Advanced Algorithms in Nonlinear Optimization

Philippe Toint

Department of Mathematics, University of Namur, Belgium

(philippe.toint@fundp.ac.be)

Belgian Francqui Chair, Leuven, April 2009

Outline

Acknowledgements

This course would not have been possible without

- the Francqui Foundation and the Katholieke Universiteit Leuven,
- Moritz Diehl, Dirk Roose and Stefan Vandewalle (the gentle organizers),
- Fabian Bastin, Stefania Bellavia, Cinzia Cirillo, Coralia Cartis, Andy Conn, Nick Gould, Serge Gratton, Sven Leyffer, Vincent Malmedy, Benedetta Morini, Mélodie Mouffe, Annick Sartenaer, Katya Scheinberg, Dimitri Tomanos, Melissa Weber-Mendonça (my patient co-authors).
- Ke Chen, Patrick Laloyaux (who supplied pictures)

My grateful thanks to them all.

What is optimization?

The best choice subject to constraints

best	$\Rightarrow$	criterion, objective function
choice	$\Rightarrow$	variables whose value may be chosen
constraints	$\Rightarrow$	restrictions on allowed values of the variables

More formally

variables	$\Rightarrow$	$x = (x_1, x_2, \ldots, x_n)$
objective function	$\Rightarrow$	minimize/maximize $f(x)$
constraints	$\Rightarrow$	$c(x) \geq 0$

Note: maximize $f(x)$ equivalent to minimize $-f(x)$.

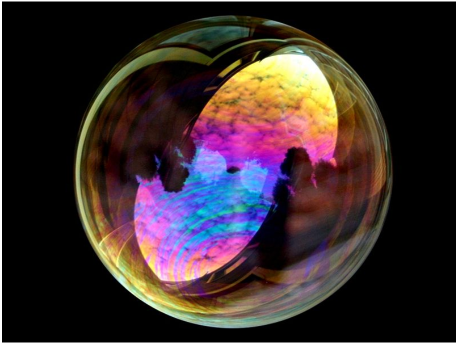
$$\min_x f(x)$$

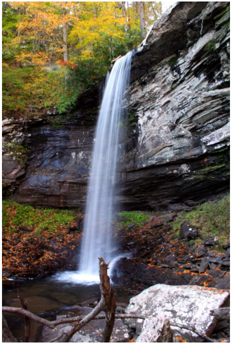
such that

$$c(x) \geq 0$$

(the general nonlinear optimization problem)
(+ conditions on x, f and c)

Nature optimizes



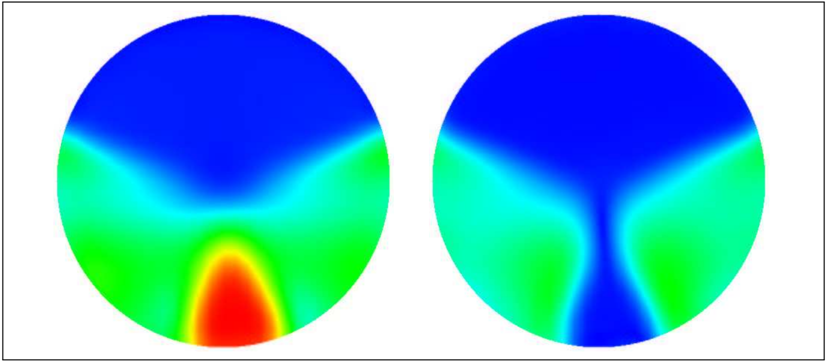


People optimize (daily)

Applications: PAL design (1)

Design of modern Progressive Adaptive Lenses:

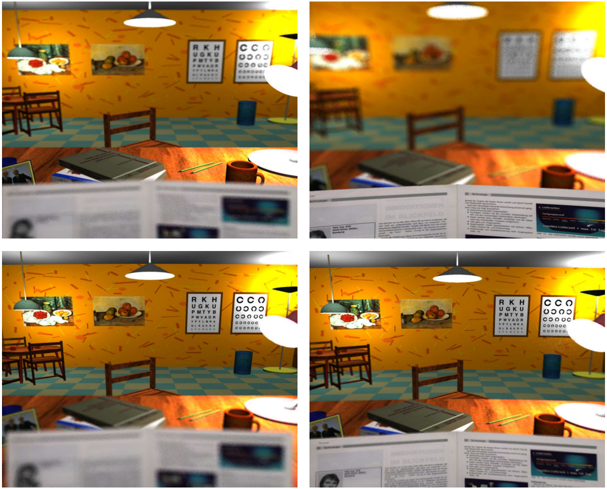
vary optical power of lenses while minimizing astigmatism



Loos, Greiner, Seidel (1997)

Applications: PAL design (2)

Achievements: Loos, Greiner, Seidel (1997)



uncorrected | short distance

long distance | PAL

Applications: PAL design (3)

Is this nonlinear ($\approx$ difficult)?

Assume the lens surface is $z = z(x, y)$. The optical power is

$$p(x,y) = \frac{N^3}{2}\left[\left(1+\left[\frac{\partial z}{\partial x}\right]^2\right)\frac{\partial^2 z}{\partial y^2} + \left(1+\left[\frac{\partial z}{\partial y}\right]^2\right)\frac{\partial^2 z}{\partial x^2} - 2\frac{\partial z}{\partial x}\frac{\partial z}{\partial y}\frac{\partial^2 z}{\partial x \partial y}\right]$$

where

$$N = N(x,y) = \frac{1}{\sqrt{1+\left[\frac{\partial z}{\partial x}\right]^2+\left[\frac{\partial z}{\partial y}\right]^2}}.$$

The surface astigmatism is then

$$a(x,y) = -2\sqrt{p(x,y) - N^4\left(\frac{\partial z}{\partial x}\frac{\partial z}{\partial y} - \left[\frac{\partial^2 z}{\partial x \partial y}\right]^2\right)}$$

Applications: Food sterilization (1)

A common problem in the food processing industry:

keep a max of vitamins while killing a prescribed fraction of the bacteria

heating in steam/hot water autoclaves

Sachs (2003)

Applications: Food sterilization (2)

Model: coupled PDEs

Concentration of micro-organisms and other nutrients:

$$\frac{\partial C}{\partial t}(x,t) = -K[\theta(x,t)]C(x,t),$$

where $\theta(x,t)$ is the temperature, and where

$$K[\theta] = K_1 e^{-K_2\left(\frac{1}{\theta}-\frac{1}{\theta_r}\right)} \quad \text{(Arrhenius equation)}$$

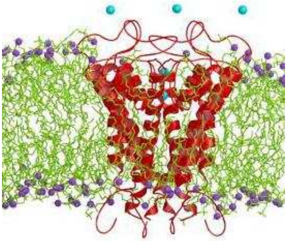
Evolution of temperature:

$$\rho c(\theta)\frac{\partial \theta}{\partial t} = \nabla \cdot [k(\theta)\nabla\theta],$$

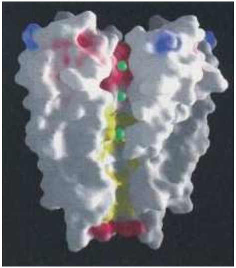
(with suitable boundary conditions: coolant, initial temperature,...)

Applications: biological parameter estimation (1)

K-channel in a the model of a neuron membrane:



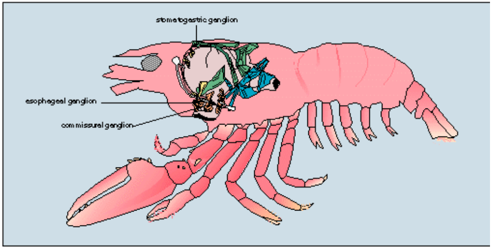
Sansom (2001)



Doyle *et al.* (1998)

Applications: biological parameter estimation (2)

Where are these neurons?



in a Pacific spiny lobster!

Simmers, Meyrand and Moulin (1995)

Applications: biological parameter estimation (3)

After gathering experimental data (applying a current to the cell):

estimate the biological model parameters that best fit experiments

Model:

- Activation: p independent gates
- Deactivation: n_h gates with different dynamics
- $n_h + 2$ coupled ODEs for the voltage, the activation level, the partial inactivations levels
- 5-points BDF for $\approx$ 50000 time steps
- $\Rightarrow$ very nonlinear!

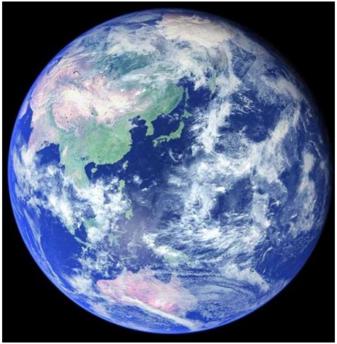
Applications: data assimilation for weather forecasting (1)

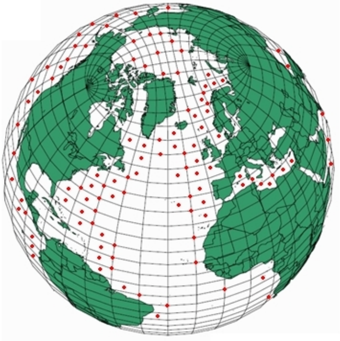
(Attempt to) predict. . .

- tomorrow's weather
- the ocean's average temperature next month
- future gravity field
- future currents in the ionosphere
- . . .

Applications: data assimilation for weather forecasting (2)

Data: temperature, wind, pressure, . . . everywhere and at all times!

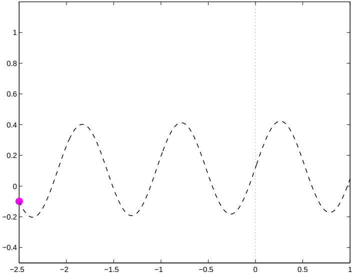




May involve up to 250000000 variables!

Applications: data assimilation for weather forecasting (3)

The principle:

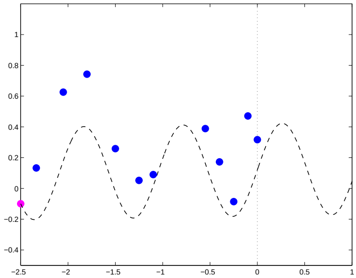


temp. vs. days

- Known situation 2.5 days ago and background prediction

Applications: data assimilation for weather forecasting (3)

The principle:



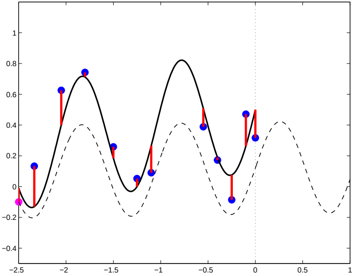
temp. vs. days

- Known situation 2.5 days ago and background prediction
- Record temperature for the past 2.5 days

Applications: data assimilation for weather forecasting (3)

The principle:

Minimize deviation between model and past observations



temp. vs. days

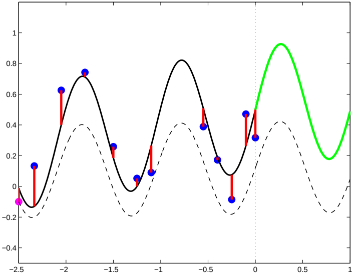
- Known situation 2.5 days ago and background prediction
- Record temperature for the past 2.5 days
- Run the model to minimize difference between model and observations

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

Applications: data assimilation for weather forecasting (3)

The principle:

Minimize deviation between model and past observations

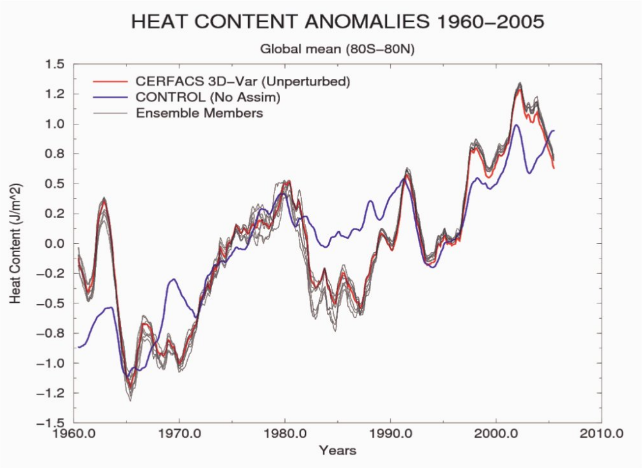


temp. vs. days

- Known situation 2.5 days ago and background prediction
- Record temperature for the past 2.5 days
- Run the model to minimize difference | between model and observations
- Predict temperature for the next day

Applications: data assimilation for weather forecasting (4)

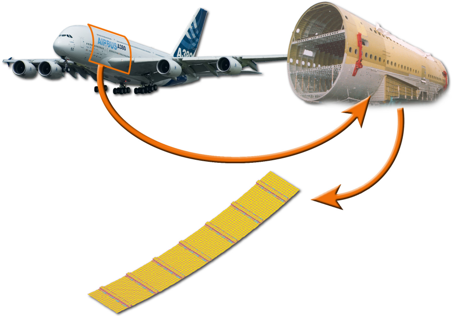
Analysis of the ocean's heat content: CERFACS (2009)



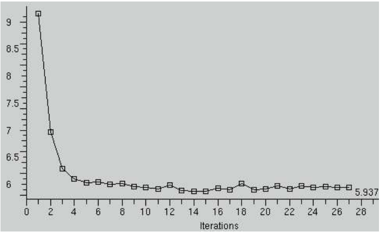
Much better fit!

Applications: aeronautical structure design

minimize weight while maintaining structural integrity



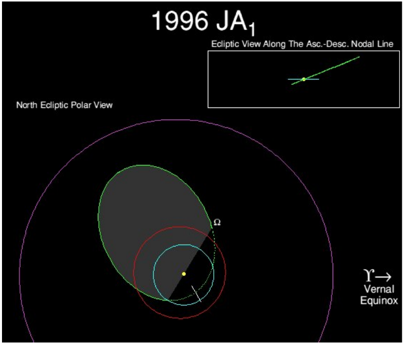
SAMTECH (2009)



mass reduction during optimization

Applications: asteroid trajectory matching

find today's asteroid whose orbital parameters
match best one observed 50 years ago



Milani, Sansaturio et al. (2005)

Applications: discrete choice modelling (1)

Context: simulation of individual choices in Transportation (or other) (mode, route, time of departure, . . .)

Random utility theory

An individual i assigns to alternative j the "utility"

$$U_{ij} = [\text{ parameters } \times \text{ explaining factors }] + [\text{ random error }]$$

Illustration :

$$U_{bus} = \text{distance} - 1.2 \times \text{price of ticket} - 2.1 \times \text{delay wrt to car travel} + \epsilon$$

Applications: discrete choice modelling (2)

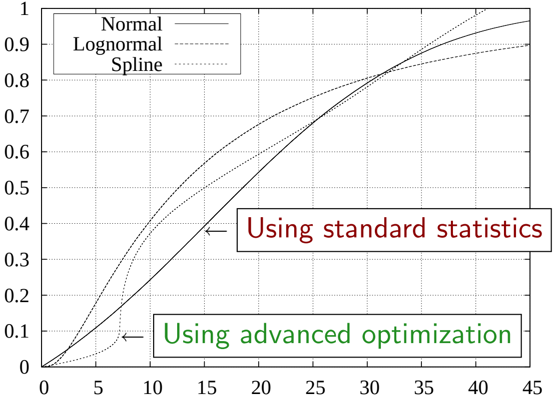
Probability that individual i chooses alternative j rather than alternative k given by

$$\text{prob}\left(U_{ij} \geq U_{ik} \quad \text{for all} \quad k\right)$$

Data: mobility surveys (MOBEL)

find the parameters in the utility function to maximize likelihood of observed behaviours

Applications: discrete choice modelling (3)



Estimation of the value of time lost in congested trafic (with and without advanced optimization)

Applications: Poisson image denoising (1)

Consider a two dimensional image with noise proportional to signal

$$z_{ij} = u_{ij} + n f(u_{ij})$$

where n is a random Gaussian noise. How to recover the original u_{ij}?

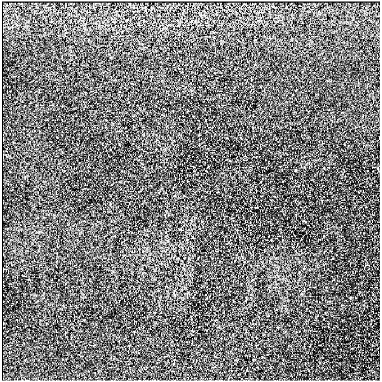
> use the pixel values as much as possible
> while minimizing sharp transitions (gradients)

This leads to the optimization problem

$$\min_u \sum_{ij\in\Omega} (u_{ij} - z_{ij}\log(u_{ij})) + \alpha \int_\Omega \|\nabla u\|$$

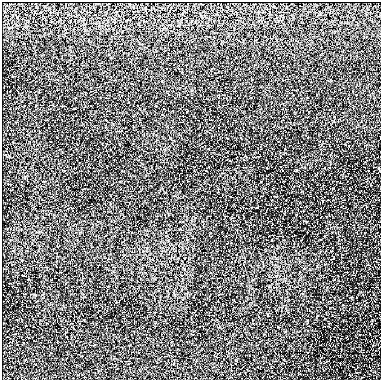
Applications: Poisson image denoising (2)

Some spectacular results: a 512×512 picture with 95% noise



Applications: Poisson image denoising (2)

Some spectacular results: a 512×512 picture with 95% noise



Chan and Chen (2007)

Applications: shock simulation in video games

Optimize the realism of the motion of multiple rigid bodies in space

$\Rightarrow$ "complementarity problem"

$$\nabla_q \Phi[q(t)] v(t) \geq 0$$

$$\Phi(q(t)) \geq 0$$

$$(q(t) = \text{positions},\ v(t) = \frac{dq}{dt}(t) = \text{velocities})$$

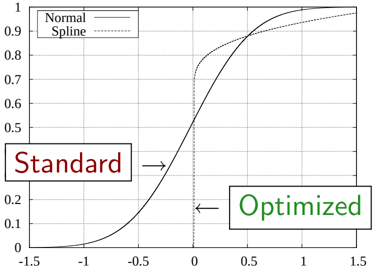
$\Rightarrow$ system of inequalities and equalities

used in realtime for video animation

Anitescu and Potra (1996)

Applications: finance

1. risk management
2. portofolio analysis
3. FX markets



Investment distribution
for the BoJ 1991-2004

4. . . .

Everybody loves
an optimizer!

Where does optimization come from?

"Nous sommes comme des nains juchés sur des épaules de géants, de telle sorte que nous puissions voir plus de choses et de plus éloignées que n'en voyaient ces derniers. Et cela, non point parce que notre vue serait puissante ou notre taille avantageuse, mais parce que nous sommes portés et exhaussés par la haute stature des géants."

"We are like dwarfs standing on the shoulders of giants, such that we can see more things and further away than they could. And this, not because our sight would be more powerful or our height more advantageous, but because we are carried and heigthened by the high stature of the giants."

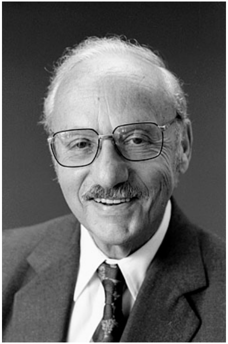
Bernard de Chartres (1130-1160)

Euclid (300 BC) Al-Khwarizmi (783-850)

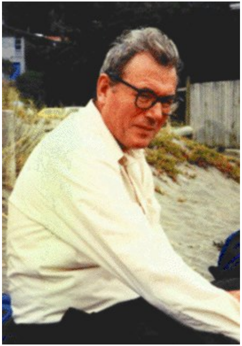
Isaac Newton (1642-1727) Leonhardt Euler (1707-1783)

J. de Lagrange (1735-1813) Friedrich Gauss (1777-1855)

Augustin Cauchy (1789-1857) George Dantzig (1914-2005)



Michael Powell Roger Fletcher



Return to the mathematical problem

$$\min_x f(x)$$

such that

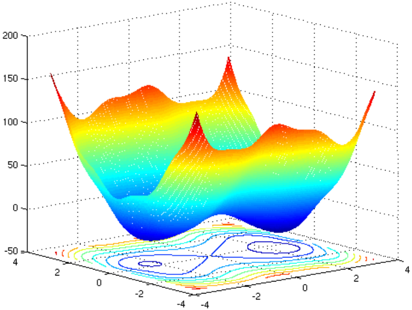
$$c(x) \geq 0$$

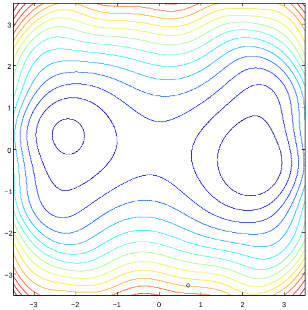
Difficulties:

- the objective function $f(x)$ is typically complicated (nonlinear)
- it is also often costly to compute
- there may be many variables
- the constraints $c(x)$ may defined a complicated geometry

An example unconstrained problem

$$\text{minimize}: \quad f(\alpha,\beta) = -10\alpha^2 + 10\beta^2 + 4\sin(\alpha\beta) - 2\alpha + \alpha^4$$





Two local minima: $(-2.20, 0.32)$ and $(2.30, -0.34)$

How to find them?

Trust-region methods

- iterative algorithms
- find local solutions only

Algorithm 1.1: The trust-region framework

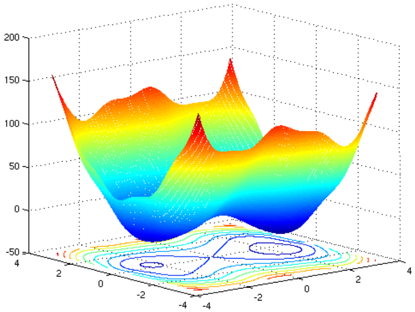
Until an (approximate) solution is found:

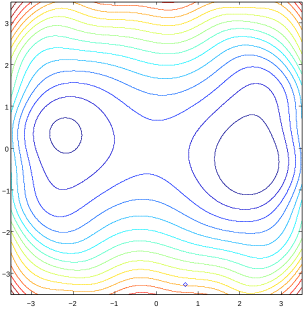
Step 1: use a model of the nonlinear function(s) within region where it can be trusted

Step 2: notion of sufficient decrease

Step 3: measure achieved and predicted reductions

Step 4: decrease the region radius if unsuccessful minimize : $f(\alpha, \beta) = -10\alpha^2 + 10\beta^2 + 4\sin(\alpha\beta) - 2\alpha + \alpha^4$

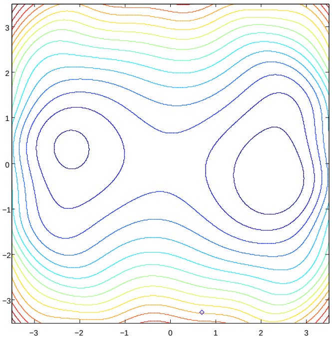




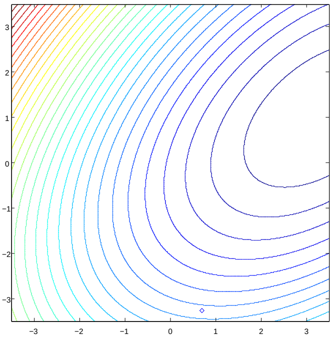
Two local minima: $(-2.20, 0.32)$ and $(2.30, -0.34)$

$$x_0 = (0.71, -3.27) \quad \text{and} \quad f(x_0) = 97.630$$

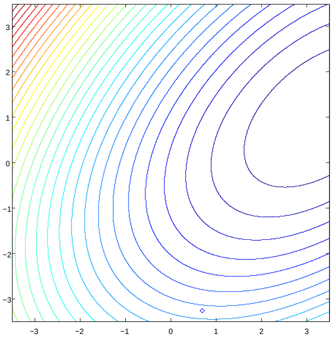
Contours of f

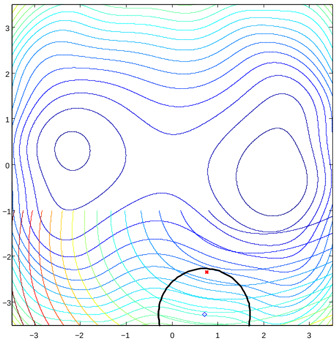


Contours of m_0 around x_0
(quadratic model)

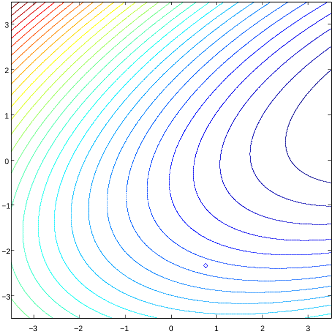


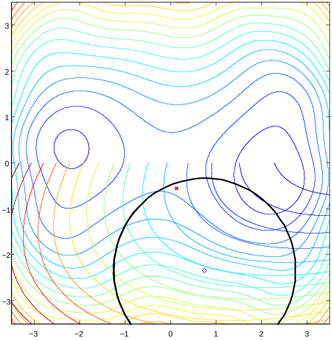
k	Δ_k	s_k	$f(x_k + s_k)$	$\Delta f / \Delta m_k$	x_{k+1}
0	**1**	$(0.05, 0.93)$	43.742	0.998	$x_0 + s_0$



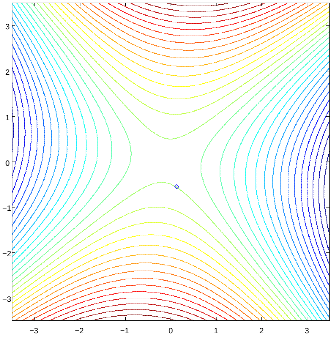


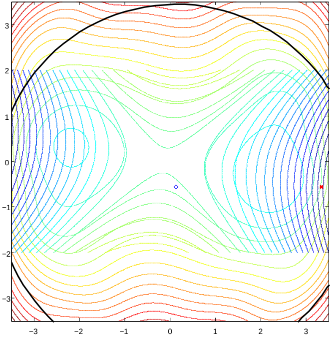
k	Δ_k	s_k	$f(x_k + s_k)$	$\Delta f / \Delta m_k$	x_{k+1}
0	**1**	$(0.05, 0.93)$	43.742	0.998	$x_0 + s_0$
1	**2**	$(-0.62, 1.78)$	2.306	1.354	$x_1 + s_1$



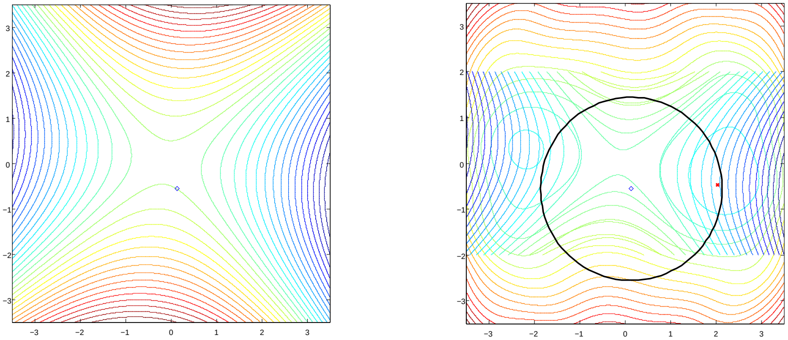


k	Δ_k	s_k	$f(x_k + s_k)$	$\Delta f / \Delta m_k$	x_{k+1}
0	**1**	$(0.05, 0.93)$	43.742	0.998	$x_0 + s_0$
1	**2**	$(-0.62, 1.78)$	2.306	1.354	$x_1 + s_1$
2	**4**	$(3.21, 0.00)$	6.295	-0.004	x_2

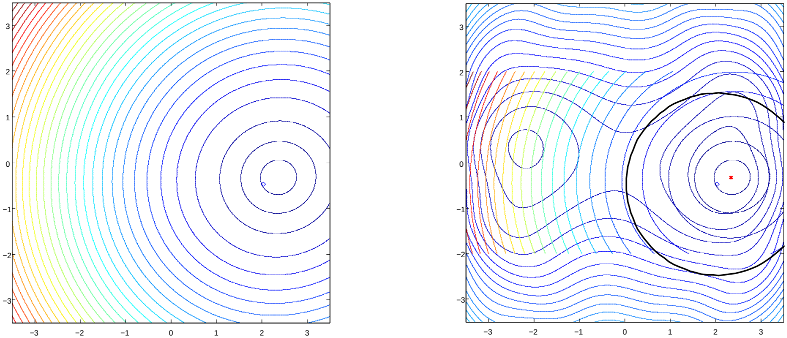




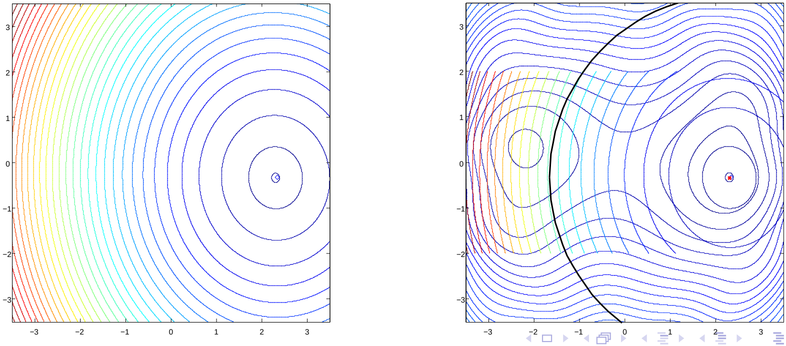
k	Δ_k	s_k	$f(x_k + s_k)$	$\Delta f / \Delta m_k$	x_{k+1}
0	**1**	$(0.05, 0.93)$	43.742	0.998	$x_0 + s_0$
1	**2**	$(-0.62, 1.78)$	2.306	1.354	$x_1 + s_1$
2	**4**	$(3.21, 0.00)$	6.295	-0.004	x_2
3	**2**	$(1.90, 0.08)$	-29.392	0.649	$x_2 + s_2$



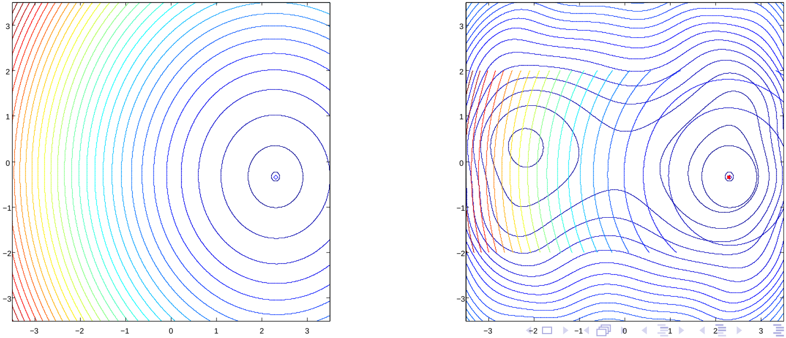
k	Δ_k	s_k	$f(x_k + s_k)$	$\Delta f / \Delta m_k$	x_{k+1}
0	**1**	$(0.05, 0.93)$	43.742	0.998	$x_0 + s_0$
1	**2**	$(-0.62, 1.78)$	2.306	1.354	$x_1 + s_1$
2	**4**	$(3.21, 0.00)$	6.295	-0.004	x_2
3	**2**	$(1.90, 0.08)$	-29.392	0.649	$x_2 + s_2$
4	**2**	$(0.32, 0.15)$	-31.131	0.857	$x_3 + s_3$



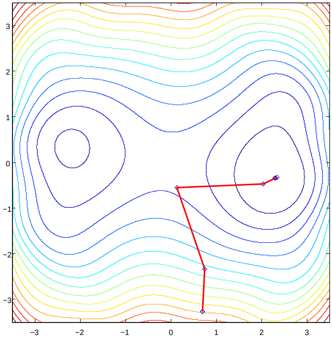
k	Δ_k	s_k	$f(x_k + s_k)$	$\Delta f / \Delta m_k$	x_{k+1}
0	**1**	$(0.05, 0.93)$	43.742	0.998	$x_0 + s_0$
1	**2**	$(-0.62, 1.78)$	2.306	1.354	$x_1 + s_1$
2	**4**	$(3.21, 0.00)$	6.295	-0.004	x_2
3	**2**	$(1.90, 0.08)$	-29.392	0.649	$x_2 + s_2$
4	**2**	$(0.32, 0.15)$	-31.131	0.857	$x_3 + s_3$
5	**4**	$(-0.03, -0.02)$	-31.176	1.009	$x_4 + s_4$



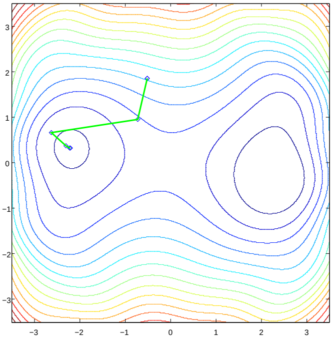
k	Δ_k	s_k	$f(x_k + s_k)$	$\Delta f / \Delta m_k$	x_{k+1}
0	**1**	$(0.05, 0.93)$	43.742	0.998	$x_0 + s_0$
1	**2**	$(-0.62, 1.78)$	2.306	1.354	$x_1 + s_1$
2	**4**	$(3.21, 0.00)$	6.295	-0.004	x_2
3	**2**	$(1.90, 0.08)$	-29.392	0.649	$x_2 + s_2$
4	**2**	$(0.32, 0.15)$	-31.131	0.857	$x_3 + s_3$
5	**4**	$(-0.03, -0.02)$	-31.176	1.009	$x_4 + s_4$
6	**8**	$(-0.02, 0.00)$	-31.179	1.013	$x_5 + s_5$



Path of iterates:



From another x_0:



And then. . .

Does it (always) work?

The answer tomorrow!

(and subsequent days for a (biased) survey of new optimization methods)

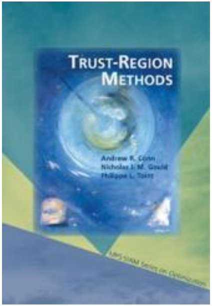
Thank you to you for your attention

H(

Lesson 2:

Trust-region methods for unconstrained problems

The basic text for this course



A. R. Conn, N. I. M. Gould and Ph. L. Toint,
Trust-Region Methods,
Nr 01 in the MPS-SIAM Series on Optimization,
SIAM, Philadelphia, USA, 2000.

2.1: Background material

Scalar mean-value theorems

Let $\mathcal{S}$ be an open subset of $\mathbb{R}^n$, and suppose $f : \mathcal{S} \rightarrow \mathbb{R}$ is continuously differentiable throughout $\mathcal{S}$. Then, if the segment $x + \theta s \in \mathcal{S}$ for all $\theta \in [0, 1]$,

$$f(x+s) = f(x) + \langle \nabla_x f(x + \alpha s), s \rangle$$

for some $\alpha \in [0, 1]$.

Let $\mathcal{S}$ be an open subset of $\mathbb{R}^n$, and suppose $f : \mathcal{S} \rightarrow \mathbb{R}$ is twice continuously differentiable throughout $\mathcal{S}$. Then, if the segment $x + \theta s \in \mathcal{S}$ for all $\theta \in [0, 1]$,

$$f(x+s) = f(x) + \langle \nabla_x f(x), s \rangle + \tfrac{1}{2} \langle s, \nabla_{xx} f(x + \alpha s) s \rangle$$

for some $\alpha \in [0, 1]$.

Vector mean-value theorem

Let $\mathcal{S}$ be an open subset of $\mathbb{R}^n$, and suppose $F : \mathcal{S} \rightarrow \mathbb{R}^m$ is continuously differentiable throughout $\mathcal{S}$. Then, if the segment $x + \theta s \in \mathcal{S}$ for all $\theta \in [0, 1]$,

$$F(x+s) = F(x) + \int_0^1 \nabla_x F(x + \alpha s) s \, d\alpha.$$

Taylor's scalar approximation theorems (1)

Let $\mathcal{S}$ be an open subset of $\mathbb{R}^n$, and suppose $f : \mathcal{S} \to \mathbb{R}$ is continuously differentiable throughout $\mathcal{S}$. Suppose further that $\nabla_x f(x)$ is Lipschitz continuous at x, with Lipschitz constant $\gamma(x)$ in some appropriate vector norm. Then, if the segment $x + \theta s \in \mathcal{S}$ for all $\theta \in [0, 1]$,

$$|f(x+s) - m(x+s)| \le \tfrac{1}{2}\gamma(x)\|s\|^2,$$

where

$$m(x+s) = f(x) + \langle \nabla_x f(x), s \rangle.$$

Taylor's scalar approximation theorems (2)

Let $\mathcal{S}$ be an open subset of $\mathbb{R}^n$, and suppose $f : \mathcal{S} \rightarrow \mathbb{R}$ is twice continuously differentiable throughout $\mathcal{S}$. Suppose further that $\nabla_{xx} f(x)$ is Lipschitz continuous at x, with Lipschitz constant $\gamma(x)$ in some appropriate vector norm and its induced matrix norm. Then, if the segment $x + \theta s \in \mathcal{S}$ for all $\theta \in [0, 1]$,

$$|f(x+s) - m(x+s)| \leq \tfrac{1}{6}\gamma(x)\|s\|^3,$$

where

$$m(x+s) =$$

$$f(x) + \langle \nabla_x f(x), s\rangle + \tfrac{1}{2}\langle s, \nabla_{xx} f(x) s\rangle.$$

Taylor's vector approximation theorem

Let $\mathcal{S}$ be an open subset of $\mathbb{R}^n$, and suppose $F : \mathcal{S} \to \mathbb{R}^m$ is continuously differentiable throughout $\mathcal{S}$. Suppose further that $\nabla_x F(x)$ is Lipschitz continuous at x, with Lipschitz constant $\gamma(x)$ in some appropriate vector norm and its induced matrix norm. Then, if the segment $x + \theta s \in \mathcal{S}$ for all $\theta \in [0, 1]$,

$$\|F(x+s) - M(x+s)\| \leq \tfrac{1}{2}\gamma(x)\|s\|^2,$$

where

$$M(x+s) = F(x) + \nabla_x F(x) s.$$

Newton's method

Solve

$$F(x) = 0$$

Idea: solve linear approximation

$$F(x) + J(x)s = 0$$

- quadratic local convergence
- ... but not globally convergent

Yet the basis of everything that follows

Unconstrained optimality conditions

Suppose that $f \in C^1$, and that x_* is a local minimizer of $f(x)$. Then

$$\nabla_x f(x_*) = 0.$$

Suppose that $f \in C^2$, and that x_* is a local minimizer of $f(x)$. Then the above holds and the objective function's Hessian at x_* is positive semi-definite, that is

$$\langle s, \nabla_{xx} f(x_*) s \rangle \geq 0 \text{ for all } s \in \mathbb{R}^n.$$

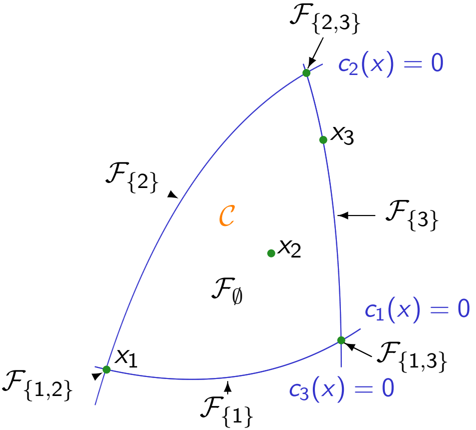
$$\langle s, \nabla_{xx} f(x_*) s \rangle > 0 \text{ for all } s \neq 0 \in \mathbb{R}^n$$

$\Rightarrow$ strict local solution

Constrained optimality conditions (1)

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & c_i(x) = 0, & \text{for } i \in \mathcal{E}, \\ \text{and} & c_i(x) \geq 0, & \text{for } i \in \mathcal{I}, \end{array}$$

Active set:



$$\mathcal{A}(x_1) = \{1, 2\}$$
$$\mathcal{A}(x_2) = \emptyset$$
$$\mathcal{A}(x_3) = \{3\}$$

Constrained optimality conditions (2): first order

Ignore constraint qualification!

Suppose that $f,\ c \in C^1$, and that x_* is a local solution. Then there exist a vector of *Lagrange multipliers* y_* such that

$$\nabla_x f(x_*) = \sum_{i \in \mathcal{E} \cup \mathcal{I}} [y_*]_i \nabla_x c_i(x_*)$$

$$\begin{aligned} c_i(x_*) &= 0 \text{ for all } i \in \mathcal{E} \\ c_i(x_*) \geq 0 \text{ and } [y_*]_i &\geq 0 \text{ for all } i \in \mathcal{I} \\ \text{and } c_i(x_*)[y_*]_i &= 0 \text{ for all } i \in \mathcal{I}. \end{aligned}$$

Lagrangian: $\ell(x, y) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} y_i c_i(x)$

Constrained optimality conditions (3): second order

Suppose that $f, c \in C^2$, and that x_* is a local minimizer of $f(x)$. Then there exist a vector of Lagrange multipliers y_* such that first-order conditions hold and

$$\langle s, \nabla_{xx}\ell(x_*, y_*)s\rangle \geq 0 \text{ for all } s \in \mathcal{N}_+$$

where $\mathcal{N}_+$ is the set of vectors s such that

$$\langle s, \nabla_x c_i(x_*)\rangle = 0 \quad \text{for all} \quad i \in \mathcal{E} \bigcup \{j \in \mathcal{A}(x_*) \bigcap \mathcal{I} \mid [y_*]_j > 0\}$$

and

$$\langle s, \nabla_x c_i(x_*)\rangle \geq 0 \quad \text{for all} \quad i \in \{j \in \mathcal{A}(x_*) \bigcap \mathcal{I} \mid [y_*]_j = 0\}$$

strict complementarity: $\langle s, \nabla_{xx}\ell(x_*, y_*)s\rangle > 0$ for all $s \in \mathcal{N}_+$ $(s \neq 0)$
$\Rightarrow$ strict local solution

Optimatity conditions (convex 1)

Assume now that $\mathcal{C}$ is convex

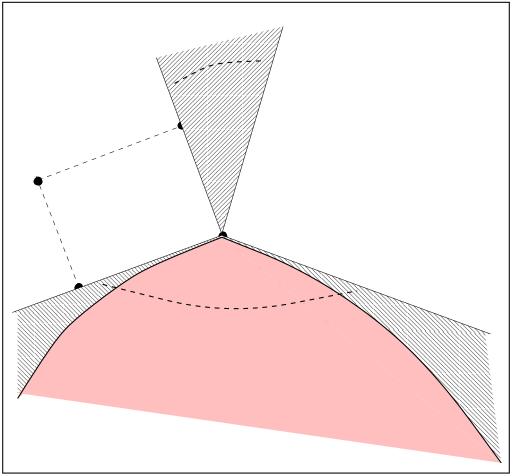
normal cone of $\mathcal{C}$ at $x \in \mathcal{C}$,

$$\mathcal{N}(x) \stackrel{\text{def}}{=} \{y \in \mathbf{R}^n \mid \langle y, u - x \rangle \leq 0, \ \forall u \in \mathcal{C}\}$$

tangent cone of $\mathcal{C}$ at $x \in \mathcal{C}$

$$\mathcal{T}(x) \stackrel{\text{def}}{=} \mathcal{N}(x)^0 = \mathrm{cl}\{\theta(u - x) \mid \theta \geq 0 \ \text{ and } \ u \in \mathcal{C}\}$$

Optimality conditions (convex 2)



The Moreau decomposition

Optimatity conditions (convex 2)

Suppose that $\mathcal{C} \neq \emptyset$ is convex, closed, that f is continuously differentiable in $\mathcal{C}$, and that x_* is a first-order critical point for the minimization of f over $\mathcal{C}$. Then, provided that constraint qualification holds,

$$-\nabla_x f(x_*) \in \mathcal{N}(x_*).$$

Conjugate gradients

Idea: minimize a convex quadratic on successive nested Krylov subspaces

Algorithm 2.1: Conjugate-gradients (CG)

Given x_0, set $g_0 = Hx_0 + c$ and let $p_0 = -g_0$.
For $k = 0, 1, \ldots,$ until convergence, perform the iteration

$$\begin{aligned}
\alpha_k &= \|g_k\|_2^2 / \langle p_k, Hp_k \rangle \\
x_{k+1} &= x_k + \alpha_k p_k \\
g_{k+1} &= g_k + \alpha_k Hp_k \\
\beta_k &= \|g_{k+1}\|_2^2 / \|g_k\|_2^2 \\
p_{k+1} &= -g_{k+1} + \beta_k p_k
\end{aligned}$$

Preconditioning

Idea: change the variables $\bar{x} = Rx$ and define $M = R^T R$.

Algorithm 2.2: Preconditioned CG

Given x_0, set $g_0 = Hx_0 + c$, and let $v_0 = M^{-1} g_0$ and $p_0 = -v_0$.
For $k = 0, 1, \ldots,$ until convergence, perform the iteration

$$\begin{aligned}
\alpha_k &= \langle g_k, v_k \rangle / \langle p_k, H p_k \rangle \\
x_{k+1} &= x_k + \alpha_k p_k \\
g_{k+1} &= g_k + \alpha_k H p_k \\
v_{k+1} &= M^{-1} g_{k+1} \\
\beta_k &= \langle g_{k+1}, v_{k+1} \rangle / \langle g_k, v_k \rangle \\
p_{k+1} &= -v_{k+1} + \beta_k p_k
\end{aligned}$$

Lanczos method

Idea: compute an orthonormal basis of the successive nested Krylov subspaces

$$\Rightarrow \text{ makes } Q_k^T H Q_k \text{ tridiagonal}$$

Algorithm 2.3: Lanczos

Given r_0, set $y_0 = r_0$, $q_{-1} = 0$.
For $k = 0, 1, \ldots$, perform the iteration,

$$\begin{aligned} \gamma_k &= \|y_k\|_2 \\ q_k &= y_k / \gamma_k \\ \delta_k &= \langle q_k, H q_k \rangle \\ y_{k+1} &= H q_k - \delta_k q_k - \gamma_k q_{k-1} \end{aligned}$$

Another view on the Conjugate-Gradients method

Conjugate Gradients = Lanczos + LDL^T (Cholesky)

$$\underbrace{\begin{array}{ccccc} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{array} \;\overset{\text{Lanczos}}{\longrightarrow}\; \begin{array}{ccccc} * & * & & & \\ * & * & * & & \\ & * & * & * & \\ & & * & * & * \\ & & & * & * \end{array} \;\overset{\text{Cholesky}}{\longrightarrow}\; \begin{array}{ccccc} * & & & & \\ * & * & & & \\ & * & * & & \\ & & * & * & \\ & & & * & * \end{array} \times \begin{array}{ccccc} * & * & & & \\ & * & * & & \\ & & * & * & \\ & & & * & * \\ & & & & * \end{array}}_{\text{Conjugate gradients in one of the Krylov subspaces}}$$

2.2: The trust-region algorithm

The trust-region idea

- use a model of the objective function
- define a trust-region where it is thought adequate

$$\mathcal{B}_k = \{x \in \mathbb{R}^n \mid \|x - x_k\|_k \leq \Delta_k\}$$

- find a trial point by sufficiently decreasing the model in $\mathcal{B}_k$
- compute the objective function at the trial point
- compare achived vs. predicted reductions
- reduce Δ_k if unsatisfactory

The basic trust-region algorithm

Algorithm 2.4: Basic trust-region algorithm (BTR)

Step 0: Initialization. x_0 and Δ_0 given, compute $f(x_0)$ and set $k = 0$.

Step 1: Model definition. Choose $\|\cdot\|_k$ and define a model m_k in $\mathcal{B}_k$.

Step 2: Step calculation. Compute s_k that sufficiently reduces the model m_k with $x_k + s_k \in \mathcal{B}_k$.

Step 3: Acceptance of the trial point. Compute $f(x_k + s_k)$ and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

If $\rho_k \geq \eta_1$, then define $x_{k+1} = x_k + s_k$; otherwise define $x_{k+1} = x_k$.

Step 4: Trust-region radius update.

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2\Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1\Delta_k, \gamma_2\Delta_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment k by one and go to Step 1.

2.3: Basic convergence theory

Assumptions

- $f \in C^2$
- $f(x) \geq \kappa_{\text{lbf}}$
- $\|\nabla_{xx} f(x)\| \leq \kappa_{\text{ufh}}$

- $m_k \in C^2(\mathcal{B}_k)$
- $m_k(x_k) = f(x_k)$
- $g_k \stackrel{\text{def}}{=} \nabla_x m_k(x_k) = \nabla_x f(x_k)$
- $\|\nabla_{xx} m_k(x)\| \leq \kappa_{\text{umh}} - 1$ for all $x \in \mathcal{B}_k$

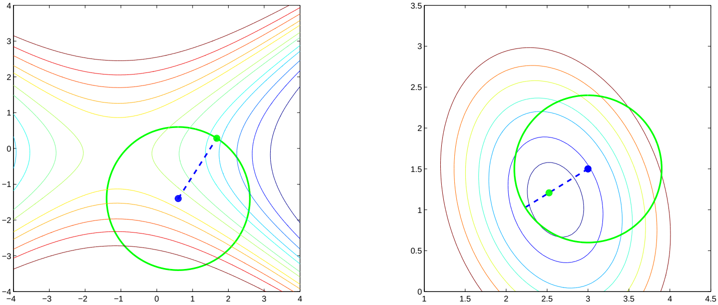
- $\frac{1}{\kappa_{\text{une}}} \|x\|_k \leq \|x\| \leq \kappa_{\text{une}} \|x\|_k$

. . . but use $\|\cdot\|_k = \|\cdot\|_2$ in what follows!

The Cauchy step

Idea: minimize m_k on the Cauchy arc

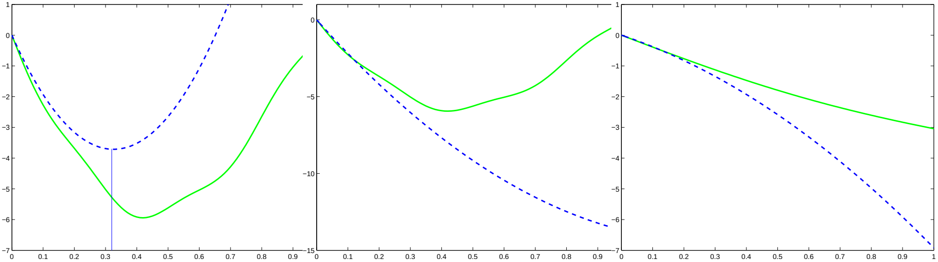
$$x_k^C(t) \stackrel{\text{def}}{=} \{x \mid x = x_k - t g_k,\ t \geq 0 \text{ and } x \in \mathcal{B}_k\}.$$



$\Rightarrow$ the Cauchy point

The Cauchy point for quadratic models

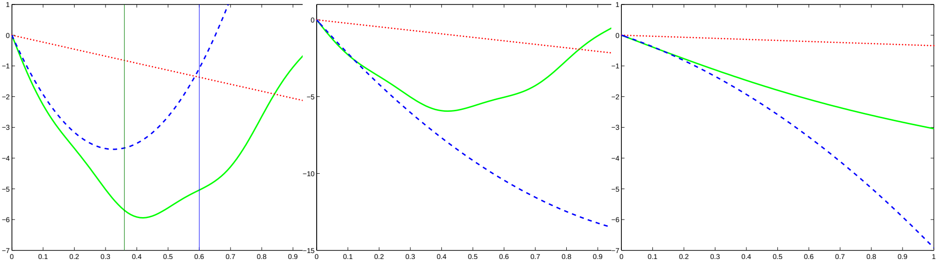
Three cases when minimizing the quadratic m_k along the Cauchy arc:



$$m_k(x_k) - m_k(x_k^C) \geq \frac{1}{2}\|g_k\| \min\left[\frac{\|g_k\|}{\beta_k}, \Delta_k\right]$$

The Cauchy point for general models

Three cases when minimizing the general m_k along the Cauchy arc:



$$m_k(x_k) - m_k(x_k^{\rm AC}) \geq \kappa_{\rm dcp} \|g_k\| \min\left[\frac{\|g_k\|}{\beta_k}, \Delta_k\right]$$

The meaning of sufficient decrease

In both cases, we get:

Sufficient decrease condition:

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}} \|g_k\| \min\left[\frac{\|g_k\|}{\beta_k}, \Delta_k\right],$$

Immediate consequence:

Suppose that $\nabla_x f(x_k) \neq 0$. Then $m_k(x_k + s_k) < m_k(x_k)$ and $s_k \neq 0$.

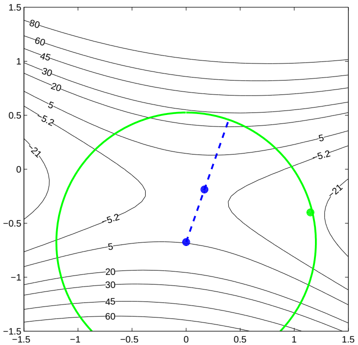
$\Rightarrow \rho_k$ is well defined!

The exact minimizer is OK

Suppose that, for all k, s_k ensures that

$$m_k(x_k) - m_k(x_k + s_k)\kappa_{\rm amm}[m_k(x_k) - m_k(x_k^{\rm M})],$$

Then sufficient decrease is obtained.



Taylor and minimum radius

For all k, $|f(x_k + s_k) - m_k(x_k + s_k)| \leq \kappa_{\text{ubh}} \Delta_k^2$,

Suppose that $g_k \neq 0$ and that

$$\Delta_k \leq \frac{\kappa_{\text{mdc}} \|g_k\| (1 - \eta_2)}{\kappa_{\text{ubh}}}.$$

Then iteration k is very successful and

$$\Delta_{k+1} \geq \Delta_k.$$

Suppose that $\|g_k\| \geq \kappa_{\text{lbg}} > 0$ for all k. Then is a constant $\kappa_{\text{lbd}} > 0$ such that, for all k

$$\Delta_k \geq \kappa_{\text{lbd}}.$$

First-order convergence (1)

Suppose that there are only finitely many successful iterations. Then $x_k = x_*$ for all sufficiently large k and x_* is first-order critical.

Suppose that there are infinitely many successful iterations. Then

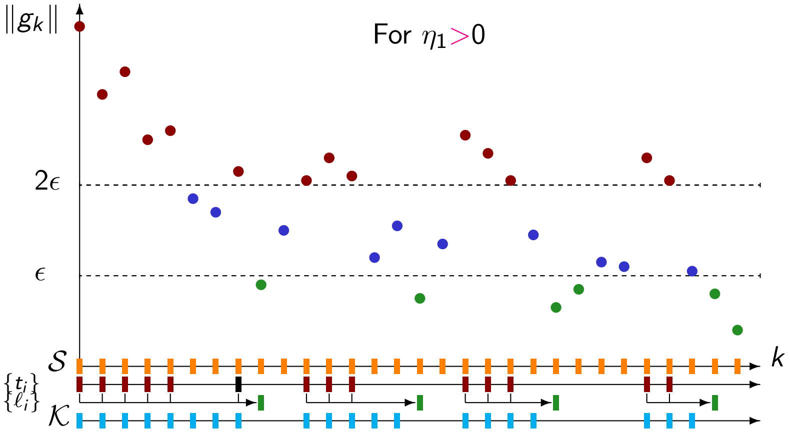
$$\liminf_{k\to\infty} \|\nabla_x f(x_k)\| = 0.$$

idea: infinite descent if not critical

First-order convergence (2)

Suppose that there are infinitely many successful iterations.

$$\text{Then} \quad \lim_{k\to\infty} \|\nabla_x f(x_k)\| = 0.$$



Convex models (1)

Suppose that $\lambda_{\min}[\nabla_{xx} m_k(x)] \geq \epsilon$ for all $x \in [x_k, x_k + s_k]$ and for some $\epsilon > 0$. Then

$$\|s_k\| \leq \frac{2}{\epsilon}\|g_k\|.$$

idea: m_k curves upwards!

Suppose that $\{x_{k_i}\} \to x_*$ and x_* is first-order critical, and that there is a constant $\kappa_{\rm smh} > 0$ such that

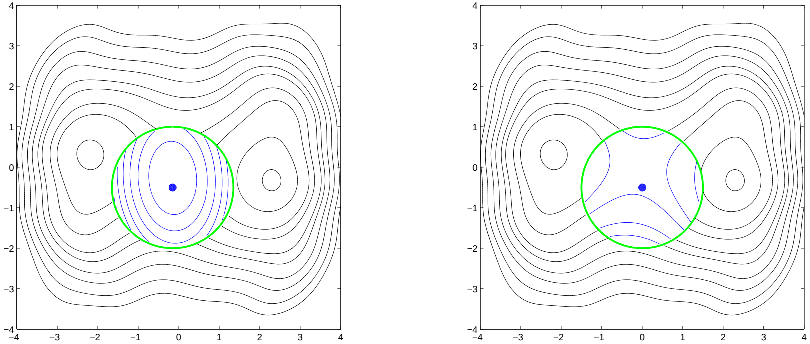
$$\min_{x \in \mathcal{B}_k} \lambda_{\min}[\nabla_{xx} m_k(x)] \geq \kappa_{\rm smh}$$

whenever x_k is sufficiently close to x_* Suppose finally that $\nabla_{xx} f(x_*)$ is nonsingular. Then the complete sequence of iterates $\{x_k\}$ converges to x_*.

idea: steps too short to escape local basin

Convex models (2)

But. . .



Asymptotically exact Hessians

Assume also that

$$\lim_{k\to\infty} \|\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)\| = 0 \text{ whenever } \lim_{k\to\infty} \|g_k\| = 0$$

Suppose that $\{x_{k_i}\} \to x_*$ and x_* is first-order critical, that $s_k \neq 0$ for all k sufficiently large, and that $\nabla_{xx} f(x_*)$ is positive definite. Then the complete sequence of iterates $\{x_k\}$ converges to x_*, all iterations are eventually very successful and the trust-region radius Δ_k is bounded away from zero.

idea: sufficient decrease implies that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mqd}} \|s_k\|^2 > 0.$$

Then $\rho_k \to 1$.

Second order: the eigen point

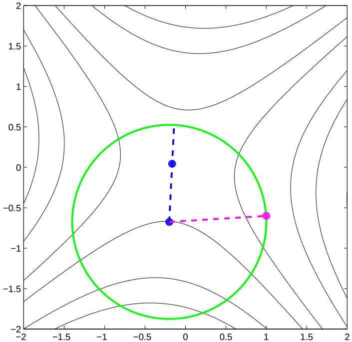
Assume $0 > \tau_k \in \sigma(H_k)$.

Then fine the eigen direction u_k such that

$$\langle u_k, g_k \rangle \le 0, \quad \|u_k\|_k = \Delta_k \quad \langle u_k, H_k u_k \rangle \le \kappa_{\rm snc} \tau_k \Delta_k^2,$$

Minimize the model along u_k to compute the eigen point:

$$m_k(x_k^{\rm E}) = m_k(x_k + t_k^{\rm E} u_k) = \min_{t \in (0,1]} m_k(x_k + t u_k)$$



Model decrease at the eigen point

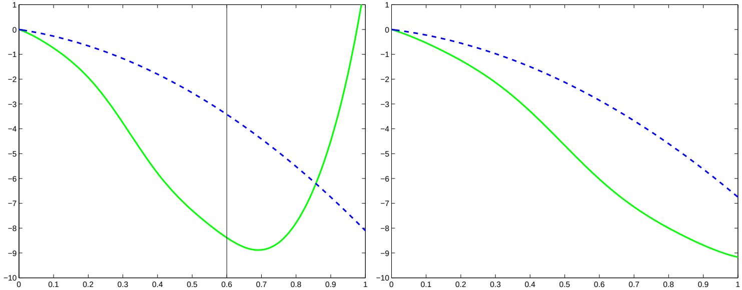
Suppose: $0 > \tau_k \in \sigma(H_k)$, u_k is an eigen direction and

$$\|\nabla_{xx} m_k(x) - \nabla_{xx} m_k(y)\| \leq \kappa_{\rm lch} \|x - y\|$$

for all $x, y \in \mathcal{B}_k$. Then

$$m_k(x_k) - m_k(x_k^{\rm E}) \geq -\kappa_{\rm sod} \tau_k \min[\tau_k^2, \Delta_k^2].$$

(quadratic or general model)



Second order: convergence theorems

$$\limsup_{k\to\infty} \lambda_{\min}[\nabla_{xx} f(x_k)] \geq 0.$$

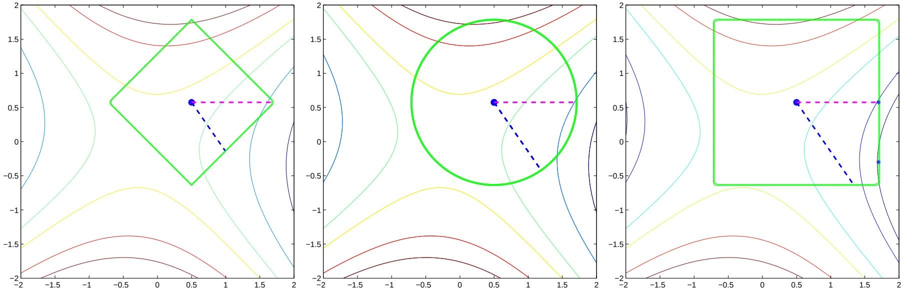
Suppose that x_* is an isolated limit point of the sequence of iterates $\{x_k\}$. Then x_* is a second-order critical point.

Assume also that, for $\gamma_3 > 1$,

$$\rho_k \geq \eta_2 \text{ and } \Delta_k \leq \Delta_{\max} \rightarrow \Delta_{k+1} \in [\gamma_3 \Delta_k, \gamma_4 \Delta_k]$$

Let x_* be any limit point of the sequence of iterates. Then x_* is a second-order critical point.

Different trust-region norms



Using norms for scaling

Idea: change the variables

$$S_k w = s$$

Then

$$m_k^S(x_k + w) \approx f(x_k + S_k w) \stackrel{\text{def}}{=} f^S(w),$$

$$\mathcal{B}_k^S = \{x_k + w \mid \|w\| \le \Delta_k\}.$$

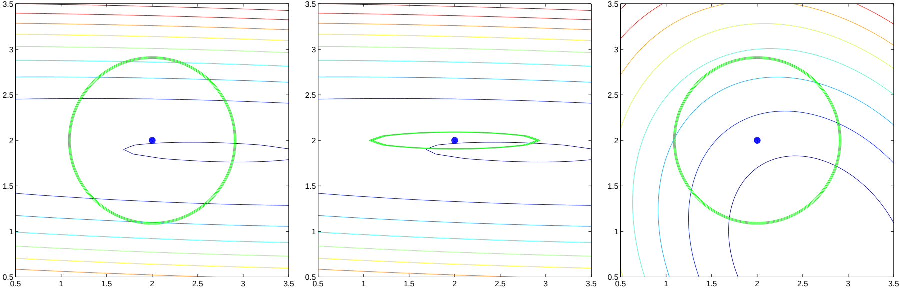
$$m_k^S(x_k) = f(x_k), \quad g_k^S = \nabla_w f^S(0) = S_k^T \nabla_x f(x_k)$$

$$H_k^S \approx \nabla_{ww} f^S(0) = S_k^T \nabla_{xx} f(x_k) S_k.$$

Thus

$$\begin{aligned} m_k^S(x_k + w) &= f(x_k) + \langle g_k^S, w\rangle + \tfrac{1}{2}\langle w, H_k^S w\rangle \\ &= f(x_k) + \langle S_k^T \nabla_x f(x_k), w\rangle + \tfrac{1}{2}\langle w, S_k^T H_k S_k w\rangle \\ &= f(x_k) + \langle \nabla_x f(x_k), S_k w\rangle + \tfrac{1}{2}\langle S_k w, H_k S_k w\rangle \\ &= f(x_k) + \langle \nabla_x f(x_k), s\rangle + \tfrac{1}{2}\langle s, H_k s\rangle \\ &= m_k(x_k + s) \end{aligned}$$

Scaling: the geometry



2.4: Solving the subproblem

The subproblem

Assume

- Euclidean norm
- quadratic model (possibly non-convex)
- (drop the index k)

$$\min_{s \in \mathbf{R}^n} \quad q(s) \equiv \langle g, s \rangle + \tfrac{1}{2} \langle s, Hs \rangle$$

$$\text{subject to} \quad \|s\|_2 \leq \Delta$$

Possible approaches

- exact minimization
- truncated conjugate-gradients
- CG + Lanczos (GLTR)
- doglegs
- eigenvalue based methods
- (projection methods)

The exact minimizer

Any global minimizer of $q(s)$ subject to $\|s\|_2 = \Delta$ satisfies the equation

$$H(\lambda^{\rm M})s^{\rm M} = -g,$$

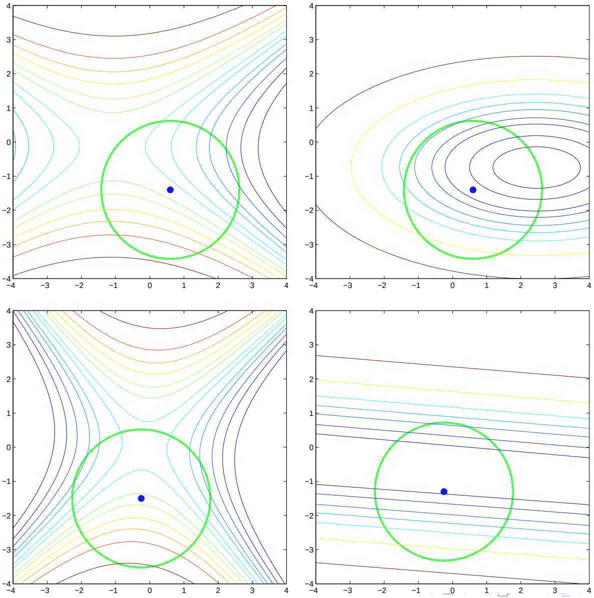
where

- $H(\lambda^{\rm M}) \stackrel{\rm def}{=} H + \lambda^{\rm M} I$ is positive semi-definite,
- $\lambda^{\rm M} \geq 0$ and
- $\lambda^{\rm M}(\|s^{\rm M}\|_2 - \Delta) = 0$.

If $H(\lambda^{\rm M})$ is positive definite, $s^{\rm M}$ is unique.

Note: $\lambda^{\rm M}$ is the Lagrange multiplier

The exact minimizer: a geometrical view



Finding the exact minimizer

Eigenvalue decomposition of H:

$$H = U^T \Lambda U$$

where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Characterization implies that

$$\lambda^{\mathrm{M}} \geq -\lambda_1$$

Suppose that $\lambda > -\lambda_1$ and define

$$s(\lambda) = -H(\lambda)^{-1} g = -U^T (\Lambda + \lambda I)^{-1} U g$$

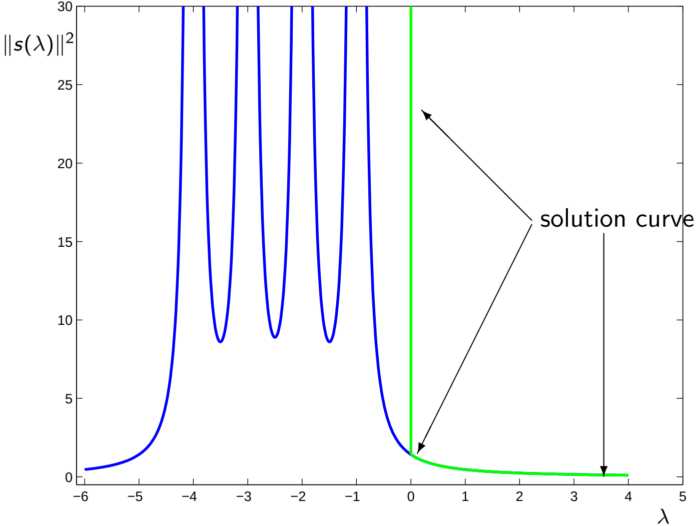
New formulation (one dimensional):

$$\|s(\lambda)\|_2 \leq \Delta$$

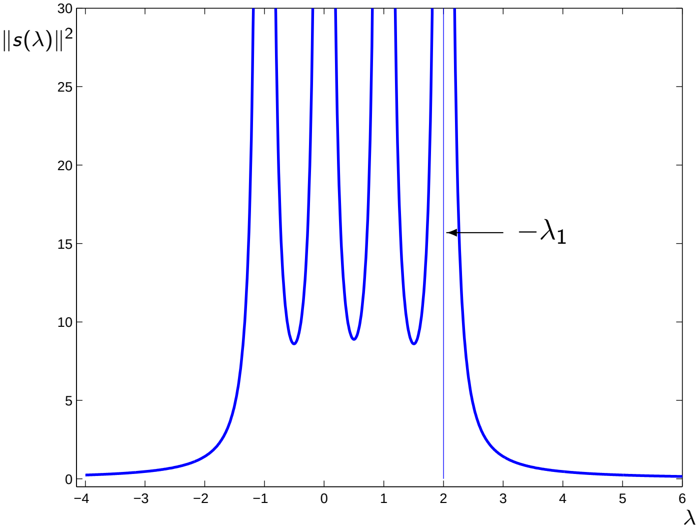
$$\|s(\lambda)\|_2^2 = \|U^T (\Lambda + \lambda I)^{-1} U g\|_2^2 = \|(\Lambda + \lambda I)^{-1} U g\|_2^2 = \sum_{i=1}^{n} \frac{\gamma_i^2}{(\lambda_i + \lambda)^2}$$

where $\gamma_i = [Ug]_i$.

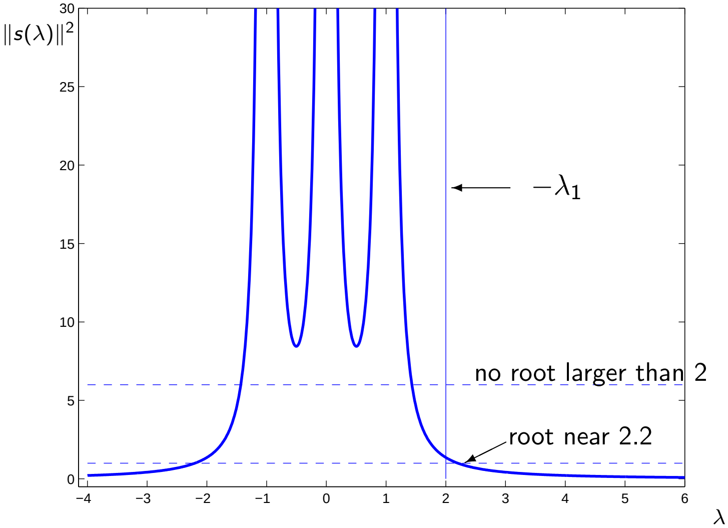
The convex case



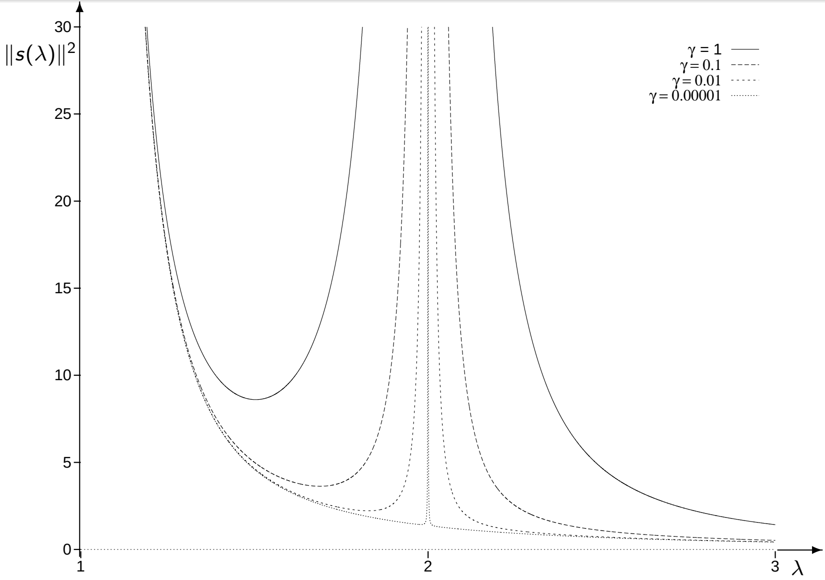
A nonconvex case



The hard case: $\gamma_1 = 0$



Near the hard case: $\gamma_1 \approx 0$

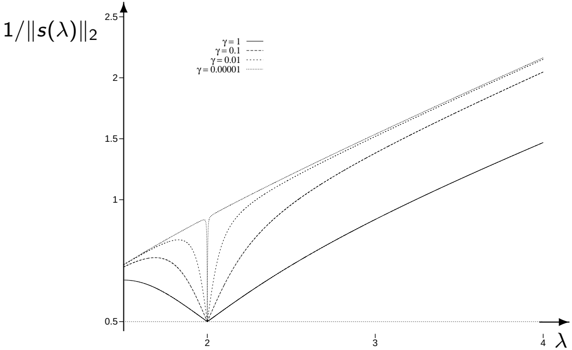


The secular equation

Idea: consider the secular equation

$$\phi(\lambda) \stackrel{\text{def}}{=} \frac{1}{\|s(\lambda)\|_2} - \frac{1}{\Delta} = 0$$

Then



$\Rightarrow$ apply Newton's method to $\phi(\lambda) = 0$: $\lambda^+ = \lambda - \phi(\lambda)/\phi'(\lambda)$

The derivatives of $\phi(\lambda)$

Suppose $g \neq 0$. Then

- $\phi(\lambda)$ is strictly increasing $(\lambda > -\lambda_1)$, and concave.
- $$\phi'(\lambda) = -\frac{\langle s(\lambda), \nabla_\lambda s(\lambda)\rangle}{\|s(\lambda)\|_2^3}$$

  where

  $$\nabla_\lambda s(\lambda) = -H(\lambda)^{-1} s(\lambda).$$

Note: if $H(\lambda) = LL^T$ and $Lw = s(\lambda)$, then

$$\langle s(\lambda), \nabla_\lambda s(\lambda)\rangle = \langle s(\lambda), L^{-T} L^{-1} s(\lambda)\rangle = \|w\|^2$$

Newton's method on the secular equation

Algorithm 2.5: Exact trust-region solver

Let $\lambda > -\lambda_1$ and $\Delta > 0$ be given.

1. Factorize $H(\lambda) = LL^T$.
2. Solve $LL^T s = -g$.
3. Solve $Lw = s$.
4. Replace λ by $\lambda + \left(\frac{\|s\|_2 - \Delta}{\Delta}\right)\left(\frac{\|s\|_2^2}{\|w\|_2^2}\right)$.

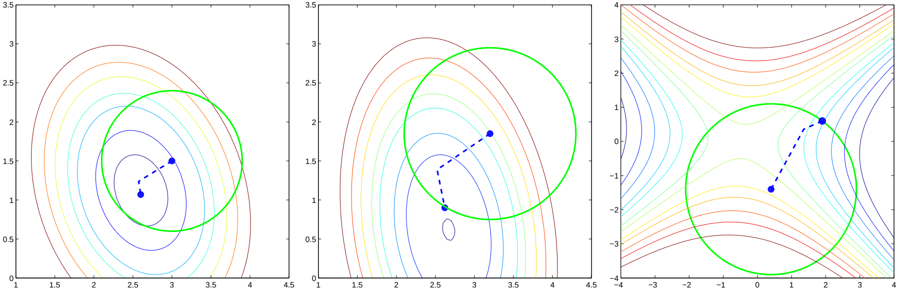
But . . . more complications due to

- bracketing the root (initial + update)
- termination rule
- may be preconditioned

Moré (1978), Moré-Sorensen (1983), Dollar-Gould-Robinson (2009)

Approximate solution by truncated CG

Fact: CG never reenters the ℓ_2 trust-region



May be preconditioned

Steihaug (1983), T. (1981)

Approximate solution by the GLTR

ST might hit the boundary for steepest descent step $\Rightarrow$ sometimes slow

Idea: solve the subproblem on the nested Krylov subspaces

Algorithm 2.6: Two-phase GLTR algorithm

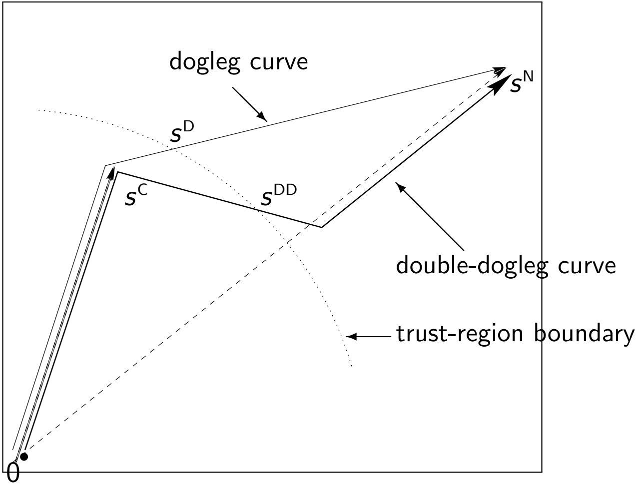
- as long as interior: conjugate-gradients
- on the boundary: Lanczos method + subproblem solution in Krylov space

(smooth transition)

Gould-Lucidi-Roma-T. (1999)

Doglegs

Idea: use steepest descent and the full Newton'step (requires convexity?)



Powell (1970), Dennis-Mei (1979)

An eigenvalue approach

Rewrite

$$(H + \lambda M)s = -g$$

as

$$(H \quad g) \begin{pmatrix} s \\ 1 \end{pmatrix} = -\lambda M s$$

or (introducing the parameter θ)

$$\begin{pmatrix} H & g \\ g^T & \theta \end{pmatrix} \begin{pmatrix} s \\ 1 \end{pmatrix} = (-\lambda) \begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s \\ 1 \end{pmatrix}$$

$\Rightarrow$ choose θ such that

- $\lambda \geq 0$,
- $H + \lambda M$ positive semi-definite
- $\lambda(\|s\|_M - \Delta) = 0$ Rendl-Wolkowicz (1997), Rojas-Santos-Sorensen (1999)

Bibliography for lesson 2 (1)

- J. E. Dennis and H. H. W. Mei,
  Two New Unconstrained Optimization Algorithms Which Use Function and Gradient Values,
  Journal of Optimization Theory and Applications, 28(4):453-482, 1979.
- H. S. Dollar, N. I. M. Gould and D. P. Robinson,
  On solving trust-region and other regularised subproblems in optimization,
  Rutherford-Appleton Laboratory, Chilton, UK, Report RAL-TR-2009-003, 2009.
- S. M. Goldfeldt, R. E. Quandt and H. F. Trotter,
  Maximization by quadratic hill-climbing,
  Econometrica, 34:541-551, 1966.
- N. I. M. Gould, S. Lucidi, M. Roma and Ph. L. Toint,
  Solving the trust-region subproblem using the Lanczos method,
  SIAM Journal on Optimization, 9(2):504-525, 1999.
- K. Levenberg,
  A Method For The Solution Of Certain Problems In Least Squares,
  Quarterly Journal on Applied Mathematics, 2:164–168, 1944.
- D. Marquardt,
  An Algorithm For Least-Squares Estimation Of Nonlinear Parameters,
  SIAM Journal on Applied Mathematics, 11:431-441, 1963.
- J. J. Moré,
  The Levenberg-Marquardt algorithm: implementation and theory,
  Numerical Analysis, Dundee 1977 (A. Watson, ed.), Springer Verlag, Heidelberg, 1978.
- J. J. Moré,
  Recent developments in algorithms and software for trust region methods,
  in "Mathematical Programming: The State of the Art" (A. Bachem, M. Grötschel and B. Korte, eds.), Springer Verlag, Heidelberg, pp. 258-287, 1983.
- J. J. Moré and D. C. Sorensen,
  Computing A Trust Region Step,
  SIAM Journal on Scientific and Statistical Computing, 4(3):553-572, 1983.

Bibliography for lesson 2 (2)

- D. D. Morrison,
  Methods for nonlinear least squares problems and convergence proofs,
  Proceedings of the Seminar on Tracking Programs and Orbit Determination, (J. Lorell and F. Yagi, eds.), Jet Propulsion Laboratory, Pasadena, USA, pp. 1-9, 1960.
- M. J. D. Powell,
  A New Algorithm for Unconstrained Optimization,
  in "Nonlinear Programming" (J. B. Rosen, O. L. Mangasarian and K. Ritter, eds.), Academic Press, London, pp. 31-65, 1970.
- F. Rendl and H. Wolkowicz,
  A Semidefinite Framework for Trust Region Subproblems with Applications to Large Scale Minimization,
  Mathematical Programming, 77(2):273-299, 1997.
- M. Rojas, S. A. Santos and D. C. Sorensen,
  A new matrix-free algorithm for the large-scale trust-region subproblem,
  CAAM, Rice University, TR99-19, 1999.
- D. Winfield,
  Function and functional optimization by interpolation in data tables,
  Ph.D. Thesis, Harvard University, Cambridge, USA, 1969.

Lesson 3:

Derivative-free optimization, infinite dimensions and filters

3.1: Derivative-free optimization

An application of trust-regions: unconstrained DFO

Consider the unconstrained problem

$$\min_x f(x)$$

Gradient (and Hessian) of $f(x)$ unavailable

- physical measurement
- object code
- typically small-scale (but not always. . .)

$\Rightarrow$ "Derivative free optimization" (DFO)

$f(x)$ typically very costly

Exploit each evaluation of $f(x)$ to the utmost possible

considerable interest of practitioners

Interpolation methods for DFO

Idea: Winfield (1973), Powell (1994)

Until "convergence":

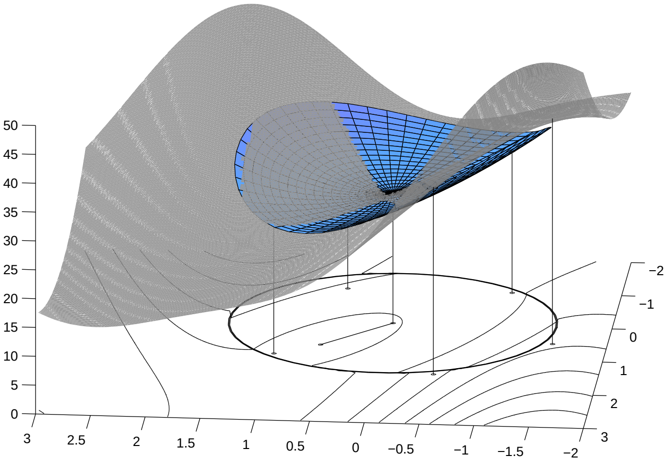
- Use the available function values to build a polynomial interpolation model m_k:

$$m_k(y_i) = f(y_i) \quad y_i \in Y;$$

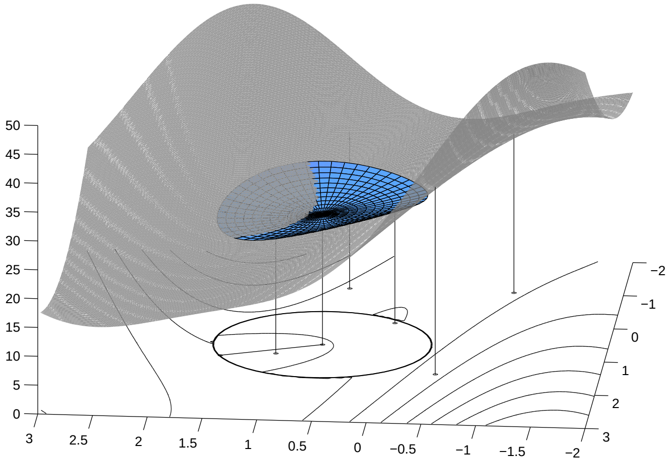
- Minimize the model in a "trust region", yielding a new potentially good point;
- Compute a new function value.

Y = interpolation set $\subseteq$ { points y_i at which $f(y_i)$ is known }

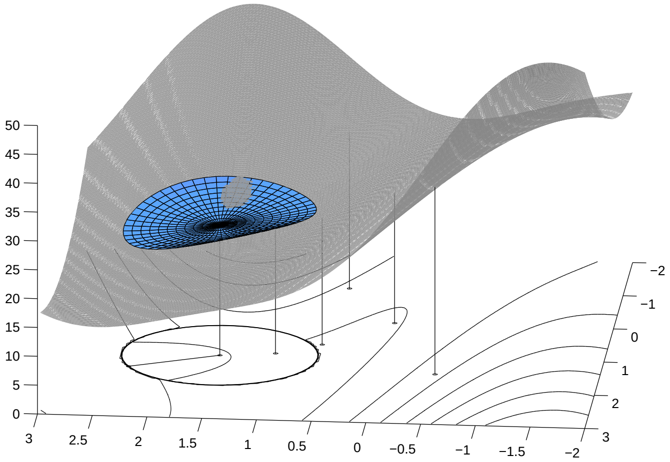
A naive trust-region method for DFO: illustration



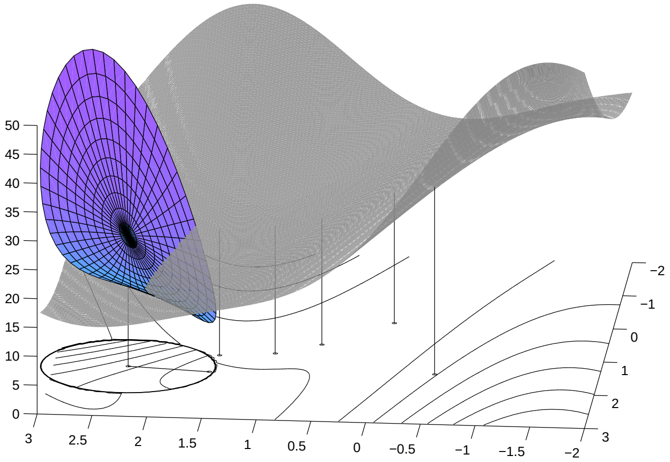
A naive trust-region method for DFO: illustration



A naive trust-region method for DFO: illustration



A naive trust-region method for DFO: illustration



Interpolation methods for DFO (2)

To be considered:

- poisedness of the interpolation set Y
- choice of models (linear, quadratic, in between, beyond)
- convergence theory
- numerical performance

Poisedness

Assume a quadratic model

$$m_k(x_k + s) = f_k + \langle g_k, s\rangle + \tfrac{1}{2}\langle s, H_k s\rangle$$

Thus

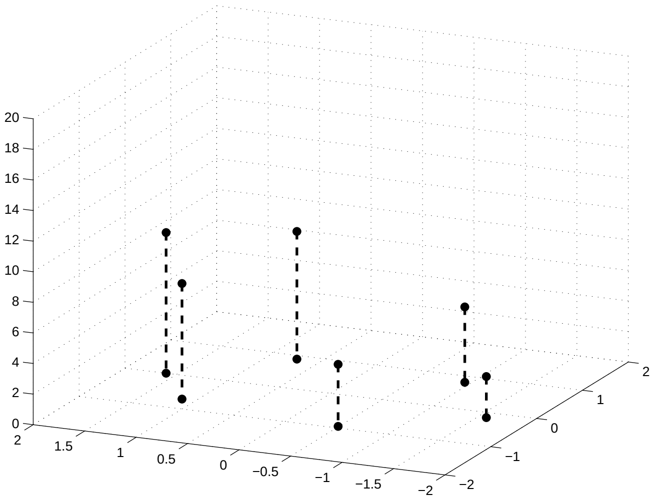
$$p = 1 + n + \tfrac{1}{2}n(n+1) = \tfrac{1}{2}(n+1)(n+2)$$

parameters to determine $\Rightarrow$ need p function values ($|Y| = p$)

Not sufficient!

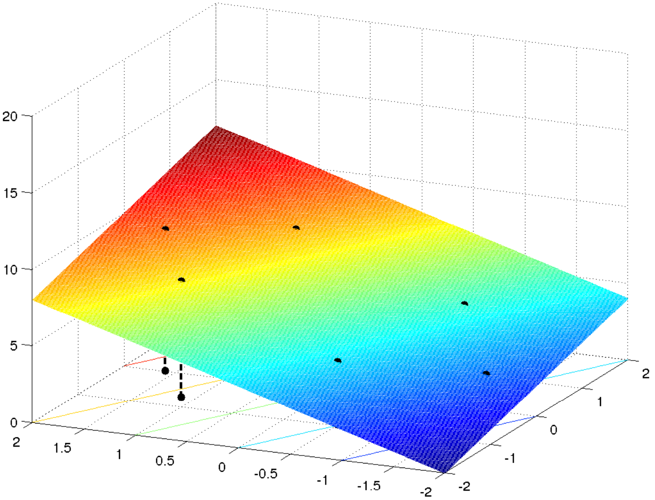
$\Rightarrow$ need geometric conditions for the points in Y ...

Poisedness: geometry with $n = 2$, $p = 6$



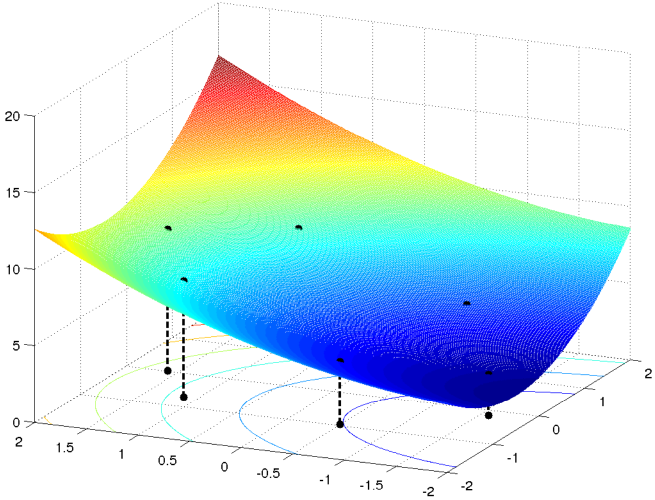
With these 6 data points in $\mathbb{R}^3$......

Poisedness: geometry with $n = 2$, $p = 6$



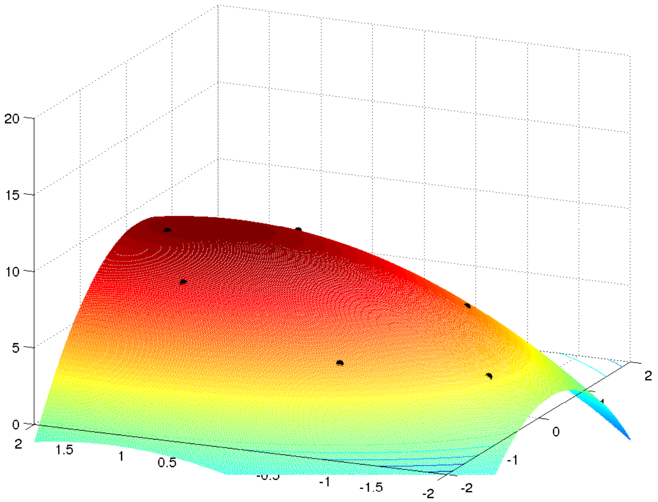
. . . is this the correct interpolation?

Poisedness: geometry with $n = 2$, $p = 6$



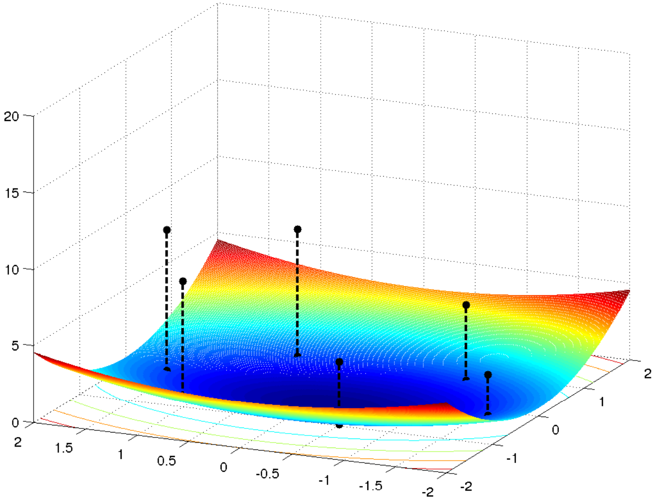
...or this?

Poisedness: geometry with $n = 2$, $p = 6$



. . . or this?

Poisedness: geometry with $n = 2$, $p = 6$



The difference ... is zero on a quadratic curve containing Y!

Poisedness: geometry (2)

If $\{\phi_i(\cdot)\}_{i=1}^p$ = basis for quadratic polynomials

$$\sum_{i=1}^{p} \alpha_i \phi_i(y_j) = f(y_j) \quad j = 1, \ldots, p$$

Possible poisedness measure:

$$\delta(Y) = \det \begin{pmatrix} \phi_1(y_1) & \cdots & \phi_p(y_1) \\ \vdots & & \vdots \\ \phi_1(y_p) & \cdots & \phi_p(y_p) \end{pmatrix}$$

Y (well) poised $\Leftrightarrow |\delta(Y)| \geq \epsilon$

- scale for the spread of the y_i's
- notion of geometry improvement

Lagrange polynomials

Remarkable: replace y_- by y_+ in Y:

$$\frac{\delta(Y_+)}{\delta(Y)} = L(y_+, y_-) \text{ is independent of the basis } \{\phi_i(\cdot)\}_{i=1}^p$$

where

$$\forall y \in Y \qquad L(y, y_-) = \begin{cases} 1 & \text{if } y = y_- \\ 0 & \text{if } y \neq y_- \end{cases}$$

is the Lagrange fundamental polynomial

Note: for quadratic interpolation, $L(\cdot, y)$ is a quadratic polynomial!

Powell (1994)

Interpolation using Lagrange polynomials

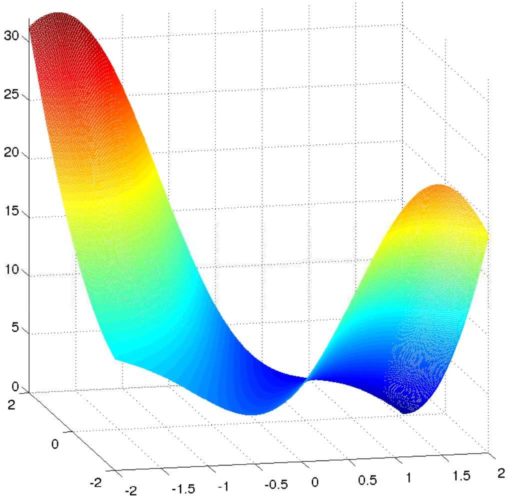
Idea: use the Lagrange polynomials to define the (quadratic) interpolant by

$$m_k(x_k+s) = \sum_{y \in Y_k} f(y) L_k(x_k+s, y)$$

And then...

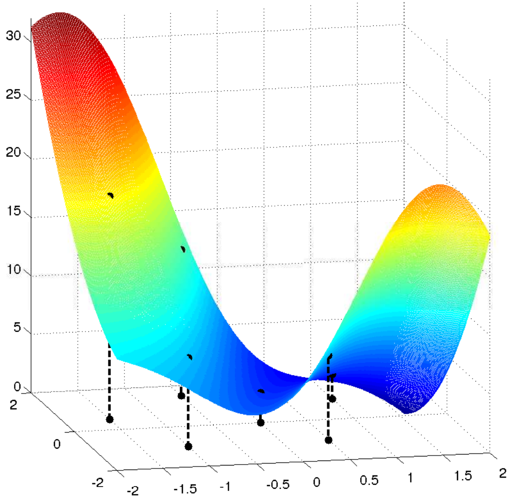
$$\|f(x_k+s) - m_k(x_k+s)\| \leq \kappa \sum_{y \in Y_k} \|x_k+s-y\|^2 |L_k(x_k+s, y)|$$

Interpolation using Lagrange polynomials: construction



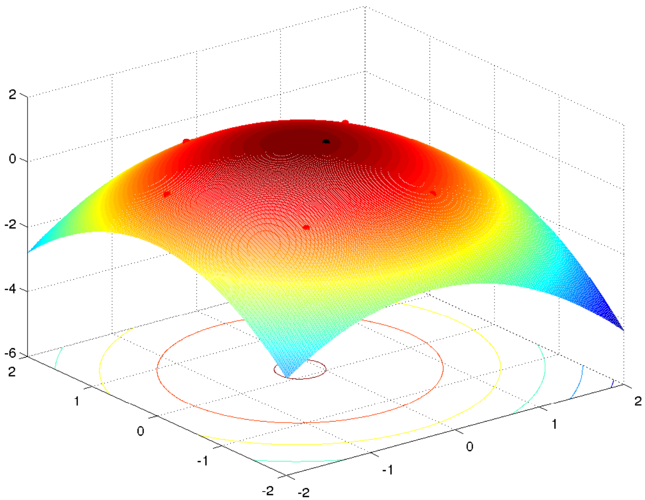
The original function. . .

Interpolation using Lagrange polynomials: construction



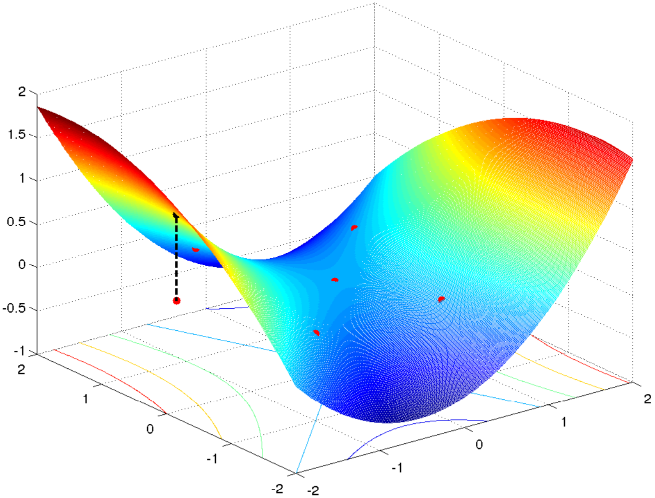
. . . and the interpolation set

Interpolation using Lagrange polynomials: construction



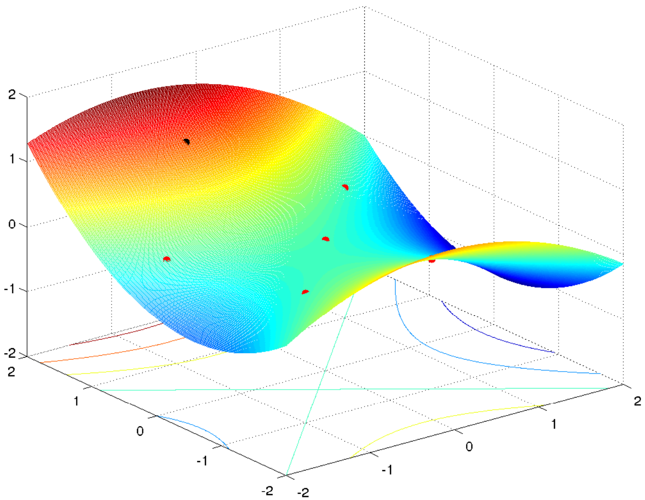
The first Lagrange polynomial

Interpolation using Lagrange polynomials: construction



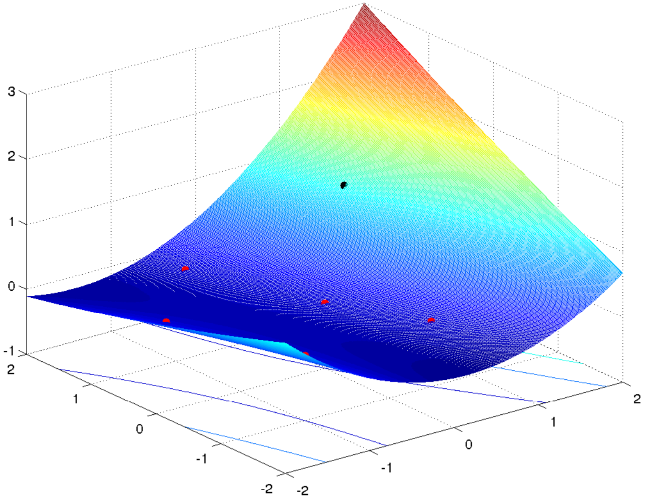
The second Lagrange polynomial

Interpolation using Lagrange polynomials: construction



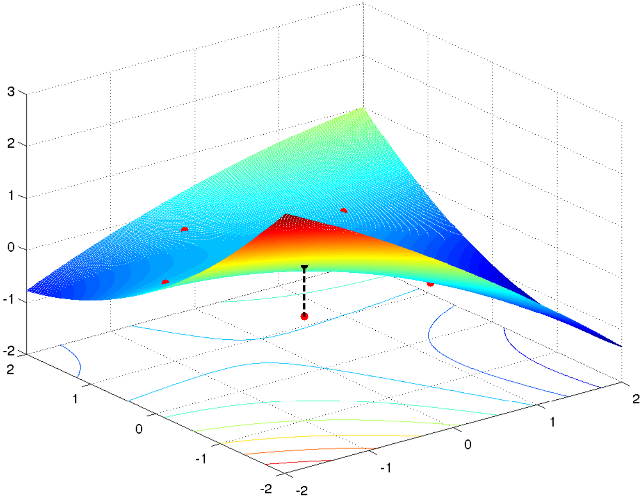
The third Lagrange polynomial

Interpolation using Lagrange polynomials: construction



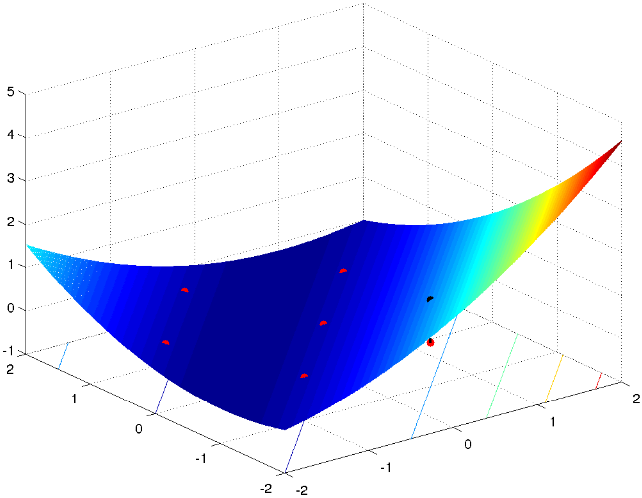
The fourth Lagrange polynomial

Interpolation using Lagrange polynomials: construction



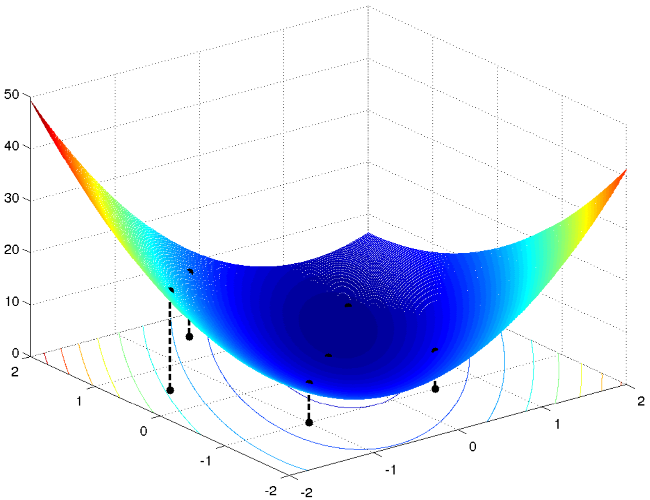
The fifth Lagrange polynomial

Interpolation using Lagrange polynomials: construction



The sixth Lagrange polynomial

Interpolation using Lagrange polynomials: construction



The final interpolating quadratic

Other algorithmic ingredients

- include a new point in the interpolation set
  - need to drop an existing interpolation point?
  - select which one to drop: make Y “as poised as possible”

Note: model/function minimizer may produce bad geometry!!

$\Rightarrow$ geometry improvement procedure . . .

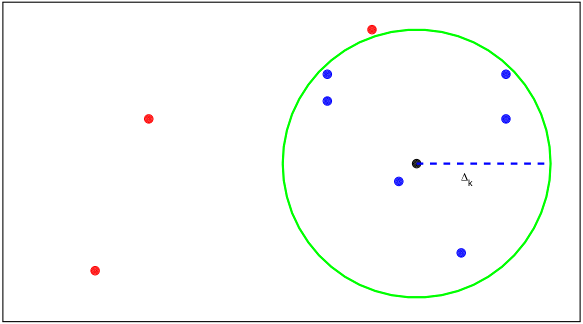
- trust-region radius management

$$\text{trust region} \quad = \mathcal{B}_k = \{x_k + s \mid \|s\| \leq \Delta_k\}$$

  - standard: reduce Δ_k when “no progress”
  - DFO: more complicated! (Could reduce Δ to fast and prevent convergence. . .)

$\Rightarrow$ verify that Y is poised before reducing Δ_k

Improving the geometry in a ball



- attempt to reuse past points that are close to x_k
- attempt to replace a distant point of Y
- attempt to replace a close point of Y

good geometry for the current Δ_k $\Leftrightarrow$ improvement impossible

Self-correction at unsuccessful iterations (1)

At iteration k, define the set of exchangeable far points:

$$\mathcal{F}_k = \{y \in Y_k \mid \|y - x_k\| > \Delta_k \text{ and } L_k(x_k + s_k, y) \neq 0\}$$

and the set of exchangeable close points (for some $\pi > 1$):

$$\mathcal{C}_k = \{y \in Y_k \setminus \{x_k\} \mid \|y - x_k\| \leq \Delta_k \text{ and } |L_k(x_k + s_k, y)| \geq \pi\}$$

Self-correction at unsuccessful iterations (2)

Remarkably,

Whenever

- iteration k is unsuccessful,
- $\mathcal{F}_k = \emptyset$
- Δ_k is small w.r.t. $\|g_k\|$,

then $\mathcal{C}_k \neq \emptyset$.

(an improvement of the geometry by a factor π is always possible at unsuccessful iterations when Δ_k is small and all exchangeable far points have been considered)

$\Rightarrow$ no need to reduce Δ_k forever!

Trust-region algorithm for DFO (1)

Algorithm 3.1: TR for DFO

Step 0: Initialization. Given: x_0, Δ_0, Y_0 ($\rightarrow L_0(\cdot, y)$). Set $k = 0$.

Step 1: Criticality test [complicated and not discussed here]

Step 2: Solve the subproblem. Compute s_k that sufficiently reduces $m_k(x_k + s)$ within the trust region,

Step 3: Evaluation. Compute $f(x_k + s_k)$ and

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

Step 4: Define the next iterate and interpolation set.

the big question

Step 5: Update the Lagrange polynomials.

Trust-region algorithm for DFO (2)

Algorithm 3.2: Step 4: Define x_{k+1} and Y_{k+1}

Step 4a: Successful iteration. If $\rho_k \geq \eta_1$, accept $x_k + s_k$, increase Δ_k and exchange $x_k + s_k$ with

$$y = \arg\max_{y \in Y_k} \|y - (x_k + s_k)\|^2 |L_k(x_k + s_k, y)|$$

Step 4b: Replace far point. If $\rho_k < \eta_1$ (+ other technical condition) and $\mathcal{F}_k \neq \emptyset$, reject $x_k + s_k$, keep Δ_k and exchange $x_k + s_k$ with

$$y = \arg\max_{y \in \mathcal{F}_k} \|y - (x_k + s_k)\|^2 |L_k(x_k + s_k, y)|$$

Step 4c: Replace close point. If $\rho_k < \eta_1$ (+ other technical condition) and $\mathcal{C}_k \neq \emptyset$, reject $x_k + s_k$, keep Δ_k and exchange $x_k + s_k$ with

$$y = \arg\max_{y \in \mathcal{C}_k} \|y - (x_k + s_k)\|^2 |L_k(x_k + s_k, y)|$$

Step 4d: Decrease the radius. Otherwise, reject $x_k + s_k$, keep Y_k, and reduce Δ_k.

Global convergence results

If the model is at least fully linear, then

$$\liminf_{k\to\infty} \|\nabla_x f(x_k)\| = \liminf_{k\to\infty} \|g_k\| = 0$$

Scheinberg and T. (2009)

With more costly algorithm:

If the model is at least fully linear, then

$$\lim_{k\to\infty} \|\nabla_x f(x_k)\| = \lim_{k\to\infty} \|g_k\| = 0$$

If the model at least fully quadratic, then iterates converge to 2nd-order critical points

For an efficient numerical method...

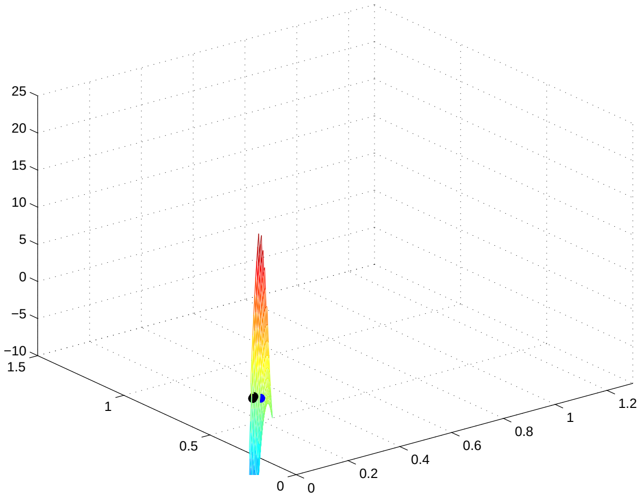
Many more issues:

- which Hessian approximation?
  (full/vs diagonal or structured)
- details of criticality tests difficult
- details for numerically handling interpolation polynomials
  (Lagrange, Newton),
- reference shifts,
- ...

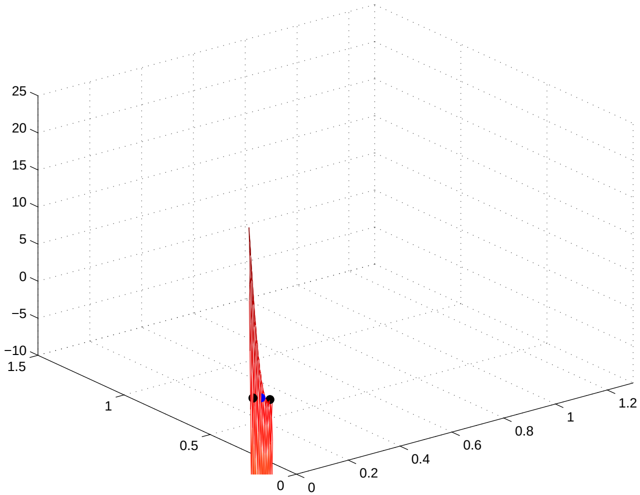
good codes around: NEWUOA, DFO $\Rightarrow$ efficient solvers

Powell (2008 and previously), Conn, Scheinberg and T. (1998)
Conn, Scheinberg and Vicente (2008)

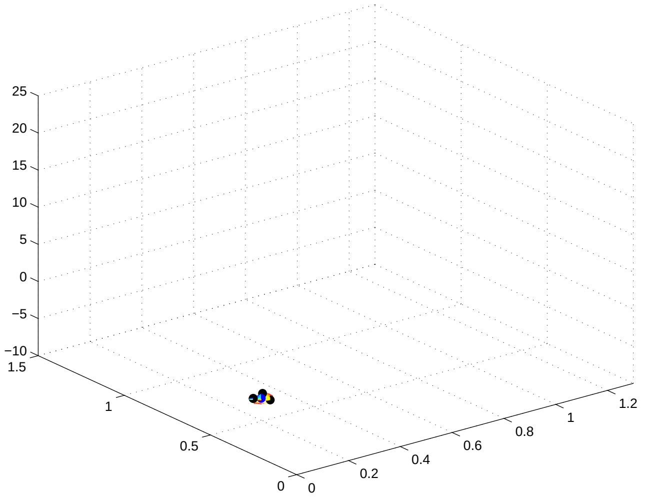
On the ever famous banana function. . .



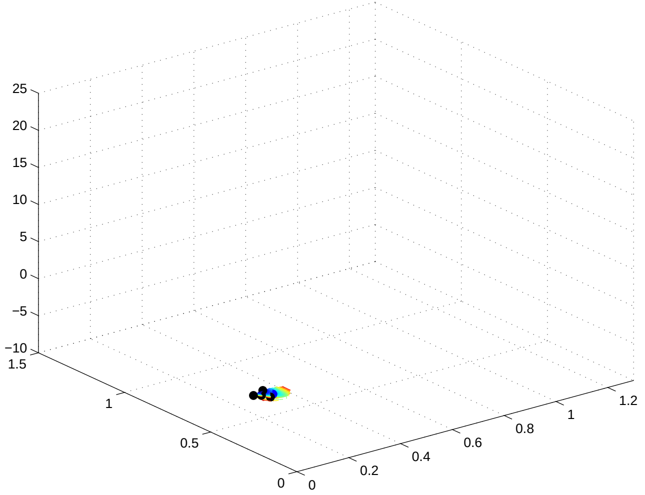
On the ever famous banana function. . .



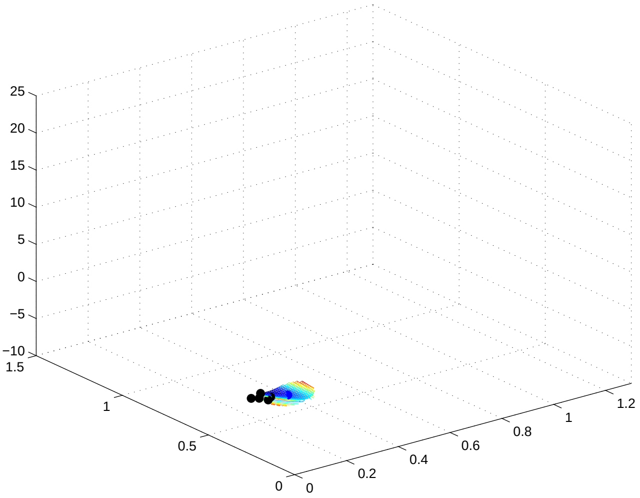
On the ever famous banana function. . .



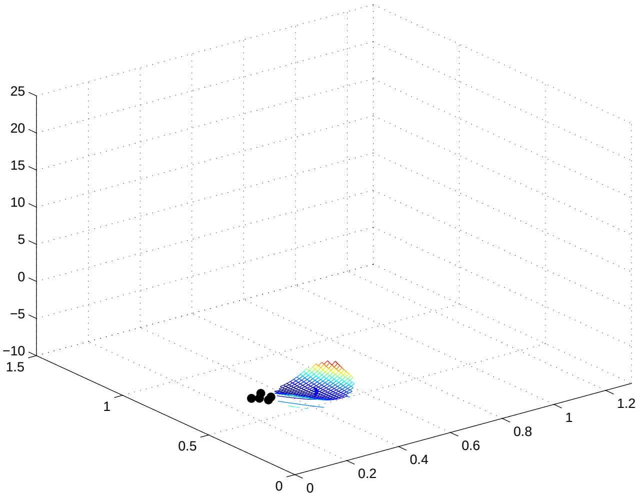
On the ever famous banana function. . .



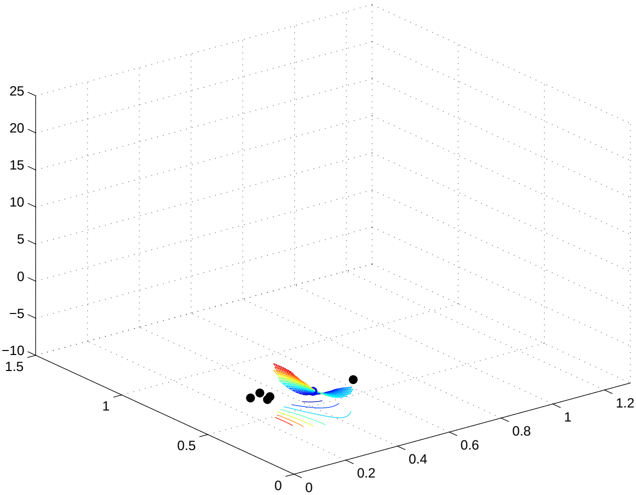
On the ever famous banana function. . .



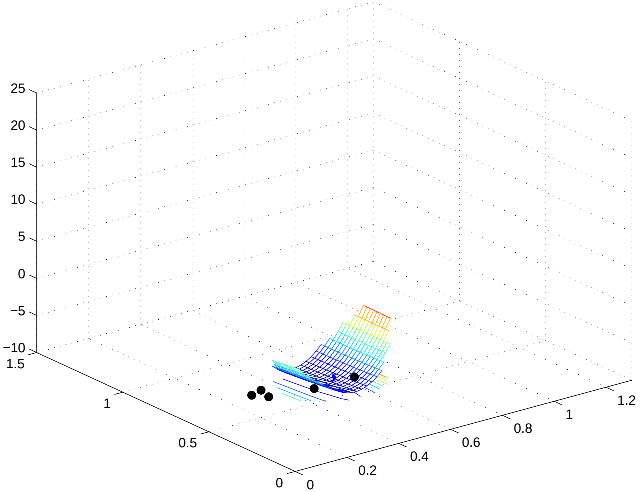
On the ever famous banana function. . .



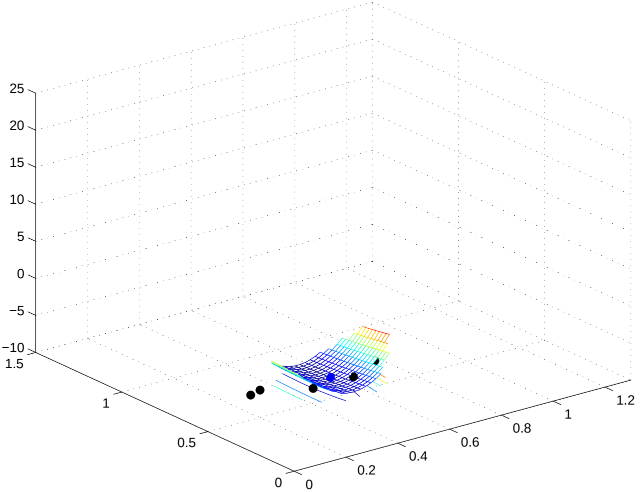
On the ever famous banana function. . .



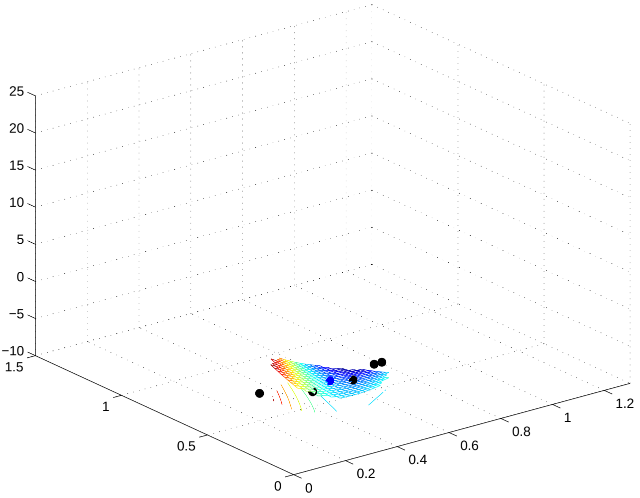
On the ever famous banana function. . .



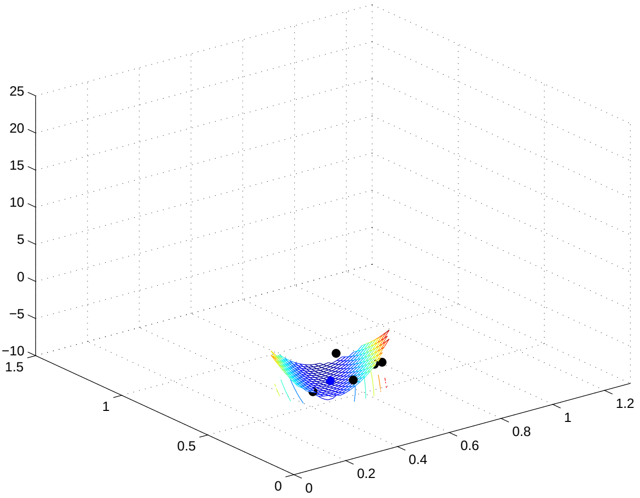
On the ever famous banana function...



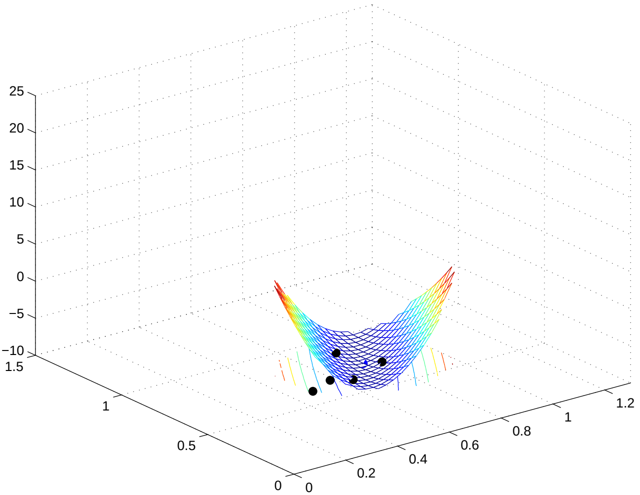
On the ever famous banana function. . .



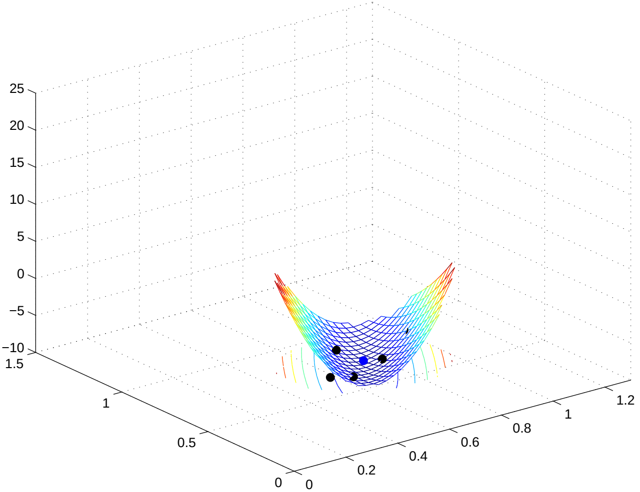
On the ever famous banana function. . .



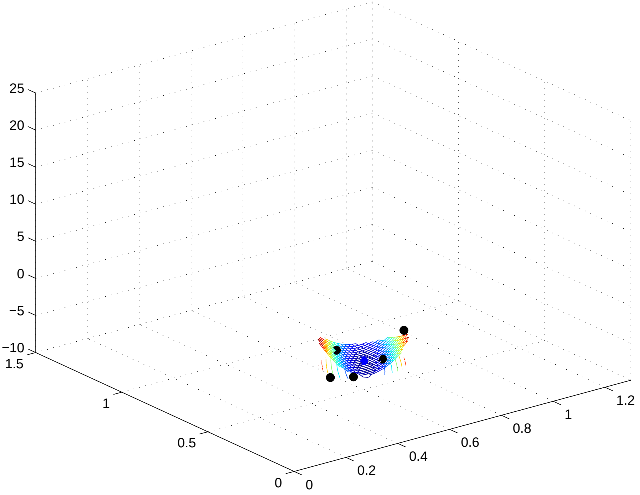
On the ever famous banana function. . .



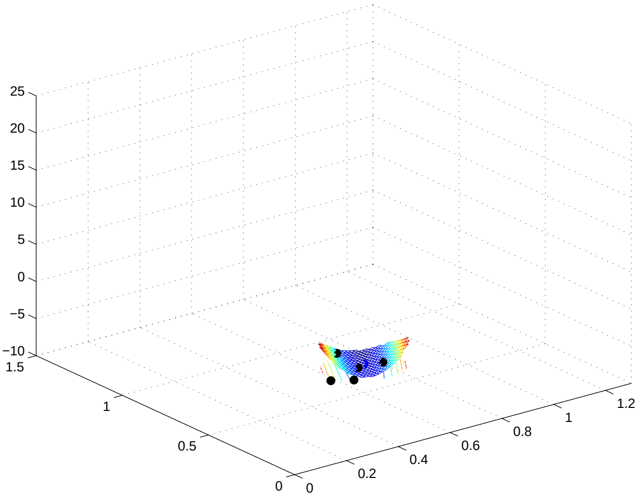
On the ever famous banana function. . .



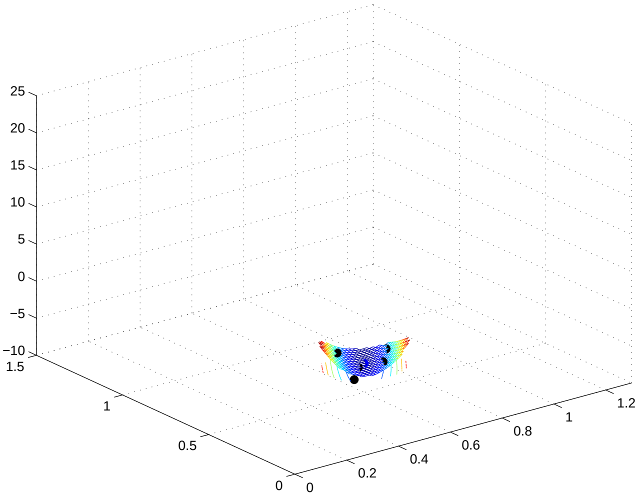
On the ever famous banana function. . .



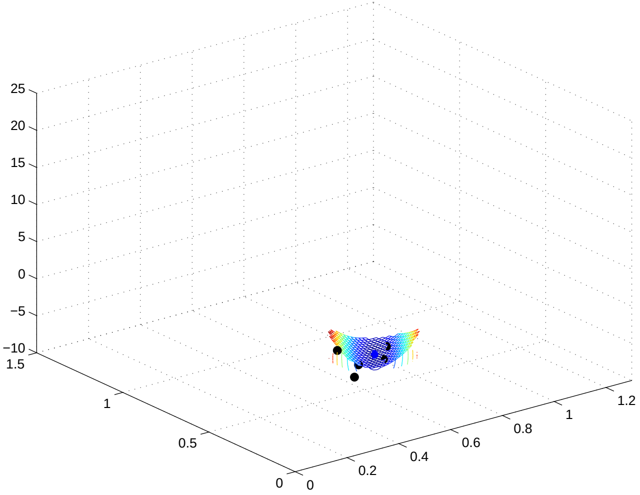
On the ever famous banana function. . .



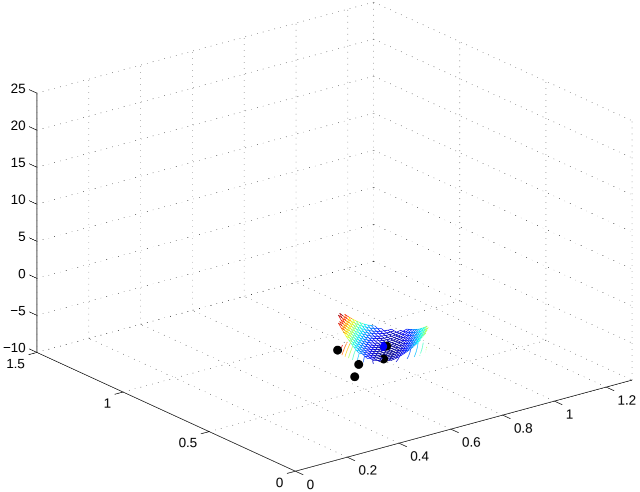
On the ever famous banana function. . .



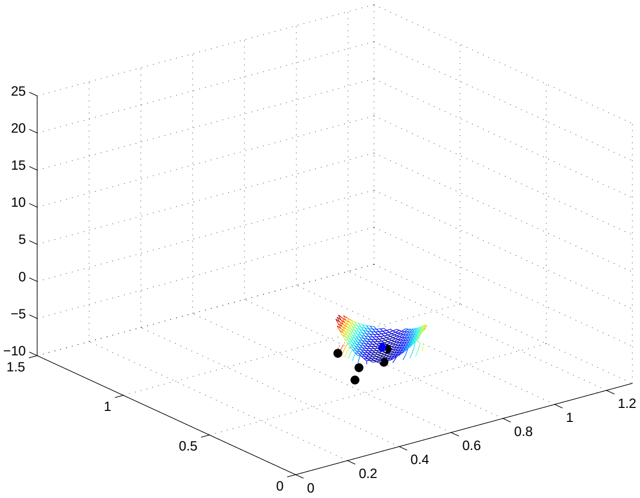
On the ever famous banana function. . .



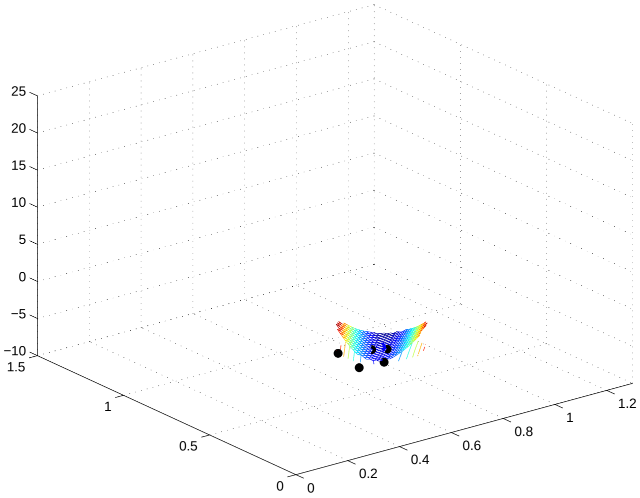
On the ever famous banana function. . .



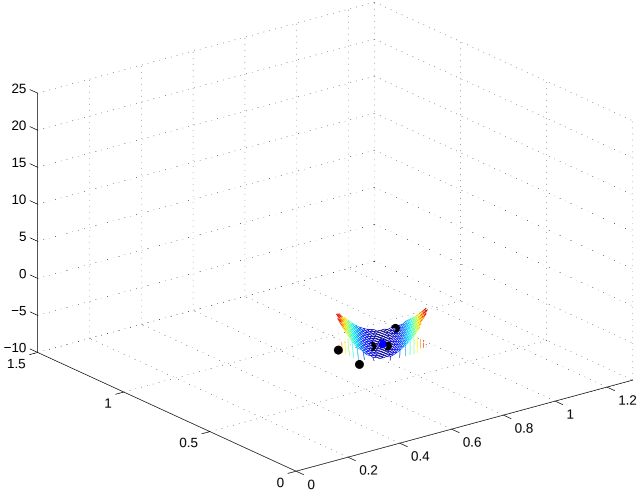
On the ever famous banana function. . .



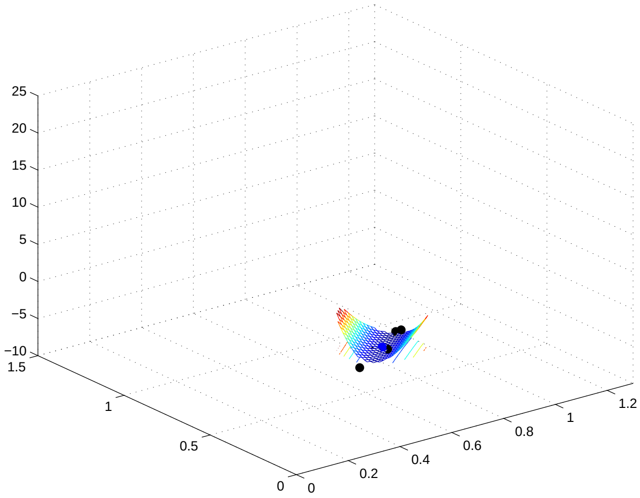
On the ever famous banana function. . .



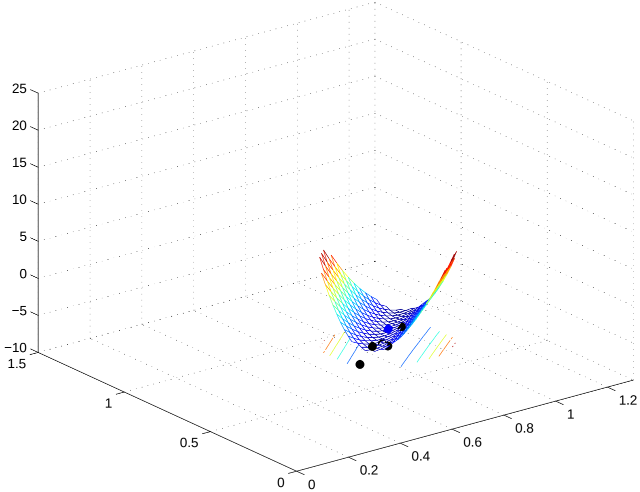
On the ever famous banana function. . .



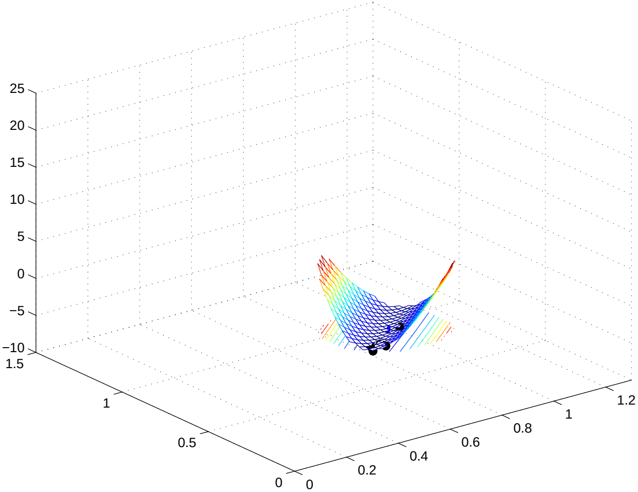
On the ever famous banana function. . .



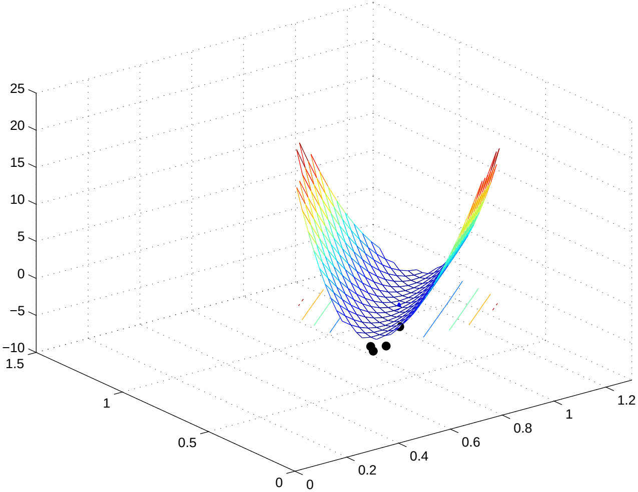
On the ever famous banana function. . .



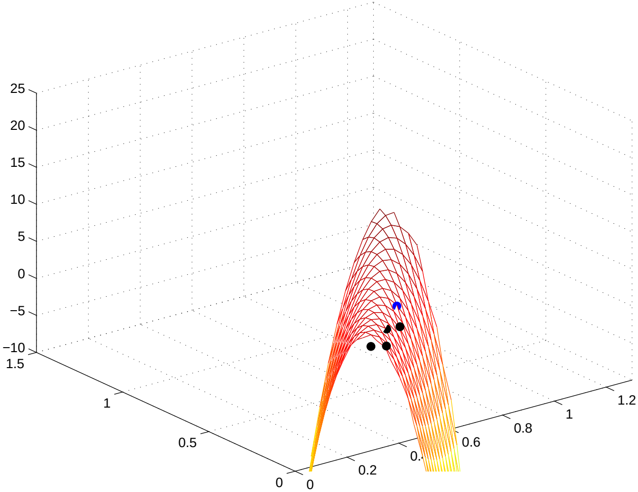
On the ever famous banana function. . .



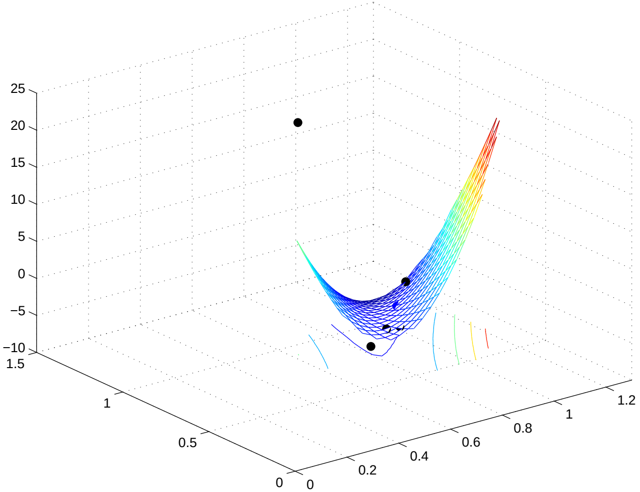
On the ever famous banana function. . .



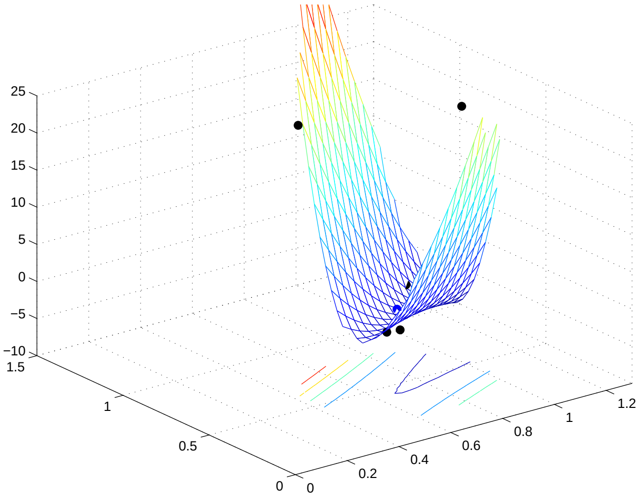
On the ever famous banana function. . .



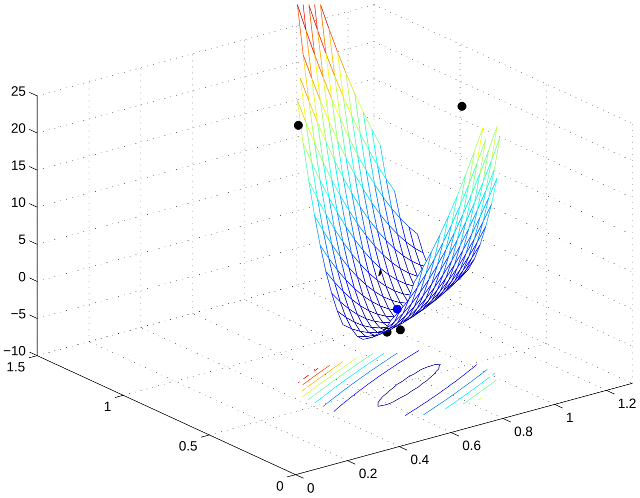
On the ever famous banana function. . .



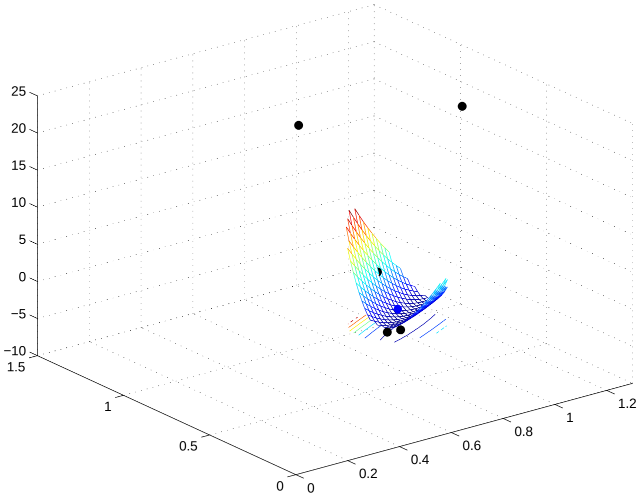
On the ever famous banana function. . .



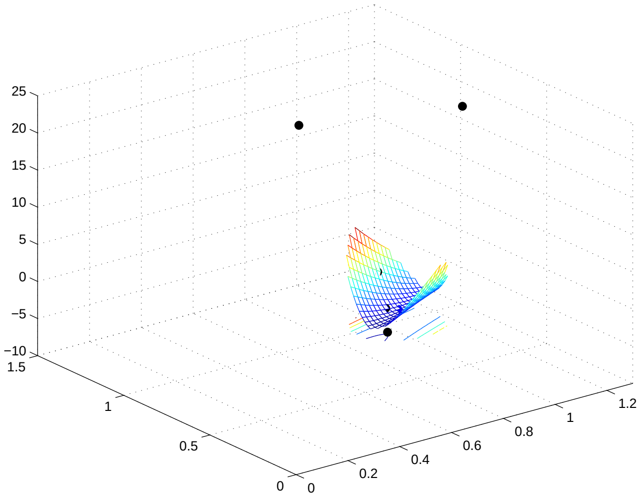
On the ever famous banana function. . .



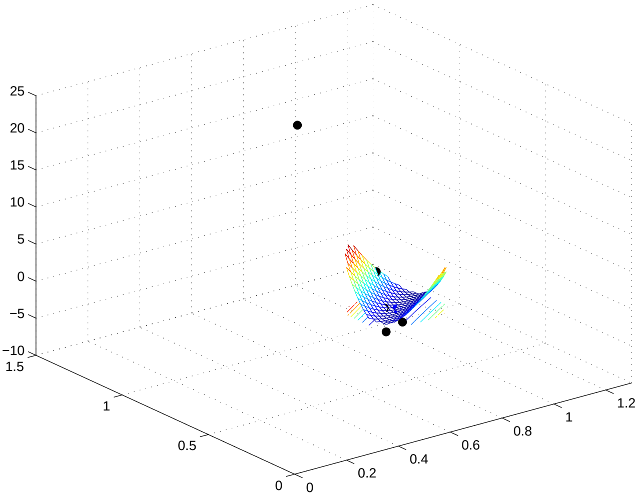
On the ever famous banana function. . .



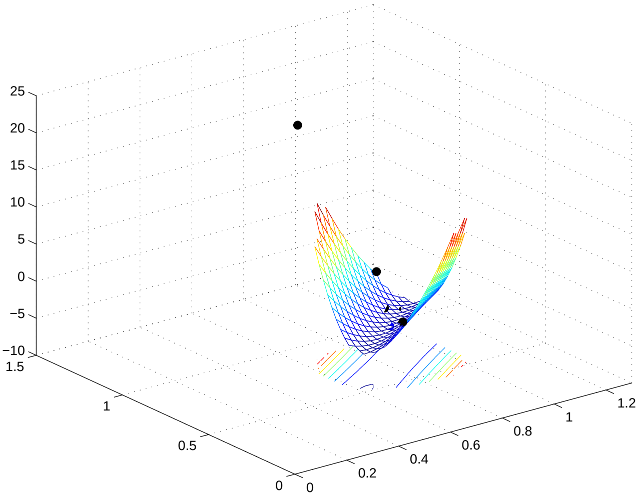
On the ever famous banana function. . .



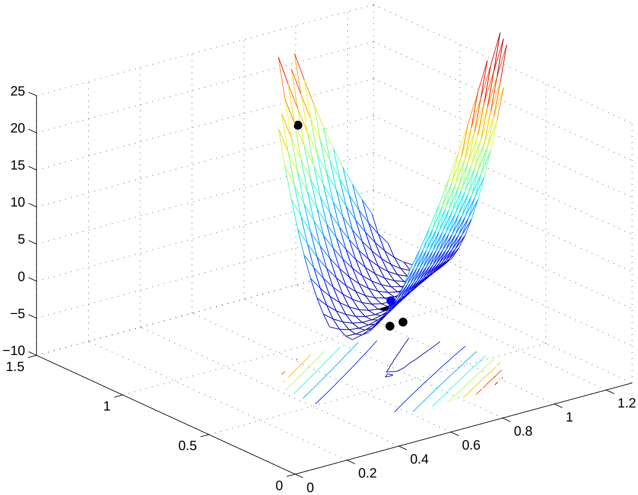
On the ever famous banana function. . .



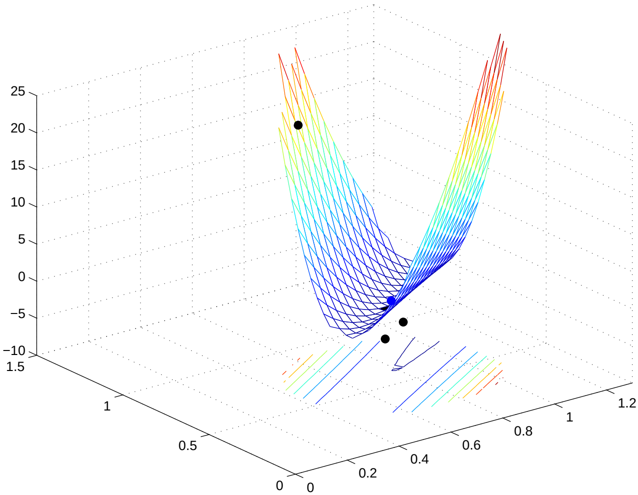
On the ever famous banana function. . .



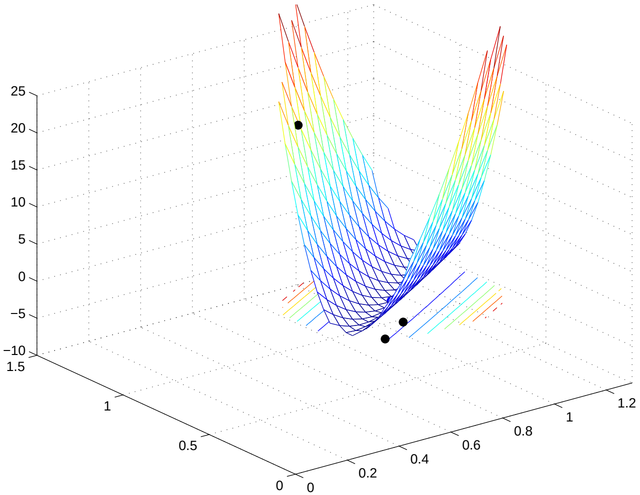
On the ever famous banana function. . .



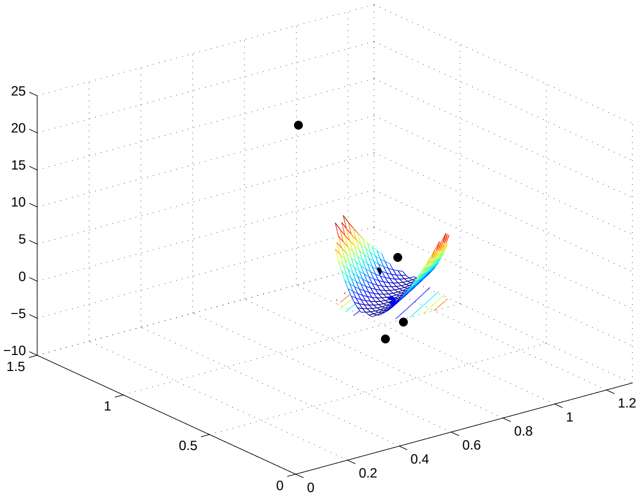
On the ever famous banana function. . .



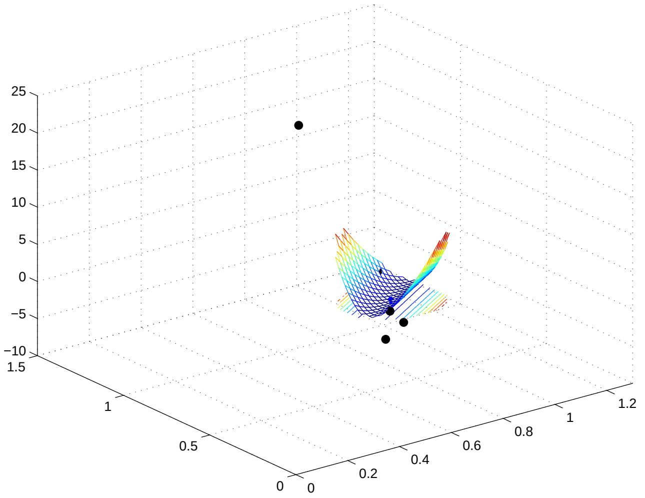
On the ever famous banana function. . .



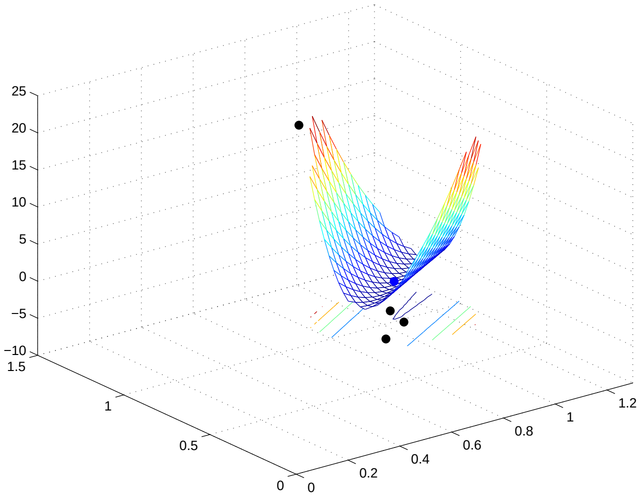
On the ever famous banana function. . .



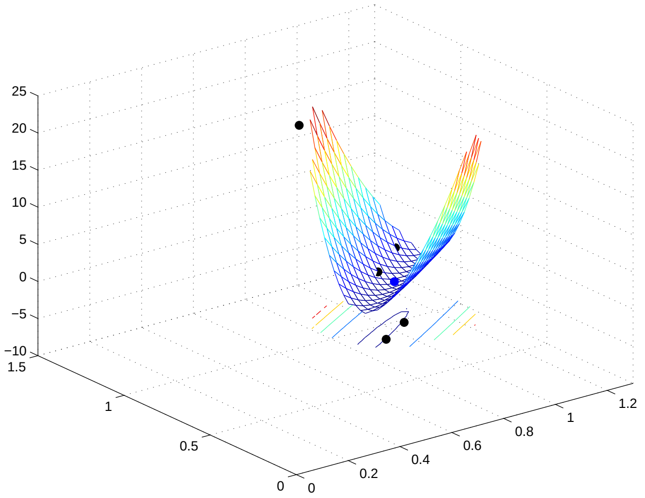
On the ever famous banana function. . .



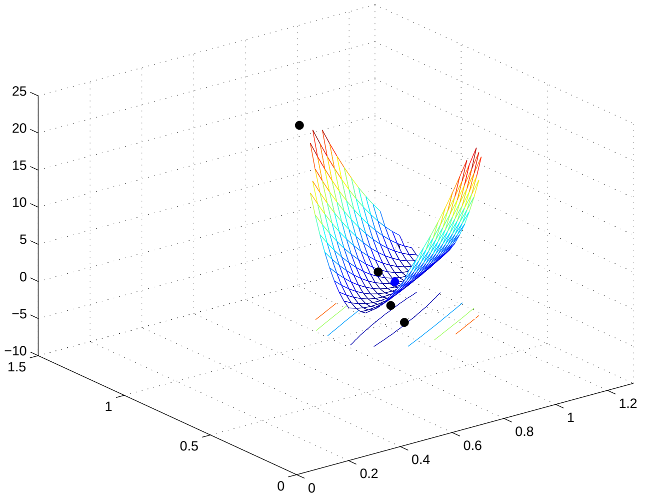
On the ever famous banana function...



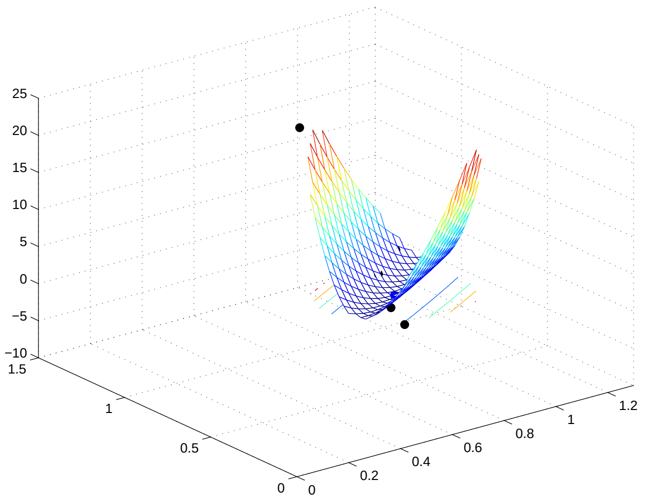
On the ever famous banana function. . .



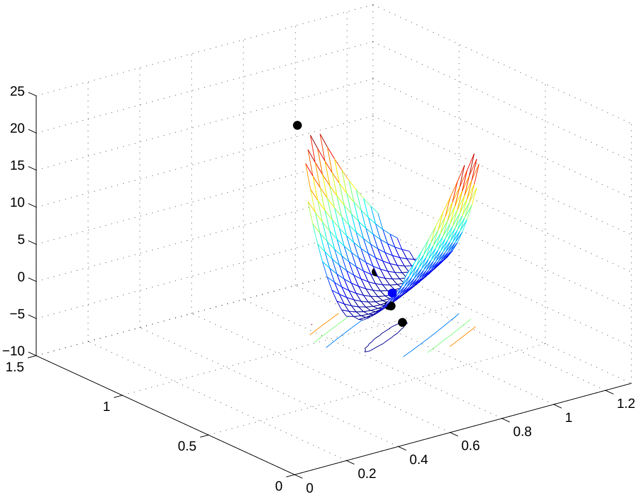
On the ever famous banana function. . .



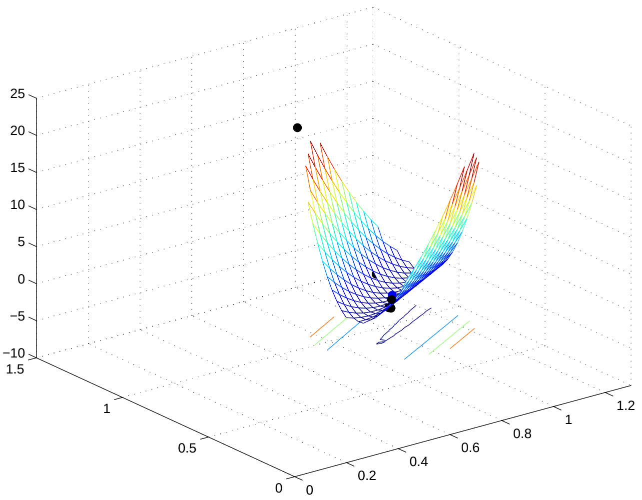
On the ever famous banana function. . .



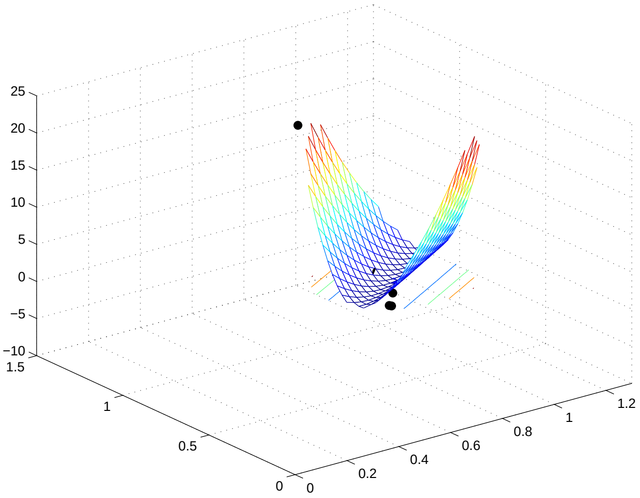
On the ever famous banana function. . .



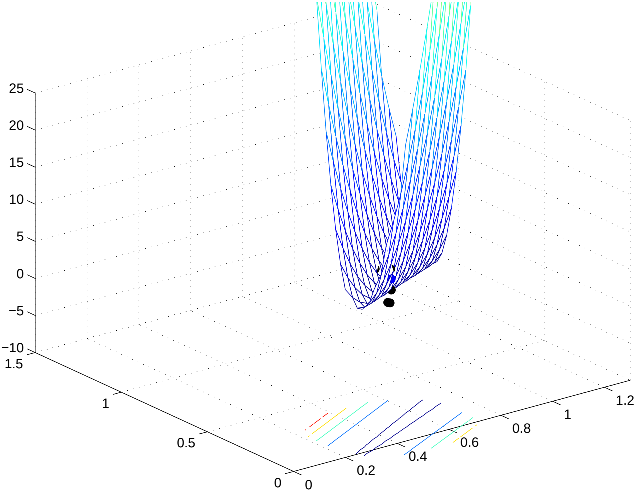
On the ever famous banana function. . .



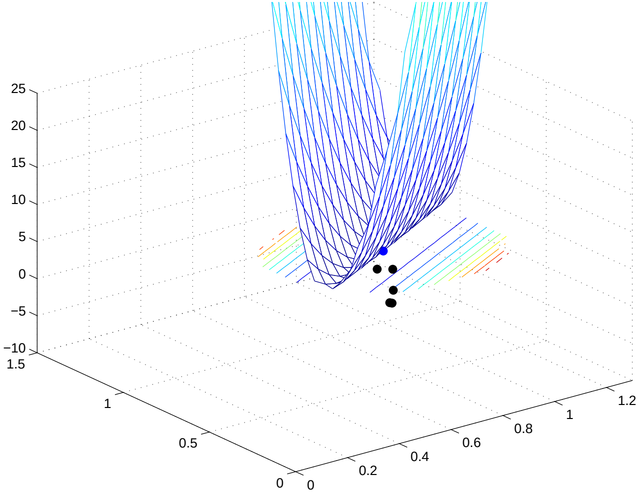
On the ever famous banana function...



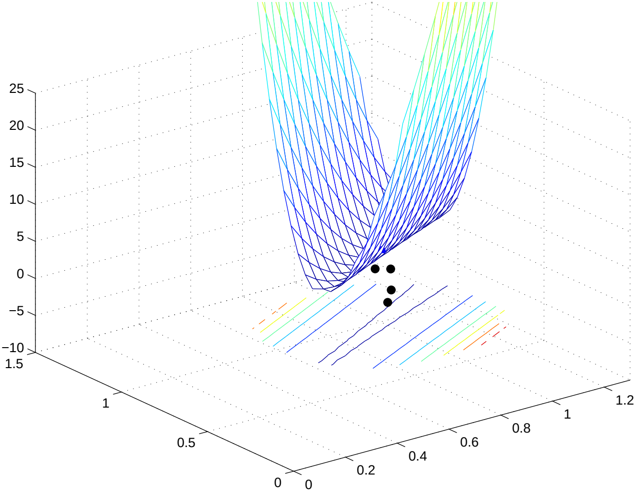
On the ever famous banana function. . .



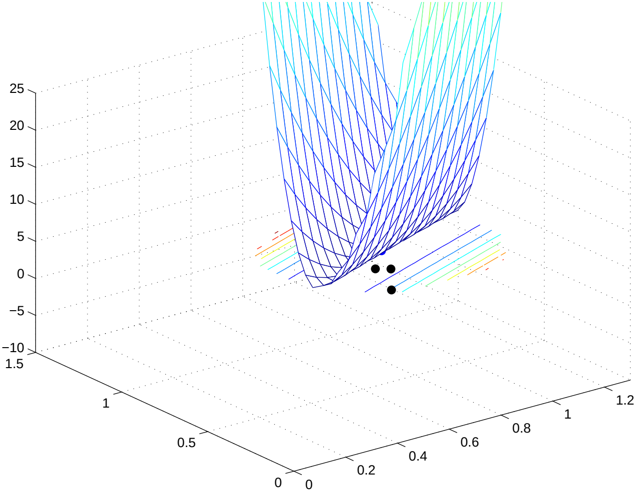
On the ever famous banana function. . .



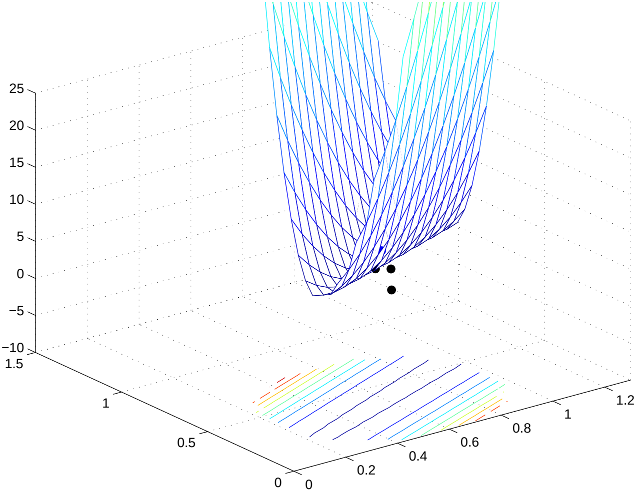
On the ever famous banana function. . .



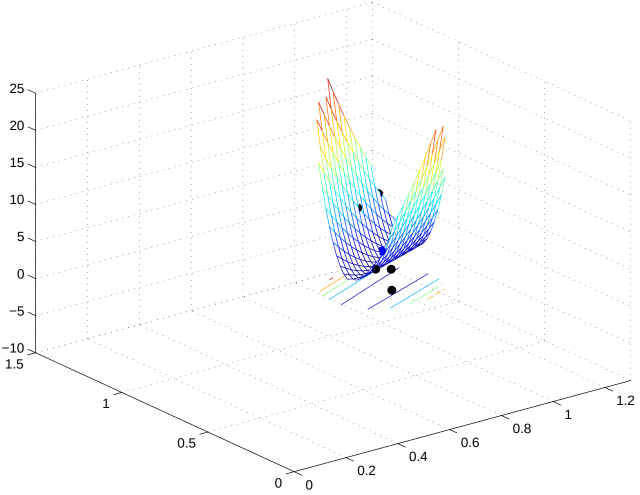
On the ever famous banana function. . .



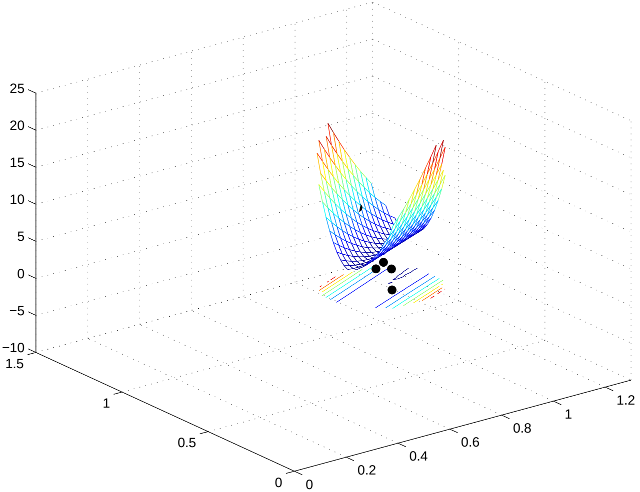
On the ever famous banana function. . .



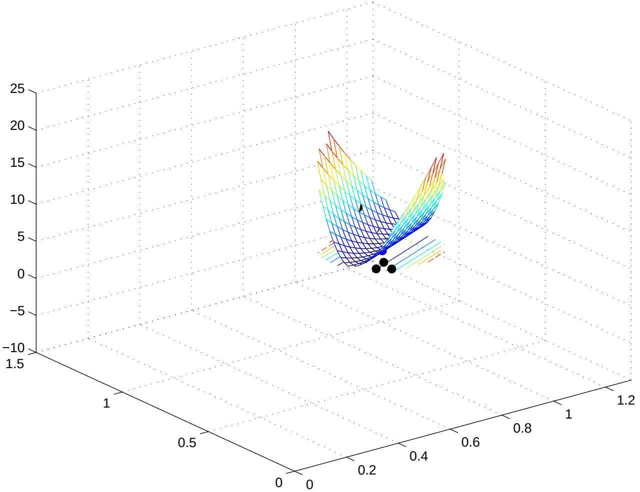
On the ever famous banana function. . .



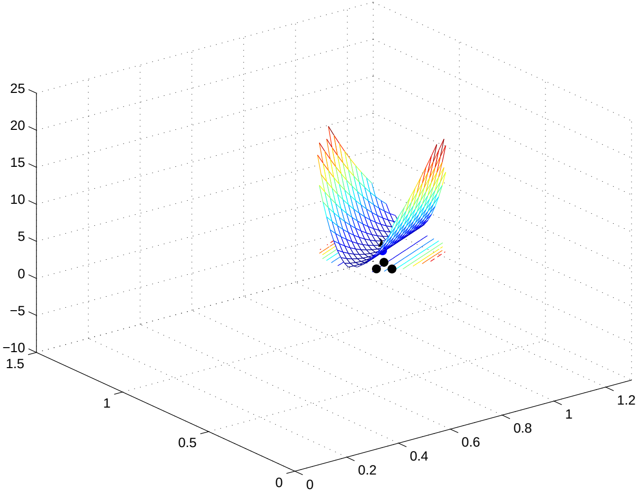
On the ever famous banana function. . .



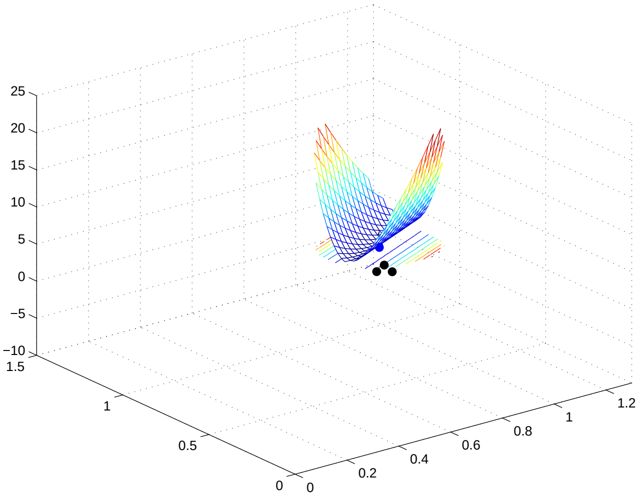
On the ever famous banana function. . .



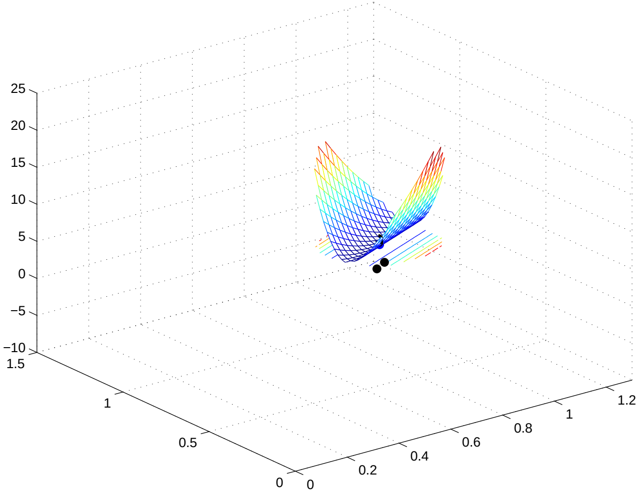
On the ever famous banana function. . .



On the ever famous banana function. . .



On the ever famous banana function. . .



3.2: Infinite dimensional problems

Why consider infinite dimensions?

Main motivation:

- large-scale finite dimensional problems often result from discretized continuous ones (surfaces, time-trajectories, optimal control, . . .)
- behaviour on these problems dominated by infinite dimensional properties

Need to investigate infinite dimensions to ensure consistency!

Two main cases: Hilbert and Banach spaces.

Convergence in Hilbert spaces

The trust-region algorithm is well-defined and globally convergent in Hilbert spaces.

- Riescz representation theorem $\Rightarrow \mathcal{V}' \approx \mathcal{V}$
- Cauchy point results from one dimensional minimization (but $x_k^{\rm M}$ may not exist!)
- $$\beta_k \stackrel{\rm def}{=} 1 + \sup_{x \in \mathcal{B}_k} \|\nabla_{xx} m_k(x)\|_{\mathcal{V},\mathcal{V}'},$$
- $$\lambda_{\min}[H] \stackrel{\rm def}{=} \inf_{d \in \mathcal{V}, d \neq 0} \frac{\langle d, Hd \rangle}{\langle d, d \rangle}$$

Wht happens in Banach spaces ?

Problem: dual space different from the primal!

Need further assumptions:

- $\nabla_x f(x) \in \mathcal{V}$ for all $x \in \mathcal{V}$.
- $\nabla_x f$ is uniformly continuous from $\mathcal{V}$ to $\mathcal{V}$.
- For every $x \in \{x \in \mathcal{V} \mid f(x) \leq f(x_0)\}$,

$$\langle \nabla_x f(x), \nabla_x f(x) \rangle \geq \phi(\|\nabla_x f(x)\|_{\mathcal{V}'}) \|\nabla_x f(x)\|_{\mathcal{V}},$$

  for some continuous monotonically increasing real ϕ from $[0, \infty]$ to itself, independent of x and such that $\phi(0) = 0$ and $\phi(t) > 0$ for $t > 0$.

Convergence in Banach spaces, nevertheless

The last assumption implies

$$\langle -g_k, g_k \rangle \le -\phi(\|g_k\|_{\mathcal{V}'})\|g_k\|_{\mathcal{V}}$$

. . . and sufficient decrease follows!

Is this realistic?

The additional assumptions always hold for $\mathcal{V} = L^p(\Omega)$ and $2 \le p < \infty$, when $\|g\|_{L^p(\Omega)} \le \kappa_{\rm ubg}$.

Under these additional assumptions, the trust-region algorithm is well-defined and globally convergent in Banach spaces.

3.3: Filter algorithms

Monotonicity (1)

Global convergence theoretically ensured by

- some global measure. . .
  - unconstrained : $f(x_k)$
  - (constrained : some merit function at x_k)
- . . . with strong monotonic behaviour (Lyapunov function)

Also practically enforced by

- algorithmic safeguards around Newton method (linesearches, trust regions)

Monotonicity (2)

But, unfortunately,

classical safeguards limit efficiency!

Of interest: design less obstructive safeguards while

- ensuring better numerical performance (the Newton Liberation Front!)
- continuing to guarantee global convergence properties

Is this possible?

Typically:

- abandon strict monotonicity of usual measures
- but insist on average behaviour instead

Non-monotone trust-regions

Idea: $f(x_{k+1}) < f(x_k)$ replaced by $f(x_{k+1}) < f_{r(k)}$

with

$$f_{r(k)} < f_{r(k-1)}$$

Further issues:

- suitably define the "reference iteration" $r(k)$
- adapt the trust-region algorithm: also compare achieved and predicted reductions since reference iteration

  T. (1997)

Non-monotone TR algorithm

Algorithm 3.3: Non monotone TR algorithm (NMTR)

Step 0: Initialization. Given: x_0, Δ_0, η_1, η_2, γ_1, γ_2. Compute $f(x_0)$, set $k = 0$.

Step 1: Model definition. Choose $\|\cdot\|_k$ and define m_k in $\mathcal{B}_k$.

Step 2: Step calculation. Compute s_k that sufficiently reduces m_k and $x_k + s_k \in \mathcal{B}_k$.

Step 3: Acceptance of the trial point. Define the reference iteration $r(k) \leq k$ and compute $f(x_k + s_k)$,

$$\sigma_k^{\rm h} = \sum_{\substack{i=r(k)\\ i\in\mathcal{S}}}^{k-1} [m_i(x_i) - m_i(x_i + s_i)],$$

Define

$$\rho_k = \max\left[\frac{f(x_{r(k)}) - f(x_k + s_k)}{\sigma_k^{\rm h} + m_k(x_k) - m_k(x_k + s_k)}, \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}\right].$$

If $\rho_k \geq \eta_1$, then define $x_{k+1} = x_k + s_k$; otherwise define $x_{k+1} = x_k$.

Step 4: Trust-region radius update. Set

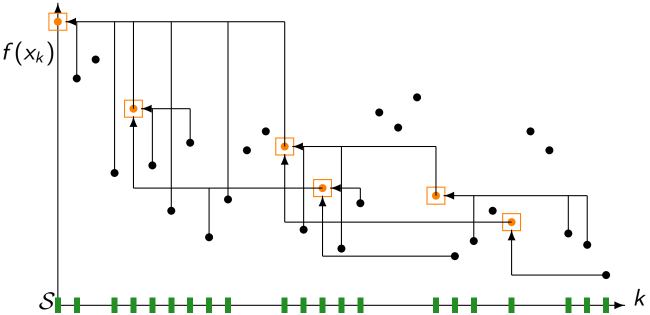
$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2\Delta_k, \Delta_k) & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1\Delta_k, \gamma_2\Delta_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment k by one and go to Step 1.

Sufficient decrease for NMTR

$$f(x_{p(k)}) - f(x_{k+1}) \geq \eta_1 \kappa_{\text{mdc}} \sum_{j=p(k),\, j\in\mathcal{S}}^{k} \|g_j\| \min\left[\frac{\|g_j\|}{\beta_j}, \Delta_j\right]$$

with $p(k) = r(k)$ when $\rho_k^{\text{h}} \geq \rho_k^{\text{c}}$, or $p(k) = k$ otherwise



$$f(x_0) - f(x_{k+1}) \geq \eta_1 \kappa_{\text{mdc}} \sum_{t=0,\, t\in\mathcal{S}}^{k} \|g_t\| \min\left[\frac{\|g_t\|}{\beta_t}, \Delta_t\right].$$

Choosing the reference iteration (1)

Algorithm 3.4: Choosing $r(k)$

Step 3: Acceptance of the trial point.

Step 3a: update the iterate. Compute $f(x_k + s_k)$ and set

$$\rho_k = \max\left[\frac{f_r - f(x_k+s_k)}{\sigma_r + m_k(x_k) - m_k(x_k+s_k)}, \frac{f(x_k) - f(x_k+s_k)}{m_k(x_k) - m_k(x_k+s_k)}\right].$$

If $\rho_k < \eta_1$, then $x_{k+1} = x_k$ and go to Step 4; otherwise $x_{k+1} = x_k + s_k$ and

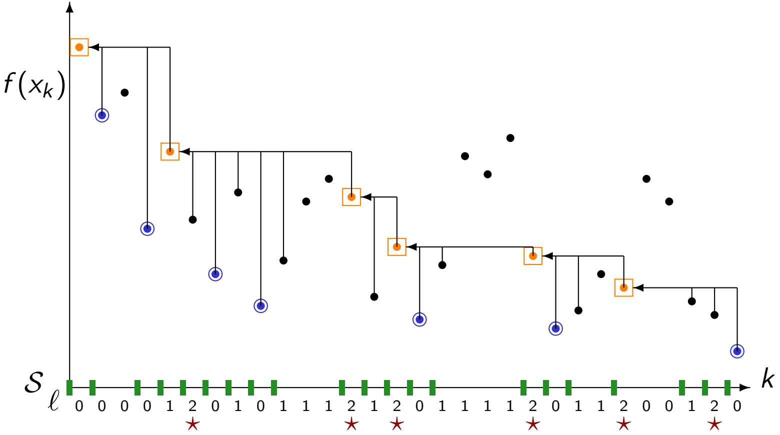
$$\sigma_c = \sigma_c + m_k(x_k) - m_k(x_{k+1}) \text{ and } \sigma_r = \sigma_r + m_k(x_k) - m_k(x_{k+1})$$

Step 3b: update the best value. If $f(x_{k+1}) < f_{\min}$ then set $f_c = f_{\min} = f(x_{k+1})$, $\sigma_c = 0$ and $\ell = 0$ and go to Step 4; otherwise, $\ell \leftarrow \ell + 1$.

Step 3c: update the reference candidate. If $f(x_{k+1}) > f_c$, set $f_c = f(x_{k+1})$ and $\sigma_c = 0$.

Step 3d: possibly reset the reference value. If $\ell = m$, set $f_r = f_c$ and $\sigma_r = \sigma_c$.

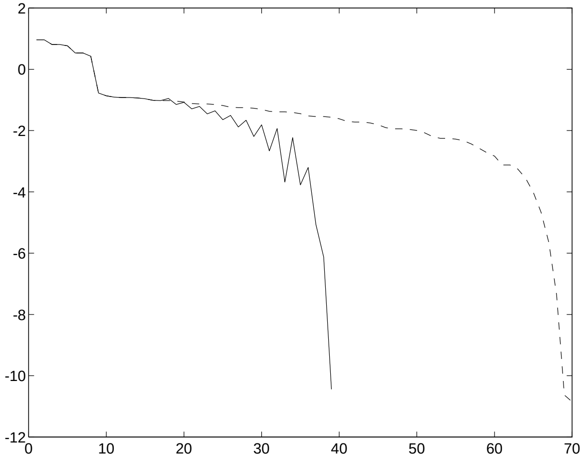
Choosing the reference iteration (2): example with $m = 2$



● : reference iteration ● : new best value

⋆ : reference iteration redefined ($l = m$)

An unconstrained example



Monotone and non-monotone TR (using LANCELOT B) on EXTROSNB

Introducing the filter

A fruitful alternative: **filter methods**

Constrained optimization :

using the SQP step, at the same time:

- reduce the objective function $f(x)$
- reduce constraint violation $\theta(x)$

$\Rightarrow$ CONFLICT

The filter point of view

Fletcher and Leyffer replace question:

What is a better point?

by:

What is a worse point?

Of course, y is worse than x when

$$f(x) \leq f(y) \;\text{ and }\; \theta(x) \leq \theta(y)$$

(y is dominated by x)

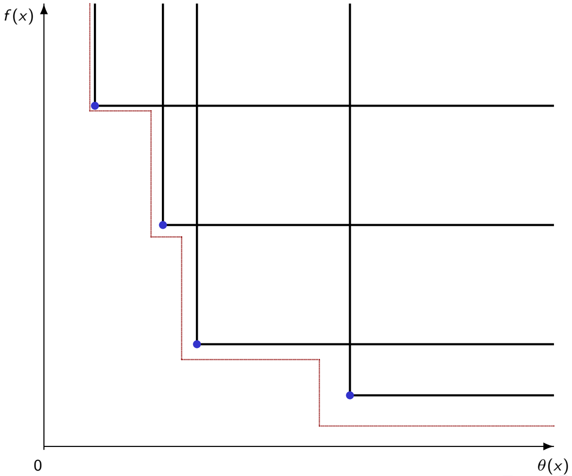
When is $x_k + s_k$ acceptable?

Fletcher and Leyffer (2002), Fletcher, Gould, Leyffer, T. and Wächter (2002)

The standard filter

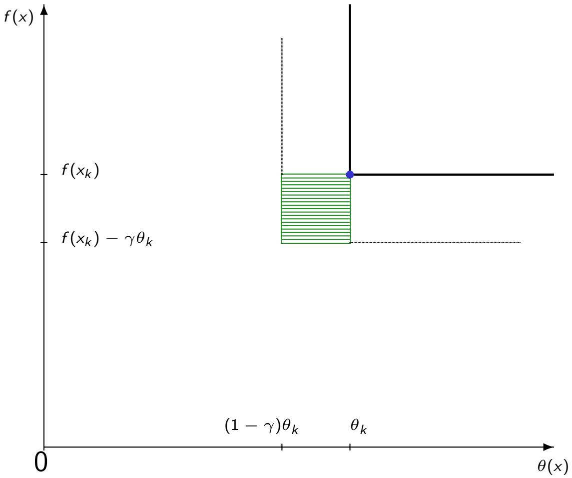
Idea: accept non-dominated points

no monotonicity of merit function implied



Filling up the standard filter

Note: filter area is bounded in the (f, θ) space!



$\Rightarrow$ filter area (non)-monotonically decreasing

The (unconstrained) feasibility problem

Feasibility

Find x such that

$$c(x) \geq 0$$

$$e(x) = 0$$

for general smooth c and e.

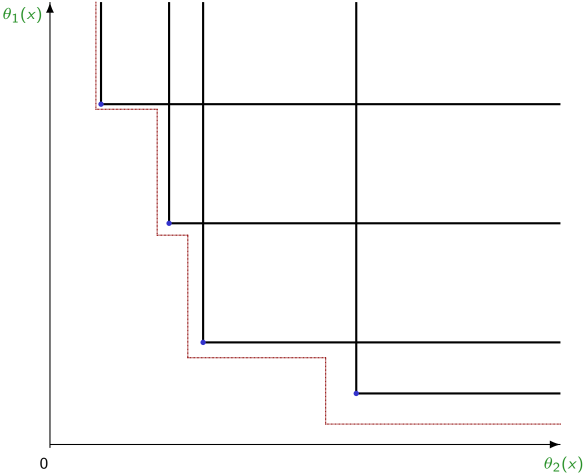
Least-squares

Find x such that

$$\min \sum \theta_i^2$$

A multidimensional filter (1)

(Simple) idea: more dimensions in filter space



(full dimension vs. grouping)

A multidimensional filter (2)

Additionally

- possibly consider unsigned filter entries
- use a trust-region algorithm when
  - trial point unacceptable
  - convergence to non-zero solution

($\Rightarrow$ "internal" restoration)

Sound convergence theory

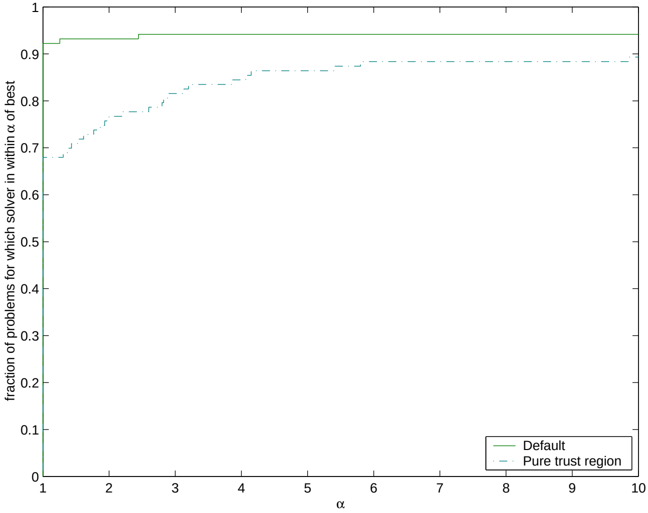
Gould, Leyffer and T. (2005)

Numerical experience: FILTRANE

- Fortran 95 package
- large scale problems (CUTEr interface)
- includes several variants of the method
  - signed/unsigned filters
  - Gauss-Newton, Newton or adaptive models
  - pure trust-region option
  - uses preconditioned conjugate-gradients + Lanczos for subproblem solution
- part of the GALAHAD library

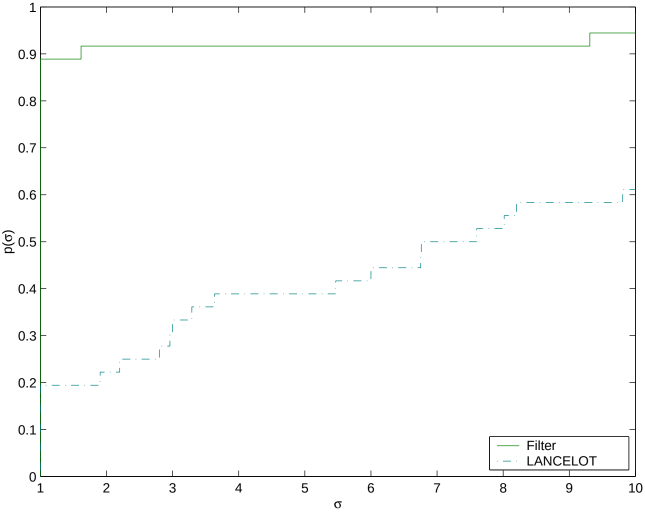
Gould, Orban and T. (2003), Gould and T. (2007)

Numerical experience (1)



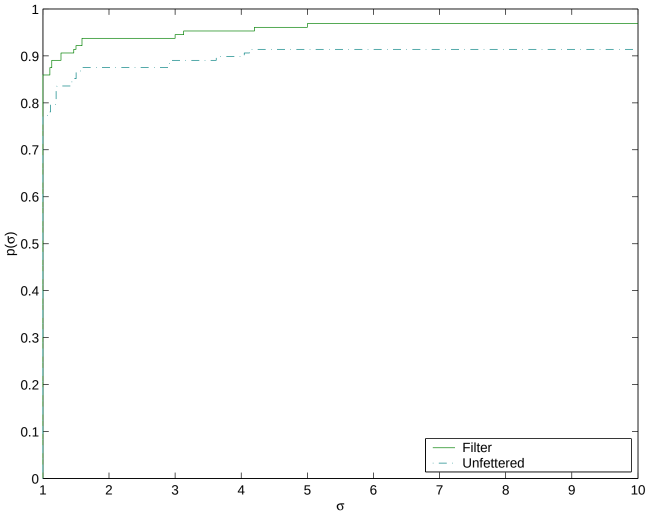
Filter vs. trust-region (CPU time)

Numerical experience (2)



Filter vs. LANCELOT B (CPU time)

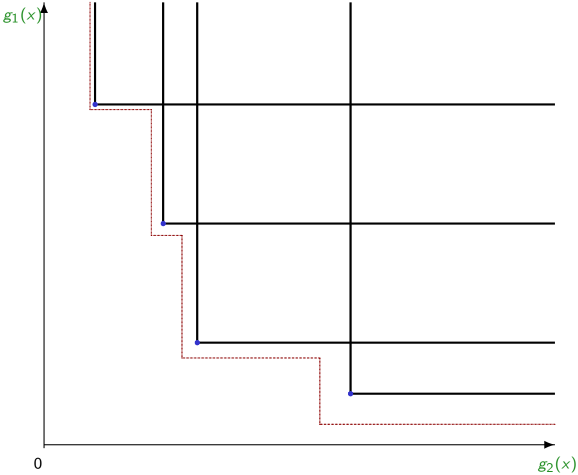
Numerical experience (3)



Filter vs. free Newton (CPU time)

Filter for unconstrained optimization

Again simple idea: use g_i instead of θ_i



(full dimension vs. grouping)

A few complications...

But ...

$g(x) = 0$ not sufficient for nonconvex problems!

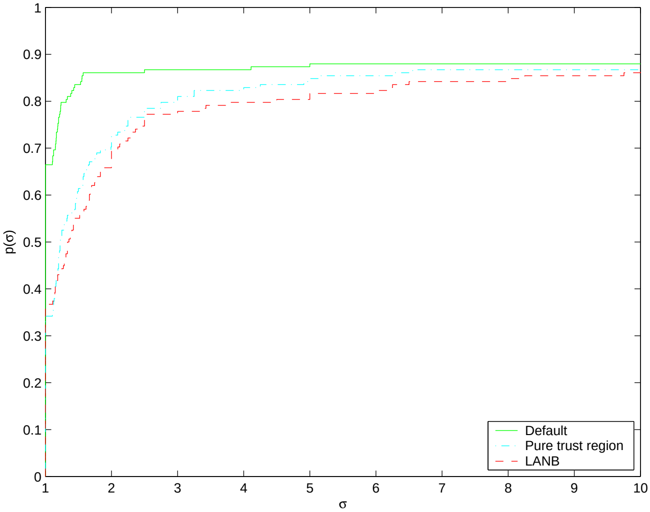
When negative curvature found:

- reset filter
- set upper bound on acceptable $f(x)$

(or... add a dimension for f in the filter)

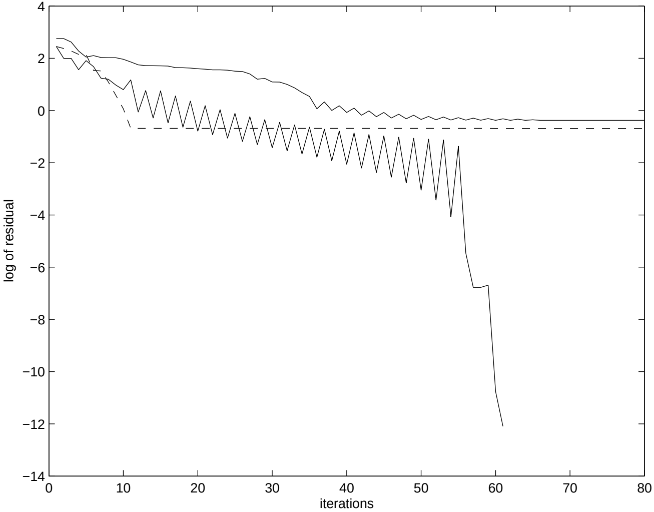
reasonable convergence theory

Numerical experience (1)



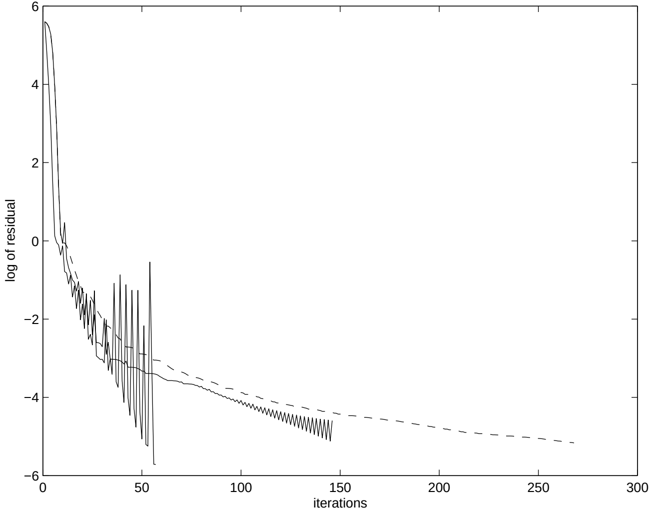
Filter vs. trust-region and LANCELOT B (iterations)

Numerical experience: HEART6



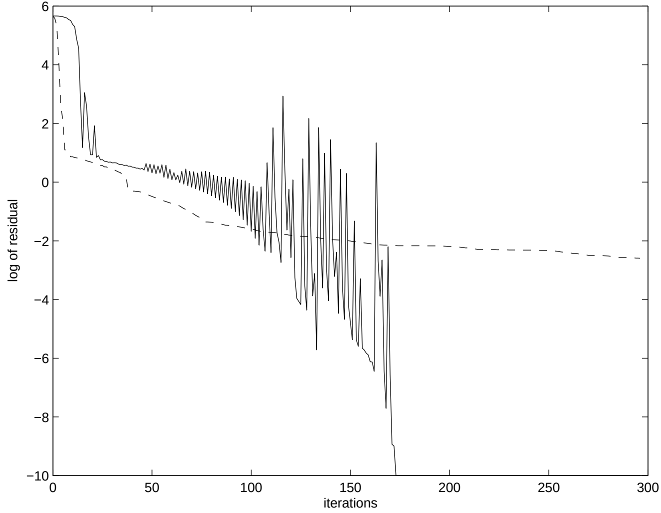
Filter vs. trust-region and LANCELOT B

Numerical experience: EXTROSNB



Filter vs. trust-region and LANCELOT B

Numerical experience: LOBSTERZ



Filter vs. trust-region

Conclusions

derivative-free optimization possible and efficient

non-monotonicity definitely helpful

filter methods very efficient

Newton's behaviour unexplained

. . . more research needed?

Bibliography for lesson 3 (1)

1. A. R. Conn, K. Scheinberg and Ph. L. Toint,
A Derivative Free Optimization Algorithm in Practice,
Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, September 1998.
2. A. R. Conn, K. Scheinberg and L. N. Vicente,
Introduction to Derivative-free Optimization,
SIAM-MPS Series on Optimization, 2008.
3. R. Fletcher and S. Leyffer,
Nonlinear Programming without a penalty function,
Mathematical Programming A, 91(2):239-269, 2002.
4. R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint and A. Wächter,
Global Convergence of Trust-Region SQP-Filter Algorithms for Nonlinear Programming,
SIAM Journal on Optimization, 13(3):635-659, 2002.
5. N. I. M. Gould and S. Leyffer and Ph. L. Toint,
A Multidimensional Filter Algorithm for Nonlinear Equations and Nonlinear Least-Squares,
SIAM Journal on Optimization, 15(1):17-38, 2005.
6. N. I. M. Gould, D. Orban and Ph. L. Toint,
GALAHAD—**a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization**, ACM Transactions on Mathematical Software, 29(4):353-372, 2003.
7. N. I. M. Gould,C. Sainvitu and Ph. L. Toint,
A Filter-Trust-Region Method for Unconstrained Optimization,
SIAM Journal on Optimization, 16(2):341-357, 2005.
8. N. I. M. Gould and Ph. L. Toint,
FILTRANE**, a Fortran 95 filter-trust-region package for solving systems of nonlinear equalities, nonlinear inequalities and nonlinear least-squares problems**,
ACM Transactions on Mathematical Software, 33(1):3-25, 2007.
9. M. J. D. Powell,
A direct search optimization method that models the objective by quadratic interpolation,
Presentation at the 5th Stockholm Optimization Days, Stockholm, 1994.

Bibliography for lesson 3 (2)

10. M. J. D. Powell,
Developments of NEWUOA for minimization without derivatives,
IMA Journal of Numerical Analysis, 28(4):649-664, 2008.

11. K. Scheinberg and Ph. L. Toint,
Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization,
Report TR09/06, FUNDP, Namur, 2009.

12. Ph. L. Toint,
A non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints,
Mathematical Programming, 77(1):69-94, 1997.

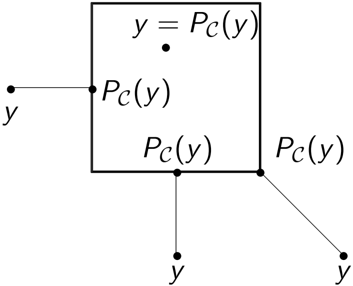
13. D. Winfield,
Function Minimization by Interpolation in a Data Table,
Journal of the IMA, 12:339-347, 1973.

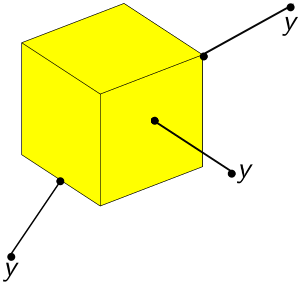
Lesson 4:

Optimization with convex constraints

4.1: Projection algorithms

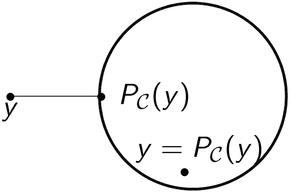
Projections on simple convex domains (1)

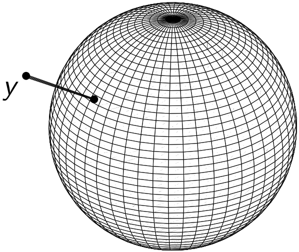


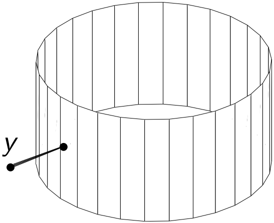


$$[P_{\mathcal{C}}(y)]_i \stackrel{\text{def}}{=} \begin{cases} [x_\ell]_i & \text{if } [y]_i \le [x_\ell]_i, \\ [y]_i & \text{if } [x_\ell]_i < [y]_i < [x_u]_i, \\ [x_u]_i & \text{if } [x_u]_i \le [y]_i \end{cases}$$

Projections on simple convex domains (2)

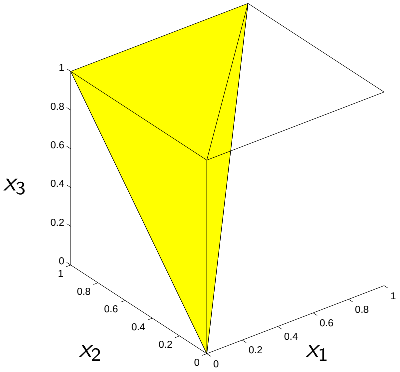






Projections on simple convex domains (2)

. . . but also the ordered simplex . . .

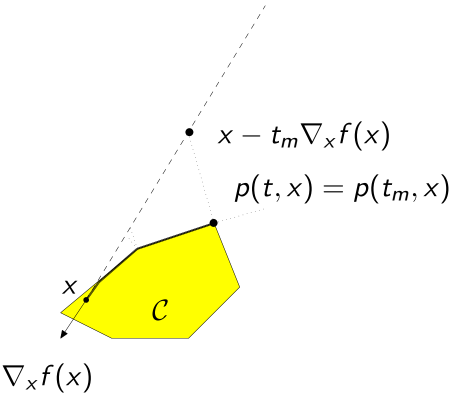


Idea: use those simple projections!

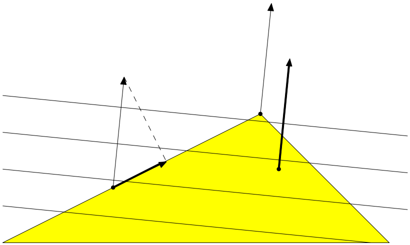
The projected gradient path

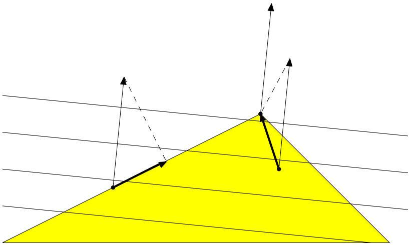
Define the projected gradient path = the Cauchy arc

$$p(t, x) = P_{\mathcal{C}}[x - t\nabla_x f(x)]$$



Two projections





$P_{\mathcal{T}(x)}[-\nabla_x f(x)] \notin C^0$

$P_{\mathcal{C}}[x - \nabla_x f(x)] - x \in C^0$

Measuring criticality

Measure the gain in linearized objective function per step of length θ:

$$\chi(x, \theta) \stackrel{\text{def}}{=} \left| \min_{x+d \in \mathcal{F}, \|d\| \leq \theta} \langle \nabla_x f(x), d \rangle \right|$$

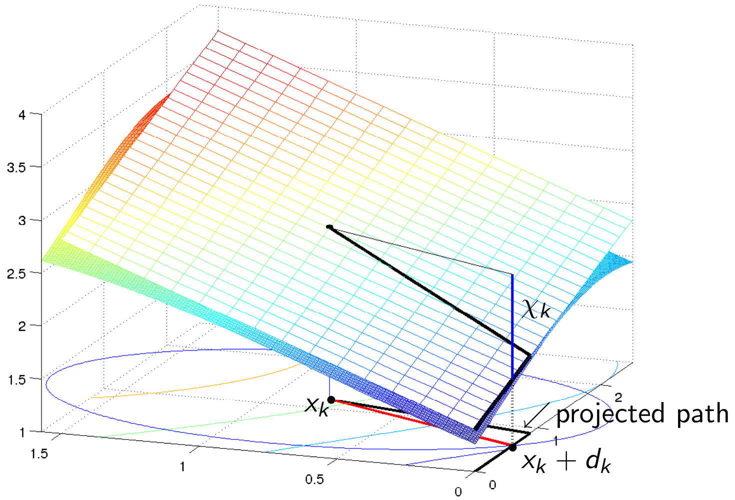
$$\theta(t) = \|P_{\mathcal{F}}(x - tg(x)) - x\| \qquad\qquad \pi(x, \theta) = \frac{\chi(x)}{\theta}$$

The χ criticality measure

$$\chi(x) \stackrel{\rm def}{=} \chi(x,1) = \left| \min_{x+d\in\mathcal{F},\|d\|\leq 1} \langle \nabla_x f(x), d\rangle \right|$$

- the feasible reduction in the linearized objective for unit steps
- reduces to $\|\nabla_x f(x)\|_2$ in the unconstrained case

The projected gradient path and χ



The generalized Cauchy point

Approximately minimize $m_k(\cdot)$ on the PG path

Find

$$x_k^{\rm GC} = P_{\mathcal{F}}[x_k - t_k^{\rm GC} g_k] \stackrel{\rm def}{=} x_k + s_k^{\rm GC} \quad (t_k^{\rm GC} > 0)$$

such that

$$m_k(x_k^{\rm GC}) \le f(x_k) + \kappa_{\rm ubs}\langle g_k, s_k^{\rm GC}\rangle \qquad \text{(below linear approximation)}$$

and either

$$m_k(x_k^{\rm GC}) \ge f(x_k) + \kappa_{\rm lbs}\langle g_k, s_k^{\rm GC}\rangle \qquad \text{(above linear approximation)}$$

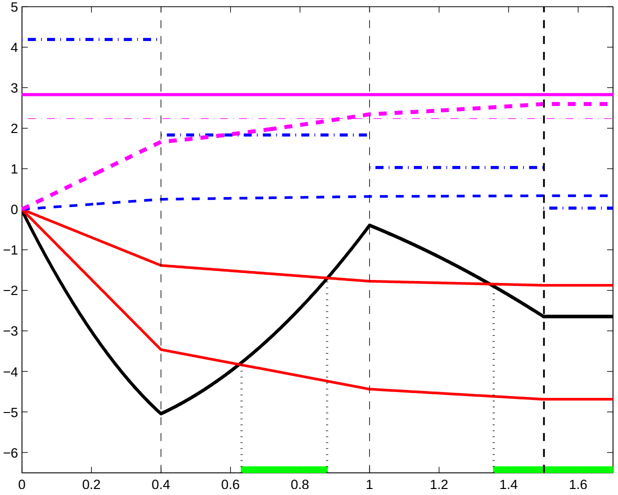
or

$$\|P_{T(x_k^{\rm GC})}[-g_k]\| \le \kappa_{\rm epp}|\langle g_k, s_k^{\rm GC}\rangle| \qquad \text{(close to path's end)}$$

or

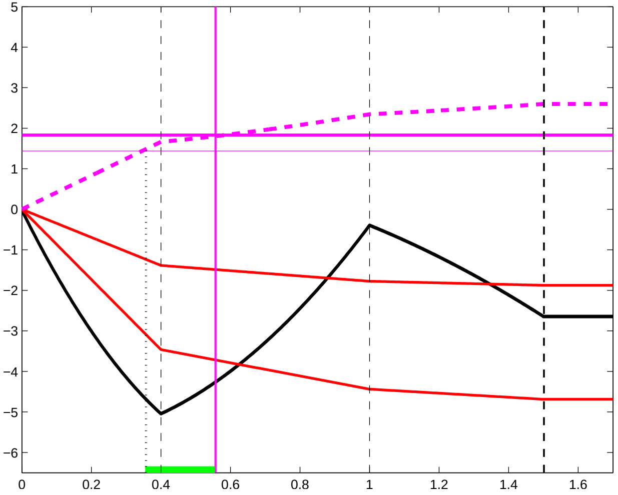
$$\|s_k^{\rm GC}\| \ge \kappa_{\rm frd}\Delta_k \qquad \text{(close to TR boundary)}$$

Searching for the GCP (1)



$$m_k(0+s) = -3.57s_1 - 1.5s_2 - s_3 + s_1 s_2 + 3s_2^2 + s_2 s_3 - 2s_3^2 \ \text{ such that } \ s \leq 1.5 \ \text{ and } \ \Delta \leq 2.8$$

Searching for the GCP (2)



$m_k(0+s) = -3.57 s_1 - 1.5 s_2 - s_3 + s_1 s_2 + 3 s_2^2 + s_2 s_3 - 2 s_3^2$ such that $s \leq 1.5$ and $\Delta \leq 1.8$

Useful properties

Piecewise search for $x_k^{\rm GC}$ well-defined and finite

1. $\theta(\cdot,\cdot)$, $\chi(\cdot,\cdot)$ and $\pi(\cdot,\cdot)$ are continuous
2. $\theta(x,\cdot)$ is non-decreasing
3. $\chi(x,\cdot)$ is non-decreasing
4. $\pi(x,\cdot)$ is non-increasing
5. $\chi(x_k) \leq \chi(x_k, \|s_k^{\rm GC}\|) + 2\|P_{T(x_k^{\rm GC})}[-g_k]\|$
6. $-\langle g_k, s_k^{\rm GC}\rangle = \chi(x_k, \|s_k^{\rm GC}\|) \geq 0$
7. $\theta(x_k, t) \geq t\,\|P_{T(x(t))}[-\nabla_x f(x_k)]\|$
8. $|\chi(x) - \chi(y)| \leq L\|x - y\|$
   if $\nabla_x f(x)$ is continuous on a bounded level set

Cauchy decrease along the projected gradient path

The Cauchy condition: minimize m_k long the projected gradient path

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\rm CR} \chi_k \min\left[\frac{\chi_k}{1 + \|H_k\|}, \Delta_k, 1\right]$$

Idea: Linesearch conditions imply

$$m_k(x_k) - m_k(x_k^{\rm GC}) \geq \kappa_{\rm ubs} |\langle g_k, s_k^{\rm GC}\rangle| = \kappa_{\rm ubs} \chi(x_k, \|s_k^{\rm GC}\|)$$

but need

$$\|P_{\mathcal{T}(P[x_k - t_j g_k])}[-g_k]\| \leq \kappa_{\rm epp} \frac{|\langle g_k, s_k(t_j)\rangle|}{\Delta_k}$$

Now define $\pi_k \stackrel{\rm def}{=} \min[1, \chi_k] \leq \chi_k$. Then

$$m_k(x_k) - m_k(x_k^{\rm GC}) \geq \kappa_{\rm dcp} \pi_k \min\left[\frac{\pi_k}{\beta_k}, \Delta_k\right]$$

How far can we turn the handle?

As above. . .

All limit points are first-order critical, i.e.

$$\lim_{k\to\infty} \pi_k = 0$$

But . . .

does the active set settle ?

(needed for 2nd-order convergence or rate)

Active constraints identification (1)

Require further assumptions: let $\mathcal{L}_* = \{\,\text{limit points of }\ \{x_k\}\,\}$

- $\forall x_* \in \mathcal{L}_*$, $\{\nabla_x c_i(x_*)\}_{i\in\mathcal{A}(x_*)}$ are linearly independent
- $\forall x_* \in \mathcal{L}_*$, $-\nabla_x f(x_*) \in \mathrm{ri}\{\mathcal{N}(x_*)\}$
- $\forall k$, $\mathcal{A}(x_k^{\mathrm{GC}}) \subseteq \mathcal{A}(x_k + s_k)$

For each connected component of limit points $\mathcal{L}(x_*) \subseteq \mathcal{L}_*$, there exists a set $\mathcal{A}_* \subseteq \{1,\ldots,m\}$ for which

$$\mathcal{A}(x_*) = \mathcal{A}_* \ \text{ for all }\ x_* \in \mathcal{L}(x_*).$$

Idea: connectivity + uniqueness of Lagrange multipliers
$\Rightarrow$ each $\mathcal{L}(x_*)$ belongs to a single facet of $\mathcal{C}$

Active constraints identification (2)

There exists a $\psi \in (0,1)$ such that

$$\mathrm{dist}(x_*, \mathcal{L}') \geq \psi$$

for every $x_* \in \mathcal{L}_*$ and each compact connected component of limit points $\mathcal{L}'$ such that $\mathcal{A}(\mathcal{L}') \neq \mathcal{A}(x_*)$.

Idea: continuity + compactness $\Rightarrow$ well separated

There exist $\delta \in (0, \frac{1}{4}\psi)$, $\psi \in (0,1)$, and $k_1 \geq 0$ such that, for $k \geq k_1$, there is a $\mathcal{L}_{*k}$ such that

$$x_k \in \mathcal{V}(\mathcal{L}_{*k}, \delta) = \{x \in \mathbf{R}^n \mid \mathrm{dist}(x, \mathcal{L}_{*k}) \leq \delta\}$$

and

$$\mathcal{A}(x) \subseteq \mathcal{A}(\mathcal{L}_{*k}) \;\text{ for all }\; x \in \mathcal{V}(\mathcal{L}_{*k}, \delta).$$

Idea: partition the complete sequence into convergent subsequences
$\Rightarrow$ each x_k near a unique $\mathcal{L}_{*k}$

Active constraints identification (3)

There exists $k_2 \geq k_1$ such that, if for some $k \geq k_2$,

$$j \in \mathcal{A}(\mathcal{L}_{*k}) \text{ and } j \notin \mathcal{A}(x_k^{\rm GC}),$$

then, for some $\epsilon_* \in (0,1)$ independent of k and j,

$$\pi_k \geq \epsilon_*.$$

Idea: complicated (uses criticality measures for incomplete constraint sets)
$\Rightarrow$ incomplete local $\mathcal{A}(x_k)$ implies not critical
(more technical arguments here)

There exists an active set $\mathcal{A}_*$, such that

$$\forall x_* \in \mathcal{L}_* \quad \mathcal{A}(x_*) = \mathcal{A}_*$$

and, for all k sufficiently large,

$$\mathcal{A}(x_k) = \mathcal{A}(x_k^{\rm GC}) = \mathcal{A}_*$$

Further convergence results

. . . and now it works in $\mathcal{T}(x_k)$ (now continuous for large k) with

$$\nabla_{xx} m_k \text{ remplaced by } \nabla_{xx} m_k^{\ell} \approx \nabla_{xx} \ell(x_k, y_k)$$

- convergence to isolated critical points
- (generalized) eigen-points for the Lagrangian (needs consistent multiplier estimates!)
- convergence to second-order points
- fast asymptotic rate of convergence

4.2: Barrier methods

A simple case

Consider $\mathcal{C} = \{x \in \mathbb{R}^n \mid x \geq 0\}$ and build

$$\phi^{\log}(x,\mu) \stackrel{\text{def}}{=} f(x) - \mu\langle e, \log(x)\rangle = f(x) - \mu \sum_{i=1}^{n} \log(x_i)$$

Under acceptable assumptions,

$$x_*(\mu) = \arg\min_x \phi^{\log}(x,\mu)$$

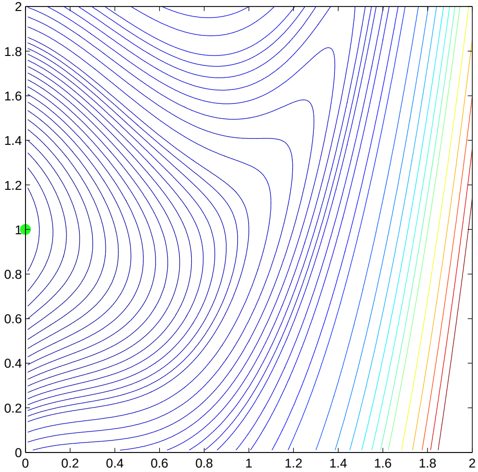
converge to the solution of the problem

$$\min_{x \in \mathcal{C}} f(x)$$

when $\mu \searrow 0$.

How it works. . .

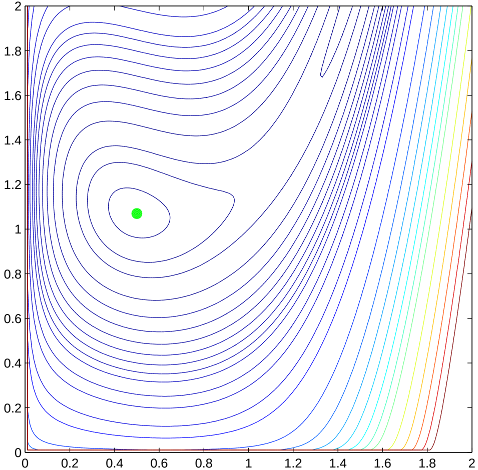
Example: $\min_{x_1,x_2\geq 0} 120\left[x_1^2(x_1-1)-x_2+1\right]^2+10(4+x_1)^2-150$



original objective function

How it works. . .

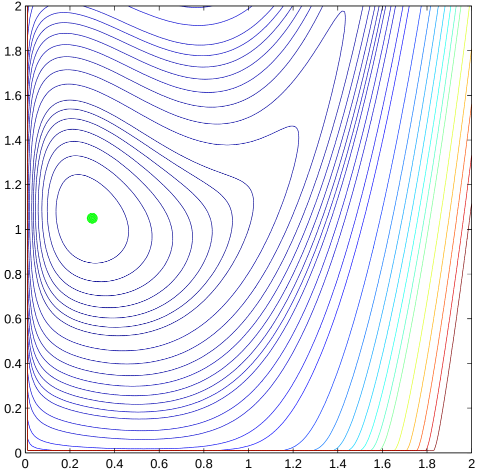
Example: $\min_{x_1,x_2 \geq 0} 120\left[x_1^2(x_1-1)-x_2+1\right]^2 + 10(4+x_1)^2 - 150$



original objective function + barrier ($\mu = 50$)

How it works. . .

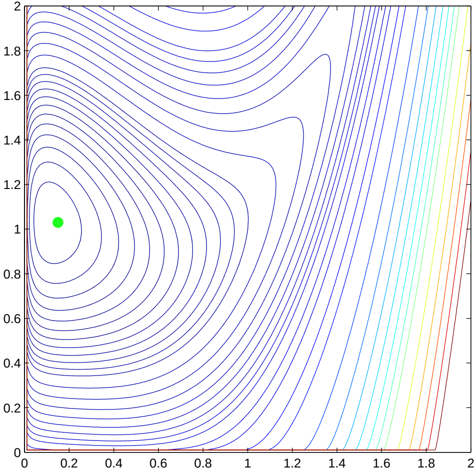
Example: $\min_{x_1,x_2\geq 0} 120\left[x_1^2(x_1-1)-x_2+1\right]^2+10(4+x_1)^2-150$



original objective function + barrier ($\mu = 25$)

How it works. . .

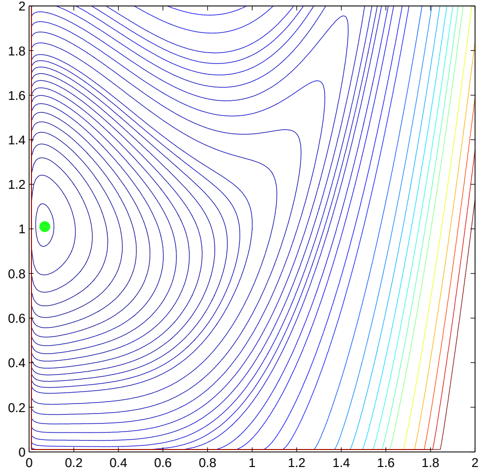
Example: $\min_{x_1,x_2\geq 0} 120\left[x_1^2(x_1-1)-x_2+1\right]^2+10(4+x_1)^2-150$



original objective function + barrier ($\mu = 10$)

How it works. . .

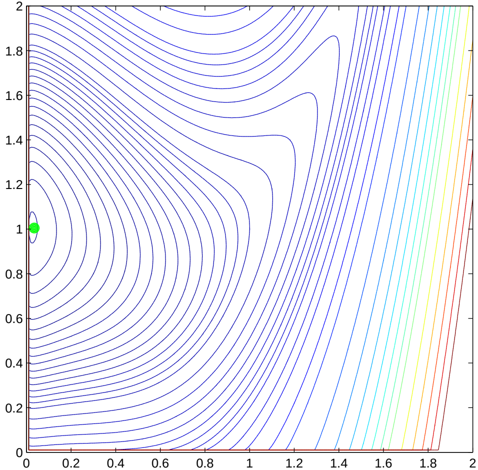
Example: $\min_{x_1,x_2\geq 0} 120\left[x_1^2(x_1-1)-x_2+1\right]^2+10(4+x_1)^2-150$



original objective function + barrier ($\mu = 5$)

How it works. . .

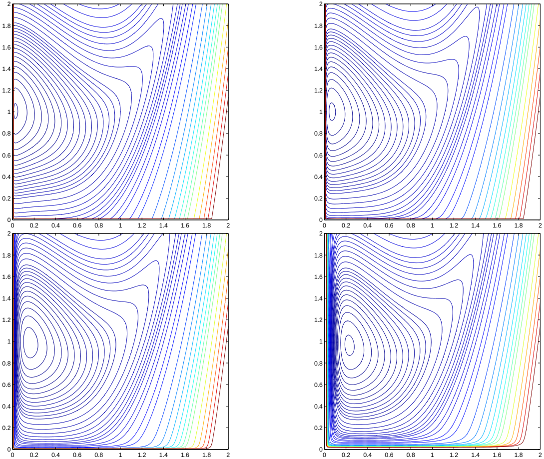
Example: $\min_{x_1,x_2\geq 0} 120\left[x_1^2(x_1-1)-x_2+1\right]^2+10(4+x_1)^2-150$



original objective function + barrier ($\mu = 2$)

Other barriers: reciprocals

$$b^{\mathrm{R}(\alpha)}(x,\mu) = \mu \sum_{i=1}^{n} \frac{1}{\alpha [x]_i^{\alpha}}$$



$(\mu = 2,\ \log + R(\tfrac{1}{2}),\ R(1)$ and $R(2))$

The barrier function

$$\phi(x,\mu) = f(x) + b(x,\mu) \stackrel{\text{def}}{=} f(x) - \mu\langle e, \log(x)\rangle$$

Assume:

- $b(x,\mu)$ is defined for all $x \in \mathrm{ri}\{\mathcal{C}\}$ and all $\mu > 0$, and is $C^2(\mathrm{ri}\{\mathcal{C}\})$ w.r.t. x.
- $\forall \mu > 0, \epsilon > 0 \; \exists \kappa_{\text{bbh}}(\epsilon,\mu) \geq 1$ such that

$$\|\nabla_{xx} b(x,\mu)\| \leq \kappa_{\text{bbh}}(\epsilon,\mu)$$

  $\forall x \in \mathcal{C}$ such that $\mathrm{dist}(x, \partial\mathcal{C}) \geq \epsilon$

- $\lim_{p\to\infty} b(y_p,\mu) = +\infty \; \forall \mu > 0$ and $\forall \{y_p\}_{p=0}^{\infty}$ such that

$$y_p \in \mathrm{ri}\{\mathcal{C}\} \;\text{ and }\; \lim_{p\to\infty} \mathrm{dist}(y_p, \partial\mathcal{C}) = 0.$$

An elementary barrier algorithm

Algorithm 4.1: A simple barrier algorithm

Step 0: Initialization. Given: $x_0 \in \mathrm{ri}\{\mathcal{C}\}, \mu_0 > 0$. Set $k = 0$.

Step 1: Inner minimization. (Approximately) solve the problem

$$\min_x \phi(x, \mu_k)$$

by applying an unconstrained (inner) algorithm, starting from a suitable starting point $x_{k,0} \in \mathrm{ri}\{\mathcal{C}\}$.
Let x_{k+1} be the corresponding (approximate) solution.

Step 2: Update the barrier parameter. Choose $\mu_{k+1} > 0$ such that

$$\lim_{k \to \infty} \mu_k = 0.$$

Increment k by one and return to Step 1.

A first inner primal algorithm

Algorithm 4.2: Inner primal 1

Step 0: Initialization. Given: $x_{k,0} \in \mathrm{ri}\{\mathcal{C}\}$, $\Delta_{k,0}$, η_1, η_2, γ_1, γ_2, $\varsigma_k \in (0,1)$.
Compute $\phi(x_0, \mu_k)$, set $j = 0$.

Step 1: Model definition. Define $m_{k,j}$ of $\phi(x_{k,j} + s, \mu_k)$ in $\mathcal{B}_{k,j}$ of the form

$$m_{k,j}(x_{k,j} + s) = m^f_{k,j}(x_{k,j} + s) + m^b_{k,j}(x_{k,j} + s),$$

Step 2: Step calculation. Compute $s_{k,j}$ that sufficiently reduces $m_{k,j}$ and such that $x_{k,j} + s_{k,j} \in \mathcal{B}_{k,j}$.

Step 3: Acceptance of the trial point. If $x_{k,j} + s_{k,j} \notin \mathcal{C}$ or if $\mathrm{dist}(x_{k,j} + s_{k,j}, \partial\mathcal{C}) < \varsigma_k \mathrm{dist}(x_{k,j}, \partial\mathcal{C})$, set $\rho_{k,j} = -\infty$, $x_{k,j+1} = x_{k,j}$ and go to Step 4.
Otherwise compute $\phi(x_{k,j} + s_{k,j}, \mu_k)$ and

$$\rho_{k,j} = \frac{\phi(x_{k,j}, \mu_k) - \phi(x_{k,j} + s_{k,j}, \mu_k)}{m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j})}.$$

Then if $\rho_{k,j} \geq \eta_1$, define $x_{k,j+1} = x_{k,j} + s_{k,j}$; otherwise define $x_{k,j+1} = x_{k,j}$.

Step 4: Trust-region radius update. Set

$$\Delta_{k,j+1} \in \begin{cases} [\Delta_{k,j}, \infty) & \text{if } \rho_{k,j} \geq \eta_2, \\ [\gamma_2 \Delta_{k,j}, \Delta_{k,j}] & \text{if } \rho_{k,j} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_{k,j}, \gamma_2 \Delta_{k,j}] & \text{if } \rho_{k,j} < \eta_1. \end{cases}$$

Increment j by one and go to Step 1.

Models and assumptions

Use separate models for f and b!

$$m_{k,j}(x_{k,j}+s) = m^f_{k,j}(x_{k,j}+s) + m^b_{k,j}(x_{k,j}+s),$$

Assume:

- $\forall k, \epsilon > 0,\ \exists \kappa_{\rm bbmh}(\epsilon,\mu_k) \geq 1\ \forall k, j \geq 0,$

  $$\|\nabla_{xx} m^b_{k,j}(x,\mu_k)\| \leq \kappa_{\rm bbmh}(\epsilon,\mu_k)$$

  $\forall x \in \mathcal{B}_{k,j} \cap \mathcal{C}$ such that $\mathrm{dist}(x,\partial\mathcal{C}) \geq \epsilon$.
- $\forall k, j \geq 0\ \forall x \in \mathcal{B}_{k,j} \cap \mathrm{ri}\{\mathcal{C}\},$

  $$\|\nabla_{xx} m^f_{k,j}(x)\| \leq \kappa_{\rm umh}$$

(Inner) convergence properties

There exists $\kappa_{\text{mdb}}(k) \in (0,1)$ such that

$$\text{dist}(x_{k,j}, \partial\mathcal{C}) \geq \kappa_{\text{mdb}}(k)$$

for all j. Moreover, for all j and all x such that $\|x - x_{k,j}\| \leq (1-\varsigma_k)\text{dist}(x_j, \partial\mathcal{C})$, we have that

$$\|\nabla_{xx} b(x,\mu)\| \leq \kappa_{\text{bbh}}(\varsigma_k\, \kappa_{\text{mdb}}(k), \mu_k)$$

and

$$\|\nabla_{xx} m^b_{k,j}(x_{k,j},\mu)\| \leq \kappa_{\text{bbmh}}(\varsigma_k\, \kappa_{\text{mdb}}(k), \mu_k)$$

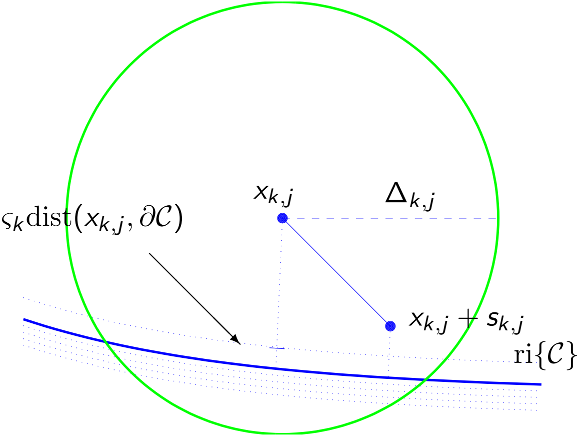
If $\Delta_{k,j} \leq (1-\varsigma_k)\kappa_{\text{mdb}}(k)$, then

$$|\phi(x_{k,j}+s_{k,j},\mu_k) - m_{k,j}(x_{k,j}+s_{k,j})| \leq \kappa_{\text{ubh}}(k)\Delta^2_{k,j}$$

...and all the nice convergence properties follow!

Constrained Cauchy and eigen-points (1)

Idea: restrict the step, not the trust region!



But . . . what of sufficient decrease ???

Constrained Cauchy and eigen-points (2)

Redefine the Cauchy arc:

$$x_{k,j}^{\rm CC}(t) \stackrel{\rm def}{=} \{x \mid x = x_{k,j} - tg_{k,j},\ t \geq 0,\ t\|g_{k,j}\| \leq (1-\varsigma_k)d_{k,j} \ \text{ and } \ x \in \mathcal{B}_k\},$$

$$m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}^{\rm CC}) \geq \tfrac{1}{2}\|g_{k,j}\| \min\left[\frac{\|g_{k,j}\|}{\beta_{k,j}}, \Delta_{k,j}, (1-\varsigma_k)d_{k,j}\right]$$

. . . etc, etc, etc . . .

A second inner primal algorithm

Algorithm 4.3: Inner primal 2

Step 0: Initialization. Given: $x_{k,0} \in \mathrm{ri}\{\mathcal{C}\}$, $\Delta_{k,0}$, η_1, η_2, γ_1, γ_2, $\varsigma_k \in (0,1)$.
Compute $\phi(x_{k,0}, \mu_k)$, set $j = 0$.

Step 1: Model definition. Define $m_{k,j}(x_{k,j}+s) = m^f_{k,j}(x_{k,j}+s) + m^b_{k,j}(x_{k,j}+s)$

Step 2: Step calculation. Define $d_{k,j} = \mathrm{dist}(x_{k,j}, \partial\mathcal{C})$. Compute $s_{k,j}$ such that

$$x_{k,j} + s_{k,j} \in \mathcal{B}_{k,j} \cap \mathcal{C} \quad \text{and} \quad \mathrm{dist}(x_{k,j}+s_{k,j}, \partial\mathcal{C}) \geq \varsigma_k d_{k,j}$$

and such that it sufficiently reduces $m_{k,j}$

Step 3: Acceptance of the trial point. Compute $\phi(x_{k,j}+s_{k,j}, \mu_k)$ and

$$\rho_{k,j} = \frac{\phi(x_{k,j}, \mu_k) - \phi(x_{k,j}+s_{k,j}, \mu_k)}{m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}+s_{k,j})}.$$

Then if $\rho_{k,j} \geq \eta_1$, define $x_{k,j+1} = x_{k,j} + s_{k,j}$; otherwise define $x_{k,j+1} = x_{k,j}$.

Step 4: Trust-region radius update. Set

$$\Delta_{k,j+1} \in \begin{cases} [\Delta_{k,j}, \infty) & \text{if } \rho_{k,j} \geq \eta_2, \\ [\gamma_2\Delta_{k,j}, \Delta_{k,j}] & \text{if } \rho_{k,j} \in [\eta_1, \eta_2), \\ [\gamma_1\Delta_{k,j}, \gamma_2\Delta_{k,j}] & \text{if } \rho_{k,j} < \eta_1. \end{cases}$$

Increment j by one and go to Step 1.

The log barrier and its derivatives

Return to:

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

The log barrier

$$b(x, \mu) = -\mu \langle e, \log(x) \rangle$$

giving

$$\phi^{\log}(x, \mu) = f(x) - \mu \langle e, \log(x) \rangle$$

Using the notation $X = \mathrm{diag}(x_1, \ldots, x_n)$, we obtain that

$$\nabla_x b(x, \mu) = -\mu X^{-1} e \quad \text{and} \quad \nabla_{xx} b(x, \mu) = \mu X^{-2} e$$

The primal log-barrier algorithm

Algorithm 4.4: Primal log-barrier algorithm

Step 0: Initialization. Given: $x_0 > 0$, $\mu_0 > 0$, and the forcing functions $\epsilon^{\rm D}(\mu)$ and $\epsilon^{\rm E}(\mu)$. Set $k = 0$.

Step 1: Inner minimization. Choose a value $\varsigma_k \in (0,1)$. Approximately minimize the log-barrier function $\phi^{\log}(x,\mu_k) = f(x) - \mu_k \langle e, \log(x) \rangle$ starting from x_k and using an inner algorithm in which

$$m^b_{k,j}(x_{k,j}+s) = \mu_k \left(-\langle e, \log(x_{k,j}) \rangle - \langle X_{k,j}^{-1} e, s \rangle + \tfrac{1}{2} \langle s, X_{k,j}^{-2} s \rangle \right)$$

Stop this algorithm as soon as an iterate $x_{k,j} = x_{k+1}$ is found for which

$$\|\nabla_x f(x_{k+1}) - \mu_k X_{k+1}^{-1} e\| \le \epsilon^{\rm D}(\mu_k),$$

$$\lambda_{\min}[\nabla_{xx} f(x_{k+1}) + \mu_k X_{k+1}^{-2}] \ge -\epsilon^{\rm E}(\mu_k)$$

and $x_{k+1} > 0$.

Step 2: Update the barrier parameter. Choose $\mu_{k+1} > 0$ such that $\lim_{k\to\infty} \mu_k = 0$. Increment k by one and return to Step 1.

Convergence of the primal log-barrier algorithm (1)

OK for first order! ... but existence of limit points not guaranteed

Define

> A subsequence $\{x_{k_j}\}$ is consistently active w.r.t. the bounds if, for each $i = 1, \ldots, n$, either
>
> $$\lim_{j\to\infty} [x_{k_j}]_i = 0 \quad \text{or} \quad \liminf_{j\to\infty} [x_{k_j}]_i > 0.$$

(Each bound constraint is asymptotically active or inactive for the complete subsequence.)

$$\mathcal{A}\{x_{k_j}\} \stackrel{\text{def}}{=} \{i \in \{1, \ldots, n\} \mid \lim_{j\to\infty} [x_{k_j}]_i = 0\}.$$

Note: finite number of such subsequences $\Rightarrow$ a partition of $\{x_k\}$

Convergence of the primal log-barrier algorithm (2)

Finally,

Under appropriate assumptions,

$$\liminf_{k\to\infty} [\nabla_x f(x_k)]_i \geq 0, \quad (i = 1, \ldots, n).$$

Furthermore, for every consistently active subsequence $\{x_{k_\ell}\}$,

$$\lim_{\ell\to\infty} [\nabla_x f(x_{k_\ell})]_i = 0, \quad (i \notin \mathcal{A}\{x_{k_\ell}\})$$

and

$$\liminf_{\ell\to\infty} \langle u, [\nabla_{xx} f(x_{k_\ell})] u\rangle \geq 0$$

for each $u \mid [u]_i = 0$ whenever $i \in \mathcal{A}\{x_{k_\ell}\}$.

The primal-dual framework (1)

In practice, as $x_k \searrow 0$, $\nabla_{xx} m_{k,j}(x_{k,j}) + \mu_k X_{k,j}^{-2}$ causes slow progress.

Idea: replace this by

$$\nabla_{xx} m_{k,j}(x_{k,j}) + X_{k,j}^{-1} Z_{k,j}$$

where $Z_{k,j}$ is a bounded positive diagonal.

Alternatively: KKT conditions for original problem:

$$\nabla_x m(x) - z = 0, \quad XZ = 0, \quad x \geq 0, \quad z \geq 0,$$

Perturb:

$$\nabla_x m(x) - z = 0, \quad XZ = \mu e \quad x \geq 0, \quad z \geq 0.$$

The primal-dual framework (2)

Now write Newton's method for the perturbed problem:

$$\begin{aligned} \nabla_{xx} m_{k,j}(x_{k,j})\Delta x_{k,j} - \Delta z_{k,j} &= -g_{k,j} + z_{k,j}, \\ X_{k,j}\Delta z_{k,j} + Z_{k,j}\Delta x_{k,j} &= \mu_k e - X_{k,j}Z_{k,j}e, \\ x_{k,j} + \Delta x_{k,j} \geq 0, &\quad z_{k,j} + \Delta z_{k,j} \geq 0. \end{aligned}$$

Substituting the 2nd equation into the 1st:

$$\left[\nabla_{xx} m_{k,j}(x_{k,j}) + X_{k,j}^{-1} Z_{k,j}\right]\Delta x_{k,j} = -\left[g_{k,j} - \mu_k X_{k,j}^{-1} e\right]$$

But

$$g_{k,j} - \mu_k X_{k,j}^{-1} e = \nabla_x \phi^{\log}(x, \mu_k)$$

Hence

$$\left[\nabla_{xx} m_{k,j}(x_{k,j}) + X_{k,j}^{-1} Z_{k,j}\right]\Delta x_{k,j} = -\nabla_x \phi^{\log}(x, \mu_k)$$

The primal-dual inner algorithm (1)

Algorithm 4.5: Inner primal-dual algorithm

Step 0: Initialization. Given: $x_{k,0} \in \mathrm{ri}\{\mathcal{C}\}$, $z_{k,0} > 0$, $\Delta_{k,0}$, η_1, η_2, γ_1,γ_2, ς_k. Compute $f(x_{k,0})$, set $j = 0$.

Step 1: Model definition. In $\mathcal{B}_{k,j}$, define

$$m_{k,j}(x_{k,j}+s) = m^f_{k,j}(x_{k,j}+s) - \mu_k \left[\langle e, \log(x_{k,j})\rangle + \langle X_{k,j}^{-1} e, s\rangle\right] - \tfrac{1}{2}\langle s, X_{k,j}^{-1} Z_{k,j} s\rangle$$

Step 2: Step calculation. Define $d_{k,j} = \mathrm{dist}(x_{k,j}, \partial\mathcal{C})$. Compute a step $s_{k,j}$ such that $x_{k,j} + s_{k,j} \in \mathcal{B}_{k,j}$, $\mathrm{dist}(x_{k,j}+s_{k,j}, \partial\mathcal{C}) \geq \varsigma_k d_{k,j}$, and

$$m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}+s_{k,j}) \geq \kappa \max\left\{\|g_{k,j}\| \min\left[\frac{\|g_{k,j}\|}{\beta_{k,j}}, \Delta_{k,j}, (1-\varsigma_k)d_{k,j}\right], -\tau_{k,j} \min\left[\tau_{k,j}^2, \Delta_{k,j}^2, (1-\varsigma_k)^2 d_{k,j}^2\right]\right\}$$

Step 3: Acceptance of the trial point. Compute $\phi^{\log}(x_{k,j}+s_{k,j}, \mu_k)$ and

$$\rho_{k,j} = \frac{\phi^{\log}(x_{k,j}, \mu_k) - \phi^{\log}(x_{k,j}+s_{k,j}, \mu_k)}{m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}+s_{k,j})}.$$

If $\rho_{k,j} \geq \eta_1$, then $x_{k,j+1} = x_{k,j} + s_{k,j}$, else $x_{k,j+1} = x_{k,j}$.

The primal-dual inner algorithm (2)

Algorithm 4.6: Inner primal-dual algorithm (2)

Step 4: Trust-region radius update. Set

$$\Delta_{k,j+1} \in \begin{cases} [\Delta_{k,j}, \infty) & \text{if } \rho_{k,j} \geq \eta_2, \\ [\gamma_2 \Delta_{k,j}, \Delta_{k,j}] & \text{if } \rho_{k,j} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_{k,j}, \gamma_2 \Delta_{k,j}] & \text{if } \rho_{k,j} < \eta_1. \end{cases}$$

Step 5: Update the dual variables. Set $z_{k,j+1} > 0$. Increment j by one,go to Step 1.

The primal-dual outer algorithm

Algorithm 4.7: Outer primal-dual algorithm

Step 0: Initialization. Given: $x_0 > 0$, $z_0 > 0$, $\mu_0 > 0$ and the forcing functions $\epsilon^{\mathrm{D}}(\mu)$, $\epsilon^{\mathrm{E}}(\mu)$, $\epsilon^{\mathrm{C}}(\mu)$. Set $k = 0$.

Step 1: Inner minimization. Choose $\varsigma_k \in (0,1)$. Approximately minimize $\phi^{\log}(x, \mu_k)$ from x_k using the primal-dual inner algorithm. Stop as soon as an iterate $(x_{k,j}, z_{k,j}) = (x_{k+1}, z_{k+1})$ is found for which

$$\|\nabla_x f(x_{k+1}) - z_{k+1}\| \leq \epsilon^{\mathrm{D}}(\mu_k),$$

$$\|X_{k+1} Z_{k+1} - \mu_k I\| \leq \epsilon^{\mathrm{C}}(\mu_k),$$

$$\lambda_{\min}[\nabla_{xx} f(x_{k+1}) + X_{k+1}^{-1} Z_{k+1}] \geq -\epsilon^{\mathrm{E}}(\mu_k)$$

and

$$x_{k+1} > 0 \text{ and } z_{k+1} > 0.$$

Step 3: Update the barrier parameter. Choose $\mu_{k+1} > 0$ such that $\lim_{k\to\infty} \mu_k = 0$. Increment k by one and return to Step 1.

Note: choosing $z_{k,j} = -\mu_k X_{k,j}^{-1} e \Rightarrow$ primal algorithm!

Updating the dual variables

How to compute $z_{k,j+1}$ in practice? Newton equations give

$$\bar{z}_{k,j+1} = \mu_k X_{k,j}^{-1} e - X_{k,j}^{-1} Z_{k,j} s_{k,j}.$$

. . . but what about $z_{k,j+1} \geq 0$?
Define

$$\mathcal{I} = \left[\kappa_{\rm zul} \min\left(e, z_{k,j}, \mu_k X_{k,j+1}^{-1} e\right), \kappa_{\rm zuu} \max\left(e, z_{k,j}, \mu_k^{-1} e, \mu_k X_{k,j+1}^{-1} e\right) \right]$$

and choose

$$z_{k,j+1} = \begin{cases} P_{\mathcal{I}}[\bar{z}_{k,j+1}] & \text{if } x_{k,j+1} = x_{k,j} + s_{k,j} \\ z_{k,j} & \text{if } x_{k,j+1} = x_{k,j}, \end{cases}$$

Properties of the dual variables

Then $z_{k,j+1} > 0$ and

$$[z_{k,j}]_i \leq \kappa_{\rm uzi} \max\left[\frac{1}{[x_{k,j}]_i}, 1\right].$$

If, furthermore,

$$\lim_{j\to\infty} \|s_{k,j}\| = 0 \;\text{ when }\; \lim_{j\to\infty} \|g_{k,j}\| = 0$$

then

$$\lim_{j\to\infty} \left\|z_{k,j} - \mu_k X_{k,j}^{-1} e\right\| = 0 \;\text{ if }\; \lim_{j\to\infty} \|g_{k,j}\| = 0.$$

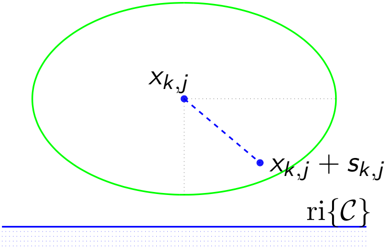
$\Rightarrow$ asymptotically exact barrier Hessian for fixed μ

Scaling of the inner iterations

In practice, scaling is crucial!
Ideally,

$$\| \cdot \|_{k,j} = \| \cdot \|_{\nabla_{xx} m_{k,j}(x_{k,j})} = \sqrt{\langle \cdot, [H_{k,j} + X_{k,j}^{-1} Z_{k,j}] \cdot \rangle}$$

Under the usual assumptions, $\| \cdot \|_{k,j}$ is uniformly equivalent to the Euclidean norm for fixed k.



$\Rightarrow$ all usual convergence properties for fixed k

Scaling of the outer iterations (1)

Scaled tests:

$$\|\nabla_x f(x_{k+1}) - z_{k+1}\|_{[k+1]} \leq \epsilon^{\mathrm{D}}(\mu_k)$$

$$\|X_{k+1} Z_{k+1} - \mu_k I\|_2 \leq \epsilon^{\mathrm{C}}(\mu_k),$$

$$\lambda_{\min}\left[M_{k+1}^{-\frac{1}{2}}(\nabla_{xx} f(x_{k+1}) + X_{k+1}^{-1} Z_{k+1}) M_{k+1}^{-\frac{1}{2}}\right] \geq -\epsilon^{\mathrm{E}}(\mu_k),$$

with

$$M_{k+1} \stackrel{\text{def}}{=} H_{k+1} + X_{k+1}^{-1} Z_{k+1}$$

But this matrix is unbounded when $k \nearrow \infty$!

Scaling of the outer iterations (2)

Fortunately,

Under the usual assumptions, the convergence properties are preserved if

$$\lim_{k\to\infty} \frac{\epsilon^{\rm D}(\mu_k)}{\mu_k} \le \kappa_\mu$$

and

$$\lim_{k\to\infty} \frac{\epsilon^{\rm C}(\mu_k)\sqrt{\mu_k}}{\min_i [x_{k+1}]_i} = 0.$$

Moreover

If exact derivatives are used, the $\epsilon^{\bullet}(\mu_k)$ can be chosen to ensure componentwise near quadratic rate of convergence.

This is quite remarkable!

Barriers for general convex constraints

Now,

$$\phi^{\log}(x,\mu) = f(x) - \mu\langle e, \log(c(x))\rangle$$

The primal-dual model becomes

$$m_{k,j}(x_{k,j}+s_{k,j}) = m^f_{k,j}(x_{k,j}+s_{k,j}) + m^b_{k,j}(x_{k,j}+s_{k,j}),$$

with

$$\begin{aligned} m^b_{k,j}(x_{k,j}+s_{k,j}) \quad = \quad & \mu_k\langle e, \log(c(x_{k,j}))\rangle - \mu_k\langle C^{-1}(x_{k,j})e, A(x_{k,j})s_{k,j}\rangle \\ & +\tfrac{1}{2}\langle A(x_{k,j})s_{k,j}, [C^{-1}(x_{k,j})Y_{k,j}]A(x_{k,j})s_{k,j}\rangle \\ & -\tfrac{1}{2}\sum_{i=1}^{m}[y_{k,j}]_i\langle s_{k,j}, \nabla_{xx}c_i(x_{k,j})s_{k,j}\rangle \end{aligned}$$

Quite a mouthful... but otherwise everything is OK!

The outer primal-dual algorithm for convex constraints

$$\nabla_{xx}\ell(x_{k,j}, y_{k,j}) = \nabla_{xx} f(x_{k,j}) - \sum_{i=1}^{m} [y_{k,j}]_i \nabla_{xx} c_i(x_{k,j}) \qquad G_{k,j} \stackrel{\text{def}}{=} A^T(x_{k,j}) C^{-1}(x_{k,j}) Y_{k,j} A(x_{k,j})$$

Algorithm 4.8: Primal-dual algorithm for convex constraints

Step 0: Initialization Given: $x_0 \mid c(x_0) > 0$, $y_0 > 0$, $\mu_0 > 0$, $\epsilon^{\mathrm{C}}(\mu)$, $\epsilon^{\mathrm{D}}(\mu)$ and $\epsilon^{\mathrm{E}}(\mu)$. Set $k = 0$.

Step 1: Inner minimization Choose $\varsigma_k \in (0,1)$. Approximately minimize

$$\phi^{\log}(x, \mu_k) = f(x) - \mu_k \langle e, \log(c(x)) \rangle$$

from x_k. Stop as soon as $(x_{k,j}, y_{k,j}) = (x_{k+1}, y_{k+1})$ is found such that

$$\|\nabla_x f(x_{k+1}) - A^T(x_{k+1}) y_{k+1}\| \le \epsilon^{\mathrm{D}}(\mu_k),$$

$$\|C(x_{k+1}) Y_{k+1} e - \mu_k I\| \le \epsilon^{\mathrm{C}}(\mu_k),$$

$$\lambda_{\min}[\nabla_{xx}\ell(x_{k+1}, y_{k+1}) + G_{k+1}] \ge -\epsilon^{\mathrm{E}}(\mu_k)$$

and

$$(c(x_{k+1}), y_{k+1}) \ge 0.$$

Step 3: Update the barrier parameter. Choose $\mu_{k+1} > 0$ such that $\lim_{k\to\infty} \mu_k = 0$. Increment k by one and return to Step 1.

Bibliography for lesson 4 (1)

1. J. V. Burke,
On the identification of active constraints,
SIAM Journal on Numerical Analysis, 25(5):1197-1211, 1988.
2. J. V. Burke,
On the identification of active constraints II: the nonconvex case,
SIAM Journal on Numerical Analysis, 27(4):1081-1102, 1990.
3. J. V. Burke and J. J. Moré,
Exposing Constraints,
SIAM Journal on Optimization, 4(3):573-595, 1994.
4. J. V. Burke, J. J. Moré and G. Toraldo,
Convergence properties of trust region methods for linear and convex constraints,
Mathematical Programming A, 47(3):305-336, 1990.
5. A. R. Conn, N. I. M. Gould, D. Orban and Ph. L. Toint,
A primal-dual trust-region algorithm for minimizing a non-convex function subject to bound and linear equality constraints,
Mathematical Programming, 87(2):215-249, 2000.
6. N. I. M. Gould, D. Orban, A. Sartenaer and Ph. L. Toint,
Superlinear Convergence of Primal-Dual Interior Point Algorithms for Nonlinear Programming,
SIAM Journal on Optimization, 11(4):974-1002, 2001.
7. A. R. Conn, N. I. M. Gould and Ph. L. Toint,
Global convergence of a class of trust region algorithms for optimization with simple bounds,
SIAM Journal on Numerical Analysis, 25(182):433-460, 1988.
8. A. R. Conn, N. I. M. Gould and Ph. L. Toint,
LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A),
Spinger Verlag, Heidelberg, 1992.
9. A. R. Conn, N. I. M. Gould, A. Sartenaer and Ph. L. Toint,
Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints,
SIAM Journal on Optimization, 3(1):164–221, 1993.

Bibliography for lesson 4 (2)

10. A. V. Fiacco and G. P. McCormick,
The Sequential Unconstrained Minimization Technique for Nonlinear Programming: a Primal-Dual Method,
Management Science, 0(2):360-366, 1964.

11. A. V. Fiacco and G. P. McCormick,
Nonlinear Programming: Sequential Unconstrained Minimization Techniques,
Wiley and Sons, Chichester (UK), 1968.

12. Ph. L. Toint,
Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space,
IMA Journal of Numerical Analysis, 8(2):231-252, 1988.

Lesson 5:

Sparsity, partial separability and multilevel methods: exploiting problem structure

Outline

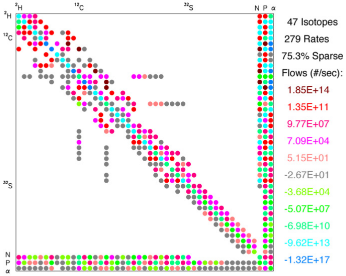
1. Sparsity and partial separability
2. Multilevel problems
3. Bibliography

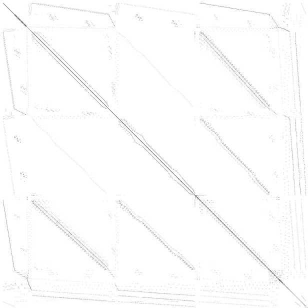
5.1: Sparsity and partial separability

Sparsity

A matrix is sparse when the proportion and/or distribution of its zero entries allows its efficient numerical usage

An (oriented) graph is asociated with every sparse (non)-symmetric matrix





Main benefits of sparsity

Sparsity and optimization $\Rightarrow$ Hessian (and) Jacobian matrices

- very important time/space savings in solving Newton's equations (unconstrained or KKT)
  1. factorizations (reduced fill-in)
  2. iterative methods (fast matrix$\times$vector products)
- sometimes important in approximations schemes
  1. derivative-free methods (makes the number of function evaluations $\approx$ linear in the number of variables)
  2. finite-difference approximations
  3. quasi Newton methods
- a path for parallel computations

exploiting sparsity = an active algorithmic industry!

The Curtis-Powell-Reid algorithm for estimating sparse Jacobians

Finite differences for a Jacobian column:

$$Je_i \approx \frac{c(x + he_i) - c(x)}{h}$$

Question: How many finite differences for estimating a 5×5 Jacobian with the structure:

$$\begin{pmatrix} \bullet & & & & \bullet \\ & \bullet & & \bullet & \\ & & \bullet & \bullet & \bullet \\ \bullet & & \bullet & & \bullet \\ \bullet & \bullet & & & \bullet \end{pmatrix} ?$$

The Curtis-Powell-Reid algorithm for estimating sparse Jacobians

Finite differences for a Jacobian column:

$$Je_i \approx \frac{c(x + he_i) - c(x)}{h}$$

Question: How many finite differences for estimating a 5×5 Jacobian with the structure:

$$\begin{pmatrix} \bullet & & & & \bullet \\ & \bullet & & \bullet & \\ & & \bullet & \bullet & \bullet \\ \bullet & & \bullet & & \bullet \\ \bullet & \bullet & & & \bullet \end{pmatrix}$$

The Curtis-Powell-Reid algorithm for estimating sparse Jacobians

Finite differences for a Jacobian column:

$$Je_i \approx \frac{c(x + he_i) - c(x)}{h}$$

Question: How many finite differences for estimating a 5×5 Jacobian with the structure:

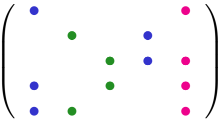
$$\begin{pmatrix} {\color{blue}\bullet} & & & & \bullet \\ & {\color{green}\bullet} & & {\color{blue}\bullet} & \\ & & {\color{green}\bullet} & {\color{blue}\bullet} & \bullet \\ {\color{blue}\bullet} & & {\color{green}\bullet} & & \bullet \\ {\color{blue}\bullet} & {\color{green}\bullet} & & & \bullet \end{pmatrix}$$

The Curtis-Powell-Reid algorithm for estimating sparse Jacobians

Finite differences for a Jacobian column:

$$Je_i \approx \frac{c(x + he_i) - c(x)}{h}$$

Question: How many finite differences for estimating a 5×5 Jacobian with the structure:



The Curtis-Powell-Reid algorithm for estimating sparse Jacobians

Finite differences for a Jacobian column:

$$Je_i \approx \frac{c(x+he_i)-c(x)}{h}$$

Question: How many finite differences for estimating a 5×5 Jacobian with the structure:

$$\begin{pmatrix} {\color{blue}\bullet} & & & & {\color{magenta}\bullet} \\ & {\color{green}\bullet} & & {\color{blue}\bullet} & \\ & & {\color{green}\bullet} & {\color{blue}\bullet} & {\color{magenta}\bullet} \\ {\color{blue}\bullet} & & {\color{green}\bullet} & & {\color{magenta}\bullet} \\ {\color{blue}\bullet} & {\color{green}\bullet} & & & {\color{magenta}\bullet} \end{pmatrix}$$

$$Je_{\color{blue}\bullet} \approx \frac{c(x+h{\color{blue}e_1}+h{\color{blue}e_4})-c(x)}{h} \qquad Je_{\color{green}\bullet} \approx \frac{c(x+h{\color{green}e_2}+h{\color{green}e_3})-c(x)}{h} \qquad Je_{\color{magenta}\bullet} \approx \frac{c(x+h{\color{magenta}e_5})-c(x)}{h}$$

Answer: 3 finite-differences! Curtis, Powell and Reid (1974), Steihaug et al.

The CPR algorithm for estimating sparse Jacobians

Algorithm 5.1: CPR algorithm

Build the column groups.

Place the columns in as few groups as possible such that two columns in the same group have their nonzero entries in different rows

Estimate the finite differences.

1. Build a difference vector $h = \sum_{\text{group}} h_i e_i$
2. Compute $v = c(x+h) - c(x)$

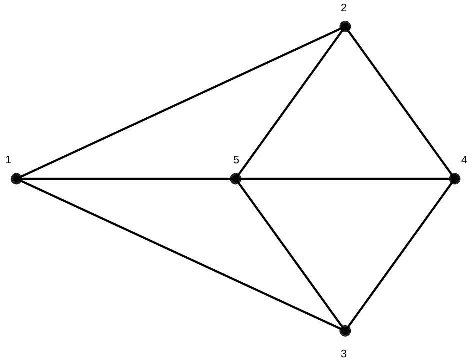
Reconstruct the Jacobian.

$$J_{ij} \approx \frac{v_i}{h_i} \quad \text{for all } j \text{ such that } j \in \text{group}$$

A graph colouring interpretation

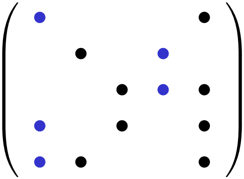
Consider the intersection graph for the columns:

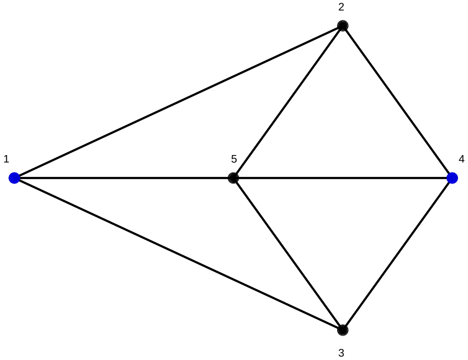
$$\begin{pmatrix} \bullet & & & & \bullet \\ & \bullet & & \bullet & \\ & & \bullet & \bullet & \bullet \\ \bullet & & \bullet & & \bullet \\ \bullet & \bullet & & & \bullet \end{pmatrix}$$



A graph colouring interpretation

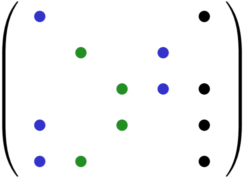
Consider the intersection graph for the columns:

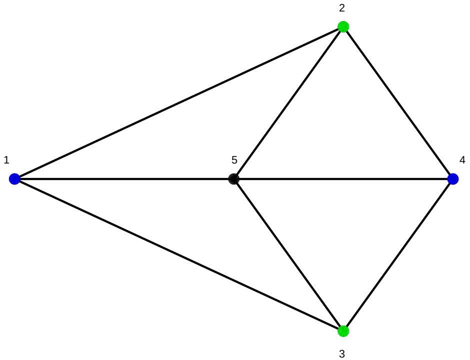




A graph colouring interpretation

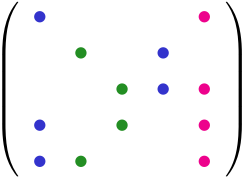
Consider the intersection graph for the columns:

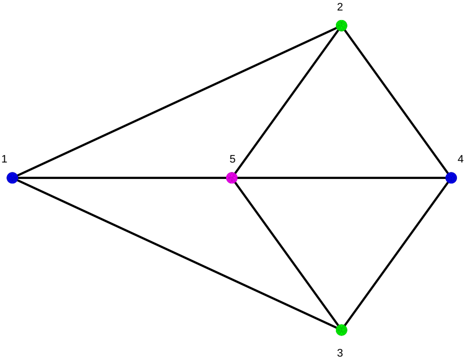




A graph colouring interpretation

Consider the intersection graph for the columns:





minimize the number of colours,
such that adjacent nodes have different colours

can build column groups using heuristic algorithms for graph colouring

Coleman and Moré, (1983)

Estimating sparse Hessians (1)

Question: How many finite differences for estimating a 8×8 symmetric Hessian with the structure:

$$\begin{pmatrix}
\bullet & & \bullet & & & \bullet & & \\
& \bullet & & \bullet & & & & \\
\bullet & & \bullet & & \bullet & \bullet & & \\
& \bullet & & \bullet & & & \bullet & \\
& & \bullet & & \bullet & & \bullet & \bullet \\
\bullet & & \bullet & & & \bullet & & \bullet \\
& & & \bullet & \bullet & & \bullet & \\
& & & & \bullet & \bullet & & \bullet
\end{pmatrix} ?$$

Estimating sparse Hessians (1)

Question: How many finite differences for estimating a 8×8 symmetric Hessian with the structure:

$$\begin{pmatrix}
\bullet & & \bullet & & & \bullet & & \\
& \bullet & & \bullet & & & & \\
\bullet & & \bullet & & \bullet & \bullet & & \\
& \bullet & & \bullet & & & \bullet & \\
& & \bullet & & \bullet & & \bullet & \bullet \\
\bullet & & \bullet & & & \bullet & & \bullet \\
& & & \bullet & \bullet & & \bullet & \\
& & & & \bullet & \bullet & & \bullet
\end{pmatrix}$$

Estimating sparse Hessians (1)

Question: How many finite differences for estimating a 8×8 symmetric Hessian with the structure:

$$\begin{pmatrix}
\bullet & & {\color{blue}\bullet} & & & \bullet & & \\
& {\color{blue}\bullet} & & {\color{blue}\bullet} & & & & \\
{\color{blue}\bullet} & & {\color{blue}\bullet} & & {\color{blue}\bullet} & {\color{blue}\bullet} & & \\
& {\color{blue}\bullet} & & \bullet & & & \bullet & \\
& & {\color{blue}\bullet} & & \bullet & & \bullet & \bullet \\
\bullet & & {\color{blue}\bullet} & & & \bullet & & \bullet \\
& & & \bullet & \bullet & & \bullet & \\
& & & & \bullet & \bullet & & \bullet
\end{pmatrix}$$

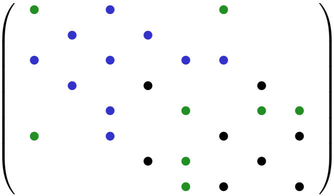
Estimating sparse Hessians (1)

Question: How many finite differences for estimating a 8×8 symmetric Hessian with the structure:

$$\begin{pmatrix}
{\color{green}\bullet} & & {\color{blue}\bullet} & & & \bullet & & \\
& {\color{blue}\bullet} & & {\color{blue}\bullet} & & & & \\
{\color{blue}\bullet} & & {\color{blue}\bullet} & & {\color{blue}\bullet} & {\color{blue}\bullet} & & \\
& {\color{blue}\bullet} & & \bullet & & & \bullet & \\
& & {\color{blue}\bullet} & & {\color{green}\bullet} & & \bullet & \bullet \\
{\color{green}\bullet} & & {\color{blue}\bullet} & & & \bullet & & \bullet \\
& & & \bullet & {\color{green}\bullet} & & \bullet & \\
& & & & {\color{green}\bullet} & \bullet & & \bullet
\end{pmatrix}$$

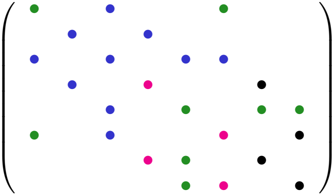
Estimating sparse Hessians (1)

Question: How many finite differences for estimating a 8×8 symmetric Hessian with the structure:



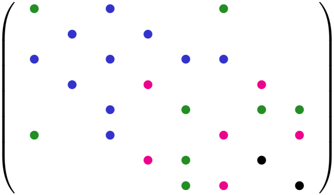
Estimating sparse Hessians (1)

Question: How many finite differences for estimating a 8×8 symmetric Hessian with the structure:



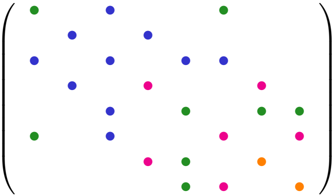
Estimating sparse Hessians (1)

Question: How many finite differences for estimating a 8×8 symmetric Hessian with the structure:



Estimating sparse Hessians (1)

Question: How many finite differences for estimating a 8×8 symmetric Hessian with the structure:



Exploiting symmetry in CPR (a direct method)

Powell and T (1979), Coleman and Moré (1984)

Estimating sparse Hessians (2)

Question: Can we do better?

$$\begin{pmatrix}
\bullet & & \bullet & & & \bullet & & \\
& \bullet & & \bullet & & & & \\
\bullet & & \bullet & & \bullet & \bullet & & \\
& \bullet & & \bullet & & & \bullet & \\
& & \bullet & & \bullet & & \bullet & \bullet \\
\bullet & & \bullet & & & \bullet & & \bullet \\
& & & \bullet & \bullet & & \bullet & \\
& & & & \bullet & \bullet & & \bullet
\end{pmatrix}$$

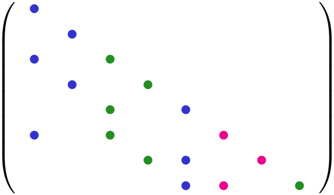
Estimating sparse Hessians (2)

Question: Can we do better?

$$\begin{pmatrix}
\bullet & & & & & & & \\
 & \bullet & & & & & & \\
\bullet & & \bullet & & & & & \\
 & \bullet & & \bullet & & & & \\
 & & \bullet & & \bullet & & & \\
\bullet & & \bullet & & & \bullet & & \\
 & & & \bullet & \bullet & & \bullet & \\
 & & & & \bullet & \bullet & & \bullet
\end{pmatrix}$$

Estimating sparse Hessians (2)

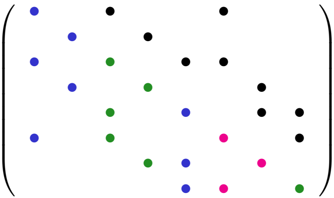
Question: Can we do better?



Apply CPR on the lower triangular part of the Hessian

Estimating sparse Hessians (2)

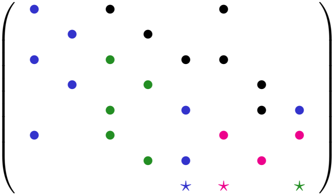
Question: Can we do better?



But what about the conflicts with the upper triangular part?

Estimating sparse Hessians (2)

Question: Can we do better?

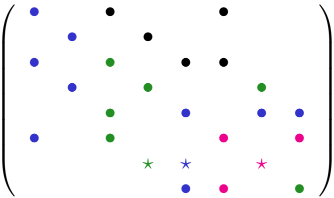


A more efficient substitution method. . .

Powell and T (1979), Coleman and Moré (1984) for a graph interpretation

Estimating sparse Hessians (2)

Question: Can we do better?

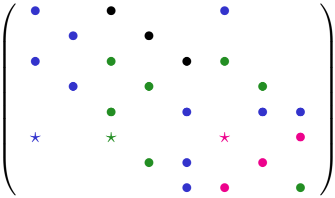


A more efficient substitution method...

Powell and T (1979), Coleman and Moré (1984) for a graph interpretation

Estimating sparse Hessians (2)

Question: Can we do better?

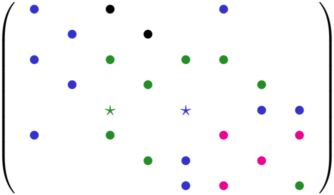


A more efficient substitution method. . .

Powell and T (1979), Coleman and Moré (1984) for a graph interpretation

Estimating sparse Hessians (2)

Question: Can we do better?

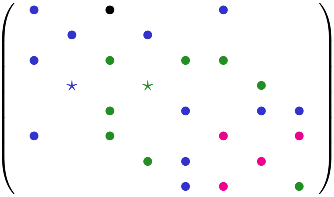


A more efficient substitution method...

Powell and T (1979), Coleman and Moré (1984) for a graph interpretation

Estimating sparse Hessians (2)

Question: Can we do better?

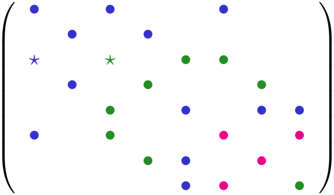


A more efficient substitution method...

Powell and T (1979), Coleman and Moré (1984) for a graph interpretation

Estimating sparse Hessians (2)

Question: Can we do better?

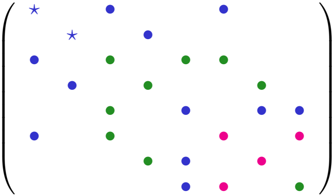


A more efficient substitution method. . .

Powell and T (1979), Coleman and Moré (1984) for a graph interpretation

Estimating sparse Hessians (2)

Question: Can we do better?

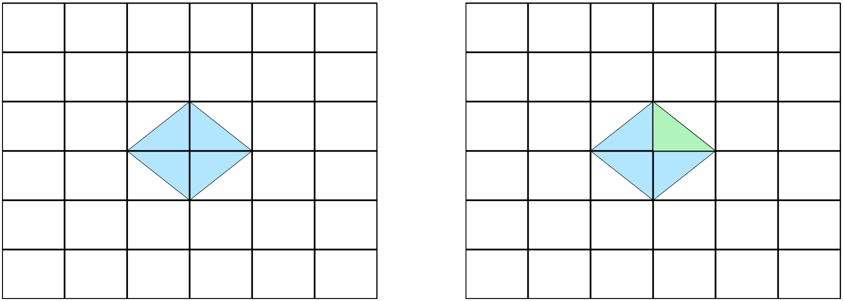


A more efficient substitution method. . .

Powell and T (1979), Coleman and Moré (1984) for a graph interpretation

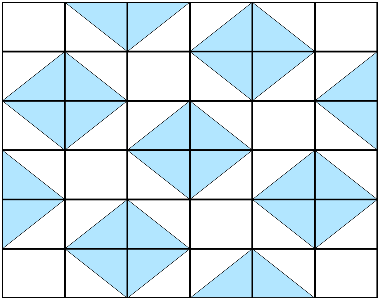
Optimized version for PDE stencils

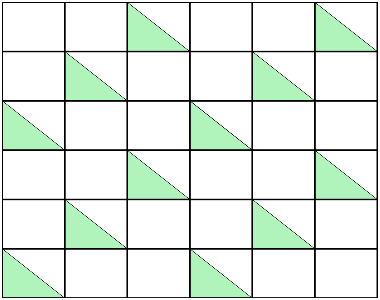
Example: the 5-points Laplacian operator in 2D (non-symmetric and symmetric)



Optimized version for PDE stencils

Example: the 5-points Laplacian operator in 2D (non-symmetric and symmetric)





Partial separability

A more geometric concept: Griewank and T. (1982)

$f(x)$ is partially separable iff

$$f(x) = \sum_{i=1}^{p} f_i(U_i x) \quad \text{where the matrices } U_i \text{ are of low rank}$$

- if U_i = disjoint columns of the identity matrix $\Rightarrow$ (totally) separable
- common case: U_i = overlapping columns of the identity matrix

$$f(x) = \sum_{i=1}^{p} f_i(x_{\mathcal{S}_i})$$

Vocabulary:

element functions, element variables, internal variables $u_i = U_i x$

Sources and examples of partially separable functions

Example 1:

$$f(x_1, x_2, x_3, x_4) = f_1(x_1, x_2) + f_2(x_2, x_3, x_4) + f_3(x_4, x_5)$$

Example 2:

$$f(x_1, x_2, x_3, x_4) = f_1(\underbrace{3x_1 + x_2}_{u_1}) + f_2(\underbrace{-2x_2 + x_3 - 2x_4}_{u_2}, \underbrace{x_4 + 3x_5}_{u_3})$$

Sources:

- (nearly) all discretized problems
- most problems in econometric modelling,
- . . . and a lot more because. . .

Properties of partially separable functions

If $f(x)$ has a sparse Hessian matrix and is sufficiently smooth, then it is partially separable

(but not conversely: ex : $f(x_1,\dots,x_n) = \sum_{i=1}^n f_i(x_i) + f_{n+1}(x_1+\cdots+x_n)$)

If $f(x) = \sum_{i=1}^p f_i(U_i x) = \sum_{i=1}^p f_i(u_i)$, then

$$\nabla_x f(x) = \sum_{i=1}^p U_i^T \nabla_x f_i(u_i)$$

$$\nabla_{xx} f(x) = \sum_{i=1}^p U_i^T \nabla_{xx} f_i(u_i) U_i$$

(easy to compute, sparsity determined by U_i)

The three points Laplacian operator

On a regular geometric grid

$$
\begin{pmatrix}
2 & -1 & & & & \\
-1 & 2 & -1 & & & \\
 & -1 & 2 & -1 & & \\
 & & -1 & 2 & -1 & \\
 & & & -1 & 2 & -1 \\
 & & & & -1 & 2
\end{pmatrix}
=
\begin{pmatrix}
2 & -1 & & & & \\
-1 & 1 & & & & \\
 & & & & & \\
 & & & & & \\
 & & & & & \\
 & & & & &
\end{pmatrix}
+
\begin{pmatrix}
 & & & & & \\
 & 1 & -1 & & & \\
 & -1 & 1 & & & \\
 & & & & & \\
 & & & & & \\
 & & & & &
\end{pmatrix}
$$

$$
+
\begin{pmatrix}
 & & & & & \\
 & & & & & \\
 & & 1 & -1 & & \\
 & & -1 & 1 & & \\
 & & & & & \\
 & & & & &
\end{pmatrix}
+
\begin{pmatrix}
 & & & & & \\
 & & & & & \\
 & & & & & \\
 & & & 1 & -1 & \\
 & & & -1 & 1 & \\
 & & & & &
\end{pmatrix}
+
\begin{pmatrix}
 & & & & & \\
 & & & & & \\
 & & & & & \\
 & & & & & \\
 & & & & 1 & -1 \\
 & & & & -1 & 2
\end{pmatrix}
$$

Note: Sum of rank one submatrices ($u_i = x_{i+1} - x_i$)!

Using the partially separable structure

Very useful for:

- quasi-Newton Hessian matrix = sum of elementwise quasi-Newton low rank submatrices (partitioned updating),
- elementwise models in DFO (number of functions evaluations only dependent of the maximum number of internal variables!),
- optimally efficient finite-difference approximations,
- (structured trust-regions),
- expressing large-scale models.

LANCELOT based on an extension of this concept

Exploitation of the computational tree

Idea: use computational tree for $f(x)$ for solving Newton's equations

- use chain-rule at the top of the computational tree
- multiplicative decompositions (and partially separable)
- often available from the problem modelling itself

Substantial computational gains

unpublished (?) by T. Coleman (2008)

5.3: Multilevel problems

Multilevel Optimization: The Problem

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

- $f : \mathbb{R}^n \to \mathbb{R}$ nonlinear, $\in \mathcal{C}^2$ and bounded below
- No convexity assumption
- 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 critical point x_* (s.t. $\nabla f(x_*) = 0$)

Hierarchy of problem descriptions

Assume now that a hierarchy of problem descriptions is available, linked by known operators

Finest problem description

Restriction $\downarrow$ R $\qquad$ P $\uparrow$ Prolongation

Fine problem description

Restriction $\downarrow$ R $\qquad$ P $\uparrow$ Prolongation

. . .

Restriction $\downarrow$ R $\qquad$ P $\uparrow$ Prolongation

Coarse problem description

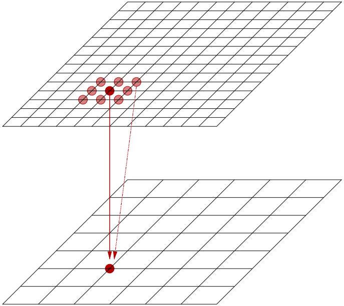
Restriction $\downarrow$ R $\qquad$ P $\uparrow$ Prolongation

Coarsest problem description

Grid transfer operators

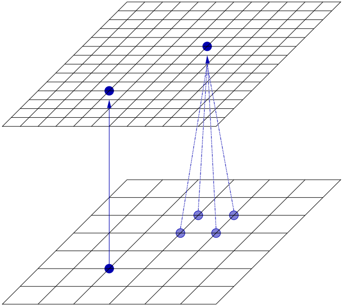
Restriction

$R_i : \mathbb{R}^{n_i} \longrightarrow \mathbb{R}^{n_{i-1}}$



Prolongation

$P_i : \mathbb{R}^{n_{i-1}} \longrightarrow \mathbb{R}^{n_i}$



$$R_i = \sigma P_i^T$$

Sources for such problems

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

The minimum surface problem

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

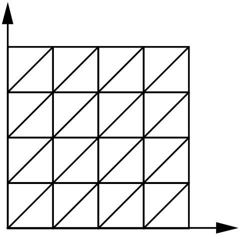
with the boundary conditions:

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

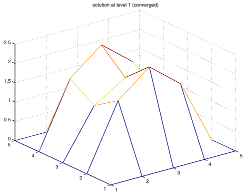
where

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

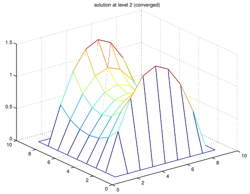
$\rightarrow$ Discretization using a finite element basis



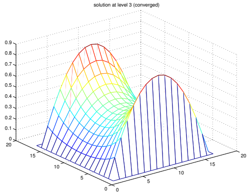
The solution at different levels



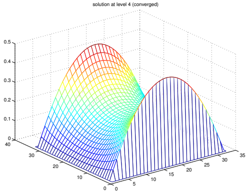
$n = 3^2 = 9$



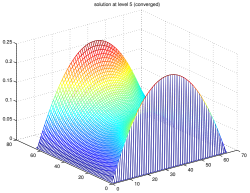
$n = 7^2 = 49$



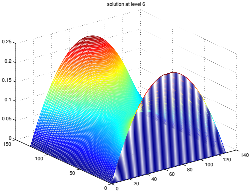
$n = 15^2 = 225$



$n = 31^2 = 961$



$n = 63^2 = 3969$



$n = 127^2 = 16129$

The main issue

Hierarchy of problem descriptions

globalization technique

Efficiency – Robustness

Illustration within a trust-region framework

(Unconstrained case)

Past and recent developments

Line-search

- Fisher (1998), Frese-Bouman-Sauer (1999), Nash (2000)
- Lewis-Nash (2000, 2002)
- Oh-Milstein-Bouman-Webb (2003)
- Wen-Goldfarb (2007, 2008)
- Gratton-T (2007)

Trust-region

- Gratton-Sartenaer-T (2006, 2008)
- Gratton-Mouffe-T-Weber Mendonça (2009)
- Gratton-Mouffe-Sartenaer-T-Tomanos (2009)
- T-Tomanos-Weber Mendonça (2009)
- Gross-Krause (2008)

On the side of multigrid methods

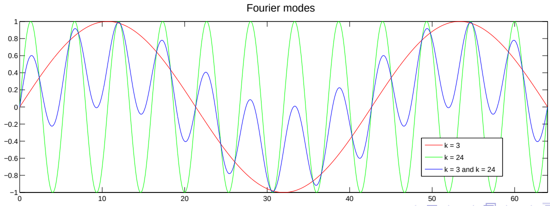
Consider the linear system (discrete Poisson equation, for instance):

$$\boxed{Ax = b} \quad \rightsquigarrow \quad \boxed{Ae = r} \qquad \text{(residual equation)}$$

where

- $e = x_* - \tilde{x}$ (error)
- $r = b - A\tilde{x}$ (residual)
- x_* (exact solution)
- $\tilde{x}$ (approximation)

Expansion of e in Fourier modes shows high (oscillatory) and low (smooth) frequency components:



Relaxation methods

Basic iterative methods:

- correct the i^{th} component of x_k in the order $1, 2, \ldots, n$
- to annihilate the i^{th} component of r_k

Jacobi

$$[x_{k+1}]_i = \frac{1}{a_{ii}} \left(- \sum_{j=1,\ j\neq i}^{n} a_{ij}[x_k]_i + [b]_i \right)$$

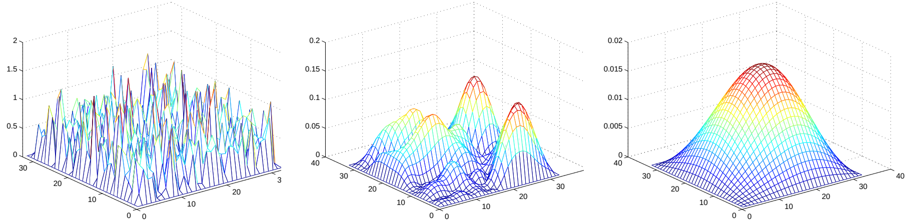
Gauss-Seidel

$$[x_{k+1}]_i = \frac{1}{a_{ii}} \left(- \sum_{j=1}^{i-1} a_{ij}[x_{k+1}]_i - \sum_{j=i+1}^{n} a_{ij}[x_k]_i + [b]_i \right)$$

$\longrightarrow$ Solve the equations of the linear system one by one

Smoothing effect

Very effective methods at "smoothing", i.e., eliminating the high-frequency (oscillatory) components of the error:



error of initial guess | error after 10 GS iterations | error after 100 GS iterations

But they leave the low-frequency (smooth) components relatively unchanged

Multigrid in linear algebar

Assume now (two levels):

- A fine grid (f) description $Ae = r \quad \longrightarrow \quad A^f e^f = r^f$
- A coarse grid (c) description $A^c e^c = r^c$
- Linked by transfer operators $A^c = RA^f P, \quad e^c = Re^f, \quad r^c = Rr^f$

Coarse grid principle

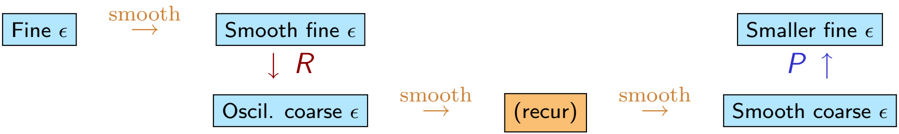
Smooth error modes on a fine grid
"look less smooth" on a coarse grid

⟶ When relaxation begins to stall at the finer level:

- Move to the coarser grid where the smooth error modes appear more oscillatory
- Apply a relaxation at the coarser level:
  - more efficient
  - substantially less expensive

Two-grid correction scheme

Annihilate oscillatory error level by level:



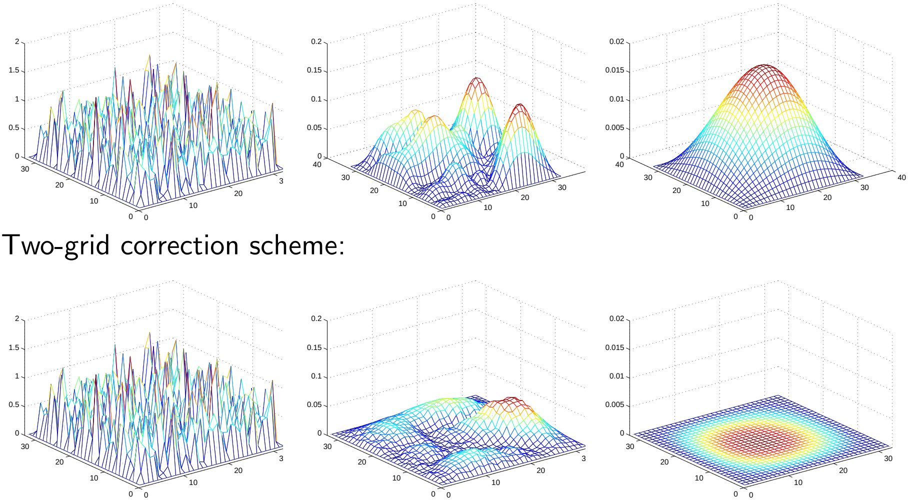
Note: P and R are not othogonal projectors!

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

Does it work?

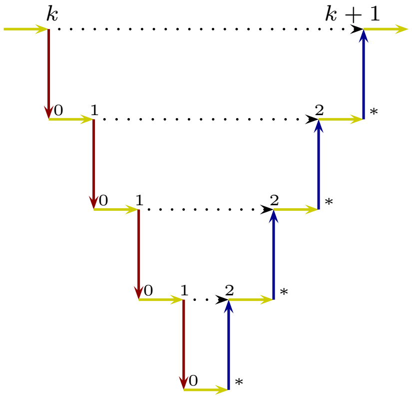
Smoothing on fine grid only:

Two-grid correction scheme:



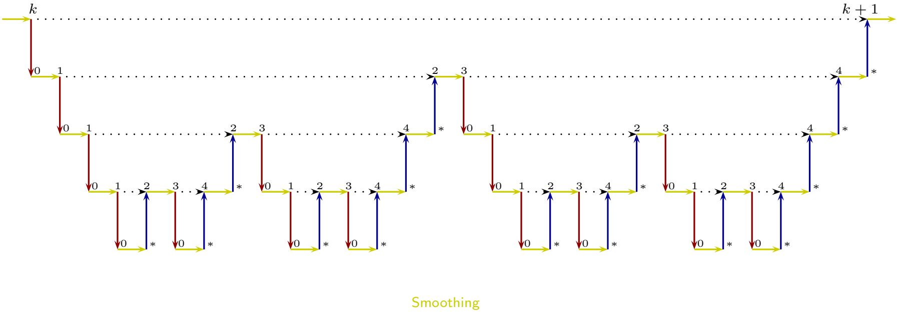
$k = 0$ $k = 10$ $k = 100$

V-cycle



Smoothing

W-cycle



Mesh Refinement

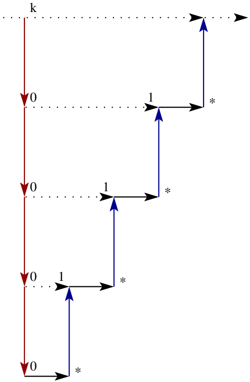
- Solve the problem on the coarsest level

  ⇒ Good starting point for the next fine level

- Do the same on each level

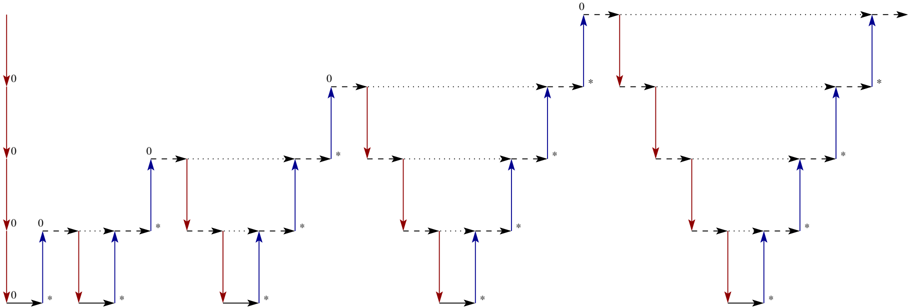
  ⇒ Good starting point for the finest level

- Finally solve the problem on the finest level



Full Multigrid Scheme

Combination of Mesh Refinement and V-cycles



Return to optimization

Hierarchy of problem descriptions ↘ Trust-region technique ↙

Efficiency – Robustness

⇓

Multilevel **optimization** method

Note: Multilevel Moré-Sorensen algorithm: $(H_k + \lambda I)\, s = -g_k$

T-Tomanos-Weber Mendonça, 2009

The framework

Assume that we have:

- A hierarchy of problem descriptions of f:

$$\{f_i\}_{i=0}^r \qquad \text{with} \qquad f_r(x) = f(x)$$

- Transfer operators, for $i = 1, \ldots, r$:
  - $R_i \colon \mathbf{R}^{n_i} \to \mathbf{R}^{n_{i-1}}$ (the restriction)
  - $P_i \colon \mathbf{R}^{n_{i-1}} \to \mathbf{R}^{n_i}$ (the prolongation)

Terminology: a particular i is referred to as a level

The idea

$$\min_{x \in \mathbf{R}^n} f_r(x) = f(x)$$

$\longrightarrow$ at x_k: minimize Taylor's model of f_r around x_k in the trust region of radius Δ_k

$\downarrow$ or (whenever suitable and desirable)

at x_k: compute $\nabla f_r(x_k)$ (possibly H_k)

Restriction $\downarrow$ R

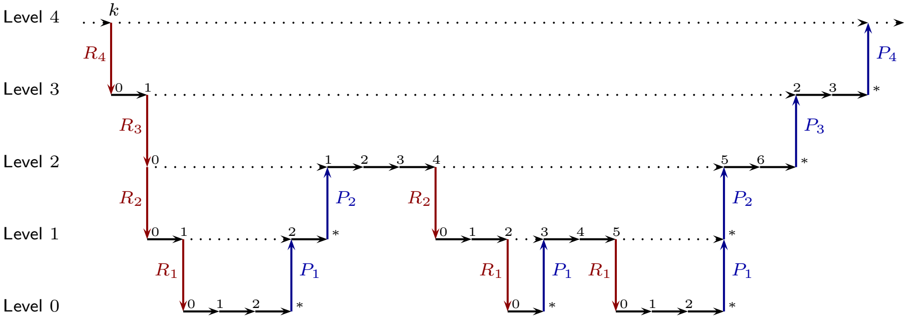
use f_{r-1} to construct a **coarse local** model of f_r and minimize it within the trust region of radius Δ_k

P $\uparrow$ Prolongation

trial step s_k

$\longrightarrow$ If more than two levels are available $(r > 1)$, do this recursively

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



Notation: $\begin{cases} i\text{: level index } (0 \le i \le r) \\ k\text{: index of the current iteration within level } i \end{cases}$

Construction of the coarse local models

If $f_i \neq 0$ for $i = 0, \ldots, r-1$

- Impose first-order coherence via a correction term:

$$g_{\rm low} = R g_{\rm up}$$

- Impose second-order coherence$^{(*)}$ via two correction terms:

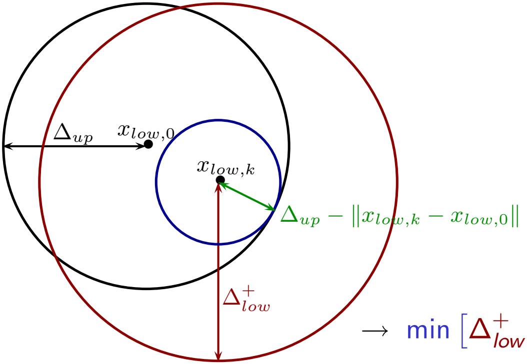
$$g_{\rm low} = R g_{\rm up} \quad \text{and} \quad H_{\rm low} = R H_{\rm up} P$$

$^{(*)}$ Not needed to derive first-order global convergence

If $f_i = 0$ for $i = 0, \ldots, r-1$

- Galerkin model: Restricted version of the quadratic model at the upper level

Preserving the trust-region constraint (1)



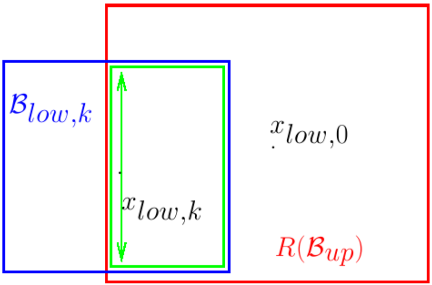
$\longrightarrow \min\left[\Delta^+_{low}, \Delta_{up} - \|x_{low,k} - x_{low,0}\|\right]$

Note: Motivation to switch to ∞-norm

Gratton, Sartenear, T (2008)

Preserving the trust-region constraint (2)

In infinity norm:



$$\min\left[\Delta_{low}^{+},\ \Delta_{up}-\|x_{low,k}-x_{low,0}\|\right]$$

Gratton, Mouffe, T, Weber Mendonça (2008)

Use the coarse model whenever suitable

- When $\|g_{\text{low}}\| \stackrel{\text{def}}{=} \|Rg_{\text{up}}\| \geq \kappa \, \|g_{\text{up}}\|$ ("Coarsening condition")

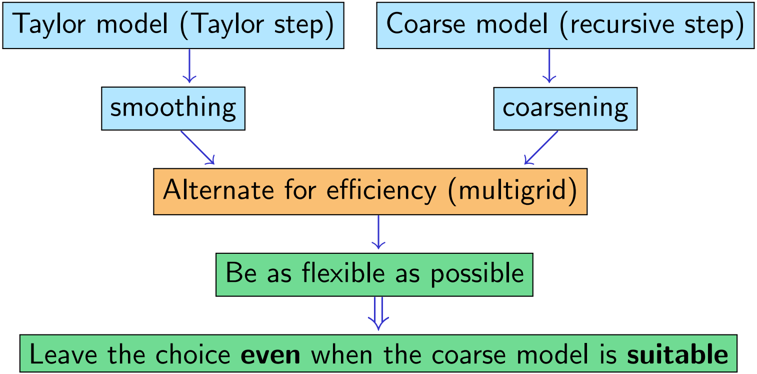
and

- When $\|g_{\text{low}}\| \stackrel{\text{def}}{=} \|Rg_{\text{up}}\| > \epsilon_{low}$

and

- When $i > 0$

Use the coarse model whenever desirable



Recursive multilevel trust-region algorithm (RMTR)

At iteration k (until convergence):

- Choose either a Taylor or (if suitable) a coarse local model (first-order coherent):
  - Taylor model: compute a Taylor step
  - Coarse local model: apply the algorithm recursively
- Evaluate the change in the objective function
- If achieved decrease $\approx$ predicted decrease, then
  - accept the trial point
  - possibly enlarge the trust region

  else

  - keep the current point
  - shrink the trust region
- Impose current trust region $\subseteq$ upper level trust region

Global convergence

Based on the trust-region technology

- Uses the sufficient decrease argument (imposed in Taylor's iterations)
- Plus the coarsening condition $(\|Rg_{\text{up}}\| \geq \kappa \, \|g_{\text{up}}\|)$

Main result

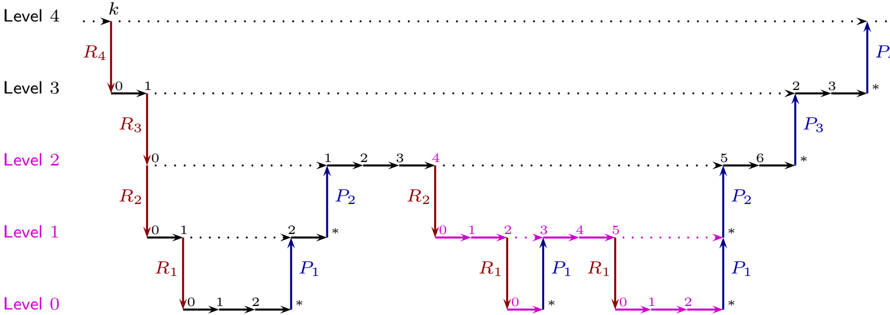
$$\lim_{k \to \infty} \|g_{r,k}\| = 0$$

Gratton, Sartenaer, (2008)

Intermediate results

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

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



Let

$$\mathcal{V}(i,k) \stackrel{\rm def}{=} \{\ (j,\ell) \in \mathcal{R}(i,k) \mid \underbrace{\Delta m_{j,\ell} \geq \kappa \|g_{i,k}\| \Delta_{j,\ell}}_{\text{“sufficient decrease”}} \}$$

Then, at a non critical point and if the trust region is small enough:

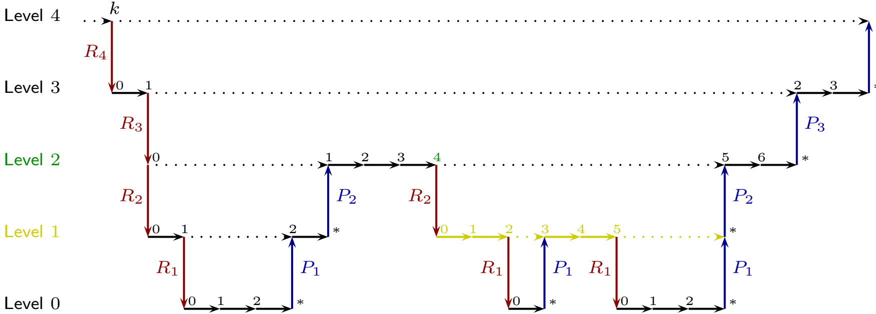
$$\mathcal{V}(i,k) = \mathcal{R}(i,k)$$

$\longrightarrow$ Back to “classical” trust-region arguments

Premature termination

For a recursive iteration (i, k):

A minimization sequence at level $i - 1$ initiated at iteration (i, k) denotes all successive iterations at level $i - 1$ until a return is made to level i



Properties of RMTR

- Each minimization sequence contains at least one successful iteration
- Premature termination in that case does not affect the convergence results at the upper level

Which allows

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

A practical RMTR algorithm: Taylor iterations

At the coarsest level

- Solve using the exact Moré-Sorensen method (small dimension)

At finer levels

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

SCM Smoothing

Adaptation of the Gauss-Seidel smoothing technique to optimization:

- Sequential Coordinate Minimization (SCM smoothing)

> Successive one-dimensional minimizations of the model along the coordinate axes when positive curvature

- 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)

Perform (at most) p SCM smoothing cycles

- while inside the trust region (reasonable cost)

Terminate

- when an approximate minimizer is found (Stop)
- when the trust-region boundary is passed (Stop at the boundary)
- when a direction of negative curvature is encountered (move to the boundary and Stop)

Convergence to weak minimizers

SCM smoothing limits its exploration of the model's curvature to the coordinate axes $\rightarrow$ only guarantees 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 $(i = 0)$

"Weak" minimizers

Gratton, Sartenaer, T (2006)

Some numerical flavor

Gratton, Mouffe, Sartenaer, T, Tomanos (2009)

All on Finest (AF)

Standard Newton trust-region algorithm (TCG)
Applied at the finest level

Multilevel on Finest (MF)

Algorithm RMTR
Applied at the finest level

Mesh Refinement (MR)

Standard Newton trust-region algorithm (TCG)
Applied successively from coarsest to finest level$^{(*)}$

Full Multilevel (FM)

Algorithm RMTR
Applied successively from coarsest to finest level$^{(*)}$

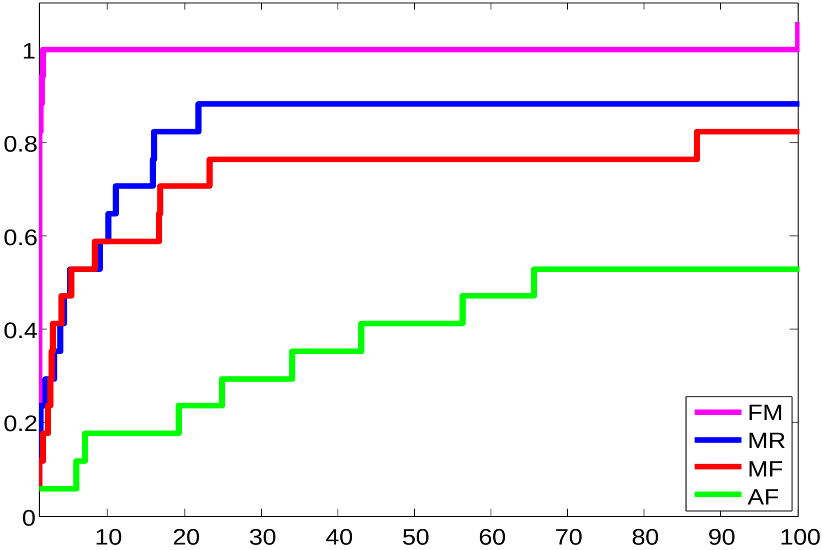
$^{(*)}$ Starting point at level $i+1$ obtained by prolongating the solution at level i

Test problem characteristics

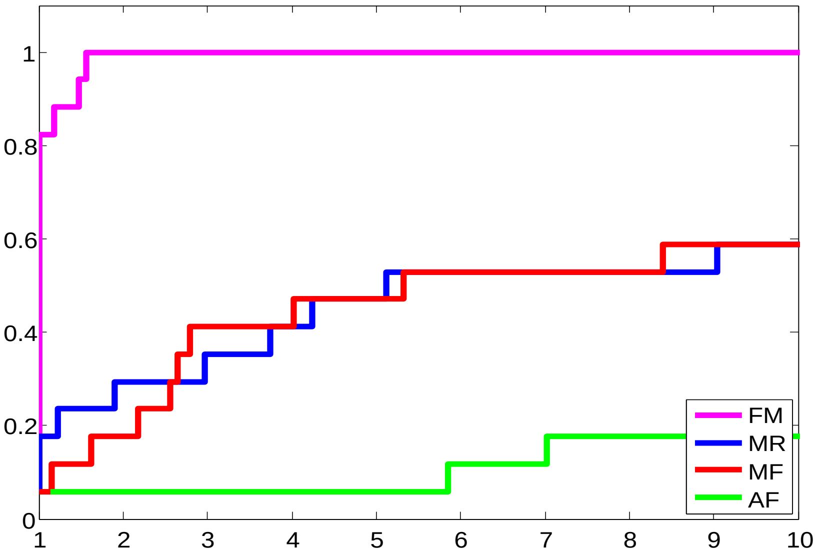
Problem name	n_r	r	Type	Description
DNT	511	8	1-D, quadratic	Dirichlet-to-Neumann transfer problem
P2D	1.046.529	9	2-D, quadratic	Poisson model problem
P3D	250.047	5	3-D, quadratic	Poisson model problem
DEPT	1.046.529	9	2-D, quadratic	Elastic-plastic torsion problem
DPJB*	1.046.529	9	2-D, quadratic	Journal bearing problem
DODC	65.025	7	2-D, convex	Optimal design problem
MINS-SB	1.046.529	9	2-D, convex	Minimium surface problem
MINS-OB	65.025	7	2-D, convex	Minimium surface problem
MINS-DMSA	65.025	7	2-D, convex	Minimium surface problem
IGNISC	65.025	7	2-D, convex	Combustion problem
DSSC	1.046.529	9	2-D, convex	Combustion problem
BRATU	1.046.529	9	2-D, convex	Combustion problem
MINS-BC*	65.025	7	2-D, convex	Minimium surface problem
MEMBR*	393.984	9	2-D, convex	Membrane problem
NCCS	103.050	7	2-D, nonconvex	Optimal control problem
NCCO	103.050	7	2-D, nonconvex	Optimal control problem
MOREBV	1.046.529	9	2-D, nonconvex	Boundary value problem

$\star$: with bound constraints

Performance profiles (CPU time)



Zoom on on efficiency (CPU time)

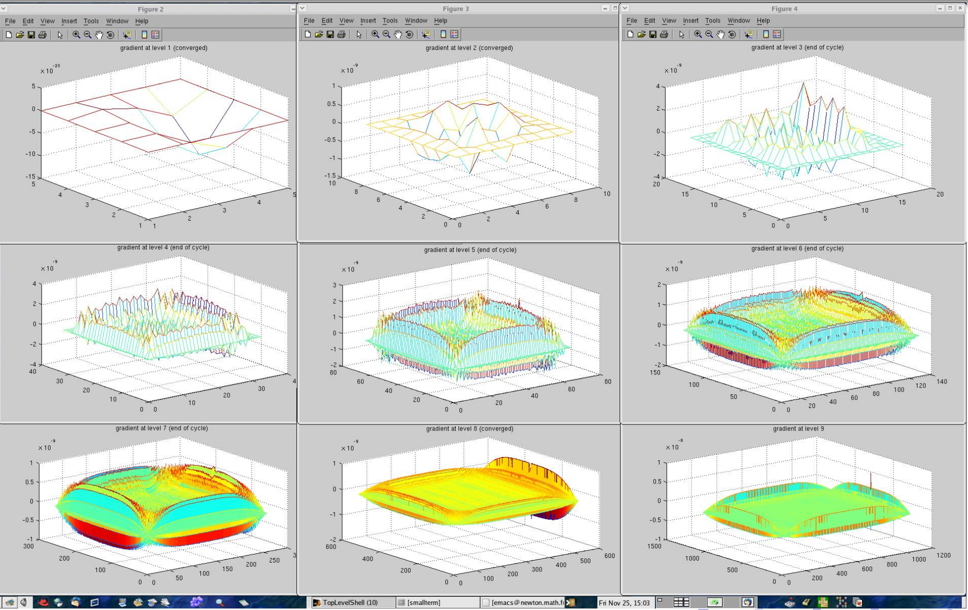


CPU times

Problem name	AF	MF	MR	FM
DNT	5.2	24.4	4.6	6.7
P2D	1122.8	72.8	569.7	26.0
P3D	626.1	47.5	18.3	28.8
DEPT	1364.4	69.5	95.4	8.6
DPJB	3600.0	1390.0	247.7	83.6
DODC	894.8	58.6	184.2	36
MINS-SB	3600.0	3600.0	3600.0	153.9
MINS-OB	1445.6	70.4	116.7	27.5
MINS-DMSA	1196.8	73.4	289.6	18.2
IGNISC	2330.4	398.3	488.2	398.2
DSSC	3183.8	1051.6	122.3	12.1
BRATU	2314.1	236.8	91.7	10.1
MINS-BC	2706.4	161.8	524.6	140.0
MEMBR	1082.0	335.2	292.4	154.0
NCCS	3600.0	3600.0	279.5	331.9
NCCO	3600.0	3600.0	3589.6	224.2
MOREBV	3600.0	704.9	3600.0	41.7

Best   Second best

A glimpse at the solution process



Bibliography for lesson 5 (1)

1. T. F. Coleman and J. J. Moré,
Estimation of sparse Jacobian matrices and graph coloring problems,
SIAM Journal on Numerical Analysis, 20:187-209, 1983.
2. T. F. Coleman and J. J. Moré,
Estimation of sparse Hessian matrices and graph coloring problems,
Mathematical Programming, 28:243-270, 1984.
3. A. R. Conn, N. I. M. Gould and Ph. L. Toint,
LANCELOT**: a Fortran package for large-scale nonlinear optimization (Release A)**,
Springer Verlag, Springer Series in Computational Mathematics 17, Heidelberg, 1992.
4. A. Curtis, M. J. D. Powell and J. Reid,
On The Estimation of Sparse Jacobian Matrices,
IMA Journal, 13:117-119, 1974.
5. M. Fisher,
Minimization Algorithms for Variational Data Assimilation,
in "Recent Developments in Numerical Methods for Atmospheric Modelling", European Center for Medium-Range Weather Forecasts, Reading, UK, pp. 364-385, 1998.
6. T. Frese, Ch. Bouman and K. Sauer,
Multiscale Bayesian Methods for Discrete Tomography,
in "Discrete Tomography: Foundations, Algorithms and Applications" (G. Herman and A. Kuba, eds.), Birkhauser, Boston, pp. 237-261, 1999.
7. D. Goldfarb and Ph. L. Toint,
Optimal Estimation of Jacobian and Hessian Matrices That Arise in Finite Difference Calculations,
Mathematics of Computation, 43(167):69-88, 1984.
8. S. Gratton, M. Mouffe, Ph. L. Toint and M. Weber-Mendonça,
A recursive trust-region method in infinity norm for bound-constrained nonlinear optimization,
IMA Journal of Numerical Analysis, 28(4):827-861, 2008.
9. S. Gratton, M. Mouffe, A. Sartenaer, Ph. L. Toint and D. Tomanos,
Numerical Experience with a Recursive Trust-Region Method for Multilevel Nonlinear Optimization,
Optimization Methods and Software, to appear, 2009.

Bibliography for lesson 5 (2)

10 S. Gratton, A. Sartenaer and Ph. L. Toint,
Recursive Trust-Region Methods for Multiscale Nonlinear Optimization, SIAM Journal on Optimization, 19(1):414-444, 2008.

11 S. Gratton, A. Sartenaer and Ph. L. Toint,
Second-order convergence properties of trust-region methods using incomplete curvature information, with an application to multigrid optimization,
Journal of Computational and Applied Mathematics, 24(6):676-692, 2006.

12 S. Gratton and Ph. L. Toint,
Multi-Secant Equations, Approximate Invariant Subspaces and Multigrid Optimization,
FUNDP, Namur, Report 07/11, 2007.

13 A. Griewank and Ph. L. Toint,
On the unconstrained optimization of partially separable functions,
in "Ninlinear Optimization 1981", (M. J. D. Powell, ed.), Academic Press, pp. 302-312, 1982.

14 S. G. Nash,
A Multigrid Approach to Discretized Optimization Problems,
Optimization Methods and Software, 14:99-116, 2000.

15 M. Lewis and S. G. Nash,
Model problems for the multigrid optimization of systems governed by differential equations,
SIAM Journal on Scientific Computing, 26(6):1811-1837, 2005.

16 S. Oh, A. Milstein, Ch. Bouman and K. Webb,
Multigrid algorithms for optimization and inverse problems,
in "Computational Imaging"(Ch. Bouman and R. Stevenson, eds.), DDM, Proceedings of the SPIE, 5016:59-70, 2003.

17 M. J. D. Powell and Ph. L. Toint,
On The Estimation of Sparse Hessian Matrices,
SIAM Journal on Numerical Analysis, 16(6):1060-1074, 1979.

18 Ph. L. Toint, D. Tomanos and M. Weber-Mendonça,
A multilevel algorithm for solving the trust-region subproblem,
Optimization Methods and Software, 24(2):299-311, 2009.

19 Ch. Gross and R. Krause,
On The Convergence of Recursive Trust-Region Methods for Multiscale Nonlinear Optimization and Applications to Nonlinear Mechanics,
University of Bonn, Germany, 2008.

Bibliography for lesson 5 (3)

20 Z. Wen and D. Goldfarb,
A Linesearch Multigrid Methods for Large-Scale Convex Optimization,
Department of Industrial Engineering and Operations Research, Columbia University, New York, July 2007.

Lesson 6:

Cubic and quadratic regularization methods: a path towards nonlinear step control

Outline

1. Regularization for unconstrained problems
    1. cubic
    2. quadratic
2. Nonlinear step control
3. Cubic regularization for constrained problems
4. Conclusions
5. Bibliography

Regularization techniques for unconstrained Problems

The problem

We return to the unconstrained nonlinear programming problem:

$$\text{minimize} \quad f(x)$$

for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ smooth.

Important special case: the nonlinear least-squares problem

$$\text{minimize} \quad f(x) = \tfrac{1}{2}\|F(x)\|^2$$

for $x \in \mathbb{R}^n$ and $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ smooth.

Unconstrained optimization — a "mature" area?

$$\underset{x \in \mathbf{R}^n}{\text{minimize}} \quad f(x) \quad \text{where} \quad f \in C^1 \quad (\text{maybe} \quad C^2 \;)$$

Currently two main competing (but similar) methodologies

- **Linesearch methods**
  - compute a descent direction s_k from x_k
  - set $x_{k+1} = x_k + \alpha_k s_k$ to improve f
- **Trust-region methods**
  - compute a step s_k from x_k to improve a model m_k of f within the trust-region $\|s_k\| \leq \Delta$
  - set $x_{k+1} = x_k + s_k$ if m_k and f "agree" at $x_k + s_k$
  - otherwise set $x_{k+1} = x_k$ and reduce the radius Δ

A useful theoretical observation

Consider trust-region method where

model = true objective function

Then

- model and objective always agree
- trust-region radius goes to infinity

$\Rightarrow$ a linesearch method

Nice consequence:

A unique convergence theory!

(Shultz/Schnabel/Byrd, 1985, T., 1988

The keys to convergence theory for trust regions

The Cauchy condition:

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\rm TR} \|g_k\| \min \left[\frac{\|g_k\|}{1 + \|H_k\|}, \Delta_k \right]$$

The bound on the stepsize:

$$\|s\| \leq \Delta$$

And we derive:

Global convergence to first/second-order critical points

Is there anything more to say?

Is there anything more to say?

Observe the following: if

- f has gradient g and globally Lipschitz continuous Hessian H with constant $2L$

Taylor, Cauchy-Schwarz and Lipschitz imply

$$\begin{aligned} f(x+s) &= f(x) + \langle s, g(x)\rangle + \tfrac{1}{2}\langle s, H(x)s\rangle \\ &\quad + \int_0^1 (1-\alpha)\langle s, [H(x+\alpha s) - H(x)]s\rangle \, d\alpha \\ &\leq \underbrace{f(x) + \langle s, g(x)\rangle + \tfrac{1}{2}\langle s, H(x)s\rangle + \tfrac{1}{3}L\|s\|_2^3}_{m(s)} \end{aligned}$$

$\Longrightarrow$ reducing m from $s = 0$ improves f since $m(0) = f(x)$.

The cubic regularization

Change from

$$\min_s \quad f(x) + \langle s, g(x)\rangle + \tfrac{1}{2}\langle s, H(x)s\rangle \;\; \text{s.t.} \;\; \|s\| \le \Delta$$

to

$$\min_s \quad f(x) + \langle s, g(x)\rangle + \tfrac{1}{2}\langle s, H(x)s\rangle + \tfrac{1}{3}\sigma\|s\|^3$$

σ is the (adaptive) regularization parameter

(ideas from Griewank, Weiser/Deuflhard/Erdmann, Nesterov/Polyak, Cartis/Gould/T)

Cubic regularization highlights

$$f(x+s) \leq m(s) \equiv f(x) + s^T g(x) + \tfrac{1}{2} s^T H(x) s + \tfrac{1}{3} L \|s\|_2^3$$

- Nesterov and Polyak minimize m globally
  - N.B. m may be non-convex!
  - efficient scheme to do so if H has sparse factors
- global (ultimately rapid) convergence to a 2nd-order critical point of f
- better worst-case function-evaluation complexity than previously known

Obvious questions:

- can we avoid the global Lipschitz requirement?
- can we approximately minimize m and retain good worst-case function-evaluation complexity?
- does this work well in practice?

Cubic overestimation

Assume

- $f \in C^2$
- f, g and H at x_k are f_k, g_k and H_k
- symmetric approximation B_k to H_k
- B_k and H_k bounded at points of interest

Use

- cubic overestimating model at x_k

$$m_k(s) \equiv f_k + s^T g_k + \tfrac{1}{2} s^T B_k s + \tfrac{1}{3} \sigma_k \|s\|_2^3$$

  - σ_k is the iteration-dependent regularisation weight
  - easily generalized for regularisation in M_k-norm $\|s\|_{M_k} = \sqrt{s^T M_k s}$ where M_k is uniformly positive definite

Adaptive Regularization with Cubic (ARC)

Algorithm 6.1: The ARC Algorithm

Step 0: Initialization: x_0 and $\sigma_0 > 0$ given. Set $k = 0$

Step 1: Step computation: Compute s_k for which $m_k(s_k) \le m_k(s_k^C)$

Cauchy point: $s_k^C = -\alpha_k^C g_k$ & $\alpha_k^C = \arg\min_{\alpha \in \mathbf{R}_+} m_k(-\alpha g_k)$

Step 2: Step acceptance: Compute $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - m_k(s_k)}$

and set $x_{k+1} = \begin{cases} x_k + s_k & \text{if } \rho_k > 0.1 \\ x_k & \text{otherwise} \end{cases}$

Step 3: Update the regularization parameter:

$$\sigma_{k+1} \in \begin{cases} (0, \sigma_k] = \frac{1}{2}\sigma_k & \text{if } \rho_k > 0.9 & \textit{very successful} \\ [\sigma_k, \gamma_1 \sigma_k] = \sigma_k & \text{if } 0.1 \le \rho_k \le 0.9 & \textit{successful} \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] = 2\sigma_k & \text{otherwise} & \textit{unsuccessful} \end{cases}$$

Local convergence theory for cubic regularization (1)

The Cauchy condition:

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\rm CR} \|g_k\| \min \left[\frac{\|g_k\|}{1 + \|H_k\|}, \sqrt{\frac{\|g_k\|}{\sigma_k}} \right]$$

The bound on the stepsize:

$$\|s_k\| \leq 3 \max \left[\frac{\|H_k\|}{\sigma_k}, \sqrt{\frac{\|g_k\|}{\sigma_k}} \right]$$

(Cartis/Gould/T)

Local convergence theory for cubic regularization (2)

And therefore. . .

$$\lim_{k\to\infty} \|g_k\| = 0$$

first-order global convergence

Under stronger assumptions can show that

If s_k minimizes m_k over subspace with orthogonal basis Q_k,

$$\lim_{k\to\infty} Q_k^T H_k Q_k \succeq 0$$

second-order global convergence

Fast convergence

For fast asymptotic convergence $\Longrightarrow$ need to improve on Cauchy point: minimize over Krylov subspaces

- g stopping-rule: $\|\nabla_s m_k(s_k)\| \leq \min(1, \|g_k\|^{\frac{1}{2}})\|g_k\|$
- s stopping-rule: $\|\nabla_s m_k(s_k)\| \leq \min(1, \|s_k\| \quad)\|g_k\|$

If B_k satisfies the Dennis-Moré condition

$$\|(B_k - H_k)s_k\|/\|s_k\| \to 0 \;\text{ whenever }\; \|g_k\| \to 0$$

and $x_k \to x_*$ with positive definite $H(x_*)$

$\Longrightarrow$ Q-superlinear convergence of x_k under the g- and s-rules

If additionally $H(x)$ is locally Lipschitz around x_* and

$$\|(B_k - H_k)s_k\| = O(\|s_k\|^2)$$

$\Longrightarrow$ Q-quadratic convergence of x_k under the s-rule

Function-evaluation complexity

How many function evaluations (iterations) are needed to ensure that

$$\|g_k\| \leq \epsilon?$$

- so long as for very successful iterations $\sigma_{k+1} \leq \gamma_3 \sigma_k$ for $\gamma_3 < 1$
  $\Longrightarrow$ basic ARC algorithm requires at most

  $$\left\lceil \frac{\kappa_{\rm C}}{\epsilon^2} \right\rceil \text{ function evaluations}$$

  for some $\kappa_{\rm C}$ independent of ϵ c.f. steepest descent
- if H is globally Lipschitz, the s-rule is applied and additionally s_k is the global (line) minimizer of $m_k(\alpha s_k)$ as a function of α
  $\Longrightarrow$ ARC algorithm requires at most

  $$\left\lceil \frac{\kappa_{\rm S}}{\epsilon^{3/2}} \right\rceil \text{ function evaluations}$$

  for some $\kappa_{\rm S}$ independent of ϵ c.f. Nesterov & Polyak

Minimizing the model

$$m(s) \equiv f + s^T g + \frac{1}{2} s^T B s + \frac{1}{3}\sigma \|s\|_2^3$$

Derivatives:

- $\lambda = \sigma \|s\|_2$
- $\nabla_s m(s) = g + Bs + \lambda s$
- $\nabla_{ss} m(s) = B + \lambda I + \lambda \left(\frac{s}{\|s\|} \right) \left(\frac{s}{\|s\|} \right)^T$

Optimality: any global minimizer s_* of m satisfies

$$(B + \lambda_* I) s_* = -g$$

- $\lambda_* = \sigma \|s_*\|_2$
- $B + \lambda_* I$ is positive semi-definite

The (adapted) secular equation

Require

$$(B + \lambda I)s = -g \quad \text{and} \quad \lambda = \sigma \|s\|_2$$

Define $s(\lambda)$:

$$(B + \lambda I)s(\lambda) = -g$$

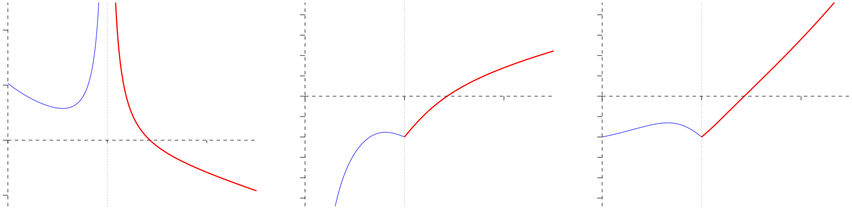
and find scalar λ as the root of secular equations

$$\|s(\lambda)\|_2 - \frac{\lambda}{\sigma} = 0 \quad \text{or} \quad \frac{1}{\|s(\lambda)\|_2} - \frac{\sigma}{\lambda} = 0 \quad \text{or} \quad \frac{\lambda}{\|s(\lambda)\|_2} - \sigma = 0$$

- values and derivatives of $s(\lambda)$ satisfy linear systems with symmetric positive definite $B + \lambda I$
- need to be able to factorize $B + \lambda I$

Plots of secular functions against λ

Example: $g = (0.25 \;\; 1)^T$, $H = \text{diag}(-1 \;\; 1)$ and $\sigma = 2$



$$\|s(\lambda)\|_2 - \frac{\lambda}{\sigma} = 0 \qquad \frac{1}{\|s(\lambda)\|_2} - \frac{\sigma}{\lambda} = 0 \qquad \frac{\lambda}{\|s(\lambda)\|_2} - \sigma = 0$$

Large problems — approximate solutions

Seek instead global minimizer of $m(s)$ in a j-dimensional ($j \ll n$) subspace $\mathcal{S} \subseteq \mathbb{R}^n$

- $g \in \mathcal{S} \implies$ ARC algorithm globally convergent
- Q orthogonal basis for $\mathcal{S} \quad \implies \quad s = Qu$ where

$$u = \arg\min_{u \in \mathbb{R}^j} f + u^T(Q^T g) + \tfrac{1}{2} u^T (Q^T B Q) u + \tfrac{1}{3} \|u\|_2^3$$

  $\implies$ use secular equation to find u

- if $\mathcal{S}$ is the Krylov space generated by $\{B^i g\}_{i=0}^{j-1}$
  $\implies Q^T B Q = T$, tridiagonal
  $\implies$ can factor $T + \lambda I$ to solve secular equation even if j is large
- using g- or s-stopping rule $\implies$ fast asymptotic convergence for ARC
- using s-stopping rule $\implies$ good function-evaluation complexity for ARC

The main features of adaptive cubic regularization

And the result is. . .

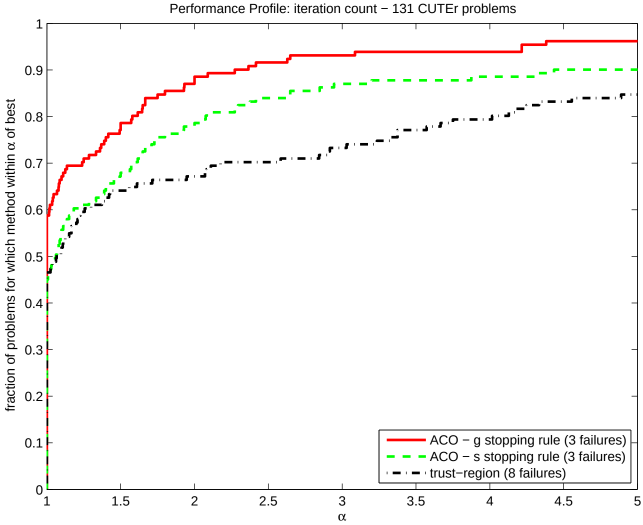
longer steps on ill-conditioned problems

similar (very satisfactory) convergence analysis

best function-evaluation complexity for nonconvex problems

excellent performance and reliability

Numerical experience — small problems using Matlab



The quadratic regularization for NLS (ARQ)

Consider the Gauss-Newton method for nonlinear least-squares problems. Change from

$$\min_s \quad \tfrac{1}{2}\|c(x)\|^2 + \langle s, J(x)^T c(x)\rangle + \tfrac{1}{2}\langle s, J(x)^T J(x) s\rangle \;\text{ s.t. }\; \|s\| \le \Delta$$

to

$$\min_s \quad \|c(x) + J(x)s\| + \tfrac{1}{2}\sigma\|s\|^2$$

σ is the (adaptive) regularization parameter

(idea by Nesterov)

Quadratic regularization: reformulation

Note that

$$\min_s \quad \|c(x) + J(x)s\| + \tfrac{1}{2}\sigma\|s\|^2$$

$$\Leftrightarrow$$

$$\min_{\nu,s} \quad \nu + \tfrac{1}{2}\sigma\|s\|^2 \quad \text{such that} \quad \|c(x) + J(x)s\|^2 = \nu^2$$

exact penalty function for the problem of minimizing $\|s\|$ subject to $c(x) + J(x)s = 0$.

Iterative techniques. . . as for the cubic case (Cartis, Gould, T.):

solve the problem in nested Krylov subspaces

- Lanczos $\rightarrow$ factorization of tridiagonal matrices
- different scalar secular equation (solution by Newton's method)

The keys to convergence theory for quadratic regularization

The Cauchy condition:

$$m(x_k) - m(x_k + s_k) \geq \kappa_{\rm QR} \frac{\|J_k^T c_k\|}{\|c_k\|} \min\left[\frac{\|J_k^T c_k\|}{1 + \|J_k^T J_k\|}, \frac{\|J_k^T c_k\|}{\sigma_k \|c_k\|}\right]$$

The bound on the stepsize:

$$\|s_k\| \leq \frac{1}{2} \frac{\|J_k^T c_k\|}{\sigma_k \|c_k\|}$$

Convergence theory for the quadratic regularization

Convergence results:

Global convergence to first-order critical points

Quadratic convergence to roots

Valid for

- general values of m and n,
- exact/approximate subproblem solution

(Bellavia/Cartis/Gould/Morini/T.)

6.2: A unifying concept: nonlinear stepsize control

Towards a unified global convergence theory

Objectives:

- recover a unified global convergence theory
- possibly open the door for new algorithms

Idea:

- cast all three methods into a generalized TR framework
- allow this TR to be updated nonlinearly

Towards a unified global convergence theory (2)

Given

- 3 continuous first-order criticality measures $\psi(x)$, $\phi(x)$, $\chi(x)$
- an adaptive stepsize parameter δ

define a generalized radius $\Delta(\delta, \chi(x))$ such that

- $\Delta(\cdot, \chi)$ is C^1, strictly increasing and concave,
- $\Delta(0, \chi) = 0$ for all χ,
- $\Delta(\delta, \cdot)$ is non-increasing
- $$\delta \frac{\partial \Delta}{\partial \delta}(\delta, \chi) \leq \kappa_\Delta \Delta(\delta, \chi)$$
- ...

6.3: Cubic regularization for constrained problems

The constrained case

Can we apply regularization to the constrained case?

Consider the constrained nonlinear programming problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ & x \in \mathcal{F} \end{aligned}$$

for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ smooth, and where

$$\mathcal{F} \text{ is convex.}$$

Main ideas:

- exploit (cheap) projections on convex sets
- define using the generalized Cauchy point idea
- prove global convergence + function-evaluation complexity

Constrained step computation (1)

$$\min_s \quad f(x) + \langle s, g(x)\rangle + \tfrac{1}{2}\langle s, H(x)s\rangle + \tfrac{1}{3}\sigma\|s\|^3$$

subject to

$$x + s \in \mathcal{F}$$

σ is the (adaptive) regularization parameter

Criticality measure: (as before)

$$\chi(x) \stackrel{\rm def}{=} \left| \min_{x+d\in\mathcal{F}, \|d\|\leq 1} \langle \nabla_x f(x), d\rangle \right|,$$

The generalized Cauchy point for ARC

Cauchy step: Goldstein-like piecewise linear seach on m_k along the gradient path projected onto $\mathcal{F}$

Find

$$x_k^{\rm GC} = P_{\mathcal{F}}[x_k - t_k^{\rm GC} g_k] \stackrel{\rm def}{=} x_k + s_k^{\rm GC} \quad (t_k^{\rm GC} > 0)$$

such that

$$m_k(x_k^{\rm GC}) \leq f(x_k) + \kappa_{\rm ubs}\langle g_k, s_k^{\rm GC}\rangle \qquad \text{(below linear approximation)}$$

and either

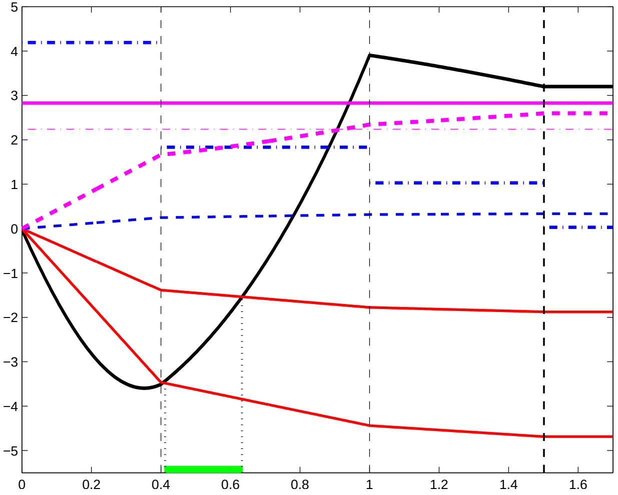
$$m_k(x_k^{\rm GC}) \geq f(x_k) + \kappa_{\rm lbs}\langle g_k, s_k^{\rm GC}\rangle \qquad \text{(above linear approximation)}$$

or

$$\|P_{T(x_k^{\rm GC})}[-g_k]\| \leq \kappa_{\rm epp}|\langle g_k, s_k^{\rm GC}\rangle| \qquad \text{(close to path's end)}$$

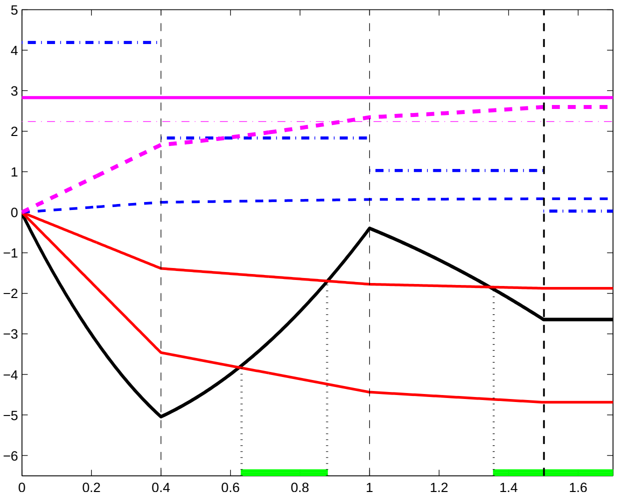
no trust-region condition!

Searching for the ARC-GCP



$$m_k(0+s) = -3.57s_1 - 1.5s_2 - s_3 + s_1s_2 + 3s_2^2 + s_2s_3 - 2s_3^2 + \frac{1}{3}\|s\|^3 \quad \text{such that} \quad s \leq 1.5$$

Remember the same for a quadratic model?



$$m_k(0+s) = -3.57s_1 - 1.5s_2 - s_3 + s_1 s_2 + 3s_2^2 + s_2 s_3 - 2s_3^2 \text{ such that } s \le 1.5 \text{ and } \Delta \le 2.8$$

A constrained regularized algorithm

Algorithm 6.2: ARC for Convex Constraints (COCARC)

Step 0: Initialization. $x_0 \in \mathcal{F}$, σ_0 given. Compute $f(x_0)$, set $k = 0$.

Step 1: Generalized Cauchy point. If x_k not critical, find the generalized Cauchy point $x_k^{\rm GC}$ by piecewise linear search on the regularized cubic model.

Step 2: Step calculation. Compute s_k and $x_k^+ \stackrel{\rm def}{=} x_k + s_k \in \mathcal{F}$ such that $m_k(x_k^+) \leq m_k(x_k^{\rm GC})$.

Step 3: Acceptance of the trial point. Compute $f(x_k^+)$ and ρ_k. If $\rho_k \geq \eta_1$, then $x_{k+1} = x_k + s_k$; otherwise $x_{k+1} = x_k$.

Step 4: Regularisation parameter update. Set

$$\sigma_{k+1} \in \begin{cases} (0, \sigma_k] & \text{if } \rho_k \geq \eta_2, \\ [\sigma_k, \gamma_1\sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1\sigma_k, \gamma_2\sigma_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Local convergence theory for COCARC

The Cauchy condition:

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\rm CR} \chi_k \min\left[\frac{\chi_k}{1 + \|H_k\|}, \sqrt{\frac{\chi_k}{\sigma_k}}, 1\right]$$

The bound on the stepsize:

$$\|s_k\| \leq 3 \max\left[\frac{\|H_k\|}{\sigma_k}, \left(\frac{\chi_k}{\sigma_k}\right)^{\frac{1}{2}}, \left(\frac{\chi_k}{\sigma_k}\right)^{\frac{1}{3}}\right]$$

And therefore...

$$\lim_{k\to\infty} \chi_k = 0$$

(Cartis/Gould/T)

Function-Evaluation Complexity for COCARC (1)

But

What about function-evaluation complexity?

If, for very successful iterations, $\sigma_{k+1} \leq \gamma_3 \sigma_k$ for $\gamma_3 < 1$, the COCARC algorithm requires at most

$$\left\lceil \frac{\kappa_{\rm C}}{\epsilon^2} \right\rceil \text{ function evaluations}$$

(for some $\kappa_{\rm C}$ independent of ϵ) to achieve $\chi_k \leq \epsilon$

c.f. steepest descent

Do the nicer bounds for unconstrained optimization extend to the constrained case?

Function-evaluation complexity for COCARC (2)

As for unconstrained, impose a termination rule on the subproblem solution:

- Do not terminate solving $\min_{x_k+s\in\mathcal{F}} m_k(x_k+s)$ before

$$\chi_k^{\mathrm{m}}(x_k^+) \leq \min(\kappa_{\text{stop}}, \|s_k\|)\, \chi_k$$

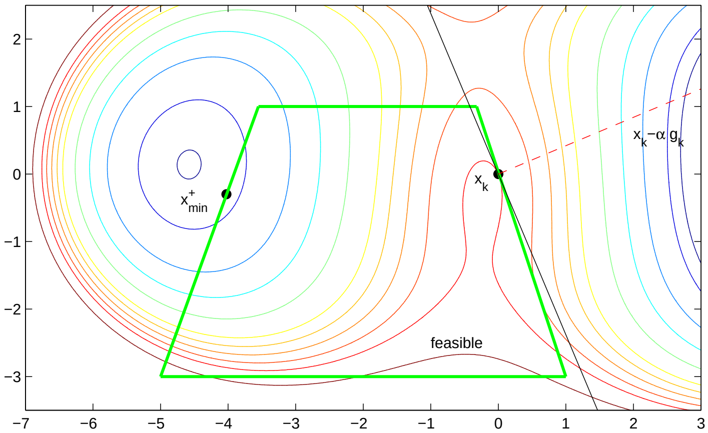
  where

$$\chi_k^{\mathrm{m}}(x) \stackrel{\text{def}}{=} \left| \min_{x+d\in\mathcal{F}, \|d\|\leq 1} \langle \nabla_x m_k(x), d \rangle \right|$$

c.f. the "s-rule" for unconstrained

Note: OK at local constrained model minimizers

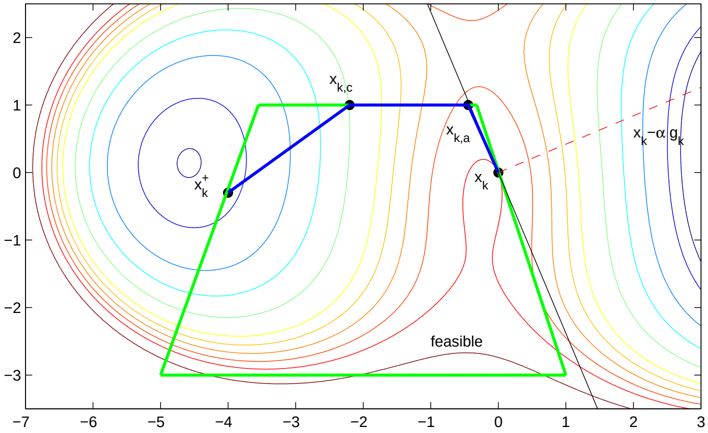
Walking through the pass...



A "beyond the pass" constrained problem with

$$m(x,y) = -x - \tfrac{42}{100}y - \tfrac{3}{10}x^2 - \tfrac{1}{10}y^3 + \tfrac{1}{3}[x^2+y^2]^{\frac{3}{2}}$$

Walking through the pass...with a sherpa



A piecewise descent path from x_k to x_k^+ on

$$m(x,y) = -x - \tfrac{42}{100}y - \tfrac{3}{10}x^2 - \tfrac{1}{10}y^3 + \tfrac{1}{3}[x^2+y^2]^{\frac{3}{2}}$$

Function-Evaluation Complexity for COCARC (2)

Assume also

- $x_k \leftarrow x_k^+$ in a bounded number of feasible descent substeps
- $\|H_k - \nabla_{xx} f(x_k)\| \leq \kappa \|s_k\|^2$
- $\nabla_{xx} f(\cdot)$ is globally Lipschitz continuous
- $\{x_k\}$ bounded

The COCARC algorithm requires at most

$$\left\lceil \frac{\kappa_{\rm C}}{\epsilon^{3/2}} \right\rceil \text{ function evaluations}$$

(for some $\kappa_{\rm C}$ independent of ϵ) to achieve $\chi_k \leq \epsilon$

Caveat: cost of solving the subproblem c.f. unconstrained case!!!

Conclusions for lesson 6

- Much left to do. . . but very interesting
- Unconstrained nonliear stepsize control could lead to very untypical methods. Example:

$$\psi_k = \phi_k = \chi_k = \|g_k\|, \qquad \Delta(\delta, \chi) = \sqrt{\delta\chi}$$

- Meaningful numerical evaluation still needed for many of these algorithms
- Many issues regarding regularizations still unresolved

Bibliography for lesson 6

- R. H. Byrd, R. B. Schnabel and G. A. Shultz,
  A trust region algorithm for nonlinearly constrained optimization,
  SIAM Journal on Numerical Analysis, 24: 1152–1170, 1987.
- Ph. L. Toint,
  Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space,
  IMA Journal of Numerical Analysis, 8(2): 231–252, 1988.
- A. Griewank,
  The modification of Newton's method for unconstrained optimization by bounding cubic terms,
  Department of Applied Mathematics and Theoretical Physics, University of Cambridge (UK), Report NA/12, 1981.
- M. Weiser, P. Deuflhard and B. Erdmann,
  Affine conjugate adaptive Newton methods for nonlinear elastomechanics,
  Optimization Methods and Software, 22(3): 413–431, 2007.
- Yu. Nesterov and B. T. Polyak,
  Cubic regularization of Newton method and its global performance,
  Mathematical Programming, 108(1): 177-205, 2006.
- C. Cartis and N. I. M. Gould and Ph. L. Toint,
  Adaptive cubic overestimation methods for unconstrained optimization,
  FUNDP, Namur, Report 07/05, 2007.
- C. Cartis, N. I. M. Gould and Ph. L. Toint,
  Trust-region and other regularisation of linear least-squares problems,
  BIT, to appear, 2009.
- Yu. Nesterov,
  Modified Gauss-Newton scheme with worst-case guarantees for global performance,
  Optimization Methods and Software, 22(3): 469–483, 2007.
- S. Bellavia, C. Cartis, N. I. M. Gould, B. Morini and Ph. L. Toint,
  Convergence of a Regularized Euclidean Residual Algorithm for Nonlinear Least-Squares,
  FUNDP, Namur, Report 08/11, 2008.
- C. Cartis, N. I. M. Gould and Ph. L. Toint,
  An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity,
  FUNDP, Namur, Report 08/05R, 2009.

Not covered in the course

- non-smooth techniques
- specifically convex problems
- penalty functions
- augmented Lagrangians
- affine scaling methods
- general sequential quadratic programming (SQP)
- systems of nonlinear equations
- . . .

Many thanks to you all for your patience!