

Multiresolution in image restoration

5- Multigrid and multilevel methods

Elisa Riccietti, Nelly Pustelnik

The concept

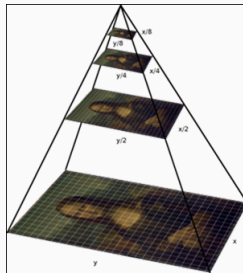
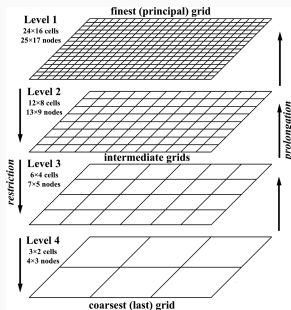
AIM? Given an optimization problem, reduce the computational cost of the solution process

HOW? Exploit the **structure** to perform dimensionality reduction

The concept

AIM? Given an optimization problem, reduce the computational cost of the solution process

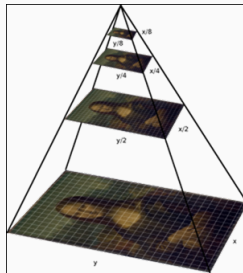
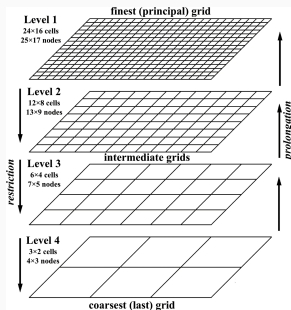
HOW? Exploit the **structure** to perform dimensionality reduction



The concept

AIM? Given an optimization problem, reduce the computational cost of the solution process

HOW? Exploit the **structure** to perform dimensionality reduction



Multilevel methods (ML)

Origins of multilevel methods

Multigrid methods

Multilevel optimization methods

Multilevel algorithms

The problem

Numerical solution of **partial differential equations** (PDEs)

Given $f : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$, $g : \partial\Omega \rightarrow \mathbb{R}$ and a differential operator D , find $u : \mathbb{R}^d \rightarrow \mathbb{R}$ that solves

$$D(u(x)) = f(x) \text{ in } \Omega$$

$$u(x) = g(x) \text{ in } \partial\Omega$$

Examples:

- Poisson's equation: $D(u(x)) = \Delta u(x) = \sum_{i=1}^d \frac{\partial^2 u(x)}{\partial x_i^2}$
- Nonlinear equation: $D(u(x)) = \sum_{i=1}^d a_i(x) \frac{\partial^2 u(x)}{\partial x_i^2} + \rho(x)$

Our model test problem

A simple 1D example: **1D Poisson's problem** with Dirichlet boundary conditions.

Given $f : \mathbb{R} \rightarrow \mathbb{R}$ and $g : \mathbb{R} \rightarrow \mathbb{R}$ find $u : \mathbb{R} \rightarrow \mathbb{R}$ such that

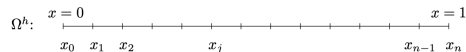
$$-u''(x) = f(x), \quad x \in (0, 1)$$

$$u(0) = g(0) = 0$$

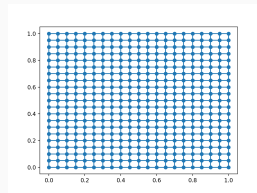
$$u(1) = g(1) = 0$$

The numerical solution of PDEs: discretization

- Build a grid on Ω . Example: equispaced points on each axis
- Discretize D on the chosen grid and obtain a discrete problem
- The size of the grids impacts the size of the discrete problem and the accuracy of the solution approximation



$$1D: h = \frac{1}{n}, x_j = jh, 0 \leq j \leq n$$



Example: 2D, $\Omega = [0, 1] \times [0, 1]$

Discretization of derivatives: the simple 1D case

Let $u : \mathbb{R} \rightarrow \mathbb{R}$

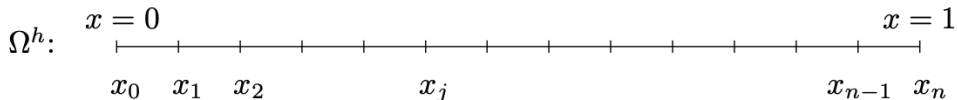
$$u'(x) = \lim_{h \rightarrow 0} \frac{u(x) - u(x+h)}{h}$$

Discretization: for fixed small h

$$u'(x) \sim \frac{u(x) - u(x+h)}{h} \text{ or } \frac{u(x-h) - u(x)}{h}$$

$$\begin{aligned} u''(x) = (u')'(x) &\sim \frac{u'(x) - u'(x+h)}{h} \sim \frac{\frac{u(x-h) - u(x)}{h} - \frac{u(x) - u(x+h)}{h}}{h} \\ &= \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} \end{aligned}$$

Discretization of the model test problem



$$1D: h = \frac{1}{n}, x_j = jh, 0 \leq j \leq n$$

- The boundary conditions: $u(x_0) = g(x_0) = 0, u(x_n) = g(x_n) = 0$
- On the interior points of the grid: $-u''(x_j) = f(x_j) \quad j = 1, \dots, n-1$
- Solution approximation: $(0, u_1, \dots, u_n, 0)$
- Discretization of $u''(x_j) \sim \frac{u(x_j-h) - 2u(x_j) + u(x_j+h)}{h^2} = \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2}$

Discretization of the model test problem

Matrix form:

$$\frac{1}{h^2} \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & & \ddots & & \\ & & & \ddots & & \\ & & & & -1 & 2 & -1 \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix} \begin{bmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_{n-1}) \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n-1}) \end{bmatrix}$$

If the PDE is linear, we obtain a **linear system**

$$Au = f, \quad A \in \mathbb{R}^{n+1 \times n+1}$$

Fixed point methods

$Au = f$ is solved using a fixed point method.

Fixed point scheme : $u^{[k+1]} = Ru^{[k]} + b$

The true solution : $u^* = Ru^* + b$

To define R , we decompose the matrix:

$$A = L + D + U$$

$$\underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}}_A = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 4 & 0 & 0 \\ 7 & 8 & 0 \end{bmatrix}}_L + \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 9 \end{bmatrix}}_D + \underbrace{\begin{bmatrix} 0 & 2 & 3 \\ 0 & 0 & 6 \\ 0 & 0 & 0 \end{bmatrix}}_U$$

Examples of fixed-point schemes

- Jacobi:

$$u^{[k+1]} = \underbrace{-D^{-1}(L+U)}_R u^{[k]} + \underbrace{D^{-1}f}_b$$

- Gauss-Seidel:

$$u^{[k+1]} = \underbrace{-(L+D)^{-1}U}_R u^{[k]} + \underbrace{(L+D)^{-1}f}_b$$

Fixed point: reduction of the error

Fixed point scheme : $u^{[k+1]} = Ru^{[k]} + b$

The true solution : $u^* = Ru^* + b$

The error : $e^{[k+1]} := u^{[k+1]} - u^* = Re^{[k]}$

After K iterations:

$$e^{[K]} := u^{[K]} - u^* = Re^{[K-1]} = R^2e^{[K-2]} = \dots = R^K e^{[0]}$$

Convergence:

$$\lim_{K \rightarrow \infty} \|e^{[K]}\| = 0 \iff \rho(R) < 1,$$

where $\rho(R) = \max\{|\lambda_1|, \dots, |\lambda_n|\}$ eigenvalues

The Jacobi scheme: study of the convergence

Let us consider or test problem:

$$A = \frac{1}{h^2} \begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & & \ddots & & & \\ & & & \ddots & & & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 & -1 \\ & & & & & & -1 & 2 \end{bmatrix}$$

We can see that

$$R := -D^{-1}(L + U) = I - \frac{1}{2}A \quad (1)$$

The Jacobi scheme: study of the convergence

Eigenvalues and eigenvectors

$$\lambda_\ell(A) = 4 \sin^2 \left(\frac{\ell\pi}{2n} \right), \quad 1 \leq \ell \leq n-1$$

$$(w_\ell)_j = \sin \left(\frac{\ell\pi j}{n} \right), \quad 1 \leq j \leq n-1$$

Thus from (1)

$$\lambda_\ell(R) = 1 - \frac{1}{2}\lambda_\ell(A) = 1 - 2 \sin^2 \left(\frac{\ell\pi}{2n} \right), \quad 1 \leq \ell \leq n-1$$

Eigenvectors are the same as those of A : $Rw_\ell = \lambda_\ell(R)w_\ell$

Notice that $|\lambda_\ell(R)| < 1$ for all k and thus the method converges, but the rate of convergence will depend on how small $|\lambda_\ell(R)|$ are

Reduction of the error

The eigenvectors form a basis.

Initial error:

$$e^{[0]} = \sum_{\ell=1}^{n-1} c_{\ell} w_{\ell}$$

After K iterations:

$$e^{[K]} := u^{[K]} - u^* = R^K e^{[0]} = \sum_{\ell=1}^{n-1} c_{\ell} R^K w_{\ell} = \sum_{\ell=1}^{n-1} c_{\ell} \lambda_{\ell}^M(R) w_{\ell}$$

- After K iterations, the ℓ th component of the initial error (modes) w_{ℓ} has been reduced by a factor of $\lambda_{\ell}^M(R) < 1$
- **Modes are not mixed:** the iteration can change the amplitude of a mode, but it cannot convert that mode into different modes

The Fourier modes

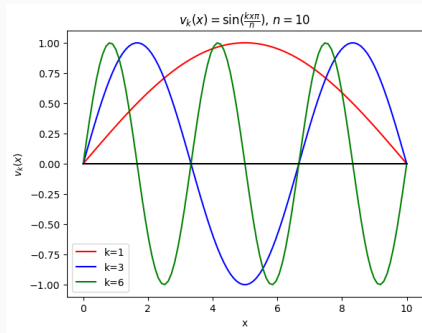
The vectors

$$(w_\ell)_j = \sin\left(\frac{k\pi j}{n}\right), \quad 1 \leq j \leq n-1$$

are special vectors called the **Fourier modes** and k is the frequency.

On a n -point grid:

- $1 \leq k < \frac{n}{2}$ low frequencies
- $\frac{n}{2} \leq k < n-1$ high frequencies



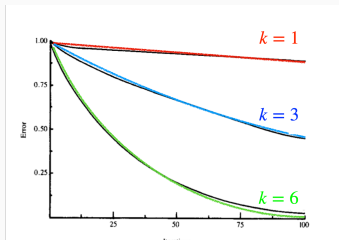
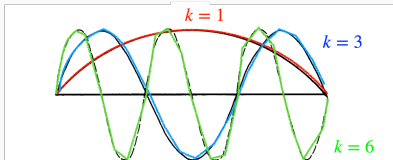
Limitation of iterative schemes: the smoothing property

$$e^{[K]} = u^{[K]} - u^* = \sum_{\ell=1}^{n-1} c_{\ell} \lambda_{\ell}^K(R) w_{\ell}$$

- $\lambda_1(R) \approx 1$
- $|\lambda_{\ell}(R)| < \frac{1}{\mu}$ for all $n/2 \leq \ell \leq n-1$
- μ is the smoothing factor (dumping coefficient of the oscillatory components)

Hard to reduce the **low frequency** components of the error

$$v_k(x) = \sin\left(\frac{xk\pi}{n}\right), 0 \leq x \leq n$$



How to make the methods efficient on all frequencies?

Frequency shift by coarsening!

- **Fine** grid Ω^h with n points: $1 \leq k \leq n-1$
- **Coarse** grid Ω^{2h} with $n/2$ points: $1 \leq k \leq n/2$

Grid points of Ω^{2h} are the even-numbered points of Ω^h .

Property: $(w_\ell^h)_{2j} = (w_\ell^{2h})_j$

Why does the coarse grid help?

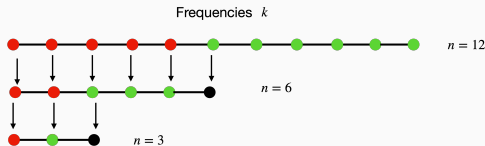
Assume that we have performed enough iterations on the fine grid to reduce all the high-frequency of the error. In this case

$$e^{[M]} \approx \sum_{\ell=1}^{n/2} c_{\ell} \lambda_{\ell}^M(R) w_{\ell}$$

and

$$\text{restriction}(e^{[M]}) \approx \sum_{\ell=1}^{n/2} c_{\ell} \lambda_{\ell}^M(R) (w_{\ell})_{2j}.$$

On the coarse grid, approximately half of the frequencies that were seen as low on the fine grid become high



Accelerate convergence by considering multiple grids

- The smoothing rate (the convergence factor for the oscillatory modes) for the standard relaxation schemes is small and independent of the grid spacing h .
- The smooth error modes, which remain after relaxation on one grid, appear more oscillatory on the coarser grids.
- Moving to successively coarser grids, all of the error components on the original fine grid eventually appear oscillatory and are reduced by relaxation.
- The overall convergence factor for a good multigrid scheme is small and independent of h .

Origins of multilevel methods

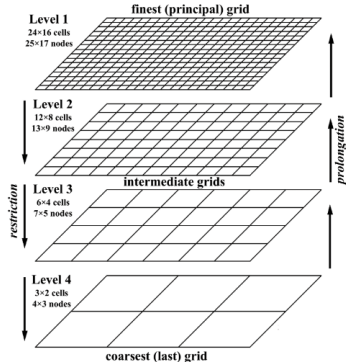
Multigrid methods

Multilevel optimization methods

Multilevel algorithms

Multigrid methods

Aim: accelerate fixed point methods: after removing all the oscillatory components, when relaxation begins to stall, signaling the predominance of smooth error modes, move to a coarser grid; there, the smooth error modes appear more oscillatory and relaxation will be more effective.



- Fine scales: eliminate **high frequency** components of the error
- Coarse scales: eliminate **low frequency** components of the error

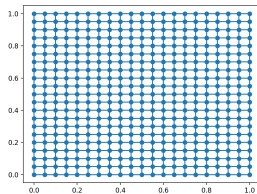
Ingredients of a two-level multigrid methods

Consider a PDE:

$$Au = f$$

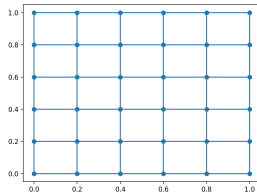
Consider two discretizations:

- Fine grid: $A^h u^h = f^h$
- Coarse grid: $A^{2h} u^{2h} = f^{2h}$



$I_h^{2h} \Downarrow$

$I_{2h}^h \Uparrow$



Transfer operators: how do we move between two grids ?

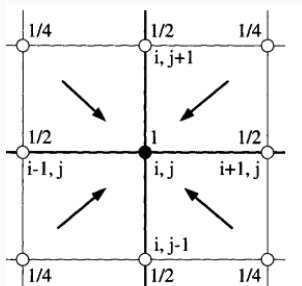
From fine to coarse: I_h^{2h} (restriction)

- Injection:

$$v_j^{2h} = v_{2j}^h$$

- Full weighting

$$v_j^{2h} = \frac{1}{4}(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h)$$

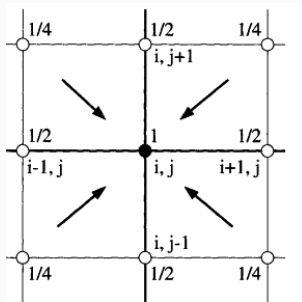


Transfer operators: how do we move between two grids ?

From coarse to fine: I_{2h}^h (prolongation)

- Interpolation

$$v_{2j}^h = v_j^{2h}$$
$$v_{2j+1}^h = \frac{1}{2}(v_j^{2h} + v_{j+1}^{2h})$$



How to define the coarse level operator A^{2h} ?

- Galerkin approximation:

$$A^{2h} = I_h^{2h} A^h I_{2h}^h$$

- Discretization of differential operator on the coarse grid

Two-level multigrid methods

Consider a PDE:

$$Au = f$$

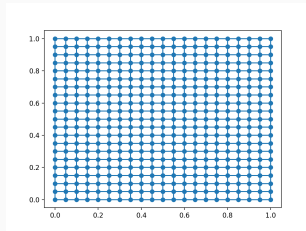
Consider two discretizations:

- Fine grid: $A^h u^h = f^h$
- Coarse grid: $A^{2h} u^{2h} = f^{2h}$

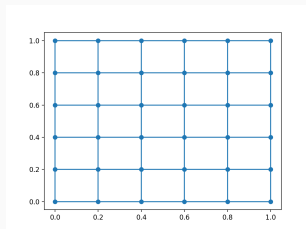
Idea: write the solution u as the sum of a fine and a coarse term:

$$u \sim \underbrace{v^h}_{\in \mathbb{R}^h} + P(\underbrace{e^{2h}}_{\in \mathbb{R}^{2h}}).$$

and update the two components in an **alternate** fashion.



$$I_h^{2h} \Downarrow \quad I_{2h}^h \Uparrow$$



Two-level multigrid methods

Update the two components in an **alternate** fashion:

$$u \sim v + e$$

- Fine level: get v^h by iterating on $A^h u = f^h$ (Jacobi or Gauss-Seidel)
-
-
-

Two-level multigrid methods

Update the two components in an **alternate** fashion:

$$u \sim v + e$$

$$r = f - Av$$

- Fine level: get v^h by iterating on $A^h u = f^h$
- Compute $r^h = f - Av^h$ and project $r^{2h} = I_h^{2h} r^h$
-
-

Two-level multigrid methods

Update the two components in an **alternate** fashion:

$$u \sim v + e$$

$Ae = r$ residual equation

- Fine level: get v^h by iterating on $A^h u = f^h$
- Compute $r^h = f - Av^h$ and project $r^{2h} = I_h^{2h} r^h$
- Coarse level: compute correction: $A^{2h} e^{2h} = r^{2h}$
-

Two-level multigrid methods

Update the two components in an **alternate** fashion:

$$u \sim v + e$$

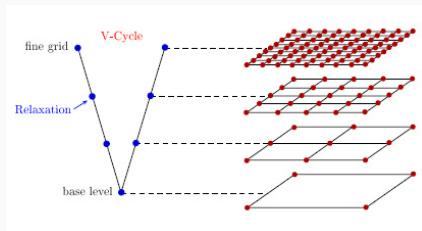
- Fine level: get v^h by iterating on $A^h u = f^h$
- Compute $r^h = f - A v^h$ and project $r^{2h} = I_h^{2h} r^h$
- Coarse level: compute correction: $A^{2h} e^{2h} = r^{2h}$
- Correct: $v^h \leftarrow v^h + I_{2h}^h(e^{2h})$

General multigrid methods - V-cycle

V-Cycle Scheme

$$\mathbf{v}^h \leftarrow V^h(\mathbf{v}^h, \mathbf{f}^h)$$

- Relax on $A^h \mathbf{u}^h = \mathbf{f}^h$ ν_1 times with initial guess $\mathbf{v}^h$.
- Compute $\mathbf{f}^{2h} = I_h^{2h} \mathbf{r}^h$.
 - Relax on $A^{2h} \mathbf{u}^{2h} = \mathbf{f}^{2h}$ ν_1 times with initial guess $\mathbf{v}^{2h} = \mathbf{0}$.
 - Compute $\mathbf{f}^{4h} = I_{2h}^{4h} \mathbf{r}^{2h}$.
 - Relax on $A^{4h} \mathbf{u}^{4h} = \mathbf{f}^{4h}$ ν_1 times with initial guess $\mathbf{v}^{4h} = \mathbf{0}$.
 - Compute $\mathbf{f}^{8h} = I_{4h}^{8h} \mathbf{r}^{4h}$.
 -
 -
 -
 - Solve $A^{Lh} \mathbf{u}^{Lh} = \mathbf{f}^{Lh}$.
 -
 -
 -
 - Correct $\mathbf{v}^{4h} \leftarrow \mathbf{v}^{4h} + I_{8h}^{4h} \mathbf{v}^{8h}$.
 - Relax on $A^{4h} \mathbf{u}^{4h} = \mathbf{f}^{4h}$ ν_2 times with initial guess $\mathbf{v}^{4h}$.
 - Correct $\mathbf{v}^{2h} \leftarrow \mathbf{v}^{2h} + I_{4h}^{2h} \mathbf{v}^{4h}$.
 - Relax on $A^{2h} \mathbf{u}^{2h} = \mathbf{f}^{2h}$ ν_2 times with initial guess $\mathbf{v}^{2h}$.
 - Correct $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h \mathbf{v}^{2h}$.
 - Relax on $A^h \mathbf{u}^h = \mathbf{f}^h$ ν_2 times with initial guess $\mathbf{v}^h$.



V-Cycle Scheme (Recursive Definition)

$$\mathbf{v}^h \leftarrow V^h(\mathbf{v}^h, \mathbf{f}^h).$$

1. Relax ν_1 times on $A^h \mathbf{u}^h = \mathbf{f}^h$ with a given initial guess $\mathbf{v}^h$.

2. If $\Omega^h =$ coarsest grid, then go to step 4.

Else

$$\mathbf{f}^{2h} \leftarrow I_h^{2h}(\mathbf{f}^h - A^h \mathbf{v}^h),$$

$$\mathbf{v}^{2h} \leftarrow \mathbf{0},$$

$$\mathbf{v}^{2h} \leftarrow V^{2h}(\mathbf{v}^{2h}, \mathbf{f}^{2h}).$$

3. Correct $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h \mathbf{v}^{2h}$.

4. Relax ν_2 times on $A^h \mathbf{u}^h = \mathbf{f}^h$ with initial guess $\mathbf{v}^h$.

General multigrid methods - other cycles

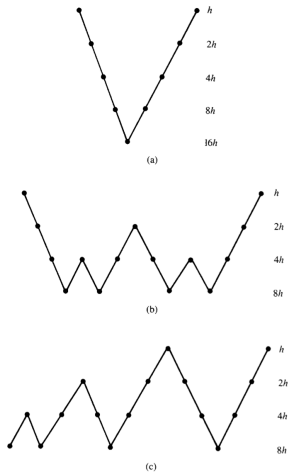


Figure 3.6: Schedule of grids for (a) V-cycle, (b) W-cycle, and (c) FMG scheme, all on four levels.

Full Multigrid V-Cycle (Recursive Form)

$$\mathbf{v}^h \leftarrow FMG^h(\mathbf{f}^h).$$

1. If $\Omega^h =$ coarsest grid, set $\mathbf{v}^h \leftarrow \mathbf{0}$ and go to step 3.

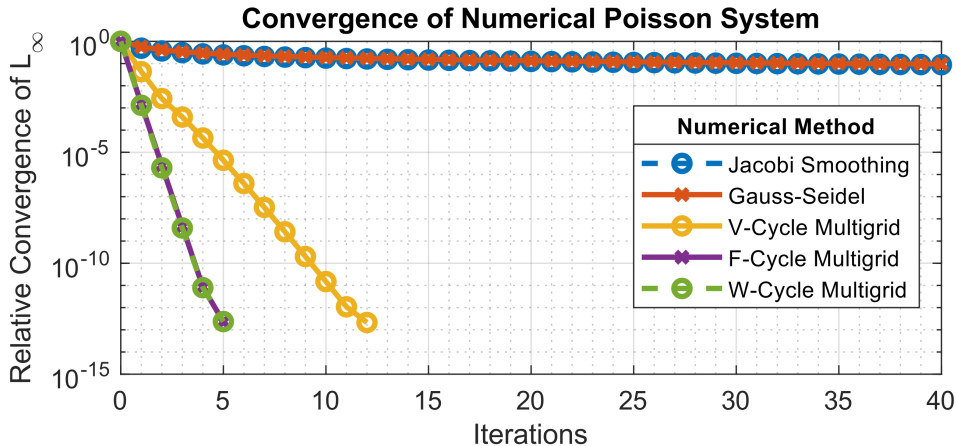
Else

$$\mathbf{f}^{2h} \leftarrow I_h^{2h}(\mathbf{f}^h),$$

$$\mathbf{v}^{2h} \leftarrow FMG^{2h}(\mathbf{f}^{2h}).$$

2. Correct $\mathbf{v}^h \leftarrow I_{2h}^h \mathbf{v}^{2h}$.
3. $\mathbf{v}^h \leftarrow V^h(\mathbf{v}^h, \mathbf{f}^h)$ ν_0 times.

Numerical example

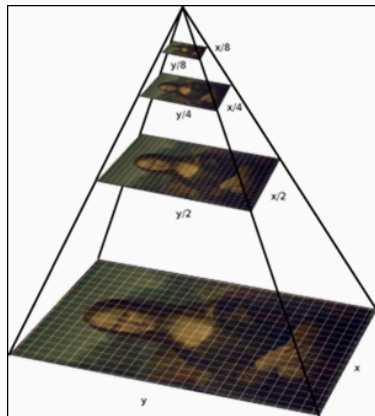


From linear to nonlinear: multilevel methods

Multigrid methods work for linear systems: $Ax = b$

An extension exists for nonlinear equations: $\mathcal{A}(x) = b$ FAS (Full Approximation Scheme)

But what happens if we want to tackle a more general **optimization problem** with a hierarchical structure ?



Origins of multilevel methods

Multigrid methods

Multilevel optimization methods

Multilevel algorithms

Classical iterative optimization methods

Large-scale **nonlinear unconstrained optimization problems**:

$$\min_x f(x)$$

- Build a **model** ($B^{[k]} \approx H_f(x_k$ or $B^{[k]} = 0$):

$$f(x^{[k]} + s) \simeq T(x^{[k]}, s) = f(x_k + s^T \nabla f(x_k + \frac{1}{2} s^T B^{[k]} s$$

Classical iterative optimization methods

Large-scale **nonlinear unconstrained optimization problems**:

$$\min_x f(x)$$

- Build a **model** ($B^{[k]} \approx H_f(x_k$ or $B^{[k]} = 0)$):

$$f(x^{[k]} + s) \simeq T(x^{[k]}, s) = f(x_k + s^T \nabla f(x_k + \frac{1}{2} s^T B^{[k]} s$$

- Compute a **step** $s^{[k]}$ ($r(\lambda_\ell)$ regularization term):

$$\min_s m(x^{[k]}, s) = T(x^{[k]}, s) + r(\lambda_\ell), \quad \lambda_\ell > 0$$

Classical iterative optimization methods

Large-scale **nonlinear unconstrained optimization problems**:

$$\min_x f(x)$$

- Build a **model** ($B^{[k]} \approx H_f(x_k$ or $B^{[k]} = 0)$):

$$f(x^{[k]} + s) \simeq T(x^{[k]}, s) = f(x_k + s^T \nabla f(x_k + \frac{1}{2} s^T B^{[k]} s$$

- Compute a **step** $s^{[k]}$ ($r(\lambda_\ell)$ regularization term):

$$\min_s m(x^{[k]}, s) = T(x^{[k]}, s) + r(\lambda_\ell), \quad \lambda_\ell > 0$$

- Update the iterate $x^{[k+1]} = x^{[k]} + s^{[k]}$

Classical examples

- Gradient method:

$$m(x^{[k]}, s) = f(x_k + s^T \nabla f(x_k + \frac{\lambda_\ell}{2} \|s\|^2)$$

- Adaptive second order method:

$$m(x^{[k]}, s) = f(x_k + s^T \nabla f(x_k + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{\lambda_\ell}{2} \|s\|^2)$$

- Adaptive Cubic Regularization (ARC):

$$m(x^{[k]}, s) = f(x_k + s^T \nabla f(x_k + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{\lambda_\ell}{3} \|s\|^3)$$

 *Cubic regularization of Newton method and its global performance*, Y. Nesterov and B. Polyak, 2006

Bottleneck: Subproblem solution

Solving

$$\min_s T_k(x^{[k]}, s) + r(\lambda_\ell)$$

represents greatest cost per iteration, which depends on the size of the problem.

Possible solution: multilevel methods

 **MG/OPT:** *A multigrid approach to discretized optimization problems*, Nash, 2000¹

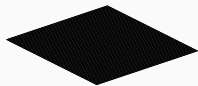
 **RMTR:** *Recursive trust-region methods*, S. Gratton, A. Sartenaer and Ph. L. Toint, 2008²

¹<https://optimization-online.org/wp-content/uploads/2012/04/3447.pdf>

²https://www.cerfacs.fr/algor/reports/2007/TR_PA_07_42.pdf

- Exploit structure of the problem to build a hierarchical representation of the problem
- Exploit lower levels to compute a **cheap** (but useful) step
- Simplify the landscape in a nonconvex setting

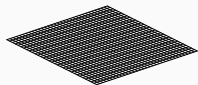
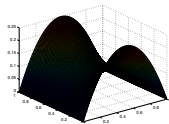
Hierarchy of problems ($n_r > n_{r-1} > \dots n_0$)



Restriction $\downarrow R_r$

Finest Level $f_r : \mathbb{R}^{n_r} \rightarrow \mathbb{R}$

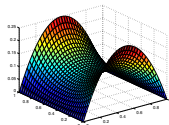
$P_r \uparrow$ Prolongation



Restriction $\downarrow R_{r-1}$

Fine Level $f_{r-1} : \mathbb{R}^{n_{r-1}} \rightarrow \mathbb{R}$

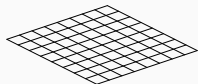
$P_{r-1} \uparrow$ Prolongation



$\vdots$

Restriction $\downarrow R_2$

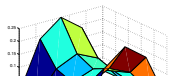
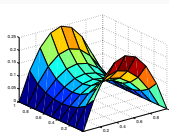
$P_2 \uparrow$ Prolongation



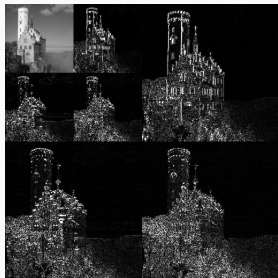
Restriction $\downarrow R_1$

Coarse Level $f_2 : \mathbb{R}^{n_2} \rightarrow \mathbb{R}$

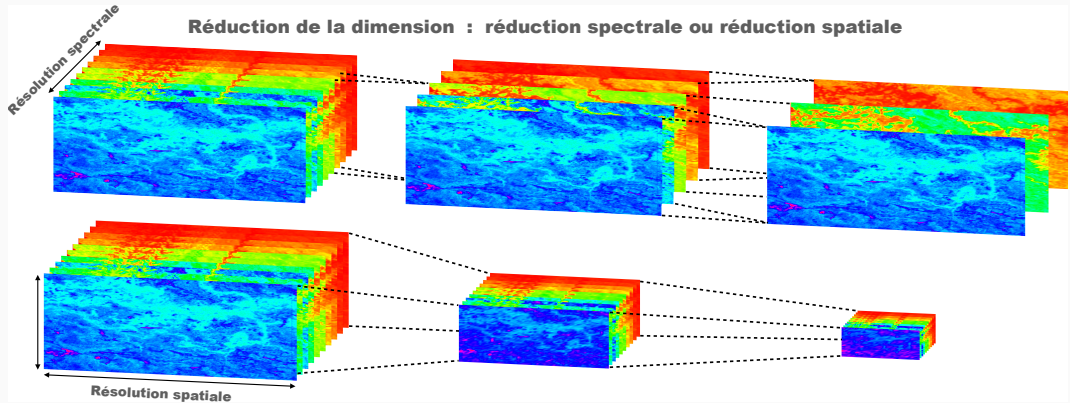
$P_1 \uparrow$ Prolongation



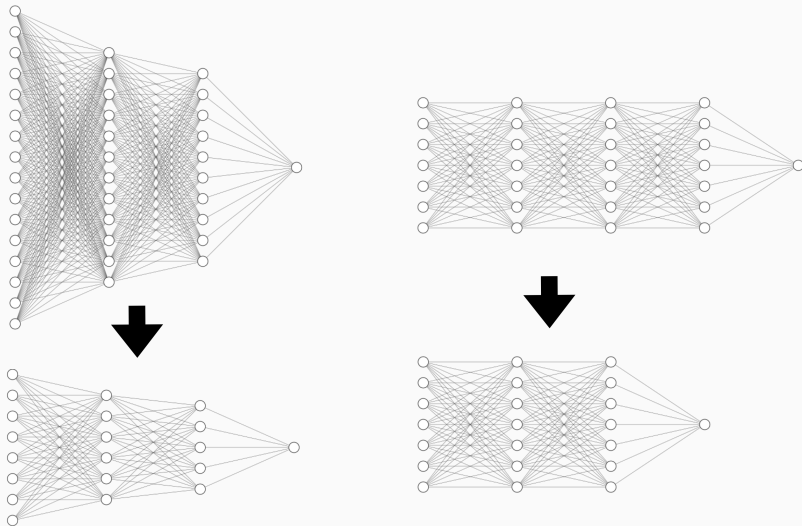
Example 1: image restoration



Example 2: hyperspectral imaging

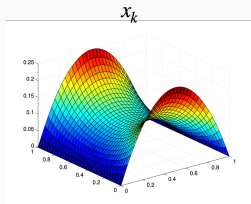


Example 3: neural networks

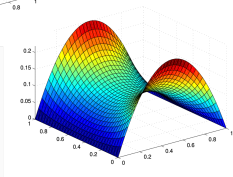
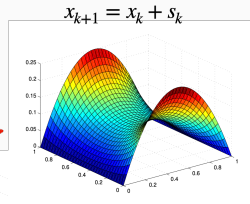


Classical one-level method

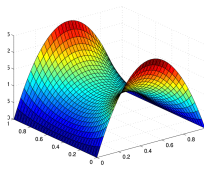
$$\min_x f(x)$$



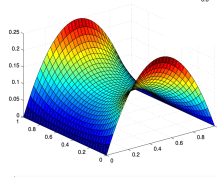
$$\min_s T_k(s) + r_k(s)$$



$$s_{k,0}$$



$$s_{k,1} \dots$$



$$s_{k,m}$$

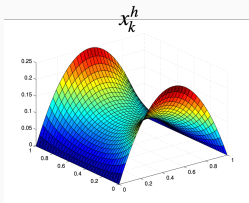
Multilevel strategy: step computation

Two choices:

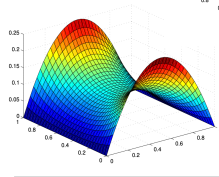
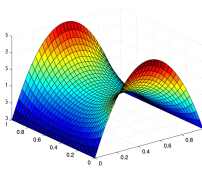
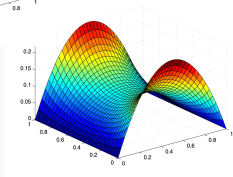
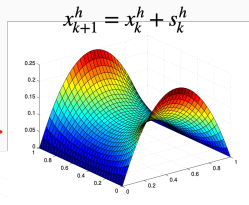
1. Classical fine step
2. Coarse step

Fine step

$$\min_{x^h} f^h(x^h)$$



$$\min_{s^h} T_k^h(s^h) + r_k^h(s^h)$$



$$s_{k,0}^h$$

$$s_{k,1}^h \dots$$

$$s_{k,m}^h$$

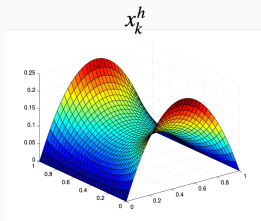
Multilevel strategy: step computation

Two choices:

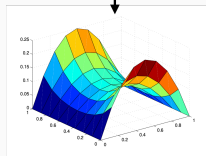
1. Classical fine step
2. Coarse step

Coarse step ($H = 2h$ in the previous notation)

$$\min_{x^h} f^h(x^h) \sim \min_{x^H} f^H(x^H)$$

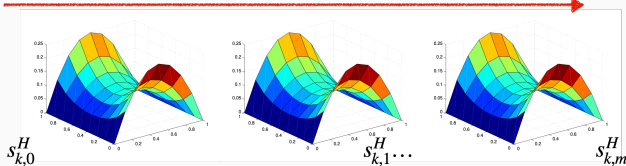


$$I_h^H$$

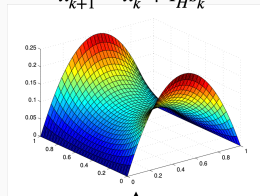


$$x_k^H$$

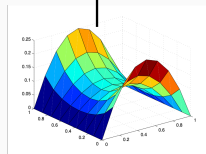
$$\min_{s^H} \mu_k^H(s^H)$$



$$x_{k+1}^h = x_k^h + I_h^H s_k^H$$

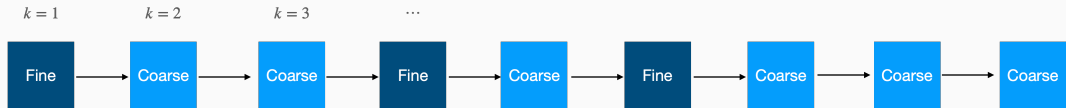


$$I_h^H$$



$$s_k^H = s_{k,m}^H$$

A possible iterative scheme



Effect of coarse projection in a nonlinear setting

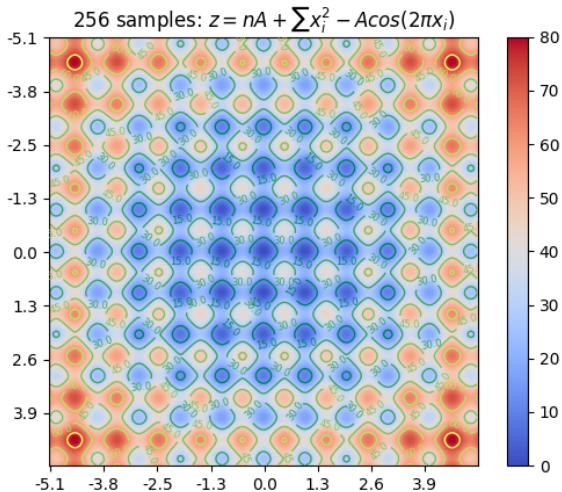
1. Cost reduction: usually a **factor 4** from one level to another in dimension 2

- Level h : $f^h(u), u \in \mathbb{R}^{n^2}$
- Level H : $f^H(u), u \in \mathbb{R}^{n^2/4}$

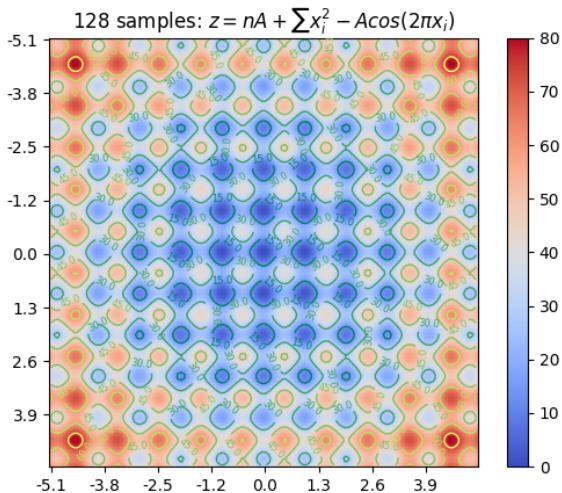
→ for the same cost, can do at coarse level 4 times the number of iterations at fine level

2. Smoothing of the landscape: escape **local minima** → better solution

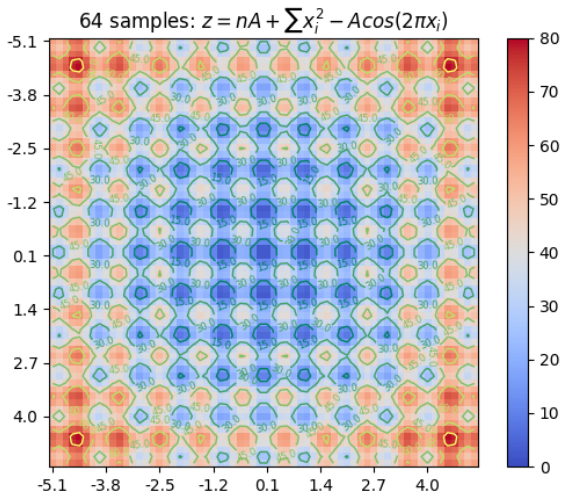
Smoothing of the landscape



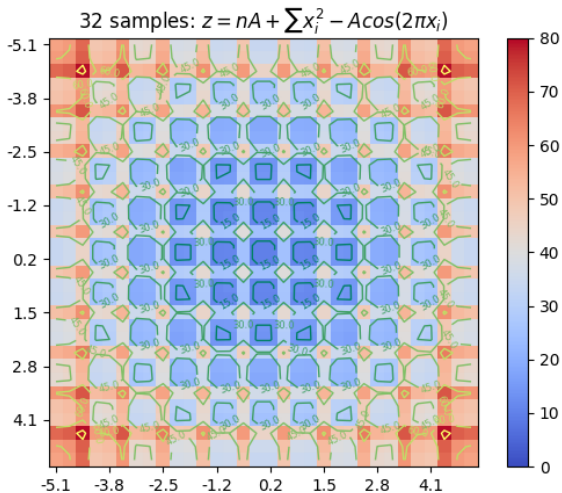
Smoothing of the landscape



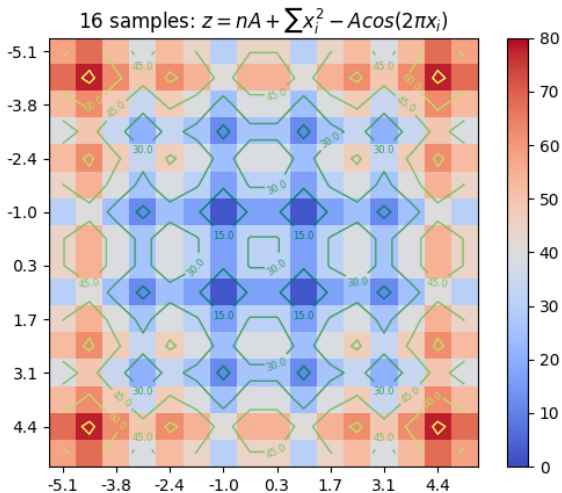
Smoothing of the landscape



Smoothing of the landscape



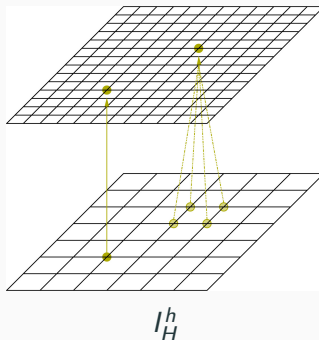
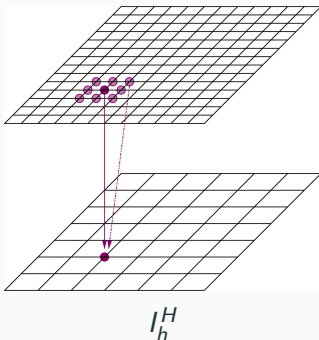
Smoothing of the landscape



What do we need to use such a method ?

Transfer operators

- $I_h^H \in \mathbb{R}^{N_H \times N_h}$: from fine to coarse ($N_H < N_h$).
- $I_H^h \in \mathbb{R}^{N_h \times N_H}$: from coarse to fine.
- Relation between the operators : $I_H^h = \alpha(I_h^H)^T$, $\alpha > 0$.



What do we need to use such a method ?

Lower level model

When to use the lower level model?

Let $x_{k,0}^H = I_h^H x_k^h$

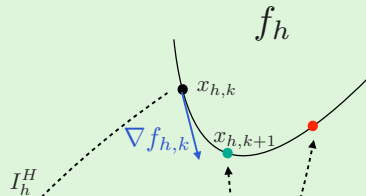
- Choose lower level model μ^H if
 - if $\|\nabla \mu_k^H(x_{k,0}^H)\| \geq \kappa \|\nabla f^h(x_k^h)\|$, $\kappa > 0$
 - if $\|\nabla \mu_k^H(x_{k,0}^H)\| > \epsilon^H$
- Minimize regularized Taylor model otherwise.

How to define the lower level model?

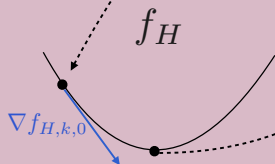
Modify f^H to ensure coherence among levels

First order coherence

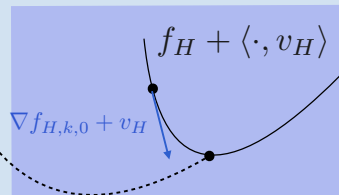
Niveau fin h



Niveau grossier H



I_H^h I_H^h



Coherence between levels: first-order coherence

Let $x_{k,0}^H = I_h^H x_k^h$. Model with first order correction:

$$\begin{aligned}\mu_k^H(s^H) &= f^H(x_{k,0}^H + s^H) + (v^H)^T s^H \\ v^H &= I_h^H \nabla f^h(x_k^h) - \nabla f^H(x_{k,0}^H)\end{aligned}$$

This ensures that

$$\nabla \mu_k^H(0^H) = I_h^H \nabla f^h(x_k^h)$$

- first-order behaviours of f^h and μ_k^H are coherent around x_k^h .
- If s^H descent direction for μ_k^H and $s^h = I_H^h s^H$, then s^h is a descent direction for f^h

$$\begin{aligned}0 > \nabla \mu_k^H(0^H)^T s^H &= (I_h^H \nabla f^h(x_k^h))^T s^H = (\nabla f^h(x_k^h))^T (I_h^H)^T s^H \\ &= \alpha (\nabla f^h(x_k^h))^T I_H^h s^H = \alpha (\nabla f^h(x_k^h))^T s^h\end{aligned}$$

Coherence between levels: second order

Let $x_{0,k}^H = I_h^H x_k^h$. We define $\mu_{2,k}^H$ as

$$\begin{aligned}\mu_{2,k}^H(x_{0,k}^H, s^H) &= f^H(x_{0,k}^H + s^H) \\ &\quad + (I_h^H \nabla f^h(x_k^h) - \nabla f^H(x_{k,0}^H))^T s^H \\ &\quad + \frac{1}{2} (s^H)^T ((I_h^H)^T \nabla^2 f^h(x_k^h) I_h^H - \nabla^2 f^H(x_{k,0}^H)) s^H\end{aligned}$$

Parameters of the coarse iterations

- Minimizer at coarse level: any minimization scheme, can be different from fine level one, can use second order
- How many iterations m ?
 - m large enough to compensate projection cost
 - m not too large to keep first order coherence

Origins of multilevel methods

Multigrid methods

Multilevel optimization methods

Multilevel algorithms

These key ingredients can be used in different frameworks to ensure that the recursive step actually decreases the function

- Nash MG/OPT: linesearch ["Convergence and Descent Properties for a Class of Multilevel Optimization Algorithms", S. G. Nash, 2010]
- RMTR: trust-region [Recursive trust-region methods for multiscale nonlinear optimization, Gratton, Sartenaer, Toint, 2008]
- MARC: adaptive regularization

Nash MG/OPT algorithm: k -th iteration

Given x_k^h

1. **Pre-smoothing:** k_1 iterations of a convergent algorithm to $\min_{x^h} f^h(x^h)$ to obtain $\bar{x}^h$
2. **Recursion:**
 - 2.1 Define $\bar{x}^h = I_h^H \bar{x}^h$
 - 2.2 Define $v^H = \nabla f^H(\bar{x}^H) - I_h^H \nabla f^h(\bar{x}^h)$
 - 2.3 Minimize $\mu^H(x^H) = f^H(\bar{x}^H) - (v^H)^T x^H$ (possibly with MG/OPT) obtaining $(x^H)^+$
 - 2.4 Define $e^H = (x^H)^+ - \bar{x}^H$ and $e_h = I_h^H e^H$
 - 2.5 Use a line search to determine $(x^h)^+ = \bar{x}^h + \gamma e_h$ such that $f((x^h)^+) < f(\bar{x}^h)$
3. **Post-smoothing:** k_2 iterations of a convergent algorithm to $\min_{x^h} f^h(x^h)$ to obtain x_{k+1}^h starting from $\bar{x}^h$

Convergence theory: assumptions

$$\text{A1 } f^h \in \mathcal{C}^{1,1}$$

$$\text{A2 } S_h := \{x^h : f^h(x^h) \leq f^h(x_0^h)\} \text{ is compact}$$

$$\text{A3 } (I_H^h)^T = \alpha I_h^H, \alpha > 0$$

Convergence theory: global convergence

Theorem

Let A1 and A2 be satisfied, then

$$\lim_{k \rightarrow \infty} \|\nabla f^h(x_k^h)\| = 0.$$

- Same convergence proof as for classical one-level method
- Does not depend on μ^H ! At worst, recursive step keeps the value of f^h the same
- Does not say anything on the practical behaviour of the method

Convergence theory: global convergence

Theorem

Let A1, A2, A3 be satisfied, then e_h is a descent direction at $\bar{x}_h$ (i.e., $\nabla f^h(\bar{x}_h)^T e_h < 0$) if

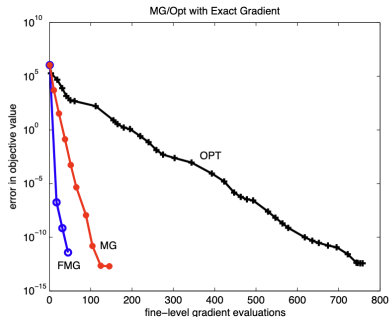
- a) $\|\nabla \mu^H((x^H)^+)\|$ is small enough
- b) $(e^H)^T H_{f^H}(\bar{x}^H + \eta e^H) e^H > 0$ for $0 \leq \eta \leq 1$.

Condition b) is hard to obtain or verify in practice.

Numerical results for 5 levels MG/OPT with line-search

Problem:

$$\min_a F(a) = \frac{1}{2} \int_0^\pi \left(\frac{\partial u}{\partial x_2}(x_1, 0) - \phi(x_1) \right)^2 dx_1,$$



RMTR: k -th iteration

RMTR(ℓ, f_ℓ) $\ell = 0$ (coarse), $\dots, L$ (fine)

Given x_k^ℓ

1. If $\ell = 0$ go to step 3. Otherwise check descent condition. If it holds decide to go to step 2 or step 3. Else go to step 2.
2. **Recursive coarse step:** call RMTR($\ell - 1, \mu^{\ell-1}$) and obtain $x^{\ell-1}$
 - 2.1 Define $s_k^\ell = I_{\ell-1}^\ell(x^{\ell-1} - I_\ell^{\ell-1}x_k^\ell)$
 - 2.2 Define $\delta = \mu^{\ell-1}(I_\ell^{\ell-1}x_k^\ell) - \mu^{\ell-1}(x^{\ell-1})$
3. **Fine step:** $\min_{\|s^\ell\| \leq \Delta} T_2^\ell(s^\ell, x_k^\ell)$ obtaining s_k^ℓ
 - 3.1 Define $\delta = T_2^\ell(x_k^\ell) - T_2^\ell(x_k^\ell + s_k^\ell)$
4. Compute $\rho_k = \frac{f^\ell(x_k^\ell) - f^\ell(x_k^\ell + s_k^\ell)}{\delta}$
5. Update Δ and decide whether to accept the step or not.

Global convergence of RMTR

Theorem

Let $f \in \mathcal{C}^2$, $f(x) \geq f_{low}$ for all $x \in \mathbb{R}^N$. Let, for all $\ell = 0, \dots, L$, $\|H_{f_\ell}(x)\| \leq \kappa_H$ for $\kappa_H > 1$ and for all $x \in \mathbb{R}^N$ and $I_\ell^{\ell-1} = c(I_{\ell-1}^\ell)^T$. Then

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$$

and the number of iterations K necessary to drive $\|\nabla f(x_k)\| < \epsilon$ is

$$K \approx 1/\epsilon^2$$

Very complicated convergence proofs, cf. [Recursive trust-region methods for multiscale nonlinear optimization, Gratton, Sartenaer, Toint, 2008]

Numerical results for 5 levels RMTR

Performance profile³ on 17 infinite-dimensional problems involving differential operators

- AF=one-level Newton
- MR=from down to up
- MF=RMTR from up to down
- FM= RMTR from down to up

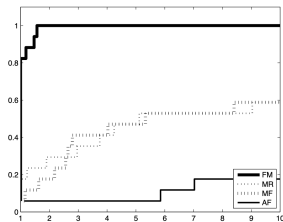


Figure 4.7: Performance profile for CPU time with variants AF, MF, MR and FM (17 test problems).



Recursive trust-region methods, S. Gratton, A. Sartenaer and Ph. L. Toint, 2008



Benchmarking Optimization Software with Performance Profiles, E. D. Dolan, J. J. Moré, 2002

³Performance profile $p_A(\tau)$ of an algorithm A at point τ shows the fraction of the test set for which the algorithm is able to solve within a factor of τ of the best algorithm for the given measure

MARC: k -th iteration

MARC(ℓ, f_ℓ) $\ell = 0$ (coarse), $\dots, L$ (fine)

Given $x_\ell^{[k]}$

1. If $\ell = 0$ go to step 3. Otherwise check descent condition. If it holds decide to go to step 2 or step 3. Else go to step 2.
2. **Recursive coarse step:** call MARC($\ell - 1, \mu^{\ell-1}$) and obtain $x^{\ell-1}$
 - 2.1 Define $d_\ell^{[k]} = I_{\ell-1}^\ell(x^{\ell-1} - I_\ell^{\ell-1}x_k^\ell)$
 - 2.2 Define $\delta = \mu^{\ell-1}(I_\ell^{\ell-1}x_k^\ell) - \mu^{\ell-1}(x^{\ell-1})$
3. **Fine step:** $\min_d T_{2,\ell}(d, x_k^\ell) + \lambda_k \|d_\ell^{[k]}\|^3$ obtaining $d_\ell^{[k]}$
 - 3.1 Define $\delta = T_{2,\ell}(x_k^\ell) - T_{2,\ell}(x_k^\ell + s_\ell^k)$
4. Compute $\rho_k = \frac{f_\ell(x_k^\ell) - f_\ell(x_k^\ell + s_\ell^k)}{\delta}$
5. Update λ_k and decide whether to accept the step or not.

Numerical results for 4 levels ARC

$$\begin{cases} -\Delta u(z) + e^{u(z)} = g(z) & \text{in } \Omega \subset \mathbb{R}^2, \\ u(z) = 0 & \text{on } \partial\Omega, \end{cases}$$

The following nonlinear minimization problem is then solved:

$$\min_{u \in \mathbb{R}^n} \frac{1}{2} u^T A u + \|e^{u/2}\|^2 - g^T u,$$

whose stationary points are solutions of the nonlinear system $Au + e^u = g$.

		$n = 1024$		$n = 4096$	
		ARC	MARC 4	ARC	MARC 4
$\bar{u}_1$	it_T/it_f	11/11	7/2	23/23	15/4
	save		2.2		4.1
$\bar{u}_2$	it_T/it_f	27/27	13/4	56/56	22/6
	save		3.9		6.1

Extension to q -order models

 *On high-order multilevel optimization strategies*, H. Calandra, S. Gratton, E. R., X. Vasseur, 2020

We define

$$\mu_{q,k}^H(s^H) = f^H(x_{0,k}^H + s^H) + \text{corr}$$
$$\text{corr} = \sum_{i=1}^q \frac{1}{i!} [\mathcal{R}(\nabla^i f^h(x_k) - \nabla^i f^H(x_k^H))] \underbrace{(s^H, \dots, s^H)}_{i \text{ times}},$$

Example

For $q = 2$

$$\begin{aligned} \text{corr} = & (l_h^H \nabla f^h(x_k^h) - \nabla f^H(x_k^H))^T x_k^h \\ & + \frac{1}{2} (x_k^h)^T (l_h^H \nabla^2 f^h(x_k^h) l_h^H - \nabla^2 f^H(x_k^H)) x_k^h \end{aligned}$$

Theoretical results: Assumptions

Assumption 1

Let us assume that for all levels the q -th derivative tensors of f^h are **Lipschitz continuous**.

Assumption 2

There exist strictly positive scalars $\kappa_{EB}, \rho > 0$ such that

$$\text{dist}(x, \mathcal{X}) \leq \kappa_{EB} \|\nabla_x f(x)\|, \quad \forall x \in \mathcal{N}(\mathcal{X}, \rho),$$

where $\mathcal{X}$ is the set of second-order critical points of f , $\text{dist}(x, \mathcal{X})$ denotes the distance of x to $\mathcal{X}$ and $\mathcal{N}(\mathcal{X}, \rho) = \{x \mid \text{dist}(x, \mathcal{X}) \leq \rho\}$.

 *On the Quadratic Convergence of the Cubic Regularization Method under a Local Error Bound Condition*, Yue, M.C. and Zhou, Z. and So, A.M.C., 2018:
generalized to higher-order methods

Theoretical results: 1) global convergence

Theorem

Let Assumption 1 hold. Then, the sequence of iterates generated by the algorithm converges globally to a first-order stationary point.

 E. G. Birgin, J. L. Gardenghi, J. M. Martinez, S. A. Santos and Ph. L. Toint,
2017: generalized to multilevel framework

Theoretical results: 2) complexity

Theorem

Let Assumption 1 hold. Let f_{low} be a lower bound on f . Then, the method requires at most

$$O\left(\epsilon^{-\frac{q+1}{q}}\right)$$

iterations to achieve an iterate $x^{[k]}$ such that $\|\nabla f(x_k)\| \leq \epsilon$

 E. G. Birgin, J. L. Gardenghi, J. M. Martinez, S. A. Santos and Ph. L. Toint,
2017: $k = O\left(\epsilon^{-\frac{q+1}{q}}\right)$ Complexity of standard method is maintained

Theoretical result: 3) local convergence

Theorem

Let Assumptions 1 and 2 hold. Assume that $\mathcal{L}(f(x_k))$ is bounded for some $k \geq 0$ and that it exists an accumulation point x^* such that $x^* \in \mathcal{X}$. Then, the whole sequence $\{x^{[k]}\}$ converges to x^* and it exist strictly positive constants $c \in \mathbb{R}$ and $\bar{k} \in \mathbb{N}$ such that:

$$\frac{\|x_{k+1} - x^*\|}{\|x^{[k]} - x^*\|^q} \leq c, \quad \forall k \geq \bar{k}.$$

Convergence proofs for AR1 - Global convergence

- On iterations where the fine model is used, we recover the results seen for the 1-level theory, cf. cours "Globalisation strategies"
- We now examine what happens on coarse iterations

We will need:

1. Upper bound for the regularization parameter λ_k
2. Link between norm on gradient and norm of step

Proof in note_multilevel.pdf

Write a multilevel cycle (1 fine step + 1 coarse step with 3 subiterations + 1 fine step)
for

$$f(x) = \frac{1}{2} \|Ax - b\|^2$$