

# Multilevel optimization using trust-region and line search approaches

S. Gratton<sup>1</sup>   M. Mouffe<sup>1</sup>   [Ph. Toint](#)<sup>2</sup>   D. Tomanos<sup>2</sup>  
M. Weber-Mendonça<sup>2</sup>

<sup>1</sup>CERFACS and CNES, Toulouse, France

<sup>2</sup>Department of Mathematics, University of Namur, Belgium  
( [philippe.toint@fundp.ac.be](mailto:philippe.toint@fundp.ac.be) )

April 2008

# Multilevel optimization using trust-region and lineasearch approaches

S. Gratton<sup>1</sup>   M. Mouffe<sup>1</sup>   [Ph. Toint](#)<sup>2</sup>   D. Tomanos<sup>2</sup>  
M. Weber-Mendonça<sup>2</sup>

<sup>1</sup>CERFACS and CNES, Toulouse, France

<sup>2</sup>Department of Mathematics, University of Namur, Belgium  
( [philippe.toint@fundp.ac.be](mailto:philippe.toint@fundp.ac.be) )

April 2008

- 1 Introduction
- 2 Recursive trust-region methods
- 3 Multigrid limited memory BFGS

# Outline

- 1 Introduction
- 2 Recursive trust-region methods
- 3 Multigrid limited memory BFGS

# Motivation

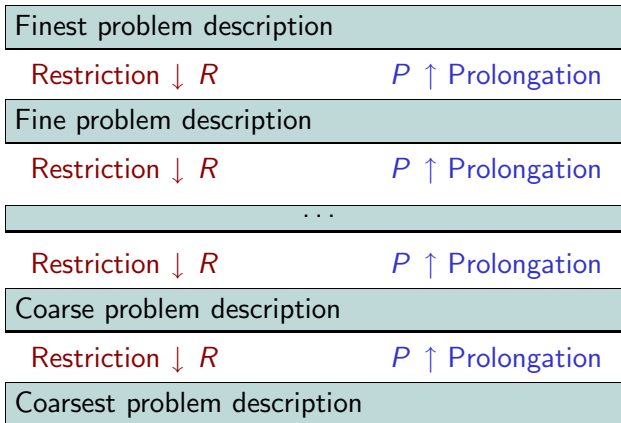
- optimization of **continuous** problems occurs in a many applications: shape optimization, data assimilation, control problems, ...
- Recent optimization methods have been designed to cope with these problems, including **multilevel/multigrid algorithms**.
- These algorithms involve the computation of a **hierarchy of problem descriptions**, linked by known operators.

**Our purpose:** review some trust-region and linesearch recent proposals for **unconstrained/ bound-constrained** optimization:

$$\min_{(x \geq 0)} f(x)$$

# Hierarchy of problem descriptions

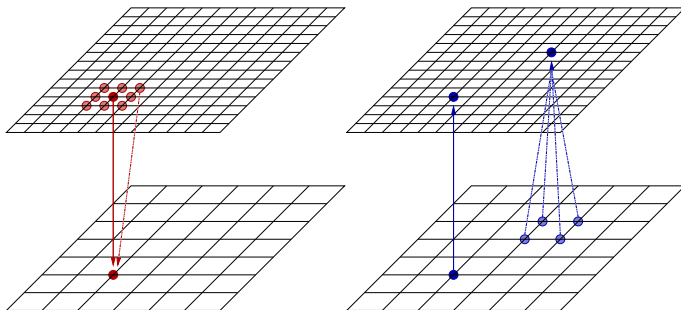
Can we use a structure of the form:



# Grid transfer operators

$$R_j : \mathbf{R}^{n_j} \rightarrow \mathbf{R}^{n_{j-1}} \quad \text{Restriction}$$

$$P_j : \mathbf{R}^{n_{j-1}} \rightarrow \mathbf{R}^{n_j} \quad \text{Prolongation}$$

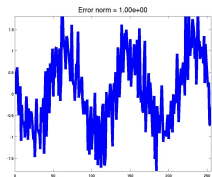


# Three keys to multigrid algorithms

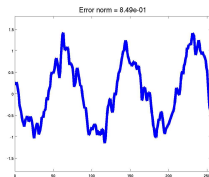
- **oscillatory** components of the error are representable on **fine** grids, but not on coarse grids
- iterative methods **reduce oscillatory components** much faster than smooth ones
- **smooth** on fine grids → **oscillatory** on coarse ones



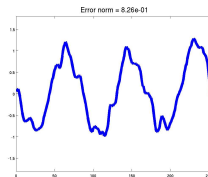
# Error at step $k$ of CG



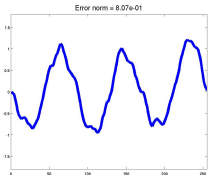
$k = 0$



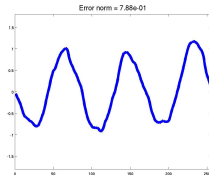
$k = 2$



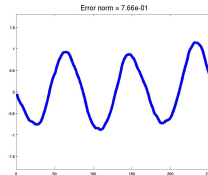
$k = 4$



$k = 6$



$k = 8$

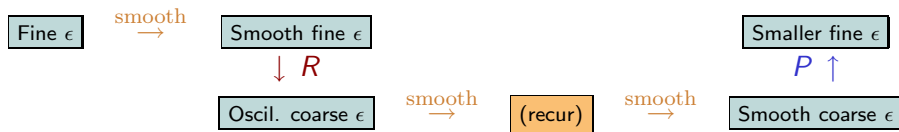


$k = 10$

Fast convergence of the oscillatory modes

# How to exploit these keys

Annihilate oscillatory error level by level:



**Note:**  $P$  and  $R$  are **not** orthogonal projectors!

A **very efficient** method for some linear systems  
 (when  $A(\text{smooth modes}) \in \text{smooth modes}$ )

# Past developments

- Fisher (1998), Nash (2000), Frese-Bouman-Sauer (1999), Nash-Lewis (2002), Oh-Milstein-Bouman-Webb (2003)  
(linesearch, no explicit smoothing, convergence?)
- Gratton-Sartenaer-T (2004), [Gratton-Mouffe-T-Weber \(2007\)](#)  
(trust-region, explicit-smoothing, convergence 1st + 2nd order, worst-case complexity)
- Wen-Goldfarb (2007)  
(linesearch, explicit smoothing, convergence on convex problems)
- [Gratton-T \(2007\)](#)  
(linesearch, implicit smoothing, convergence?)

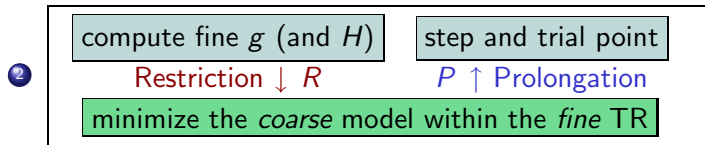
# Outline

- 1 Introduction
- 2 Recursive trust-region methods
- 3 Multigrid limited memory BFGS

# Recursive multilevel trust region

At each iteration at the **fine** level:

- 1 consider a **coarser description** model with a **trust region**



- 3 evaluate  $f$  at the trial point
- 4 if **achieved decrease**  $\approx$  **predicted decrease**:
  - **accept** the trial point
  - (possibly) **enlarge** the trust region
- 5 else:
  - **keep** current point
  - **shrink** the trust region

Until **convergence** :

- Choose either a Taylor or recursive model
  - Taylor model: compute a Taylor step
  - Recursive: **apply the Algo recursively**
- Evaluate change in the objective function
- If achieved reduction  $\approx$  predicted reduction,
  - accept trial point as new iterate
  - (possibly) enlarge the trust region
- else
  - reject the trial point
  - shrink the trust region
- Impose: **current TR  $\subseteq$  upper level TR**

# RMTR - Criticality Measure

- We only use **recursion** if:

$$\|g_{\text{low}}\| \stackrel{\text{def}}{=} \|Rg_{\text{up}}\| \geq \kappa_g \|g_{\text{up}}\| \quad \text{and} \quad \|g_{\text{low}}\| > \epsilon^g$$

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

$$\|g_i\| < \epsilon_i^g$$

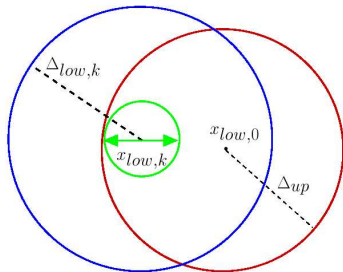
- BUT:** we must stop before we reach the border, or the inner trust region becomes too small

$$\|x_{\text{low}}^+ - x_{\text{low}}^0\|_{\text{low}} = \|P(x_{\text{low}}^+ - x_{\text{low}}^0)\|_{\text{up}} > (1 - \epsilon_\Delta)\Delta_{\text{up}}$$

# Why Change?

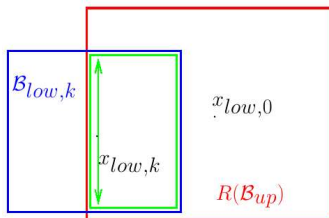
## RMTR

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



## RMTR- $\infty$

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





$\infty$ -norm in trust regions

- Possibility for **asymmetric** trust regions (more freedom)
- In lower levels: a **bound constrained subproblem**
- We will impose that the **lower level steps** must remain inside the **restriction** of the upper level trust region: If

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

then

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

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

$$\|s_{\text{up}}\|_{\infty} \leq \|P\|_{\infty} \|R\|_{\infty} \Delta_{\text{up}}.$$

# 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

$$\chi(x_{\text{low}}) = \left| \min_{\substack{d \in RB_{\text{up}} \\ \|d\| \leq 1}} \langle g_{\text{low}}, d \rangle \right|.$$

- 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

$$\chi < \epsilon_i^{\chi}.$$

# Model Reduction

- Taylor iterations in the 2-norm version satisfy the sufficient decrease condition

$$m_i(x) - m_i(x + s) \geq \kappa_{red} g(x) \min \left[ \frac{g(x)}{\beta}, \Delta \right].$$

- Taylor iterations in the  $\infty$ -norm are constrained; they satisfy

$$h_i(x) - h_i(x + s) \geq \kappa_{red} \chi_i(x) \min \left[ 1, \frac{\chi_i(x)}{\beta}, \Delta \right].$$

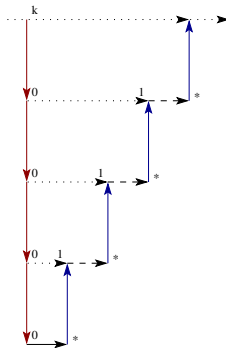
Until convergence :

- Choose either a Taylor or recursive model
  - Taylor model: compute a Taylor step ( $\infty$ -norm)
  - Recursive: apply the Algo recursively
- Evaluate change in the objective function
- If achieved reduction  $\approx$  predicted reduction,
  - accept trial point as new iterate
  - (possibly) enlarge the trust region
- else
  - reject the trial point
  - shrink the trust region
- Impose: current TR  $\subseteq$  Restricted upper level TR

# Mesh refinement, as different from...

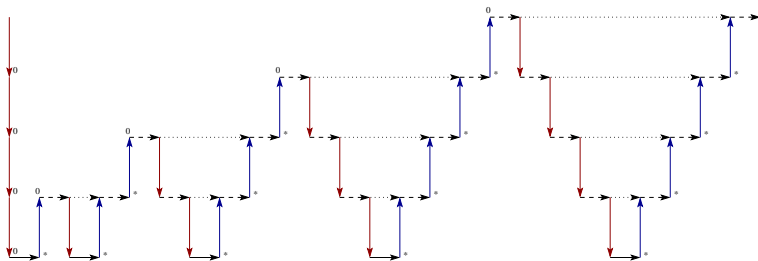
Computing **good starting points**:

- Solve the problem on the coarsest level  
 $\Rightarrow$  Good starting point for the next fine level
- Do the same on each level  
 $\Rightarrow$  Good starting point for the finest level
- Finally solve the problem on the finest level



# ... V-cycles and Full Multigrid (FMG)

- FMG : Combination of mesh refinement and V-cycles



# A first test case: the minimum surface problem (MS)

Consider the minimum surface problem

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

where  $K = \{v \in H^1(S_2) \mid v(x, y) = v_0(x, y) \text{ on } \partial S_2\}$  with

$$v_0(x, y) = \begin{cases} f(x), & y = 0, & 0 \leq x \leq 1, \\ 0, & x = 0, & 0 \leq y \leq 1, \\ f(x), & y = 1, & 0 \leq x \leq 1, \\ 0, & x = 1, & 0 \leq y \leq 1, \end{cases}$$

where  $f(x) = x(1 - x)$ .

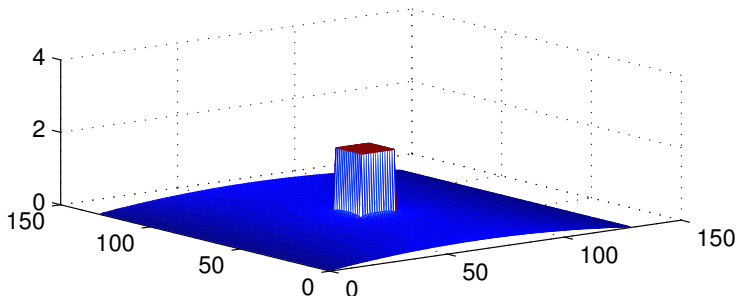
Finite element basis (P1 on triangles)  $\rightarrow$  convex problem.

Some typical results on MS ( $n = 127^2$ , 6 levels)

unconstrained

bound-constrained

	Mesh ref.	RMTR <sub>2</sub>	RMTR <sub>∞</sub>	Mesh ref.	RMTR <sub>∞</sub>
nit	1057	23	10	2768	214
nf	23	38	15	649	240
ng	16	28	14	640	236
nH	17	20	6	32	101





# RMTR- $\infty$ in practice

- Excellent numerical experience !
- Adaptable to bound-constrained problems
- Fully supported by (simpler?) theory
- Fortran code in the polishing stages ( $\rightarrow$  GALAHAD)

# Outline

- 1 Introduction
- 2 Recursive trust-region methods
- 3 Multigrid limited memory BFGS**

# Line search quasi-Newton method

Until **convergence** :

- Compute a **search direction**  $d = -Hg$
- Perform a **line search** along  $d$ , yielding

$$f(x^+) \leq f(x) + \alpha \langle g, d \rangle \quad \text{and} \quad \langle g^+, d \rangle \geq \beta \langle g, d \rangle$$

- **Update** the Hessian approximation to satisfy

$$H^+(g^+ - g) = x^+ - x \quad (\text{secant equation})$$

**BFGS** update:

$$H^+ = \left( I - \frac{ys^T}{y^T s} \right) H \left( I - \frac{ys^T}{y^T s} \right) + \frac{ss^T}{y^T s}$$

with

$$y = g^+ - g \quad \text{and} \quad s = x^+ - x$$

# Generating new secant equations

The fundamental secant equation:  $H^+ y = s$

Motivation:

$$G^{-1}y = s \quad \text{where} \quad G = \int_0^1 \nabla_{xx} f(x + ts) dt$$

Assume:

- known **invariants subspaces**  $\{S_i\}_{i=1}^p$  of  $G$ .
- known orthogonal projectors onto  $S_i$

$$G^{-1}S_i y = S_i G^{-1}y = S_i s$$

$\Rightarrow$  **new secant equation**:  $H^+ y_i = s_i$  with  $s_i = S_i s$  and  $y_i = S_i y$

## (Limited-memory) multi-secant variant

Until **convergence** :

- Compute a **search direction**  $d = -Hg$
- Perform a **linesearch** along  $d$ , yielding

$$f(x^+) \leq f(x) + \alpha \langle g, d \rangle \quad \text{and} \quad \langle g^+, d \rangle \geq \beta \langle g, d \rangle$$

- **Update** the Hessian approximation to satisfy

$$H^+ y = s \quad \text{and} \quad H^+ y_i = s_i \quad (i = 1, \dots, p)$$

Natural setting: limited-memory (BFGS) algorithm

$\Rightarrow$  apply L-BFGS with **secant pairs**  $(s_1, y_1), \dots, (s_p, y_p), (s, y)$

# Multigrid and invariant subspaces

Are they reasonable settings where the  $S_i$  are known?

**Idea:** Grid levels may provide invariant subspace information!

Fine grid: all modes

Less fine grid: all but the most oscillatory modes

Coarser grid: relatively smooth modes

Coarsest grid: smoothest modes

$P^i R^i$  provides a (cheap) approximate  $S_i$  operator!

# Multigrid multi-secant LBFGS... questions

How to *order* the secant pairs?

Update for lower grid levels (smooth modes) first or last?

How *exact* are the secant equations derived from the grid levels?

Measure by a the **norm of the perturbation** to true Hessian  $G$  for the secant equation to hold exactly:

$$\frac{\|E\|}{\|G\|} \leq \frac{\|Gs_i - y_i\|}{\|s_i\| \|G\|}$$

Should we control *collinearity*?

remember **nested structure** of the  $\mathcal{S}_i$  subspaces...

test cosines of angles between  $s$  and  $s_i$ ?

What information should we remember?

a memory-less BFGS method is possible!

Many possible choices!

# A second test case: Dirichlet-to-Neumann transfer (DN)

- It consists [Lewis,Nash,04] in finding the function  $a(x)$  defined on  $[0, \pi]$ , that minimizes

$$\int_0^\pi (\partial_y u(x, 0) - \phi(x))^2 dx,$$

where  $\partial_y u$  is the partial derivative of  $u$  with respect to  $y$ ,

- and where  $u$  is the solution of the boundary value problem

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



# A third test case: the multigrid model problem (MG)

- Consider here the two-dimensional model problem for multigrid solvers in the unit square domain  $S_2$

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

- $f$  such that the analytical solution is  $u(x, y) = 2y(1 - y) + 2x(1 - x)$ .
- 5-point finite-difference discretization
- Consider the variational formulation

$$\min_{x \in R^{nr}} \frac{1}{2} x^T A_r x - x^T b_r,$$

# Data assimilation: the 4D-Var functional

- Consider a **dynamical system**  $\dot{x} = f(t, x)$  with solution operator  $x(t) = \mathcal{M}(t, x_0)$ .
- **Observations**  $b_i$  at time  $t_i$  modeled by  $b_i = \mathcal{H}x(t_i) + \epsilon$ , where  $\epsilon$  is a Gaussian noise with covariance matrix  $R_i$ .
- The *a priori* error error covariance matrix on  $x_0$  is  $B$ .
- We wish to **find**  $x_0$  which minimizes

$$\frac{1}{2} \|x_0 - x_b\|_{B^{-1}}^2 + \frac{1}{2} \sum_{i=0}^N \|\mathcal{H}\mathcal{M}(t_i, x_0) - b_i\|_{R_i^{-1}}^2,$$

- The first term in the cost function is the background term, the second term is the observation term.

# A fourth test case: the shallow water system (SW)

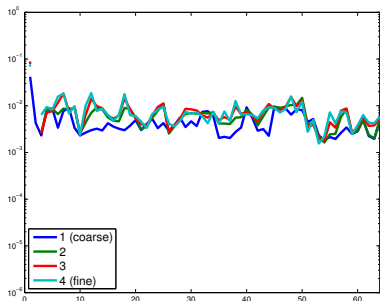
- The shallow system is often considered as a good approximation of the dynamical systems used in [ocean modeling](#).
- It is based on the Shallow Water equations

$$\begin{cases} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial v}{\partial y} - fv + g \frac{\partial z}{\partial x} = \lambda \Delta u \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + fu + g \frac{\partial z}{\partial y} = \lambda \Delta v \\ \frac{\partial z}{\partial t} + u \frac{\partial z}{\partial x} + v \frac{\partial z}{\partial y} + z \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = \lambda \Delta z \end{cases}$$

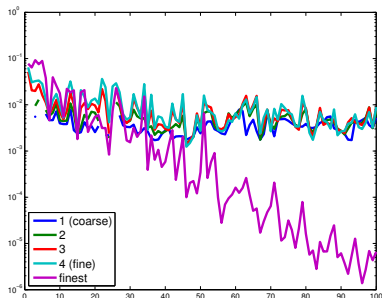
- Observations: every 5 points in the physical domain at every 5 time steps
- The a priori term is modeled using a diffusion operator [Weaver, Courtier, 2001]
- The system is time integrated using a leapfrog scheme.
- The damping in  $\lambda \Delta$  improves spatial solution smoothness

# Relative accuracy of the multigrid secant equations

Plot  $\|E\|/\|G\|$  against  $k$



MG (quadratic)



MS (nonquadratic)

$\Rightarrow$  size of perturbation **marginal**

# Testing a few variants

In our tests:

- old approximate secant pairs are discarded
- the LM updates are started with  $\frac{\langle y, s \rangle}{\|y\|^2}$  times the identity
- L-BFGS + 8 algorithmic variants:

	collinearity control (0.999)			
	no		yes	
Update order	mem	nomem	mem	nomem
Coarse first	CNM	CNN	CYM	CYN
Fine first	FNM	FNN	FYM	FYN

Memory management:

- \*M: past “exact” secant pairs are used (mem)
- \*N: past “exact” secant pairs are not used (nomem)

## The results

Algo levels/mem	DN ( $n = 255$ ) 7/10	MG ( $n = 127^2$ ) 6/9	SW ( $n = 63^2$ ) 3/5	MS ( $n = 127^2$ ) 4/5
L-BFGS	330/319	308/299	64/61	387/378
CNM	94/84	137/122	83/81	224/192
CNN	125/100	174/134	57/55	408/338
CYM	110/92	123/104	83/81	196/170
CYN	113/89	138/107	57/55	338/267
FNM	120/100	172/144	63/57	241/208
FNN	137/89	151/120	65/62	280/221
FYM	90/76	149/128	63/57	211/176
FYN	140/107	153/120	65/62	283/216

(NF/NIT)

# Further developments (not covered in this talk)

## Observations:

- L-BFGS acts as a smoother
- the step is asymptotically very smooth
- the eigenvalues associated with the smooth subspace are (relatively) close to each other
- the step is asymptotically an **approximate eigenvector**
- an equation of the form

$$Hs_i = \frac{\langle y_i, s_i \rangle}{\|y_i\|^2} s_i$$

can also be included...

⇒ **more (efficient) algorithmic variants!**

# Conclusions

Multilevel/multigrid optimization useful and interesting

Much remains to be explored

Recursive trust-region methods often very effective

Invariant subspace information useful for some problems

Multilevel quasi-Newton information exploitable



# Perspectives

- More complicated **constraints**
- Better understanding of **approximate** secant/eigen information
- **Invariant subspaces** without grids?
- Multilevel L-BFGS in RMTR?
- Combination with ACO methods?
- More test problems?

Thank you for your attention!