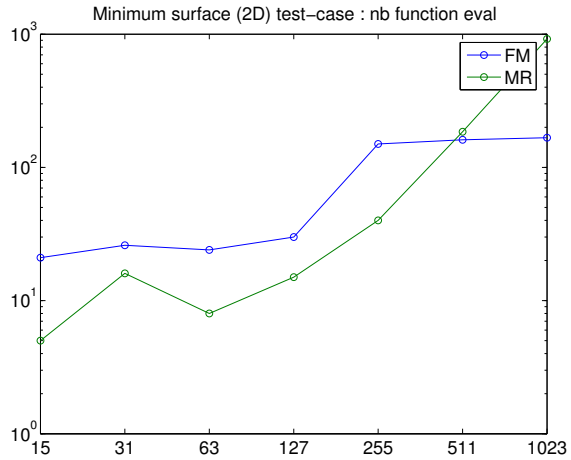
On quadratic problems: the number of smoothing cycles is fairly independent of the mesh size and dimension

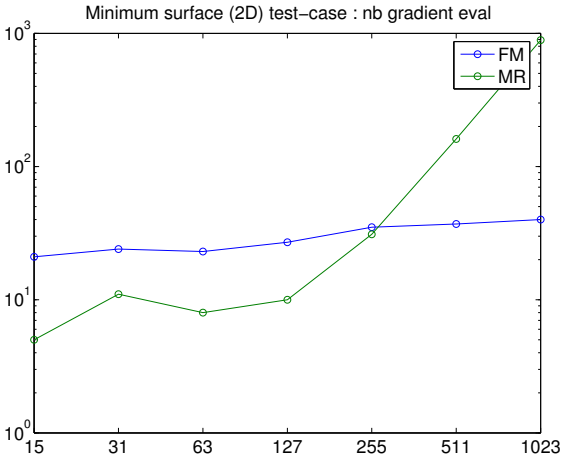
$\longrightarrow$ Similar behaviour as the linear multigrid approach

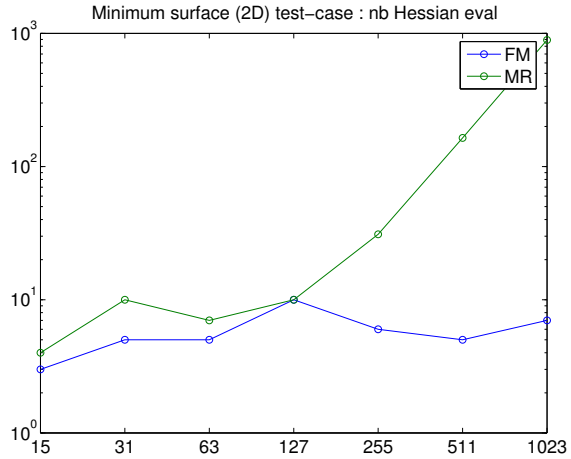
$\longrightarrow$ The trust-region machinery introduced in the multigrid setting does not alter the property

Performance results on Surf

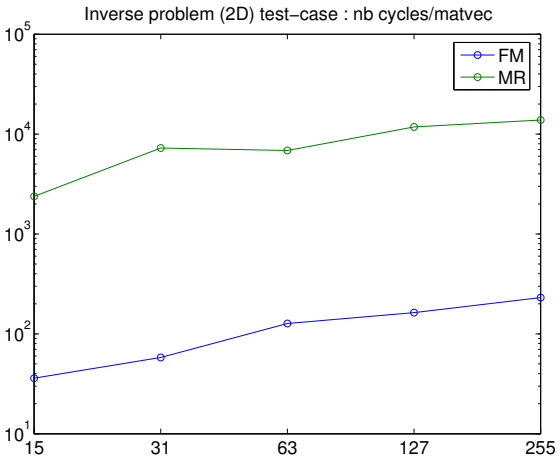


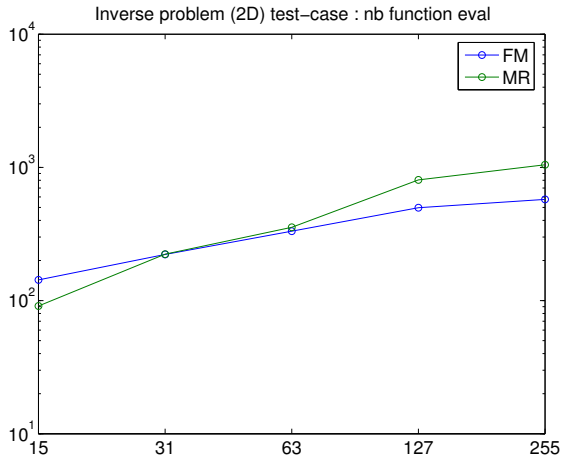


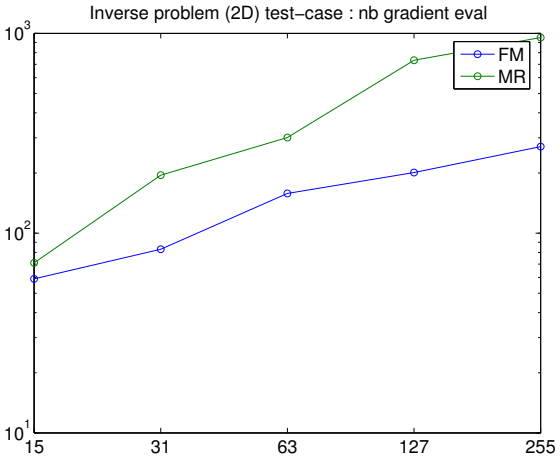


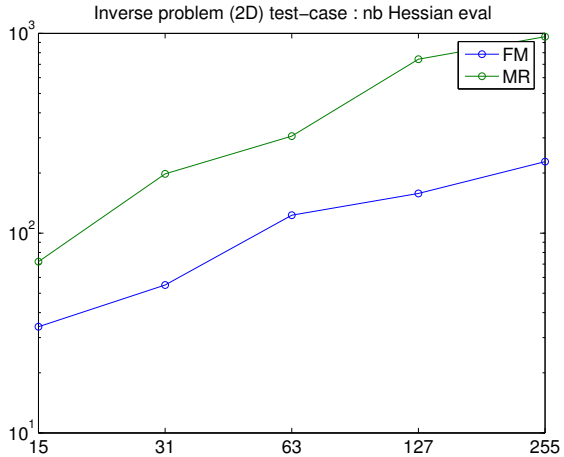


Performance results on Inv

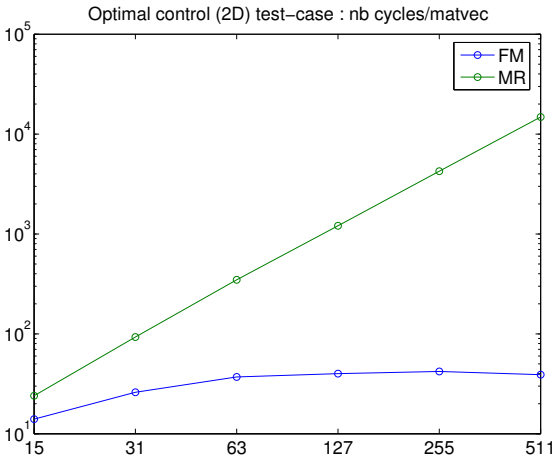


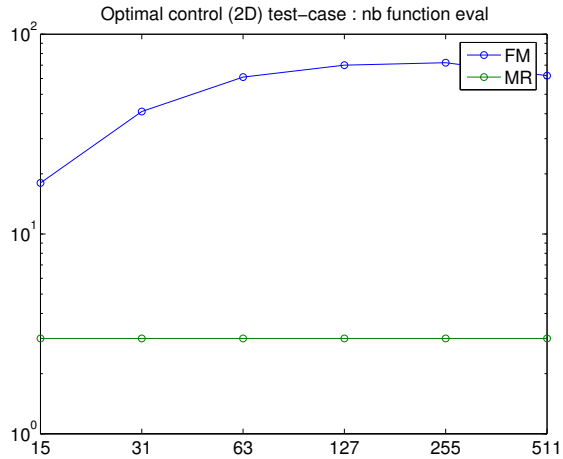


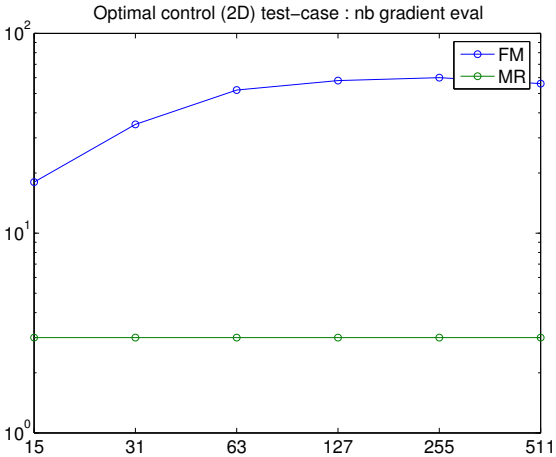


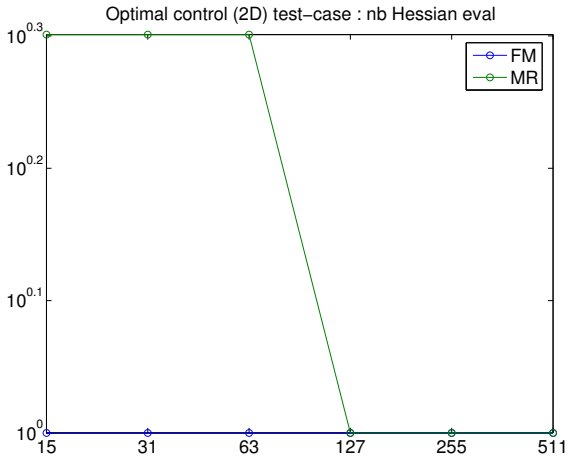


Performance results on Opt

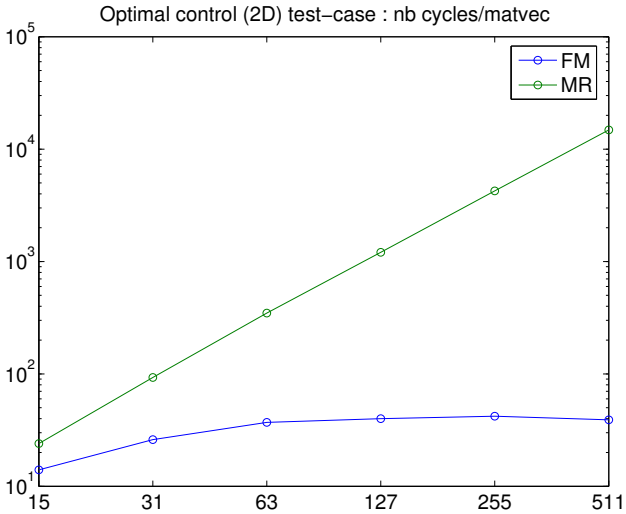


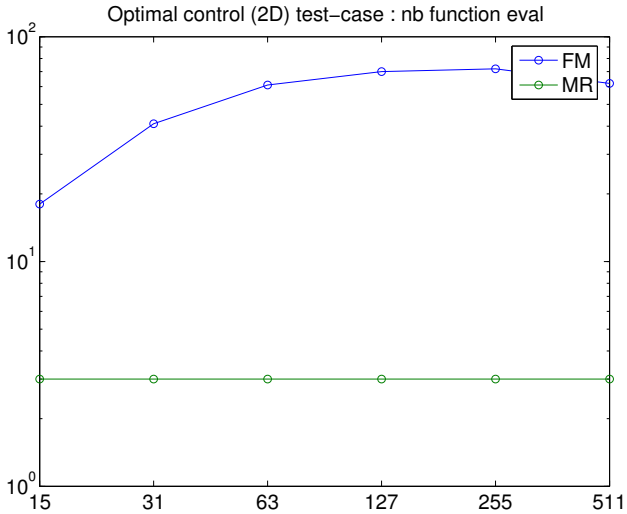






Operations counts for Opt (at the finest level)



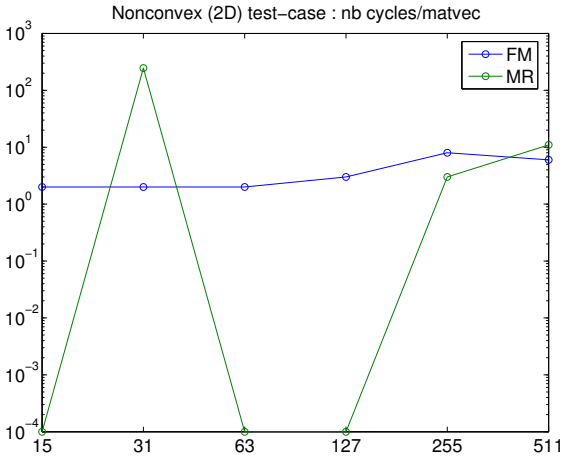


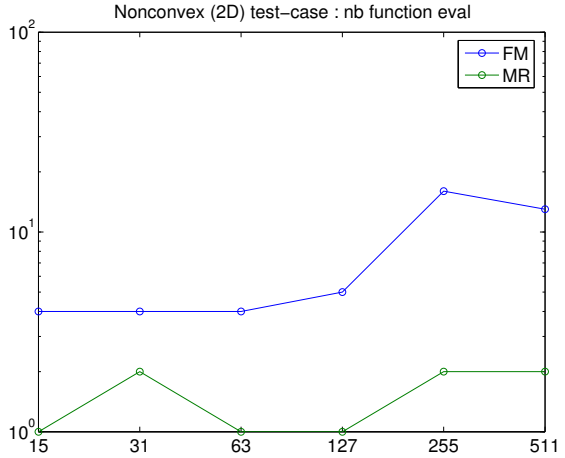
- One eval of $f = 14n_r$ flops
- One eval of $g = 56n_r$ flops
- One cycle/matvec = $10n_r$ flops

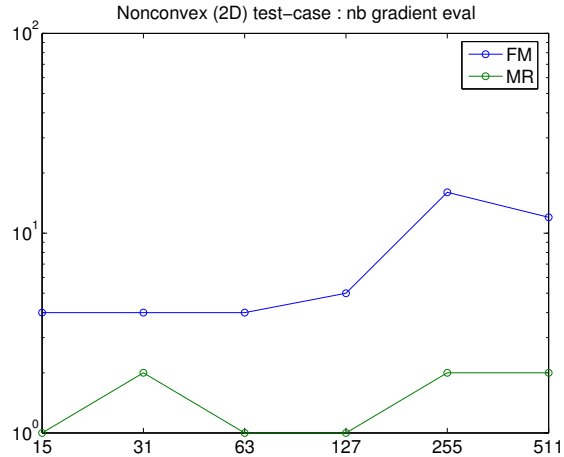
- **FM**: 4394 flops
- **MR**: 148470 flops

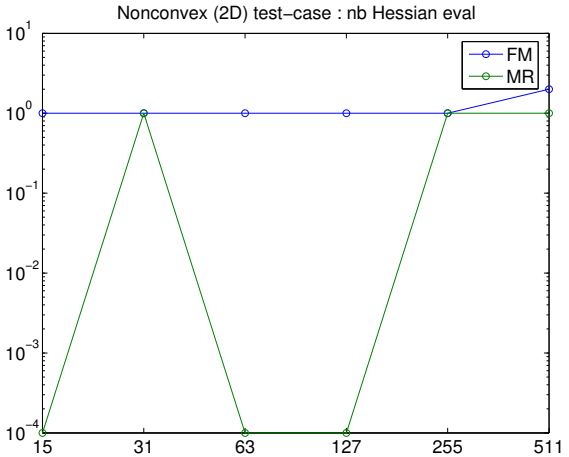
$\longrightarrow$ **MR** much more expensive than **FM** because the gain in the number of smoothing cycles is much superior to the loss in f and g evaluations

Performance results on NC









Comparison of algorithmic variants

$\longrightarrow$ 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 conclusions

$\longrightarrow$ 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

Perspectives

- More numerical experiments
- A (more natural) ℓ_∞ version
- Hessian approximation schemes
- Multigrid-type developments: algebraic multigrid, p-multigrids, ...
- Constrained problems: bounds, equalities, ...
- Combination with non-monotone techniques, filter methods, ...
- ...and much more !