Derivative-Free Optimization: Basic Principles and State of the Art

Luís Nunes Vicente, University of Coimbra

Vienna Graduate School On Computational Optimization

November 27 – December 1, 2017, Universität Wien

http//www.mat.uc.pt/~lnv

Presentation outline

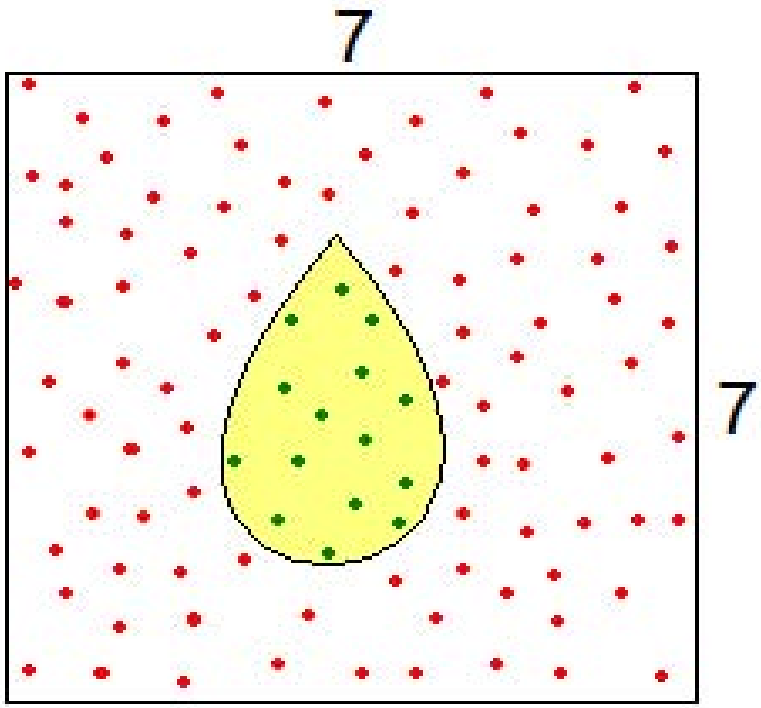
Why derivative-free optimization

Some of the reasons to apply Derivative-Free Optimization are the following:

- Nowadays computer hardware and mathematical algorithms allows increasingly large simulations.
- Functions are noisy (one cannot trust derivatives or approximate them by finite differences).
- Binary codes (source code not available) and random simulations — making automatic differentiation impossible to apply.
- Legacy codes (written in the past and not maintained by the original authors).
- Lack of sophistication of the user (users need improvement but want to use something simple).

Examples of problems where derivatives are unavailable

Computation of areas of figures by random generation of points (the derivatives of the area function are clearly unavailable):



$$\text{Area} = 7*7*\frac{15}{90+15} = 7$$

Examples of problems where derivatives are unavailable

Tuning of algorithmic parameters:

- Most numerical codes depend on a number of critical parameters.
- One way to automate the choice of the parameters (to find optimal values) is to solve:

$$\min_{p \in \mathbb{R}^{n_p}} f(p) = CPU(p; solver) \quad \text{s.t.} \quad p \in P,$$

or

$$\min_{p \in \mathbb{R}^{n_p}} f(p) = \#iterations(p; solver) \quad \text{s.t.} \quad p \in P,$$

where n_p is the number of parameters to be tuned and P is of the form $\{p \in \mathbb{R}^{n_p} : \ell \leq p \leq u\}$.

- It is hard to calculate derivatives for such functions f (which are likely noisy and non-differentiable).

Examples of problems where derivatives are unavailable

Automatic error analysis:

- A process in which the computer is used to analyze the accuracy or stability of a numerical computation.
- How large can the growth factor for GE be for a pivoting strategy? Given n and a pivoting strategy, one maximizes the growth factor:

$$\max_{A \in \mathbb{R}^{n \times n}} f(A) = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}.$$

  A starting point could be $A_0 = I_n$.
- When no pivoting is chosen, f is defined and continuous at all points where GE does not break down (possibly non-differentiable).
- For partial pivoting, the function f is defined everywhere (because GE cannot break down) but it can be discontinuous when a tie occurs.

Examples of problems where derivatives are unavailable

A list of known applications:

- Engineering design. A case study is the helicopter rotor blade design problem. Other examples are wing platform design, aeroacoustic shape design, and hydrodynamic design.
- Circuit design (tuning parameters of relatively small circuits).
- Molecular geometry optimization.
- Groundwater community problems.
- Medical image registration.
- Dynamic pricing.
- Earth imaging in Geophysics (full-waveform inversion).
- Process design of material science applications.

Limitations of derivative-free optimization

iteration	$\|x_k - x_*\|$
0	1.8284e+000
1	5.9099e-001
2	1.0976e-001
3	5.4283e-003
4	1.4654e-005
5	1.0737e-010
6	1.1102e-016

- Newton methods converge quadratically (locally) but require first and second order derivatives (gradient and Hessian).

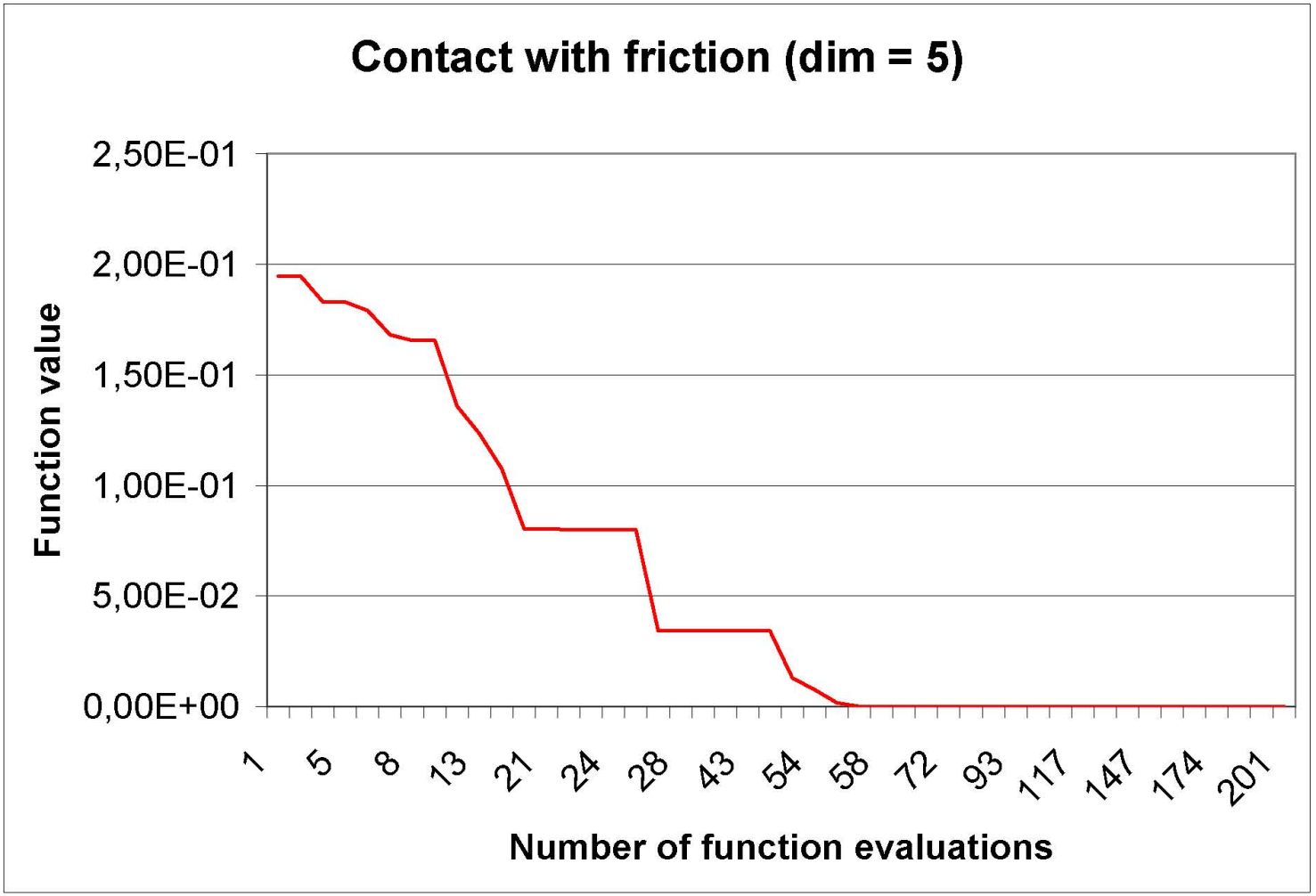
Limitations of derivative-free optimization

iteration	$\|x_k - x_*\|$
0	3.0000e+000
1	2.0002e+000
2	6.4656e-001
⋮	⋮
6	1.4633e-001
7	4.0389e-002
8	6.7861e-003
9	6.5550e-004
10	1.4943e-005
11	8.3747e-008
12	8.8528e-010

- Quasi Newton (secant) methods converge superlinearly (locally) but require first order derivatives (gradient).

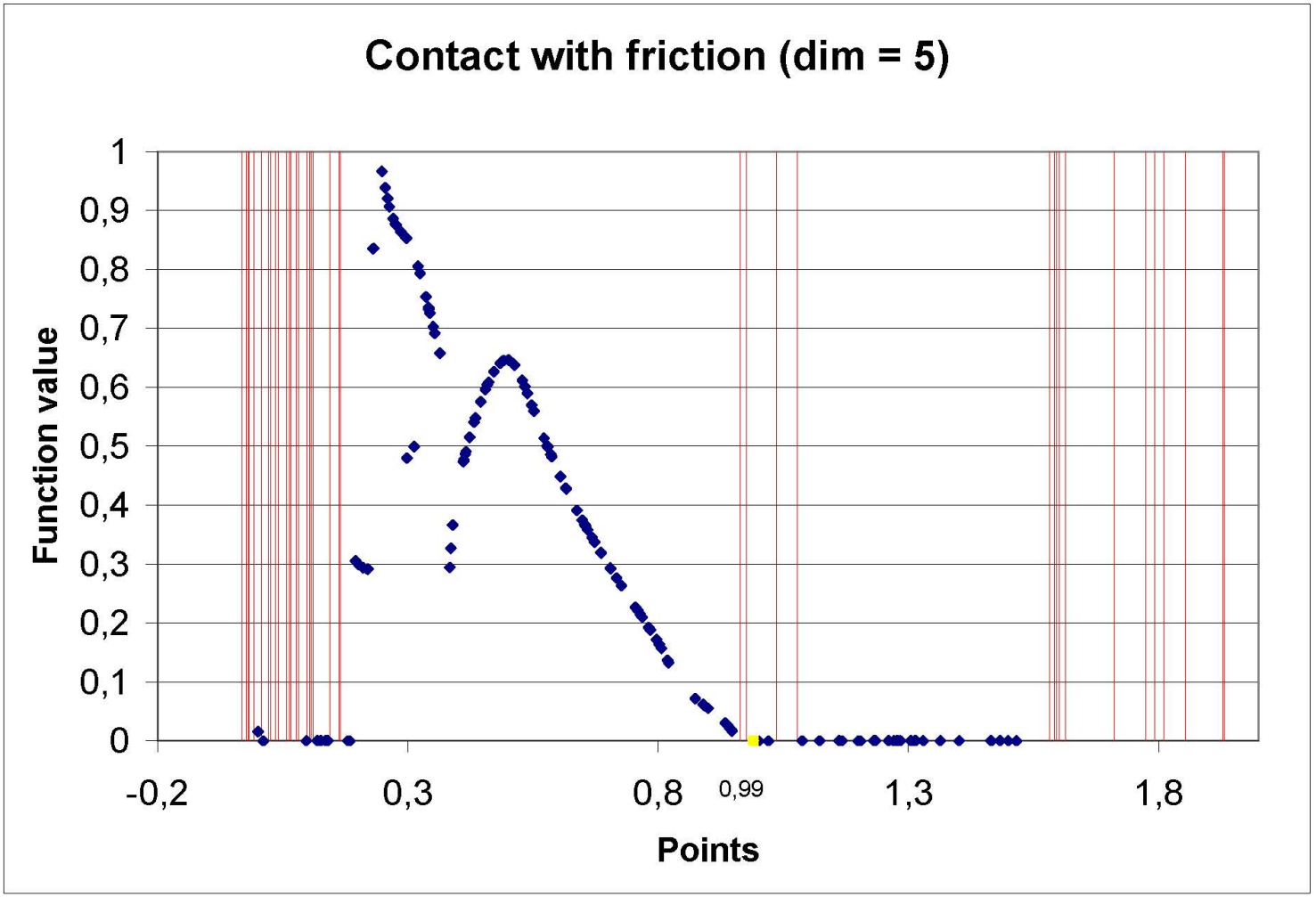
Limitations of derivative-free optimization

In DFO convergence/stopping is typically slow (per function evaluation):



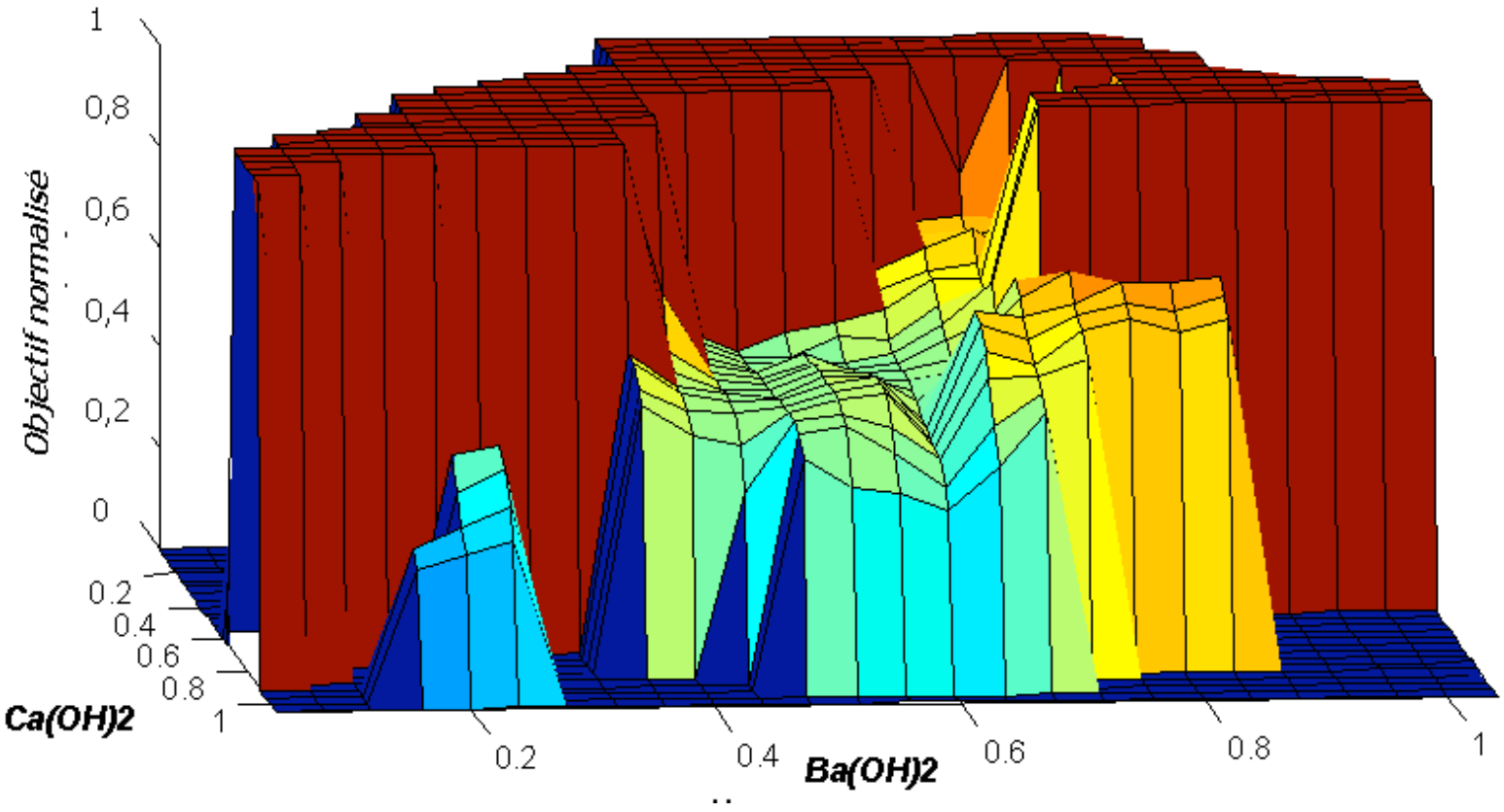
Pitfalls

The objective function might not be continuous or even well defined:



Pitfalls

The objective function might not be continuous or even well defined:



What can we solve

With the current state-of-the-art DFO methods one can expect to successfully address problems where:

- The evaluation of the function is expensive and/or computed with noise (and for which accurate finite-difference derivative estimation is prohibitive and automatic differentiation is ruled out).
- The number of variables does not exceed, say, a hundred (in serial computation).
- The functions are not excessively non-smooth.
- Rapid asymptotic convergence is not of primary importance.
- Only a few digits of accuracy are required.

Illustration of course of dimensionality

Number of points needed to build a complete/determined quadratic polynomial interpolant model:

n	10	20	50	100	200
$(n+1)(n+2)/2$	66	231	1326	5151	20301

What can we solve

In addition we can expect to solve problems:

- With hundreds of variables using a parallel environment or exploiting problem information.
- With a few integer or categorical variables.
- With a moderate level of multimodality:

  It is hard to minimize non-convex functions without derivatives.

  However, it is generally accepted that derivative-free optimization methods have the ability to find 'good' local optima.

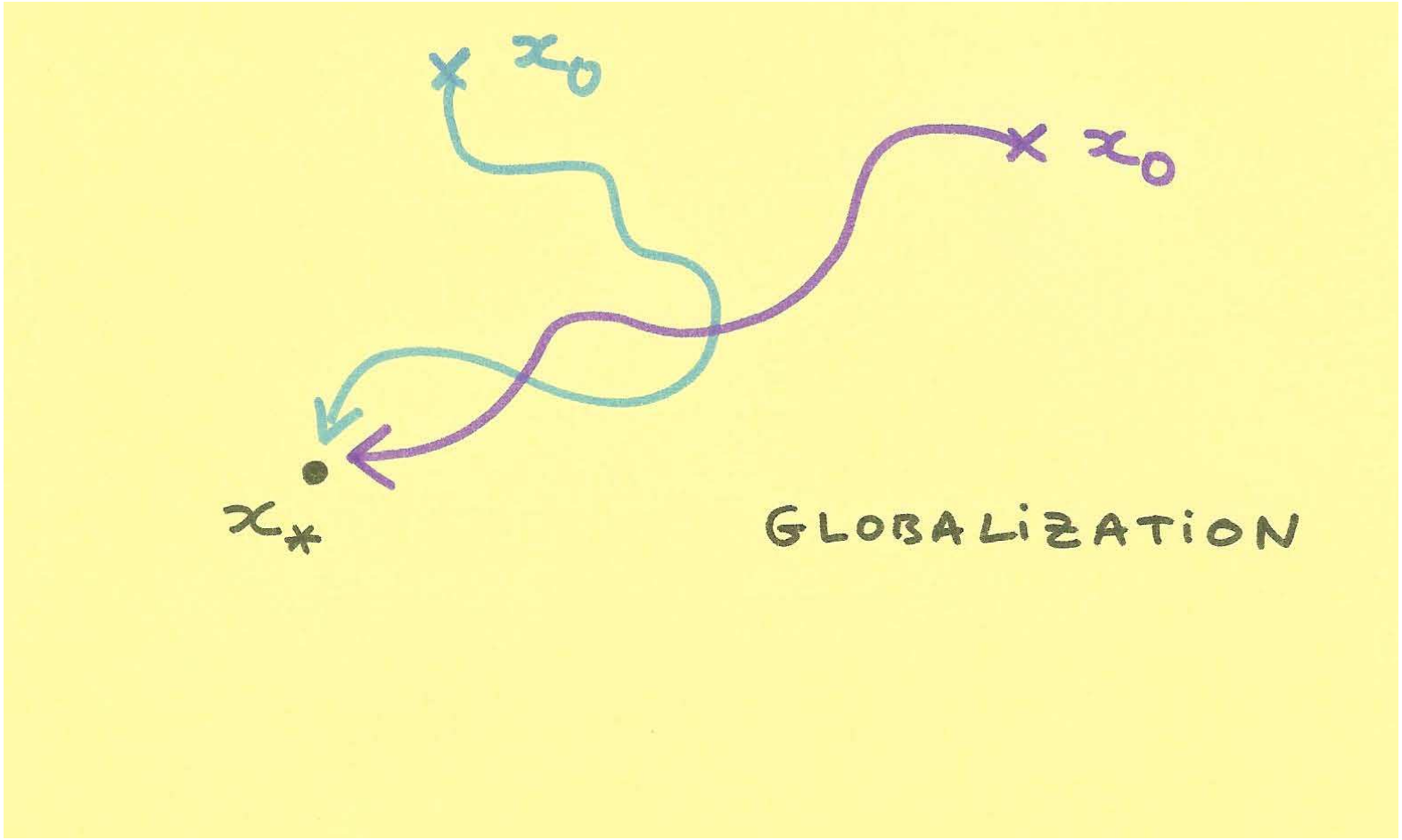
  DFO methods have a tendency to: (i) go to generally low regions in the early iterations; (ii) 'smooth' the function in later iterations.

Analysis of algorithms for Nonlinear Optimization

- Global convergence: convergence to some form of stationarity independently of the starting point.
- Global rates (worst case complexity): no assumption on the starting point, amount of work needed to reach some threshold of stationarity.
- Local rates of convergence: rates of convergence, like superlinear or quadratic, in a neighborhood of a minimizer.

This course will focus on: Derivative-Free Optimization (DFO), zero-order methods.

Global convergence of algorithms



By global convergence we mean convergence to some form of stationarity from arbitrary starting points.

ATTENTION!!!

Global and local applied to algorithms and to its convergence properties

is not the same as

global and local applied to minimizers (i.e, absolute and relative).

Classes of algorithms (globally convergent)

Over-simplifying, all globally convergent DFO algorithms must:

- Achieve (deterministically or probabilistically) some form of descent away from stationarity

  ... by guaranteeing (deterministically or probabilistically) some control of the geometry of the sample sets (or directions) where (or along which) the objective function is evaluated.

- Imply convergence of step size parameters to zero, indicating global convergence to a stationary point.

There are two main classes of rigorous methods in DFO

- Directional methods, like direct search.
- Model-based methods, like trust-region methods.

Classes of algorithms (globally convergent)

(Directional) Direct Search:

- Achieve descent by moving in the direction of potentially better points.
- In the smooth and deterministic case, these points are defined by directions in positive spanning sets.
- Examples are coordinate search, pattern search, generalized pattern search (GPS), generating set search (GSS), mesh adaptive direct search (MADS).

Classes of algorithms (globally convergent)

(Simplicial) Direct Search:

- Ensure descent from simplex operations like reflections, by moving in the direction away from the point with the worst function value.
- Examples are the Nelder-Mead method and several modifications to Nelder-Mead.

Their global convergence is though viewed in a directional sense.

Classes of algorithms (globally convergent)

Line-Search Methods:

- Aim to get descent along negative simplex gradients (which are intimately related to polynomial models).
- Examples are the implicit filtering method.

Their global convergence is viewed also in a directional sense.

Classes of algorithms (globally convergent)

Trust-Region Methods:

- Minimize trust-region subproblems defined by fully linear or fully quadratic models (typically built from interpolation or regression).
- Examples are methods based on polynomial models or radial basis functions models.

Indication of typical behavior

A first problem is the minimization of the Rosenbrock function:

$$\min_{(x_1,x_2)\in\mathbb{R}^2} 100(x_1^2 - x_2)^2 + (1 - x_1)^2,$$

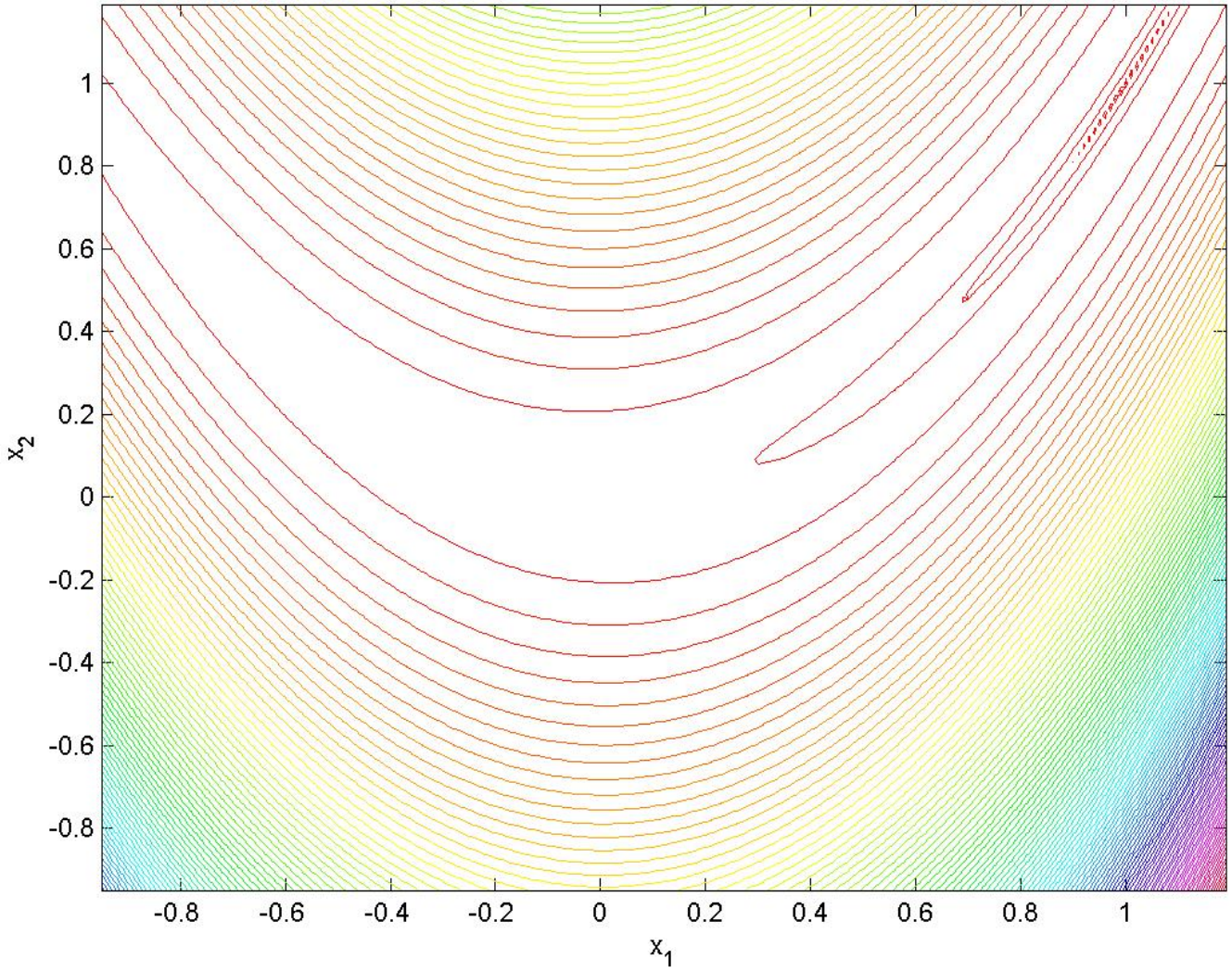
which has a unique minimizer at $(1, 1)$.

The level curves of this function describe a strongly curved valley with steep sides.

Depending on the starting point picked, methods which do not explore the curvature of the function might be extremely slow.

For instance, if one starts around $(-1, 1)$ one has to follow a curved valley with relatively steep sides in order to attain the minimum.

Rosenbrock function (curvature)



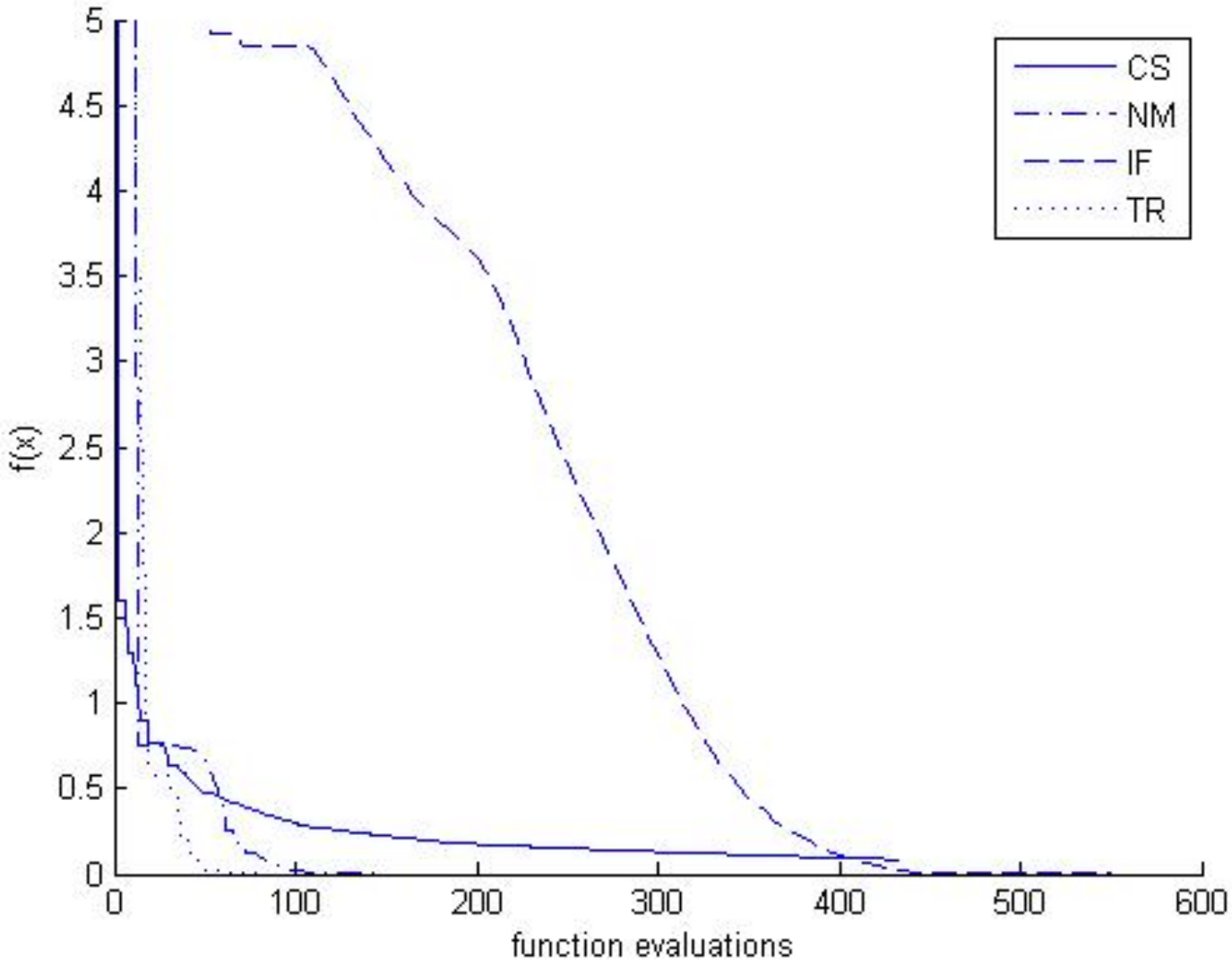
Indication of typical behavior

We chose four methods:

- Directional Direct Search: Coordinate Search (CS).
- Simplicial Direct Search: Nelder-Mead (NM).
- Line-Search Method: Implicit Filtering (IF).
- Trust-Region Method: DFO code (TR).

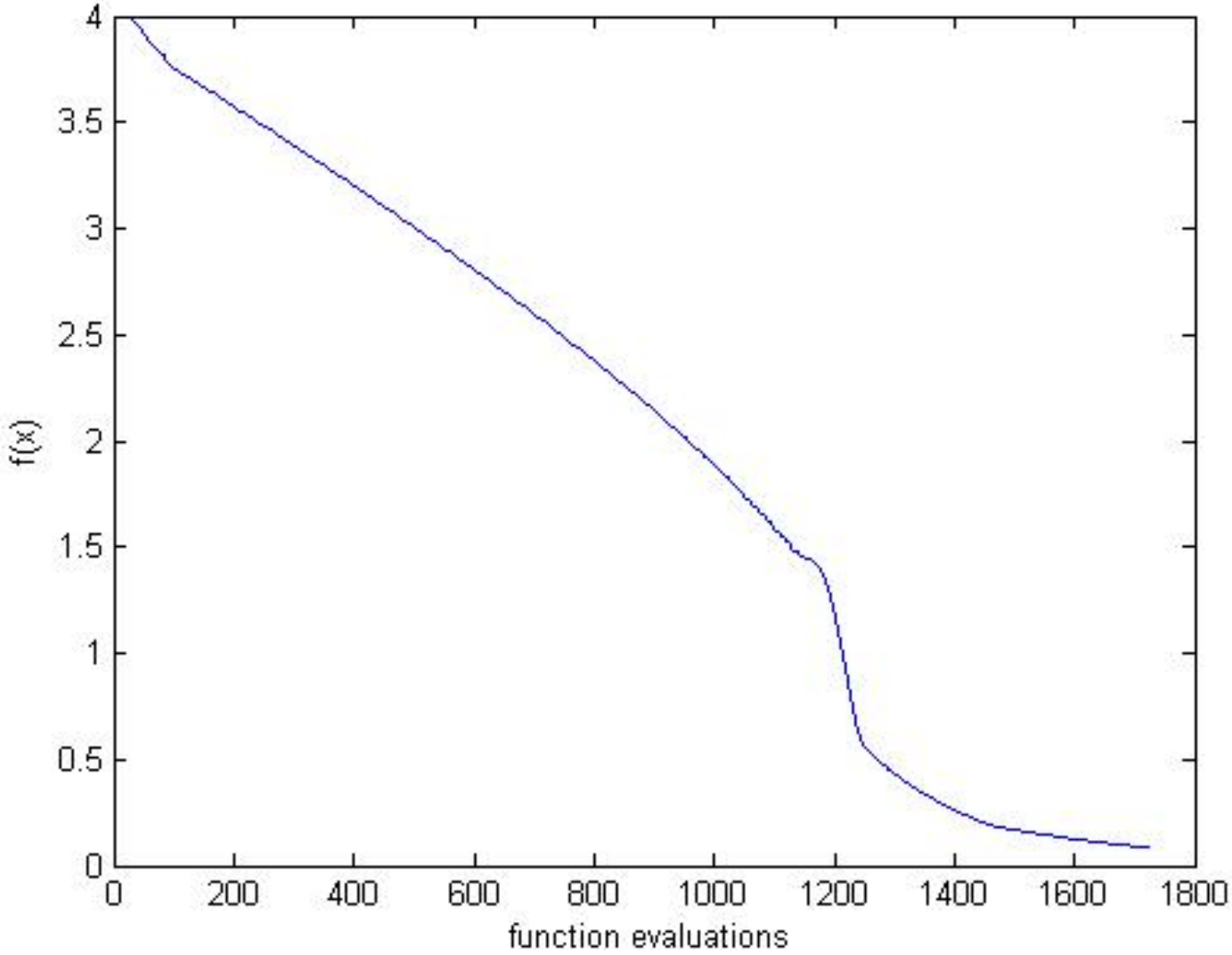
Four methods on Rosenbrock

We started the four methods at $(1.2, 0)$.



Coordinate search on Rosenbrock

Now we started CS at $(-1, 1)$.



Indication of typical behavior

The second problem involves the minimization of a simple, perturbed quadratic function.

The perturbation involves cosine functions with periods of $2\pi/70$ and $2\pi/100$:

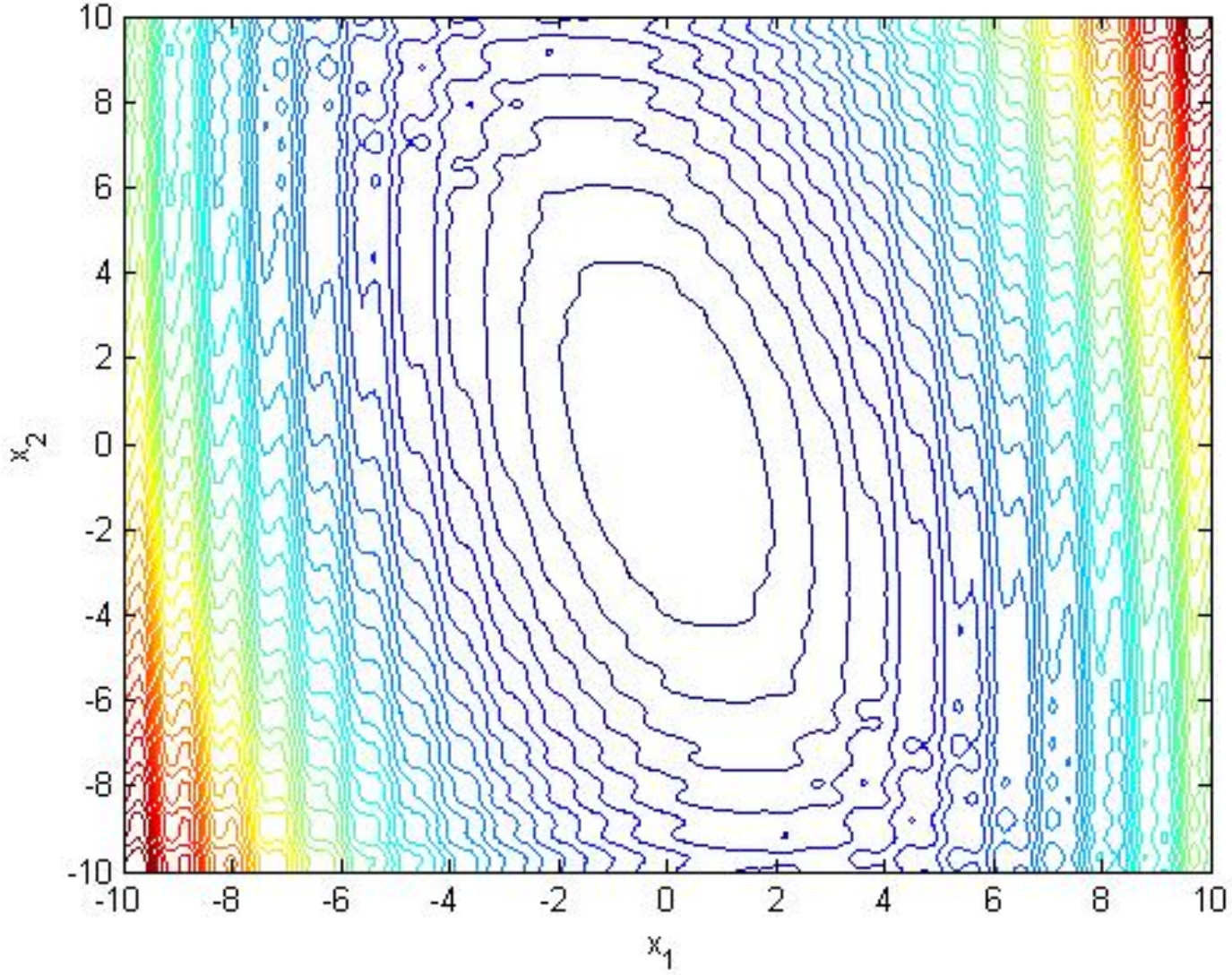
$$\min_{(x_1,x_2)\in\mathbb{R}^2} \quad 10(x_1^2)(1+0.75\cos(70x_1)/12)+\cos(100x_1)^2/24+ \\ 2(x_2^2)(1+0.75\cos(70x_2)/12)+\cos(100x_2)^2/24+4x_1x_2.$$

The unique minimizer (of the underlying quadratic function) is at $(0,0)$.

As opposed to the Rosenbrock function, the underlying smooth function here has a mild curvature.

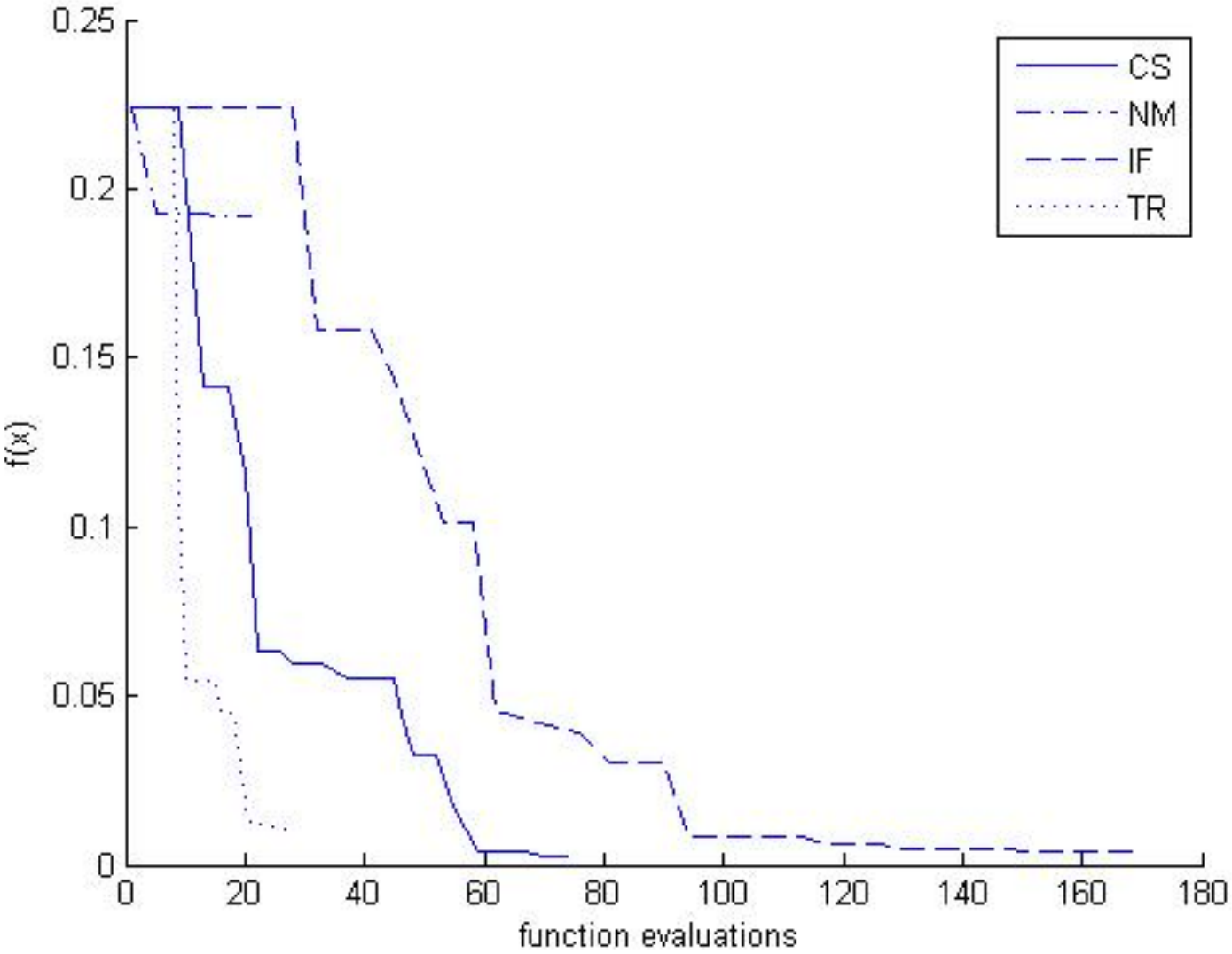
However, the perturbed function has been contaminated with noise which will then pose different difficulties to algorithms.

Perturbed quadratic (noise)



Four methods on perturbed quadratic

We started the four methods at $(0.1, 0.1)$.



Summary

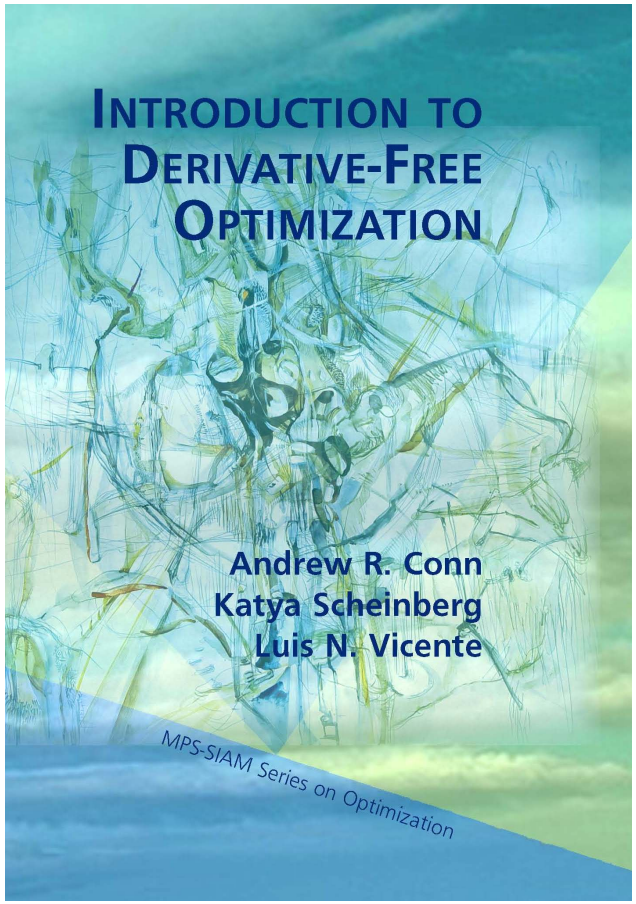
- Model-based methods (like interpolation with trust regions) can be very efficient.
- Directional direct search looses in comparison when the function is smooth and the curvature is adverse, but:

  offers a valid alternative for noisy or non-smooth problems and can be successfully combined with model-based techniques,

  and is easy to parallelize and to adapt for constraints.
- Simplicial direct search (Nelder-Mead) can be efficient too (when it works).
- There is certainly room for other methods like modified, safeguarded Nelder-Mead methods and for approaches particularly tailored for noisy problems and easy to parallelize like implicit filtering.

The IDFO book

- A. R. Conn, K. Scheinberg, and L. N. Vicente, Introduction to Derivative-Free Optimization, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2009.



Presentation outline

Positive spanning sets

The positive span of a set of vectors $[v_1 \cdots v_r]$ in $\mathbb{R}^n$ is the convex cone

$$\{v \in \mathbb{R}^n : \ v = \alpha_1 v_1 + \cdots + \alpha_r v_r, \quad \alpha_i \geq 0, \ i = 1, \ldots, r\}.$$

Definition

A positive spanning set in $\mathbb{R}^n$ is a set of vectors whose positive span is $\mathbb{R}^n$.

The set $[v_1 \cdots v_r]$ is said to be positively dependent if one of the vectors is in the convex cone positively spanned by the remaining vectors, i.e., if one of the vectors is a positive combination of the others; otherwise the set is positively independent.

Positive bases

Definition

A positive basis in $\mathbb{R}^n$ is a positively independent set whose positive span is $\mathbb{R}^n$.

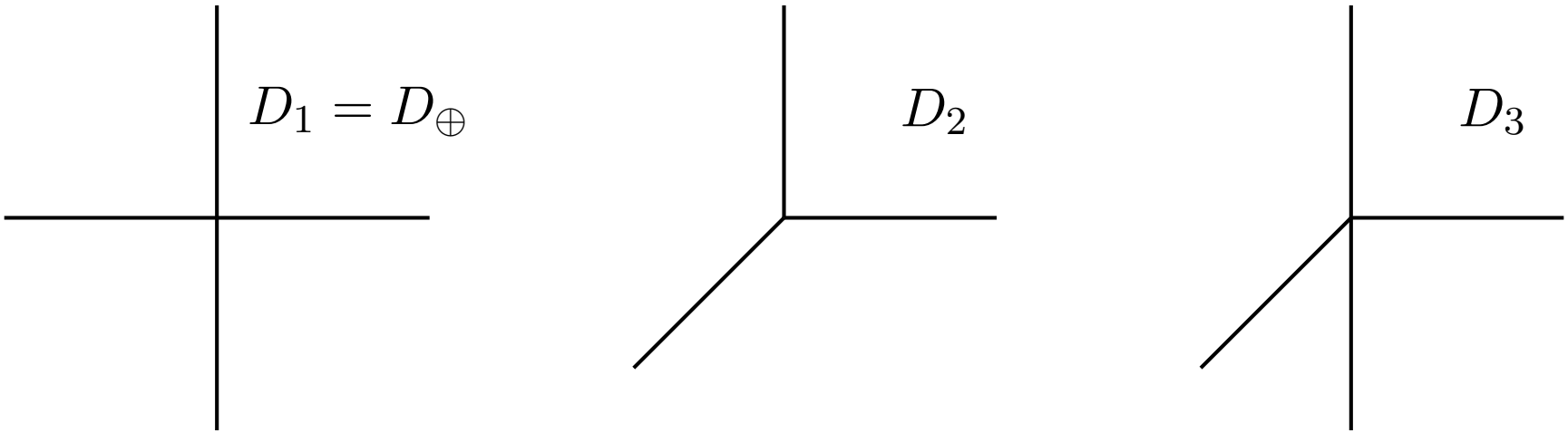
Equivalently, a positive basis for $\mathbb{R}^n$ can be defined as a set of nonzero vectors of $\mathbb{R}^n$ whose positive combinations span $\mathbb{R}^n$, but for which no proper subset exhibits the same property.

A positive spanning set contains at least $n+1$ vectors in $\mathbb{R}^n$.

It can also be shown that a positive basis cannot contain more than $2n$ elements.

Positive bases with $n+1$ and $2n$ elements are referred to as minimal and maximal positive bases, respectively.

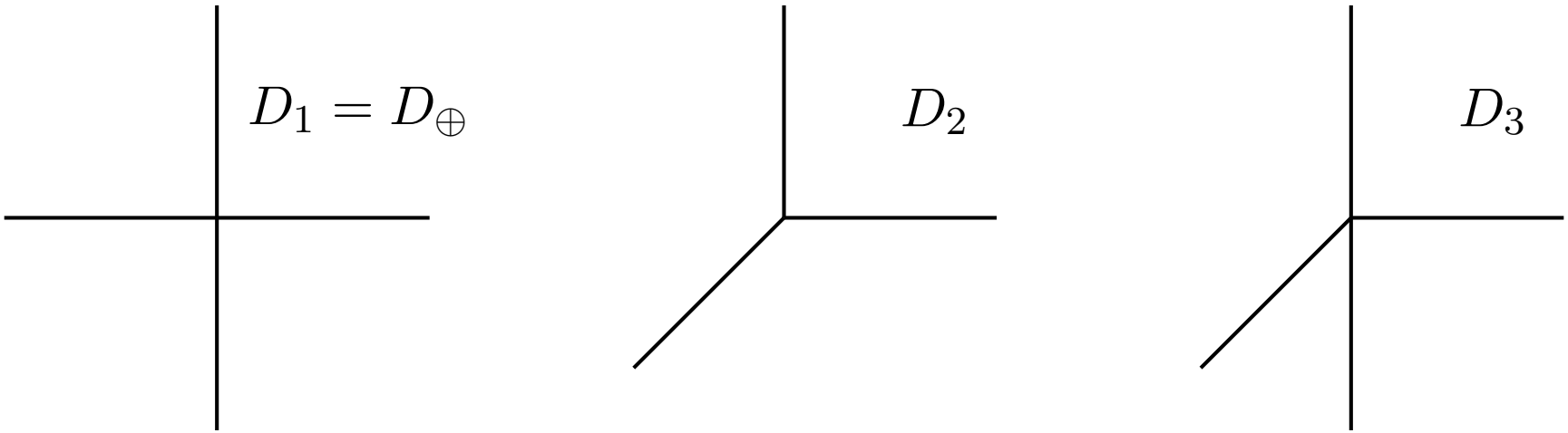
Examples



$$D_\oplus = D_1 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$ maximal positive basis

$$D_2 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \quad D_2 = \begin{bmatrix} 1 & 0 & -\sqrt{2}/2 \\ 0 & 1 & -\sqrt{2}/2 \end{bmatrix}$$ minimal positive basis

Examples



$$D_3 = \begin{bmatrix} 1 & 0 & -\sqrt{2}/2 & 0 \\ 0 & 1 & -\sqrt{2}/2 & -1 \end{bmatrix}$$

positive spanning set, not positive basis

Necessary and sufficient characterizations

We present now three necessary and sufficient characterizations for a set that spans $\mathbb{R}^n$ to also span $\mathbb{R}^n$ positively.

Theorem

Let $[v_1 \cdots v_r]$, with $v_i \neq 0$ for all $i \in \{1, \ldots, r\}$, span $\mathbb{R}^n$. Then the following are equivalent:

(i) $[v_1 \cdots v_r]$ *spans* $\mathbb{R}^n$ *positively.*

(ii) *For every $i = 1, \ldots, r$, the vector $-v_i$ is in the convex cone positively spanned by the remaining $r - 1$ vectors.*

(iii) *There exist real scalars $\alpha_1, \ldots, \alpha_r$ with $\alpha_i > 0$, $i \in \{1, \ldots, r\}$, such that $\sum_{i=1}^{r} \alpha_i v_i = 0$.*

(iv) *For every nonzero vector $w \in \mathbb{R}^n$, there exists an index i in $\{1, \ldots, r\}$ for which $w^\top v_i > 0$.*

Consequences (construction)

Corollary

Suppose $[v_1 \cdots v_r]$ *is a positive basis for* $\mathbb{R}^n$ *and* $W \in \mathbb{R}^{n\times n}$ *is a nonsingular matrix. Then* $[Wv_1 \cdots Wv_r]$ *is also a positive basis for* $\mathbb{R}^n$.

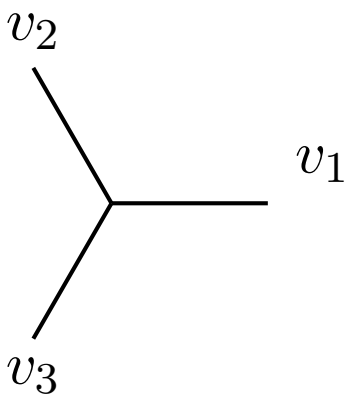
One can easily prove that $D_\oplus = [I \ -I]$ is a (maximal) positive basis. Thus, $[W \ -W]$ is also a (maximal) positive basis, for any choice of the nonsingular matrix $W \in \mathbb{R}^{n\times n}$.

Corollary

(i) $[I \ -e]$ *is a (minimal) positive basis.*

(ii) Let $W = [w_1 \cdots w_n] \in \mathbb{R}^{n\times n}$ *be a nonsingular matrix. Then* $[W \ -\sum_{i=1}^n w_i]$ *is a (minimal) positive basis for* $\mathbb{R}^n$.

Consequences (construction)



$$[\, v_1 \; v_2 \; v_3 \,] = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3/4} & -\sqrt{3/4} \end{bmatrix}$$

One can also build (minimal) positive bases with uniform amplitudes in $\mathbb{R}^n$.

Consequences (descent)

The characterization (iv) is at the heart of directional direct search methods.

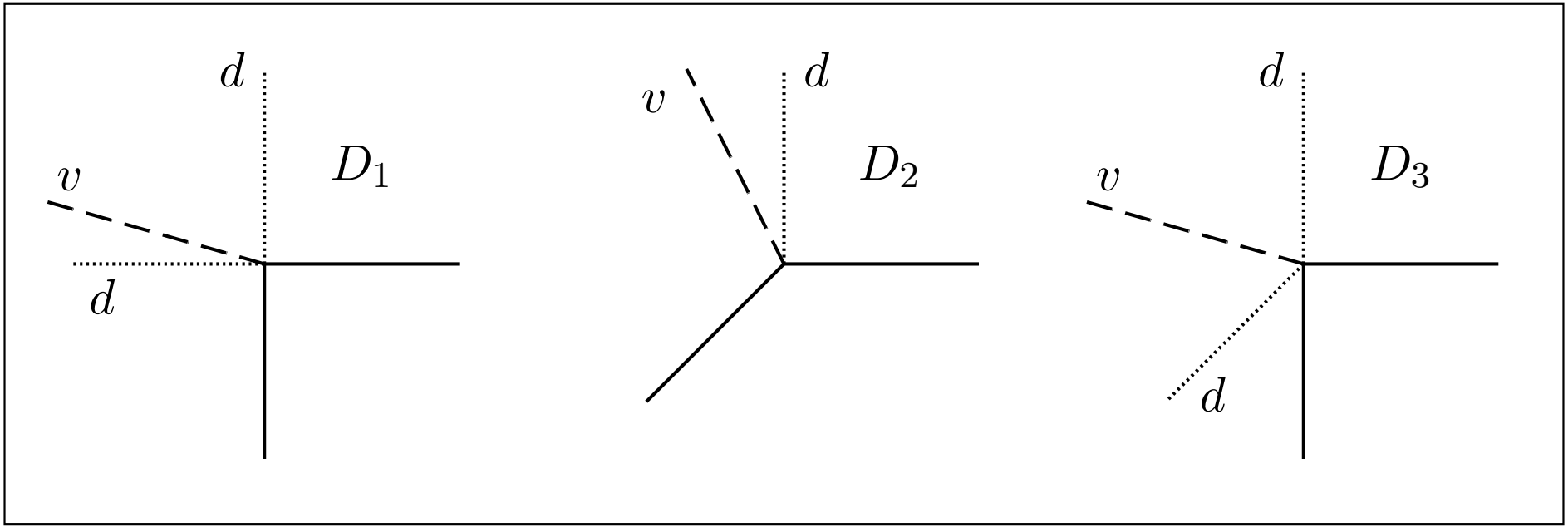
It implies that, given a continuously differentiable function f at some given point x where $\nabla f(x) \neq 0$, there must always exist a vector d in a given positive spanning set (or in a positive basis) such that

$$-\nabla f(x)^\top d \; > \; 0.$$

In other words, there must always exist a direction of descent in such a set.

We identify such a vector d for the three spanning sets D_1, D_2, and D_3 given before.

Consequences (descent)



Cosine measure

Definition

The cosine measure of a positive spanning set (with nonzero vectors) or of a positive basis D is defined by

$$\mathrm{cm}(D) = \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|}.$$

Given any positive spanning set, it necessarily happens that

$$\mathrm{cm}(D) > 0.$$

Values of the cosine measure close to zero indicate a deterioration of the positive spanning property.

Cosine measure (examples)

For example, the maximal positive basis $D_{\oplus} = [I \; -I]$ has cosine measure equal to $1/\sqrt{n}$. When $n = 2$ we have $\mathrm{cm}(D_{\oplus}) = \sqrt{2}/2$.

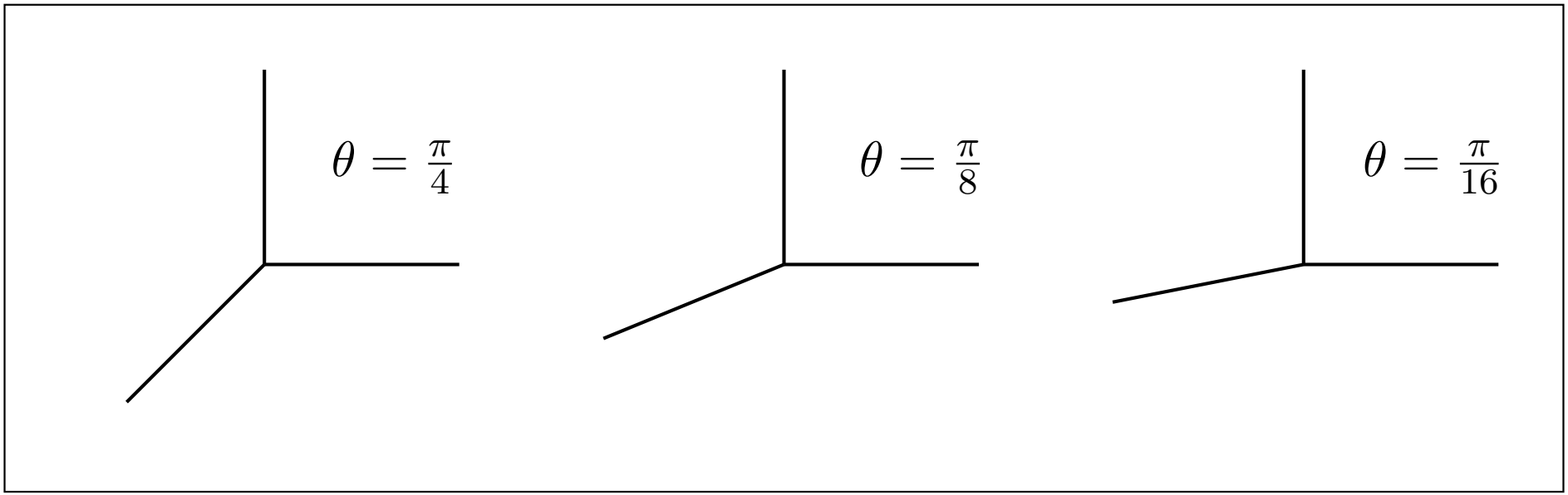
Now let us consider the following example. Let θ be an angle in $(0, \pi/4]$ and D_θ be a positive basis defined by

$$D_\theta = \begin{bmatrix} 1 & 0 & -\cos(\theta) \\ 0 & 1 & -\sin(\theta) \end{bmatrix}.$$

Observe that $D_{\frac{\pi}{4}}$ is just the positive basis D_2 considered before.

The cosine measure of D_θ is given by $\cos((\pi - \theta)/2)$ and it converges to zero when θ tends to zero.

Cosine measure (examples)



$$\mathrm{cm}(D_\theta) \longrightarrow 0.$$

Gradient estimate in direct search

Given a positive spanning set D, a point x, and a positive value for the parameter α, we are interested in looking at the points of the form

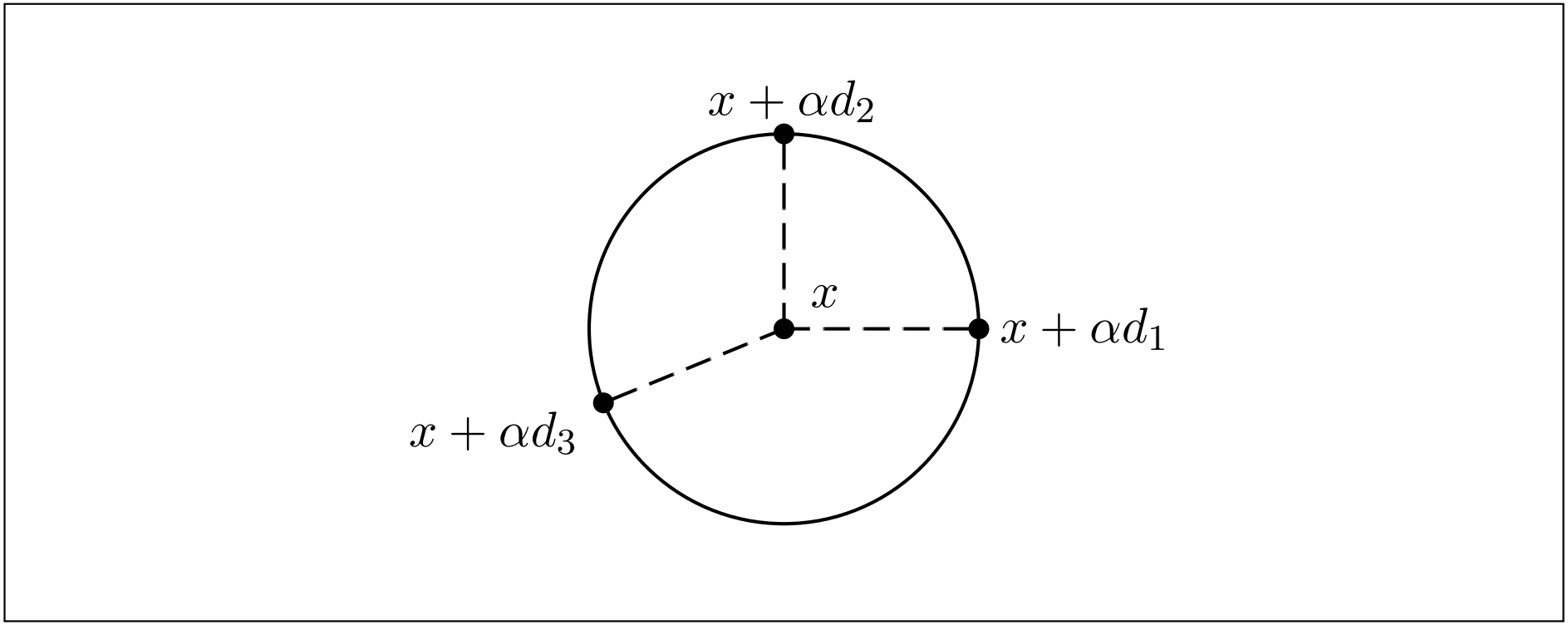
$$x + \alpha d, \quad \text{for all} \quad d \in D.$$

These points are in a ball centered at x, of radius Δ defined by:

$$\Delta \;=\; \alpha \max_{d \in D} \|d\|.$$

If only a finite number of positive spanning sets are used in an algorithm, then Δ tends to zero if and only if α tends to zero.

Polling points in a ball



A ball of radius $\Delta = \alpha \max_{d \in D} \|d\|$ centered at x.

Gradient estimate in direct search

If we sample $n+1$ points of the form $x+\alpha d$ defined by a positive basis D, and their function values are no better than the function value at x, then the size of the gradient is $\mathcal{O}(\alpha) = \mathcal{O}(\Delta)$:

Theorem

Let D be a positive spanning set and $\alpha > 0$ be given. Assume that ∇f is Lipschitz continuous (with constant $\nu > 0$) in an open set containing the ball $B(x; \Delta)$.

If $f(x) \leq f(x+\alpha d)$, for all $d \in D$, then

$$\|\nabla f(x)\| \leq \frac{\nu}{2} \operatorname{cm}(D)^{-1} \max_{d \in D} \|d\| \, \alpha.$$

Gradient estimate in direct search

The bound can be rewritten in the form (for normalized d's):

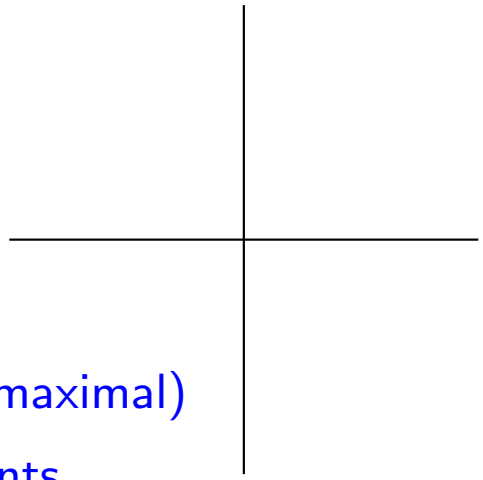
$$\|\nabla f(x)\| \leq \kappa_{eg}\Delta.$$

where $\kappa_{eg} = \nu\,\mathrm{cm}(D)^{-1}/2$. This bound has the same structure as other bounds used in different derivative-free methods.

The bound is basically given by Δ times a constant that depends on the nonlinearity of the function (expressed by the Lipschitz constant ν) and on the geometry of the sample set (measured by $\mathrm{cm}(D)^{-1}$).

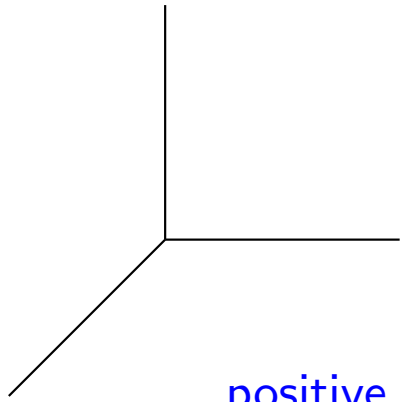
If a directional direct-search method is able to generate a sequence of points x satisfying the conditions of this theorem for which α (and thus Δ) tends to zero, then clearly the gradient converges to zero along this sequence.

Recapitulation



positive basis (maximal)

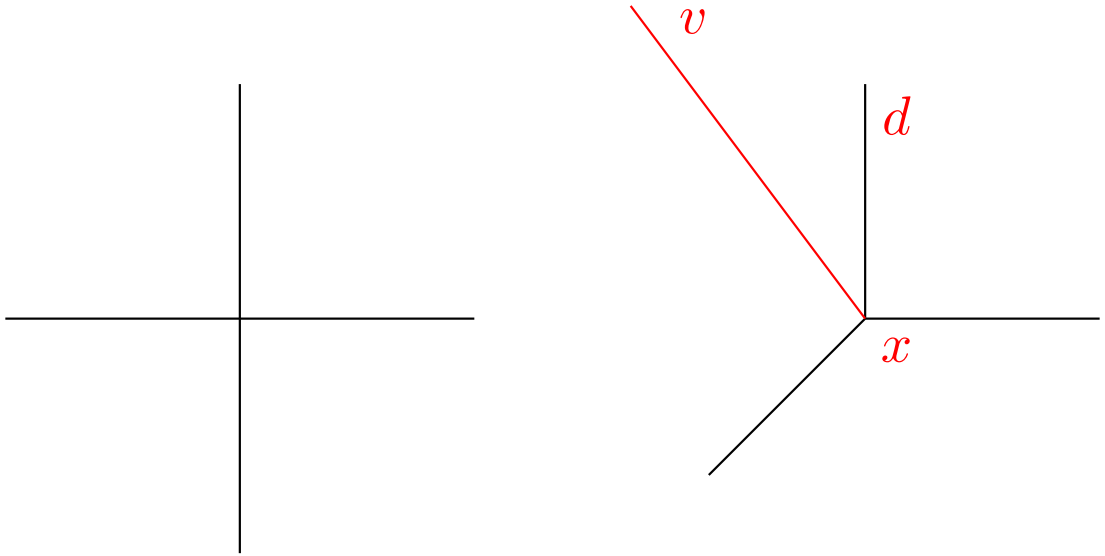
$2n$ elements



positive basis (minimal)

$n + 1$ elements

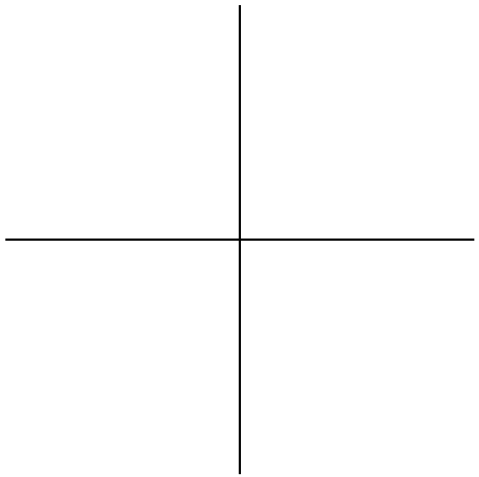
Recapitulation

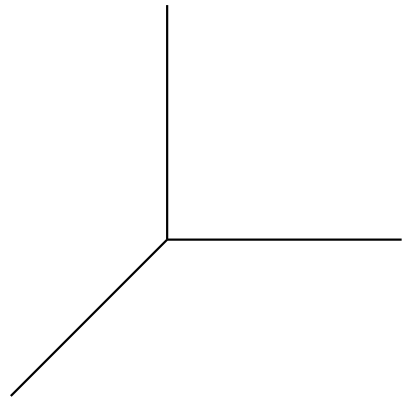


If $v = -\nabla f(x)$ then d is a descent direction.

Recapitulation

$$D = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \qquad D = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$$



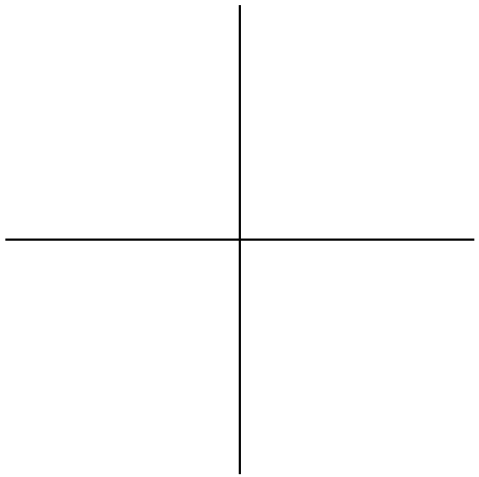


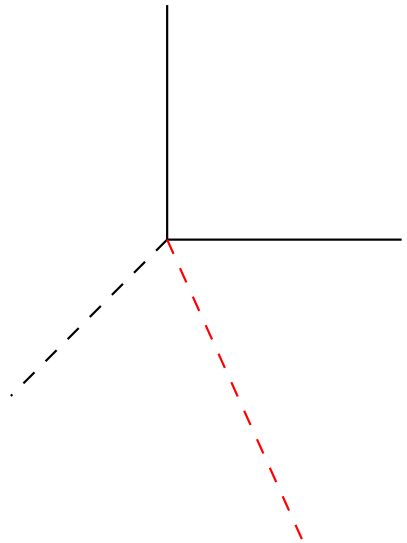
$$\mathrm{cm}(D) \;=\; \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|} \;>\; 0.$$

Recapitulation

$$D = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$$

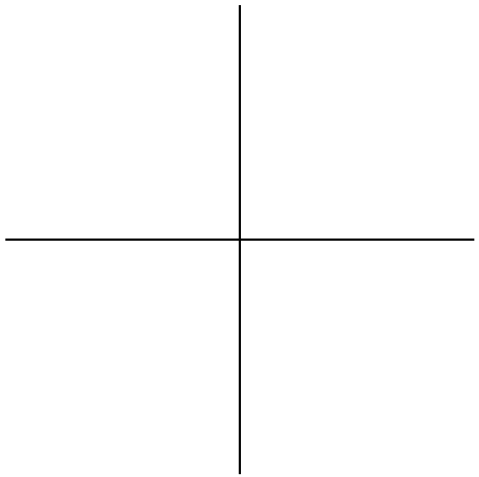


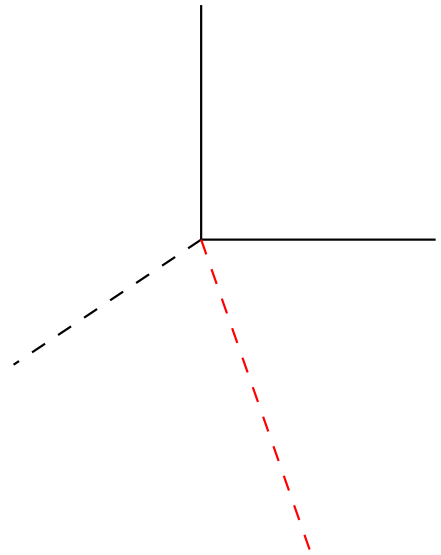


$$\mathrm{cm}(D) \;=\; \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|} \;>\; 0.$$

Recapitulation

$$D = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \qquad D = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$$



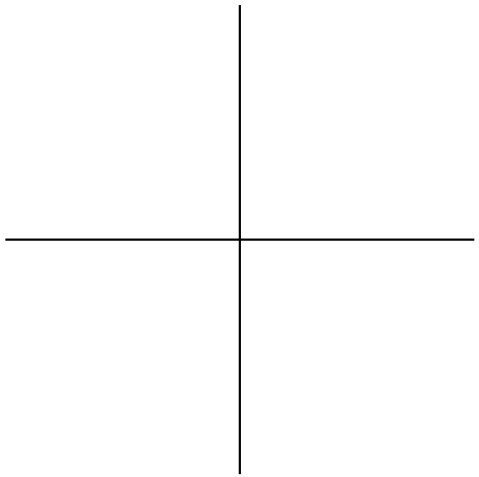


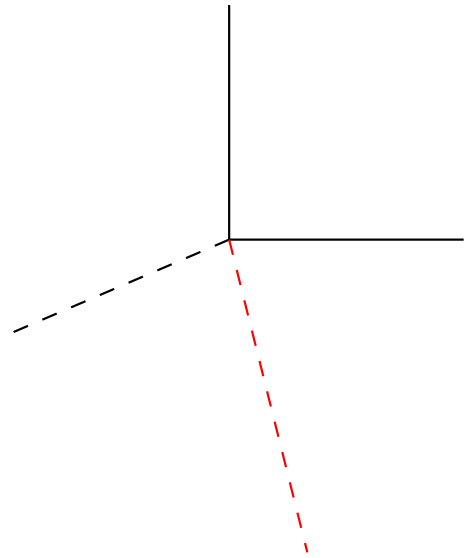
$$\mathrm{cm}(D) \;=\; \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|} \;>\; 0.$$

Recapitulation

$$D = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$$



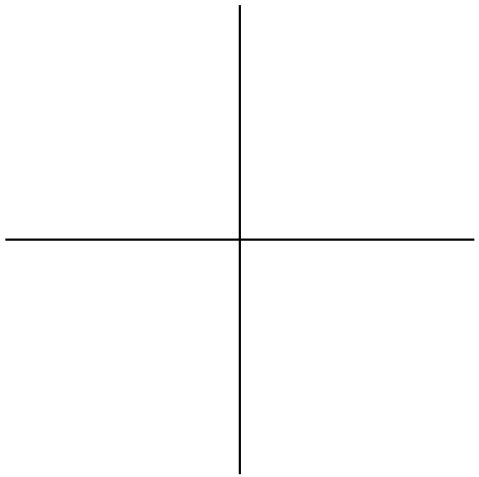


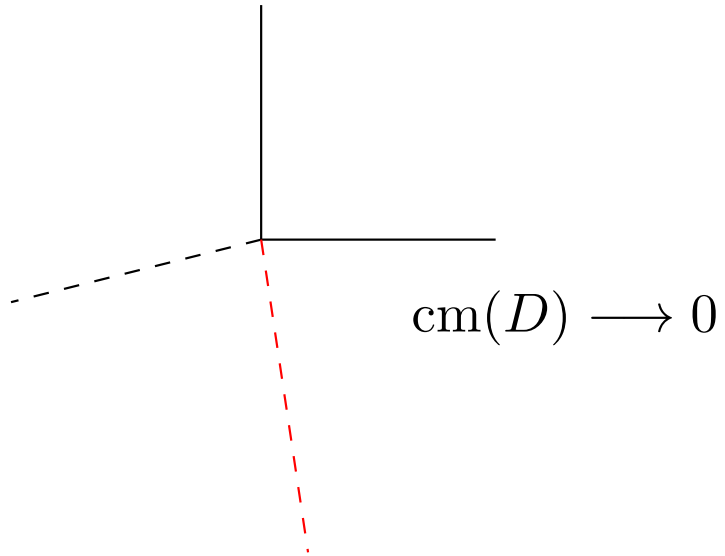
$$\mathrm{cm}(D) \;=\; \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|} \;>\; 0.$$

Recapitulation

$$D = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$$





$$\mathrm{cm}(D) \;=\; \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|} \;>\; 0.$$

Linear interpolation

Now we turn our attention to sample sets not necessarily formed by a predefined set of directions. Consider a sample set $Y = \{y^0, y^1, \ldots, y^p\}$ in $\mathbb{R}^n$.

The simplest model based on $n+1$ sample points ($p = n$) that we can think of is an interpolation model.

Let $m(x)$ denote a polynomial of degree $d = 1$ interpolating f at the points in Y:

$$m(y^i) \;=\; f(y^i), \quad i = 0, \ldots, n.$$

We can express $m(x)$ in the form

$$m(x) \;=\; \alpha_0 + \alpha_1 x_1 + \cdots + \alpha_n x_n,$$

using $\phi = \{1, x_1, \ldots, x_n\}$ as a basis for $\mathcal{P}_n^1$.

Linear interpolation conditions

We can then rewrite the interpolation conditions

$$\begin{bmatrix} 1 & y_1^0 & \cdots & y_n^0 \\ 1 & y_1^1 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^n & \cdots & y_n^n \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} f(y^0) \\ f(y^1) \\ \vdots \\ f(y^n) \end{bmatrix}.$$

The matrix of this linear system is denoted by

$$M = M(\phi, Y) = \begin{bmatrix} 1 & y_1^0 & \cdots & y_n^0 \\ 1 & y_1^1 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^n & \cdots & y_n^n \end{bmatrix}.$$

We write M as $M(\phi, Y)$ to highlight the dependence of M on the basis ϕ and on the sample set Y.

Poisedness for linear interpolation

Definition

The set $Y = \{y^0, y^1, \ldots, y^n\}$ is poised for linear interpolation in $\mathbb{R}^n$ if the corresponding matrix $M(\phi, Y)$ is nonsingular.

The definition of poisedness is independent of the basis chosen. In other words, if Y is poised for a basis ϕ then it is poised for any other basis in $\mathcal{P}_n^1$.

The definition of $m(x)$ is also independent of the basis chosen.

The sample set Y is poised for linear interpolation if and only if the linear interpolating polynomial $m(x) = \alpha_0 + \alpha_1 x_1 + \cdots + \alpha_n x_n$ is uniquely defined.

Linear regression

When $p > n$, it might not be possible to fit a linear polynomial.

One option is to use linear regression and to compute the coefficients of the linear (least-squares) regression polynomial

$$m(x) = \alpha_0 + \alpha_1 x_1 + \cdots + \alpha_n x_n$$

as the least-squares solution of the system

$$\begin{bmatrix} 1 & y_1^0 & \cdots & y_n^0 \\ 1 & y_1^1 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^p & \cdots & y_n^p \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} f(y^0) \\ f(y^1) \\ \vdots \\ \vdots \\ f(y^p) \end{bmatrix}.$$

Again, we denote the matrix of this (possibly overdetermined) system of linear equations by $M = M(\phi, Y)$.

Poisedness for linear regression

The definition of poisedness generalizes easily from linear interpolation to linear regression.

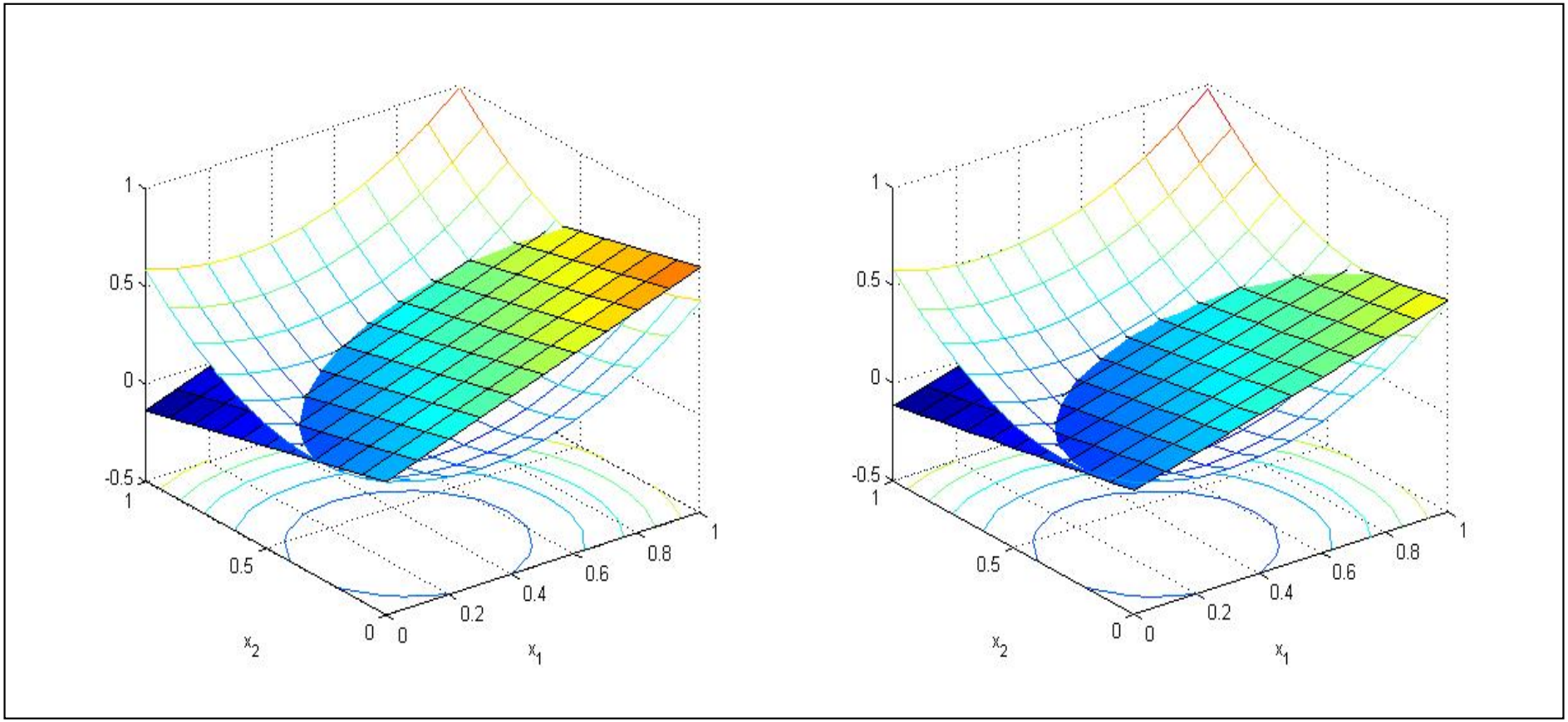
Definition

The set $Y = \{y^0, y^1, \ldots, y^p\}$, with $p > n$, is poised for linear regression in $\mathbb{R}^n$ if the corresponding matrix $M(\phi, Y)$ has full (column) rank.

If a set Y is poised for a basis ϕ then it is also poised for any other basis in $\mathcal{P}_n^1$. Also, $m(x)$ is independent of the basis chosen.

The sample set is poised for linear regression if and only if the linear regression polynomial $m(x)$ is uniquely defined.

Example of linear interpolation and regression



Error bounds for linear interpolation

We consider the interpolation points $y^0, y^1, \ldots, y^n$ in $B(y^0; \Delta)$, where:

$$\Delta \;=\; \Delta(Y) \;=\; \max_{1 \le i \le n} \|y^i - y^0\|.$$

We are interested in the quality of $\nabla m(y)$ and $m(y)$ in $B(y^0; \Delta)$.

Assumption

We assume that $Y = \{y^0, y^1, \ldots, y^n\} \subset \mathbb{R}^n$ is a poised set of sample points (in the linear interpolation sense) contained in the ball $B(y^0; \Delta(Y))$.

Further, we assume that the function f is continuously differentiable in an open domain Ω containing $B(y^0; \Delta)$ and ∇f is Lipschitz continuous in Ω with constant $\nu > 0$.

Error bounds for linear interpolation

The derivation of the error bounds is based on the application of one step of Gaussian elimination to the matrix $M = M(\phi, Y)$:

$$\begin{bmatrix} 1 & y_1^0 & \cdots & y_n^0 \\ 0 & y_1^1 - y_1^0 & \cdots & y_n^1 - y_n^0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & y_1^n - y_1^0 & \cdots & y_n^n - y_n^0 \end{bmatrix} = \begin{bmatrix} 1 & y_0^\top \\ 0 & L \end{bmatrix},$$

with

$$L = \left[y^1 - y^0 \cdots y^n - y^0\right]^\top = \begin{bmatrix} (y^1 - y^0)^\top \\ \vdots \\ (y^n - y^0)^\top \end{bmatrix}.$$

L is nonsingular if and only if M is nonsingular, since $\det(L) = \det(M)$. Notice that the points appear listed in L by rows, which favors factorizations by rows.

Error bounds for linear interpolation

It turns out that the error bounds for the approximation which we derive are in terms of the scaled matrix

$$\hat{L} = \frac{1}{\Delta} L = \frac{1}{\Delta} \left[y^1 - y^0 \cdots y^n - y^0 \right]^\top = \begin{bmatrix} \frac{y_1^1 - y_1^0}{\Delta} & \cdots & \frac{y_n^1 - y_n^0}{\Delta} \\ \vdots & \vdots & \vdots \\ \frac{y_1^n - y_1^0}{\Delta} & \cdots & \frac{y_n^n - y_n^0}{\Delta} \end{bmatrix}.$$

This matrix $\hat{L}$ appears in

$$M(\phi, Y_{scaled}) = \begin{bmatrix} 1 & 0^\top \\ e & \hat{L} \end{bmatrix}$$

for the shifted and scaled sample set $Y_{scaled} \subset B(0; 1)$.

Shifting and scaling

Given any sample set written as

$$Y \;=\; \{y^0, y^1, \ldots, y^p\},$$

one can shift it by $-y^0$ to center the new set at the origin:

$$\{0, y^1 - y^0, \ldots, y^p - y^0\}.$$

Then, one can consider

$$\Delta \;=\; \Delta(Y) \;=\; \max_{1 \le i \le p} \|y^i - y^0\|$$

and scale the set by Δ:

$$\{0, \hat{y}^1, \ldots, \hat{y}^p\} \;=\; \{0, (y^1 - y^0)/\Delta, \ldots, (y^p - y^0)/\Delta\} \;\subset\; B(0;1).$$

The resulting sample set Y_{scaled} is contained in $B(0;1)$ and has at least one point on the ball boundary.

Error bounds for linear interpolation

Theorem

The gradient of the linear interpolation model satisfies, for all points y in $B(y^0; \Delta)$, an error bound of the form

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\, \Delta,$$

where $\kappa_{eg} = \nu(1 + p^{\frac{1}{2}} \|\hat{L}^{-1}\|/2)$ and $\hat{L} = L/\Delta$.

The linear interpolation model satisfies, for all points y in $B(y^0; \Delta)$, an error bound of the form

$$|f(y) - m(y)| \leq \kappa_{ef}\, \Delta^2,$$

where $\kappa_{ef} = \kappa_{eg} + \nu/2$.

Error bounds for linear regression

In the regression case we are considering a sample set $Y = \{y^0, y^1, \ldots, y^p\}$ with more than $n+1$ points, contained in the ball $B(y^0; \Delta(Y))$ of radius

$$\Delta \;=\; \Delta(Y) \;=\; \max_{1 \le i \le p} \|y^i - y^0\|.$$

Assumption

We assume that $Y = \{y^0, y^1, \ldots, y^p\} \subset \mathbb{R}^n$, with $p > n$, is a poised set of sample points (in the linear regression sense) contained in the ball $B(y^0; \Delta(Y))$.

Further, we assume that the function f is continuously differentiable in an open domain Ω containing $B(y^0; \Delta)$ and ∇f is Lipschitz continuous in Ω with constant $\nu > 0$.

Error bounds for linear regression

The error bounds for the approximation are also derived in terms of the scaled matrix

$$\hat{L} = \frac{1}{\Delta} L = \frac{1}{\Delta} \left[y^1 - y^0 \cdots y^p - y^0 \right]^\top .$$

This matrix $\hat{L}$ appears in

$$\hat{M} = M(\phi, Y_{scaled}) = \begin{bmatrix} 1 & 0^\top \\ e & \hat{L} \end{bmatrix}$$

for the shifted and scaled sample set $Y_{scaled} \subset B(0; 1)$.

There is an ERROR in the statement and derivation of the errors bounds for linear regression in the IDFO book. See the errata of the first printing at `http://www.mat.uc.pt/~lnv/idfo`

Error bounds for linear regression

Theorem

The gradient of the linear regression model satisfies, for all points y in $B(y^0; \Delta)$, an error bound of the form

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\, \Delta,$$

where $\kappa_{eg} = \nu(1 + p^{\frac{1}{2}} \|\hat{M}^\dagger\|/2)$.

The linear regression model satisfies, for all points y in $B(y^0; \Delta)$, an error bound of the form

$$|f(y) - m(y)| \leq \kappa_{ef}\, \Delta^2,$$

where $\kappa_{ef} = 2\kappa_{eg} + \nu/2$.

In the current context where $\hat{M}$ is full column rank, we have $\hat{M}^\dagger = (\hat{M}^\top \hat{M})^{-1} \hat{M}^\top$.

Affine independence

The notion of poisedness for linear interpolation is closely related to the concept of affine independence in convex analysis.

The affine hull of a given set $S \subset \mathbb{R}^n$ is the smallest affine set containing S (meaning that is the intersection of all affine sets containing S).

The affine hull of a set is always uniquely defined and consists of all linear combinations of elements of S whose scalars sum up to one.

Definition

A set of $m+1$ points $Y = \{y^0, y^1, \ldots, y^m\}$ is said to be affinely independent if its affine hull $\operatorname{aff}(y^0, y^1, \ldots, y^m)$ has dimension m.

Affine independence

The dimension of an affine set is the dimension of the linear subspace parallel to it. So, we cannot have an affinely independent set in $\mathbb{R}^n$ with more than $n+1$ points.

Given an affinely independent set of points $\{y^0, y^1, \ldots, y^m\}$, we have that

$$\text{aff}(y^0, y^1, \ldots, y^m) \;=\; y^0 + \mathcal{L}(y^1 - y^0, \ldots, y^m - y^0),$$

where $\mathcal{L}(y^1 - y^0, \ldots, y^m - y^0)$ is the linear subspace of dimension m generated by the vectors $y^1 - y^0, \ldots, y^m - y^0$.

Associated with an affinely independent set of points $\{y^0, y^1, \ldots, y^m\}$ is the matrix

$$\left[y^1 - y^0 \cdots y^m - y^0\right] \;=\; L^\top,$$

whose rank must be equal to m.

Simplex

Similarly, the convex hull of a given set $S \subset \mathbb{R}^n$ is the smallest convex set containing S (meaning that is the intersection of all convex sets containing S).

The convex hull of a set is always uniquely defined and consists of all convex combinations of elements of S, i.e., of all linear combinations of elements of S whose scalars are nonnegative and sum up to one.

Definition

Given an affinely independent set of points $Y = \{y^0, y^1, \ldots, y^m\}$, its convex hull is called a simplex of dimension m.

Simplex

A simplex of dimension 0 is a point, of dimension 1 is a closed line segment, of dimension 2 is a triangle, and of dimension 3 is a tetrahedron.

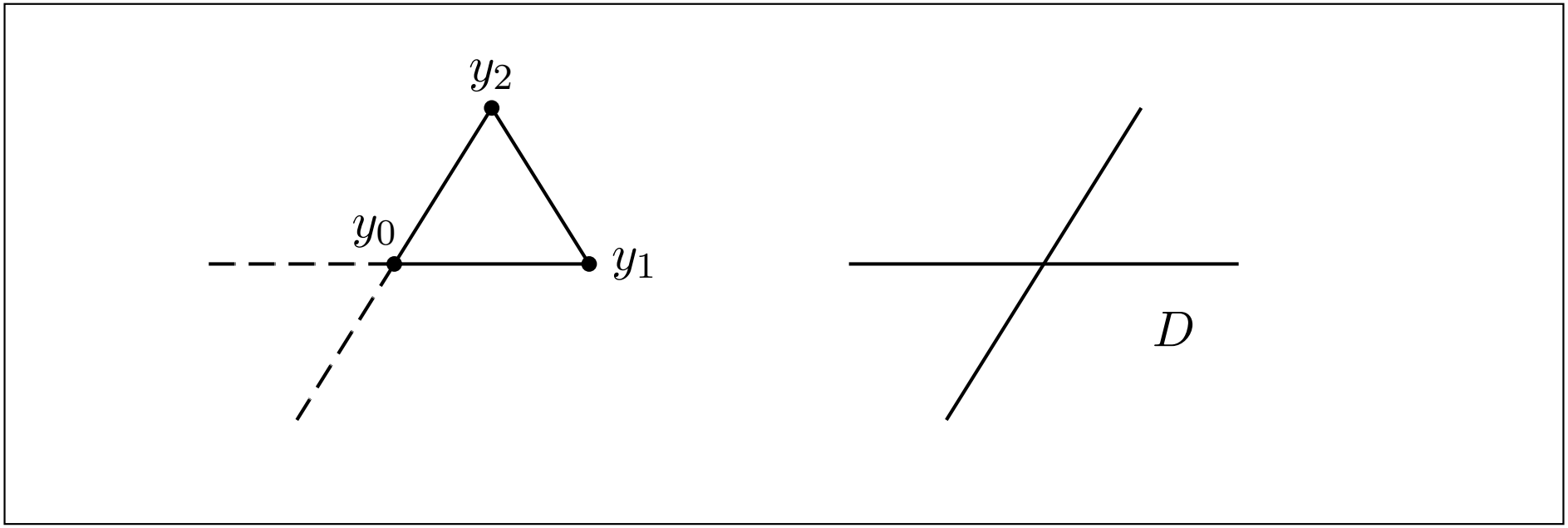
The vertices of a simplex are the elements of Y. A simplex in $\mathbb{R}^n$ cannot have more than $n+1$ vertices.

When there are $n+1$ vertices, its dimension is n. In this case,

$$\left[\, y^1 - y^0 \;\cdots\; y^n - y^0 \;\; -(y^1 - y^0) \;\cdots\; -(y^n - y^0) \,\right]$$

forms a (maximal) positive basis in $\mathbb{R}^n$.

Simplices and positive bases



Simplices

The diameter of a simplex Y of vertices $y^0, y^1, \ldots, y^m$ is defined by

$$\mathrm{diam}(Y) = \max_{0 \le i < j \le m} \|y^i - y^j\|.$$

One way of approximating $\mathrm{diam}(Y)$ at y^0 is by computing the less expensive quantity

$$\Delta(Y) = \max_{1 \le i \le n} \|y^i - y^0\|.$$

Clearly, we can write $\Delta(Y) \le \mathrm{diam}(Y) \le 2\Delta(Y)$.

By the shape of a simplex it is meant its equivalent class under similarity: the simplices of vertices Y and λY, $\lambda > 0$, share the same shape.

Simplices

The volume of a simplex of $n+1$ vertices $Y = \{y^0, y^1, \ldots, y^n\}$ is defined by

$$\mathrm{vol}(Y) \;=\; \frac{|\det(L)|}{n!},$$

where

$$L \;=\; L(Y) \;=\; \left[y^1 - y^0 \cdots y^n - y^0\right]^\top.$$

Since the vertices of a simplex form an affinely independent set, one clearly has that $\mathrm{vol}(Y) > 0$.

Simplices

The volume of a simplex is not a good measure for geometry since it is not scaling independent.

To see this let

$$Y_t = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} t \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ t \end{bmatrix} \right\},$$

with $t > 0$.

It is easy to see that

$$\mathrm{vol}(Y_t) \to 0 \quad \text{when} \quad t \to 0.$$

However, the angles between the vectors formed by the vertices are the same for all positive values of t (or putting it differently all these simplices have the same shape).

Simplices

A measure of the quality of a simplex geometry must be independent of the scale of the simplex, given by either $\Delta(Y)$ or $\mathrm{diam}(Y)$.

One such measure is given by

$$\|[L(Y)/\Delta(Y)]^{\dagger}\|,$$

which reduces to

$$\|[L(Y)/\Delta(Y)]^{-1}\|$$

for simplices of $n+1$ vertices.

One alternative when there are $n+1$ vertices is to work with the normalized volume

$$\mathrm{von}(Y) \;=\; \mathrm{vol}\left(\frac{1}{\mathrm{diam}(Y)}Y\right) \;=\; \frac{|\det(L(Y))|}{n!\,\mathrm{diam}(Y)^n}.$$

Poisedness and positive spanning

For a positive spanning set D formed by nonzero normalized vectors:

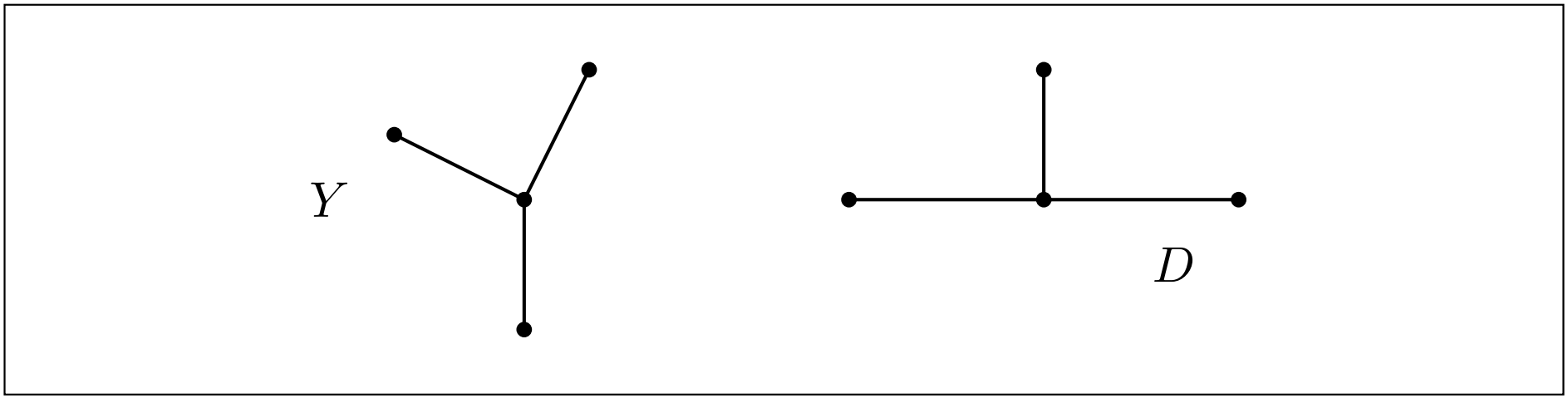
$$\mathrm{cm}(D) \;=\; \min_{\|v\|=1} \max_{d\in D} v^\top d \;\le\; \min_{\|v\|=1} \max_{d\in D} |v^\top d| \;\le\; \min_{\|v\|=1} \|D^\top v\|.$$

Thus, if $\mathrm{cm}(D) > 0$ then D has full row rank.

(next LEFT) Thus, given a point y^0 and a positive spanning set D for which $\mathrm{cm}(D) > 0$, we know that the sample set $\{y^0\} \cup \{y^0 + d : d \in D\}$ is poised for linear regression.

(next RIGHT) The contrary, however, is not true. Given a poised set $Y = \{y^0, y^1, \ldots, y^p\}$ for linear regression, the set of directions $\{y^1 - y^0, \ldots, y^p - y^0\}$ might not be a positive spanning set.

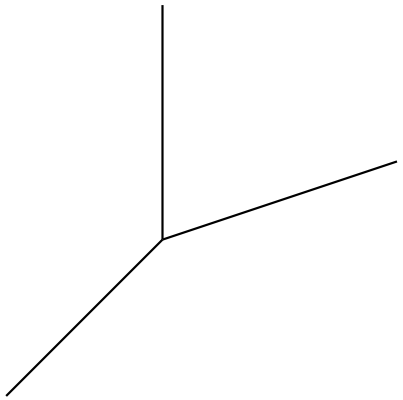
Poisedness and positive spanning



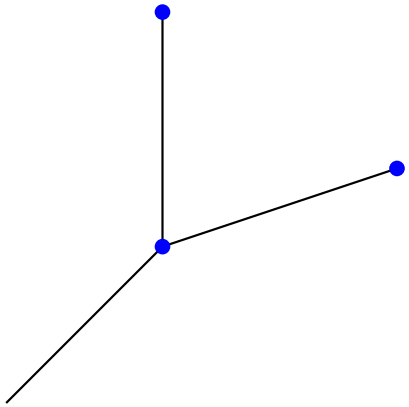
It is trivial to construct a counterexample (RIGHT). For instance, let us take $n = 2$, $p = 3$, $y^0 = (0, 0)$, and

$$\begin{bmatrix} y^1 & y^2 & y^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}.$$

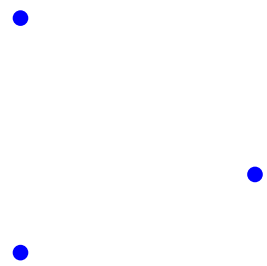
Recapitulation



Recapitulation



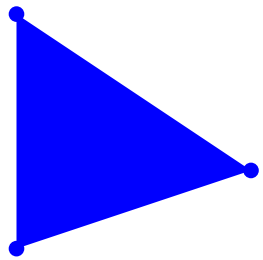
Recapitulation



The $3 = n + 1$ points form an affinely independent set.

A set of $m + 1$ points $Y = \{y^0, y^1, \ldots, y^m\}$ is said to be affinely independent if its affine hull $\mathrm{aff}(y^0, y^1, \ldots, y^m)$ has dimension m.

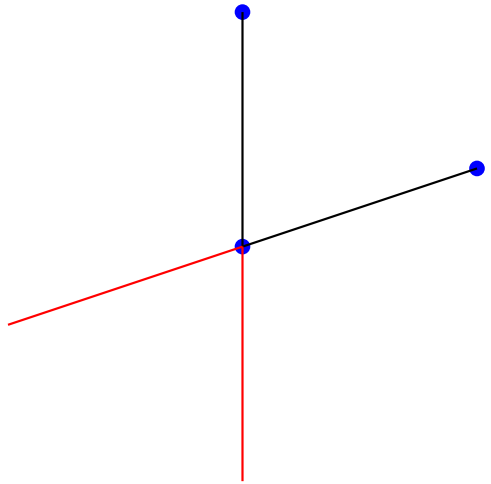
Recapitulation



Their convex hull is a simplex of dimension $n = 2$.

Given an affinely independent set of points $Y = \{y^0, y^1, \ldots, y^m\}$,
its convex hull is called a simplex of dimension m.

Recapitulation



Its reflection produces a (maximal) positive basis.

Simplex gradient

Given a set $Y = \{y^0, y^1, \ldots, y^n\}$ with $n+1$ sample points and poised for linear interpolation, the simplex gradient at y^0 is defined by

$$\nabla_s f(y^0) \;=\; L^{-1}\delta f(Y),$$

where

$$L \;=\; \left[y^1 - y^0 \cdots y^n - y^0\right]^\top \quad \text{and} \quad \delta f(Y) \;=\; \begin{bmatrix} f(y^1) - f(y^0) \\ \vdots \\ f(y^n) - f(y^0) \end{bmatrix}.$$

The simplex gradient is nothing else than the gradient of the linear interpolation model $m(x) = c + g^\top x$:

$$\nabla_s f(y^0) \;=\; g.$$

Simplex gradient

The simplex gradient (based on $n+1$ affinely independent points) is the gradient of the corresponding linear interpolation model:

$$\begin{aligned} f(y^0) + \langle \nabla_s f(y^0), y^i - y^0 \rangle &= f(y^0) + \left(L^{-1}\delta f(Y)\right)^\top \left(L^\top e_i\right) \\ &= f(y^0) + \delta f(Y)_i \\ &= f(y^i). \end{aligned}$$

$\longrightarrow$ Simplex derivatives are the derivatives of the polynomial models.

Simplex gradient

When the number of sample points exceeds $n+1$, simplex gradients are defined in a regression sense as the least-squares solution of

$$L\nabla_s f(y^0) = \delta f(Y),$$

where

$$L = \left[y^1 - y^0 \cdots \cdots y^p - y^0\right]^\top \text{ and } \delta f(Y) = \begin{bmatrix} f(y^1) - f(y^0) \\ \vdots \\ \vdots \\ f(y^p) - f(y^0) \end{bmatrix}.$$

Again, one points out that a simplex gradient defined in this way is the gradient g of the linear regression model $m(x) = c + g^\top x$.

We note that simplex gradients when $p > n$ are also referred as stencil gradients. The set $\{y^1, \ldots, y^p\}$ is a stencil centered at y^0.

Error bounds for simplex gradients

Under the assumptions stated for linear interpolation and linear regression, the simplex gradient satisfies an error bound of the form

$$\|\nabla f(y^0) - \nabla_s f(y^0)\| \ \leq \ \kappa_{eg}\, \Delta,$$

where $\kappa_{eg} = p^{\frac{1}{2}} \nu \|\hat{L}^{\dagger}\|/2$ and $\hat{L} = L/\Delta$.

In the case $p = n$, one has $\|\hat{L}^{\dagger}\| = \|\hat{L}^{-1}\|$.

Presentation outline

Direct-search methods

Definition (general)

Direct-search methods are DFO methods that sample the objective function at a finite number of points at each iteration and decide which actions to take next solely based on those function values and without any explicit or implicit derivative approximation or model building.

Definition (of directional type)

- *Sample the objective function at a finite number of points at each iteration.*
- *Achieve descent by moving in directions of potential descent.*
- *In the smooth case, and when the methods are deterministic, these directions lie in positive spanning sets (PSS).*

A class of direct-search methods

Choose: x_0 and α_0.

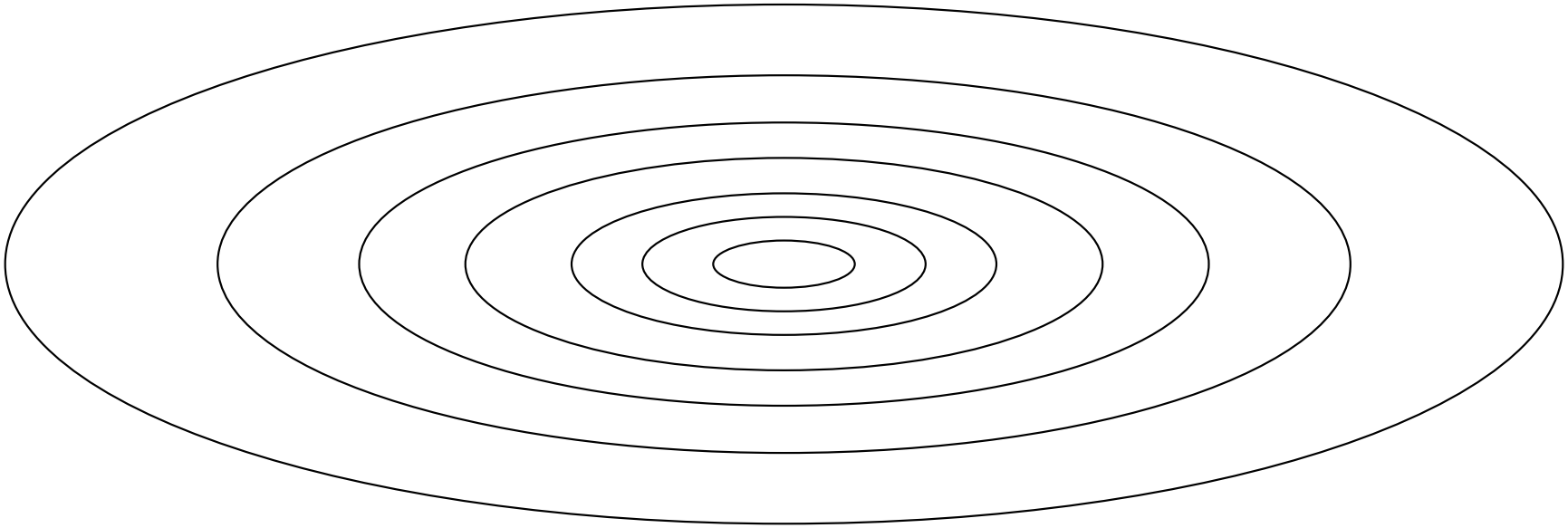
For $k = 0, 1, 2, \ldots$ (Until α_k is suff. small)

- **Search step (optional)**
- **Poll step:** Select D_k PSS and find $x_k + \alpha_k d_k$ ($d_k \in D_k$):

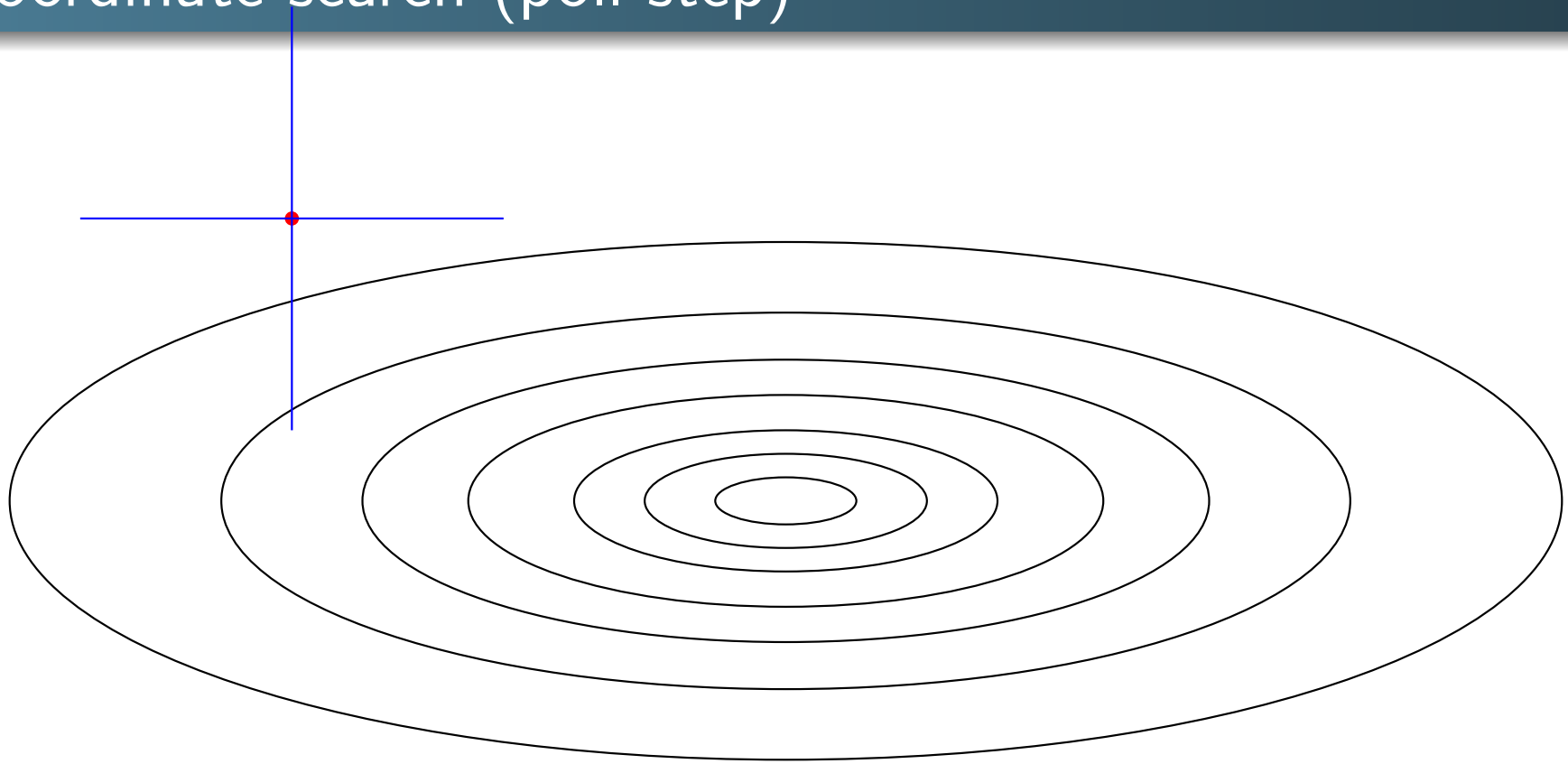
$$f(x_k + \alpha_k d_k) < f(x_k) - \rho(\alpha_k).$$

- Update the new iterate x_{k+1} (stay at x_k is unsuccessful).
- Update the step size α_{k+1}.
  Possible increase if iteration is successful. Decrease otherwise.

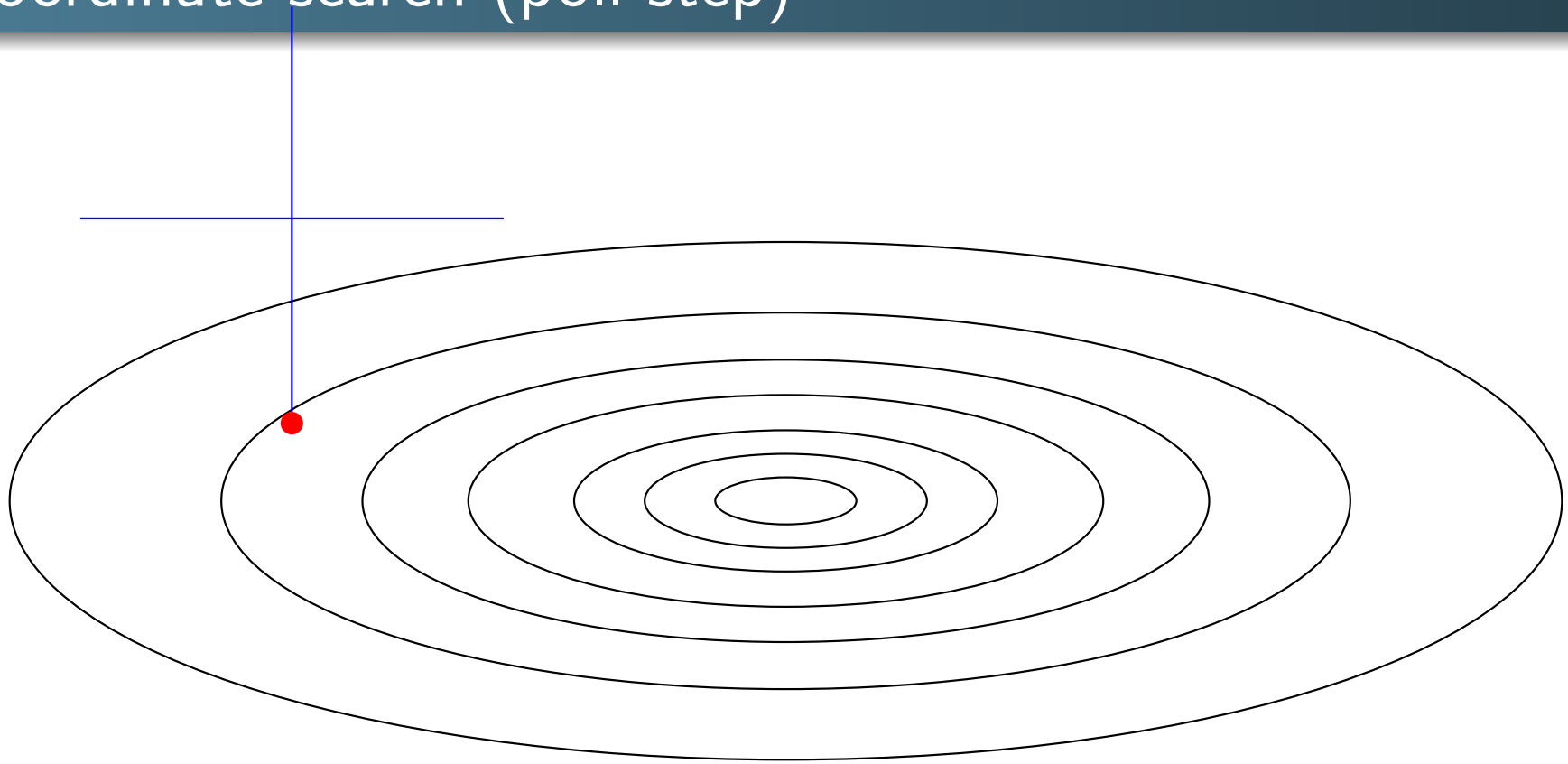
Coordinate search (poll step)



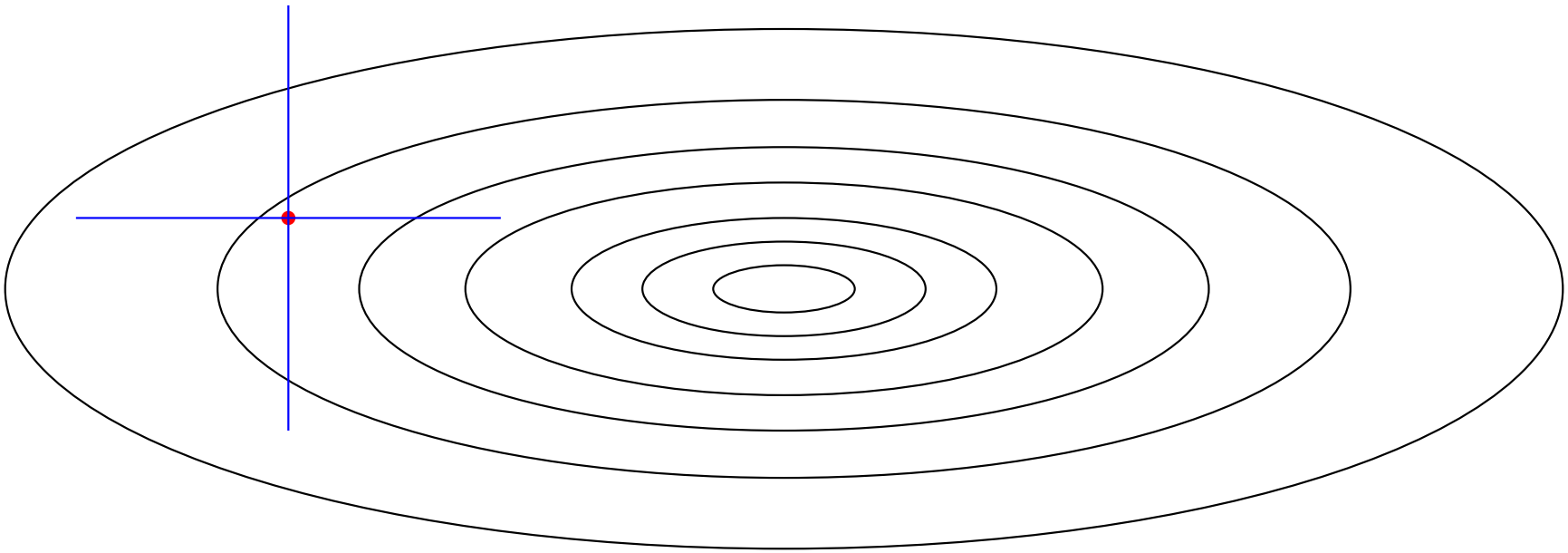
Coordinate search (poll step)



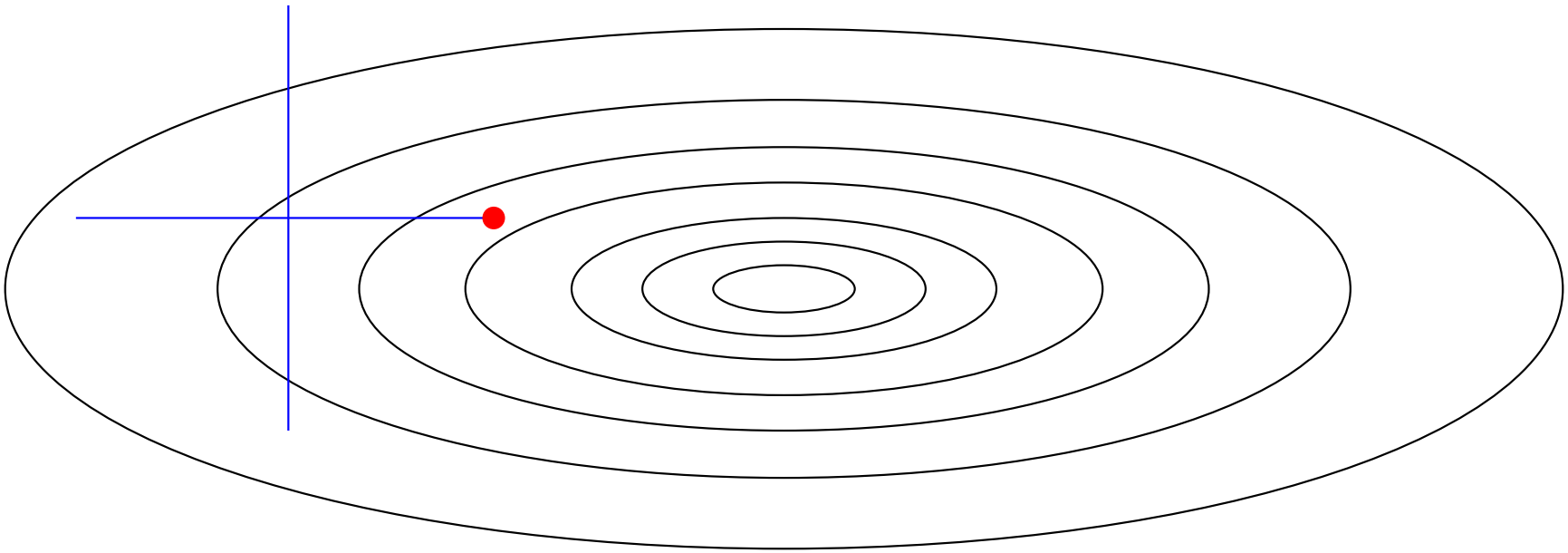
Coordinate search (poll step)



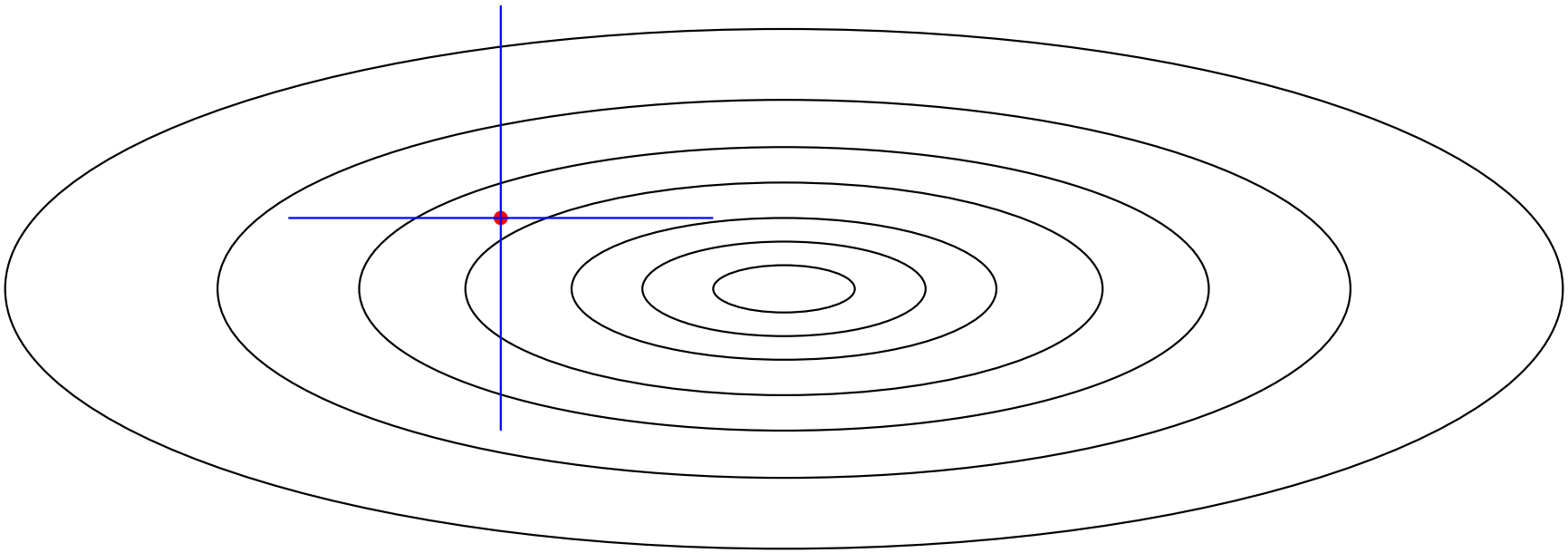
Coordinate search (poll step)



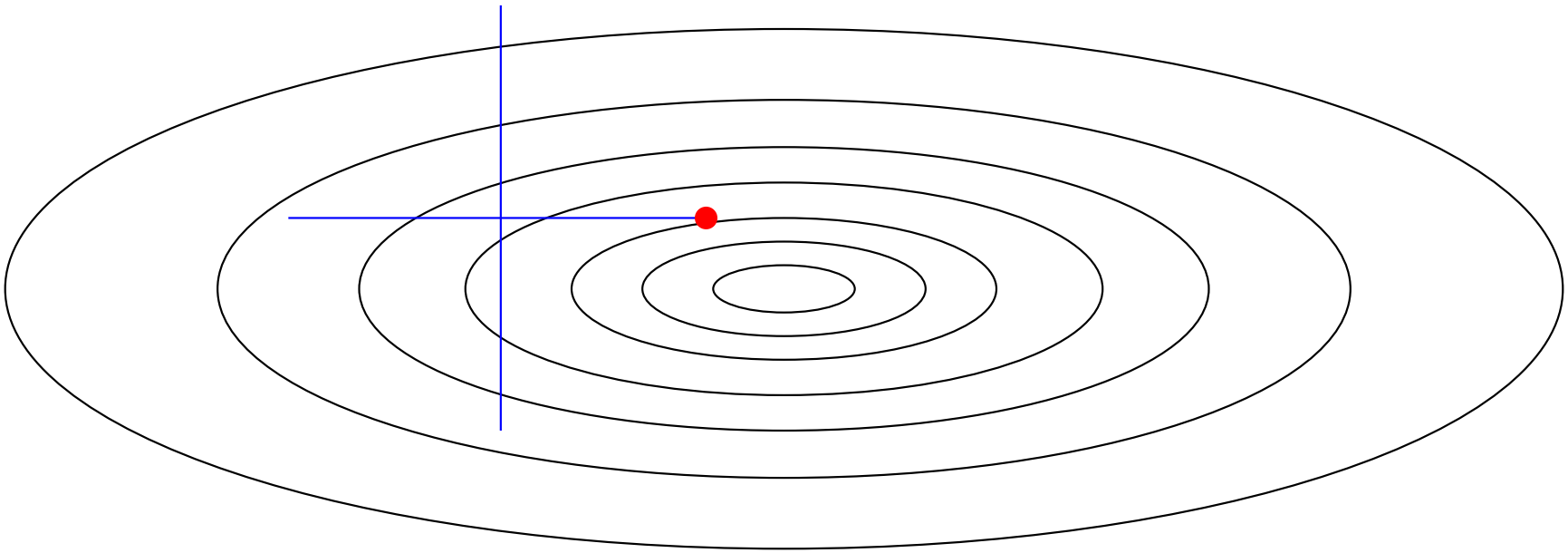
Coordinate search (poll step)



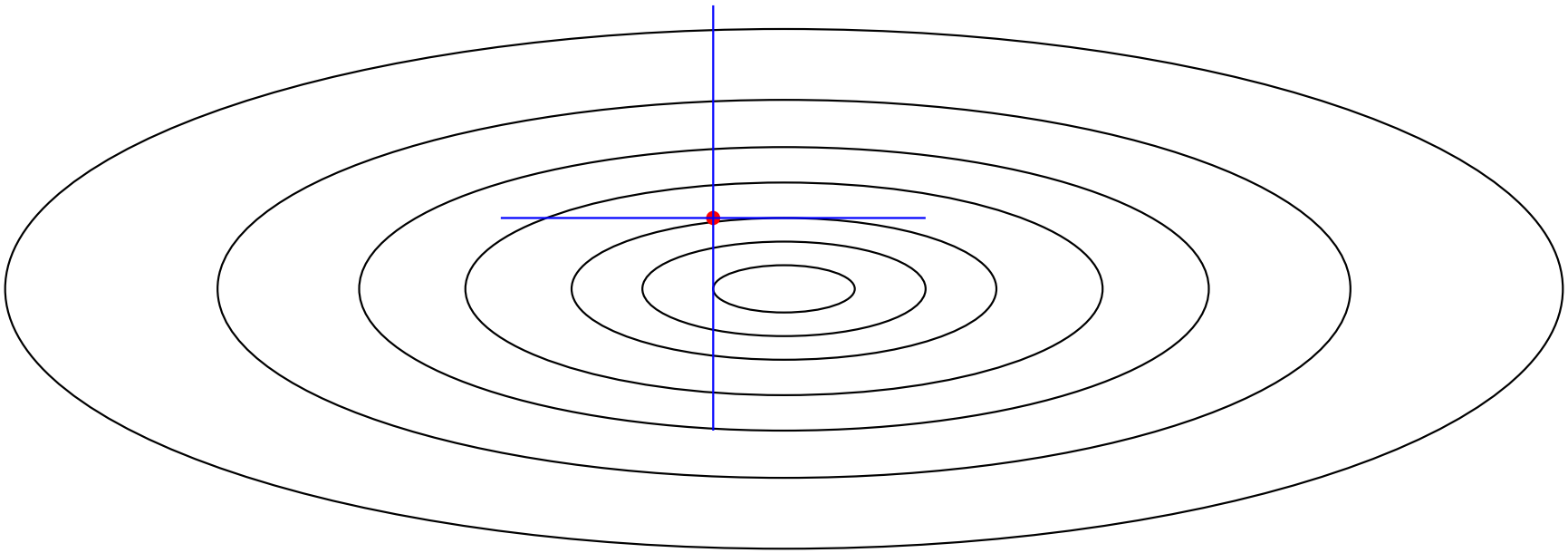
Coordinate search (poll step)



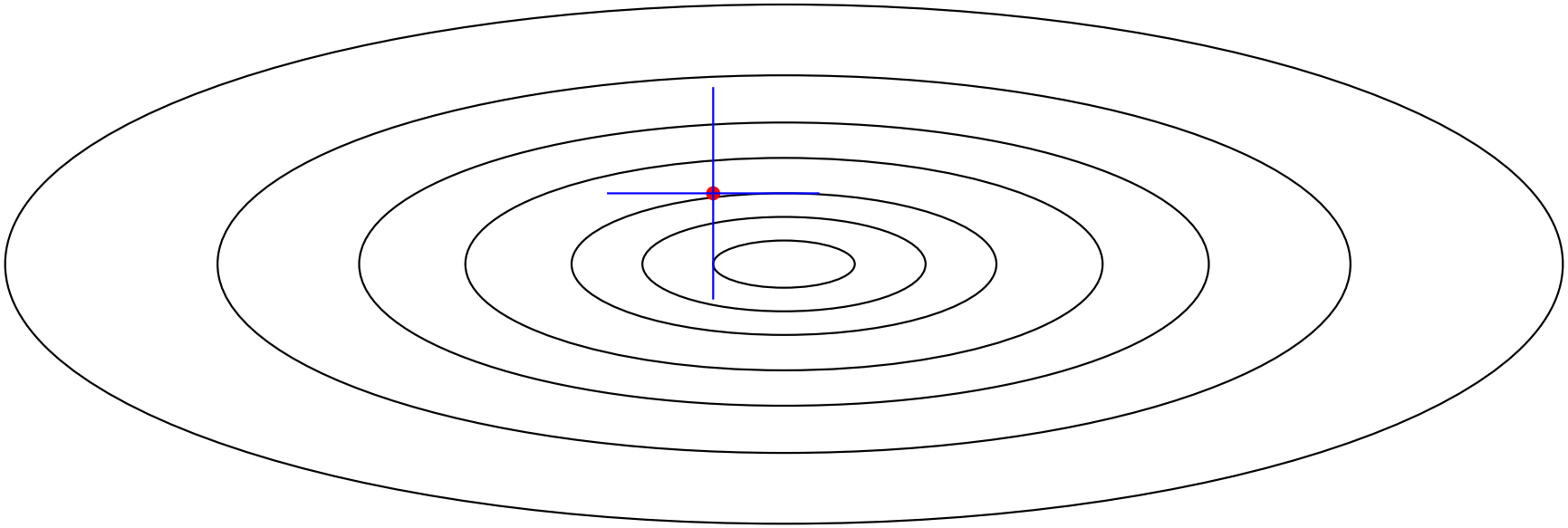
Coordinate search (poll step)



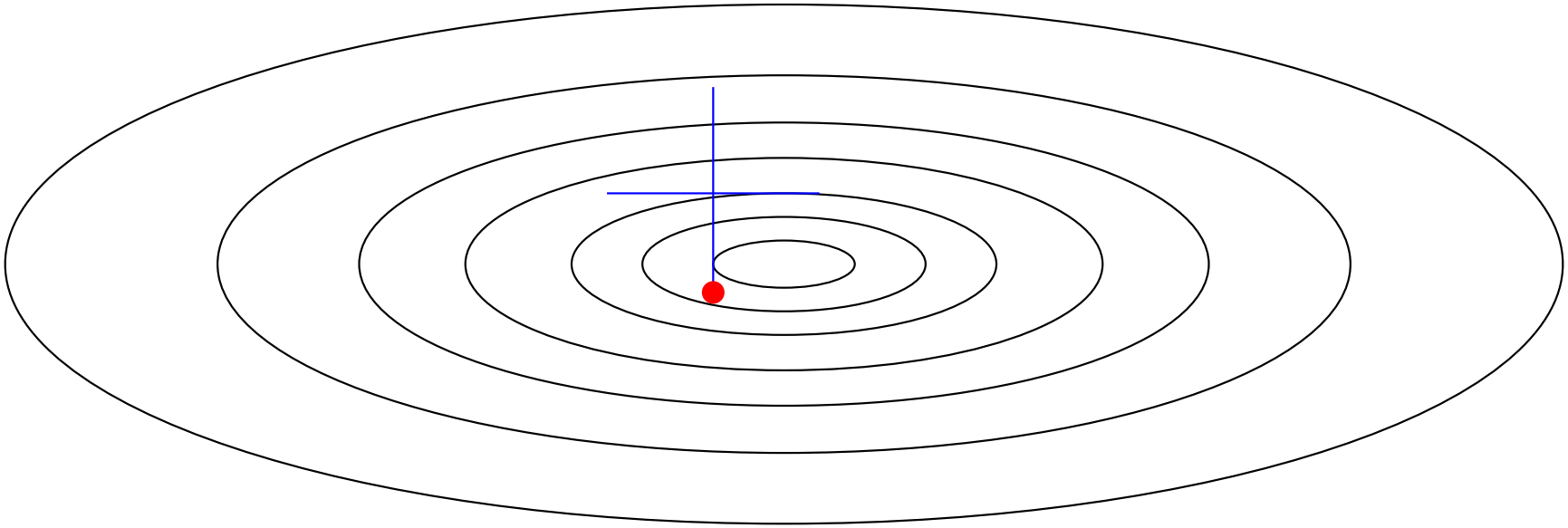
Coordinate search (poll step)



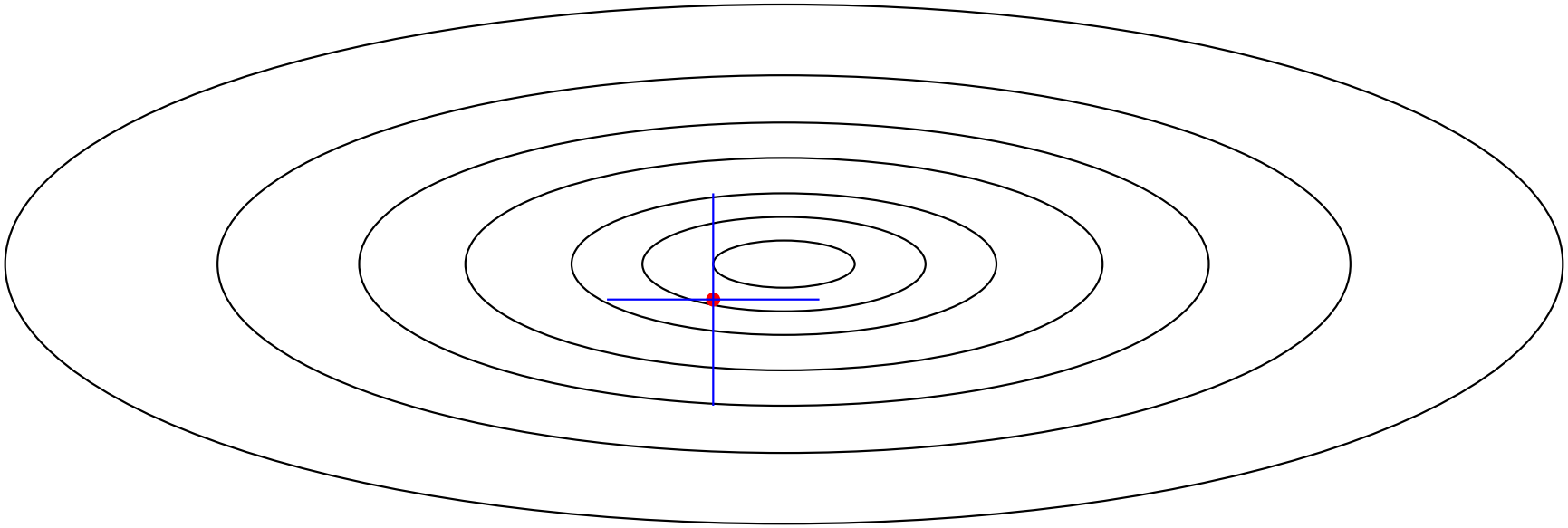
Coordinate search (poll step)



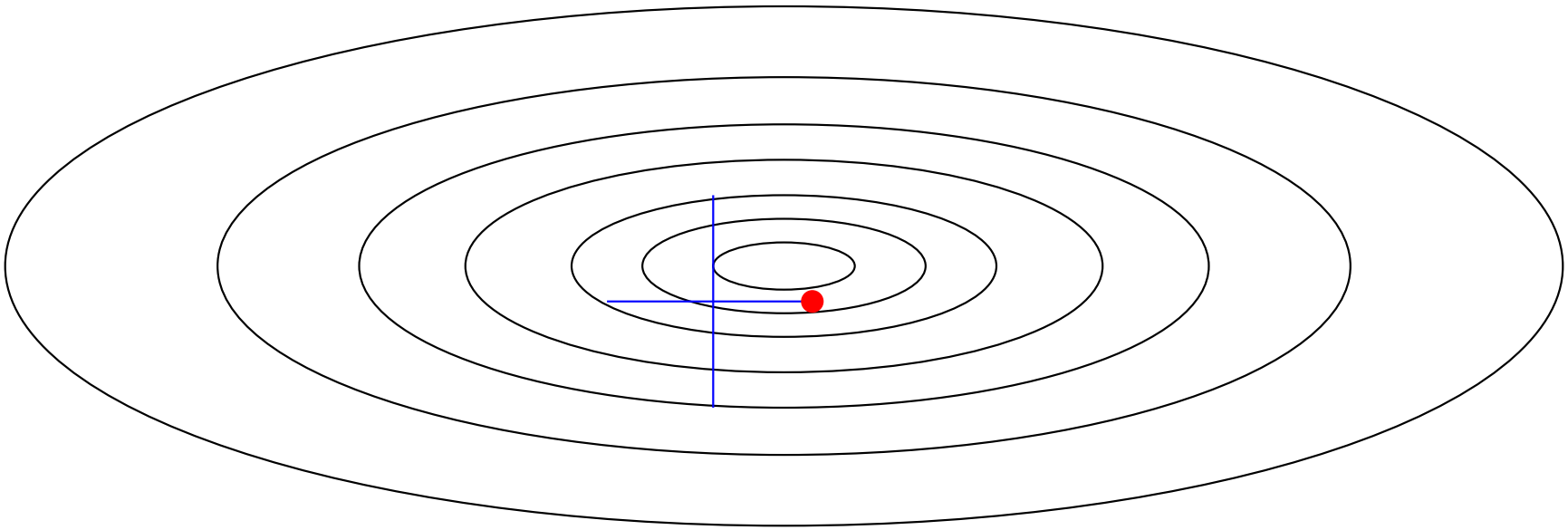
Coordinate search (poll step)



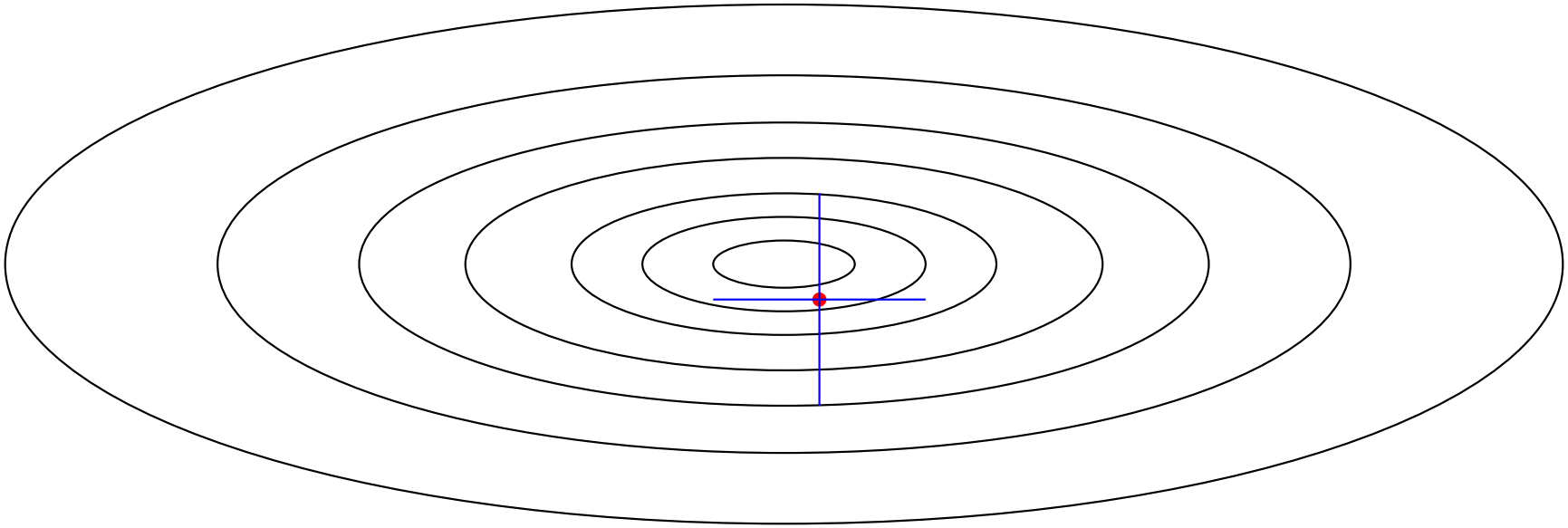
Coordinate search (poll step)



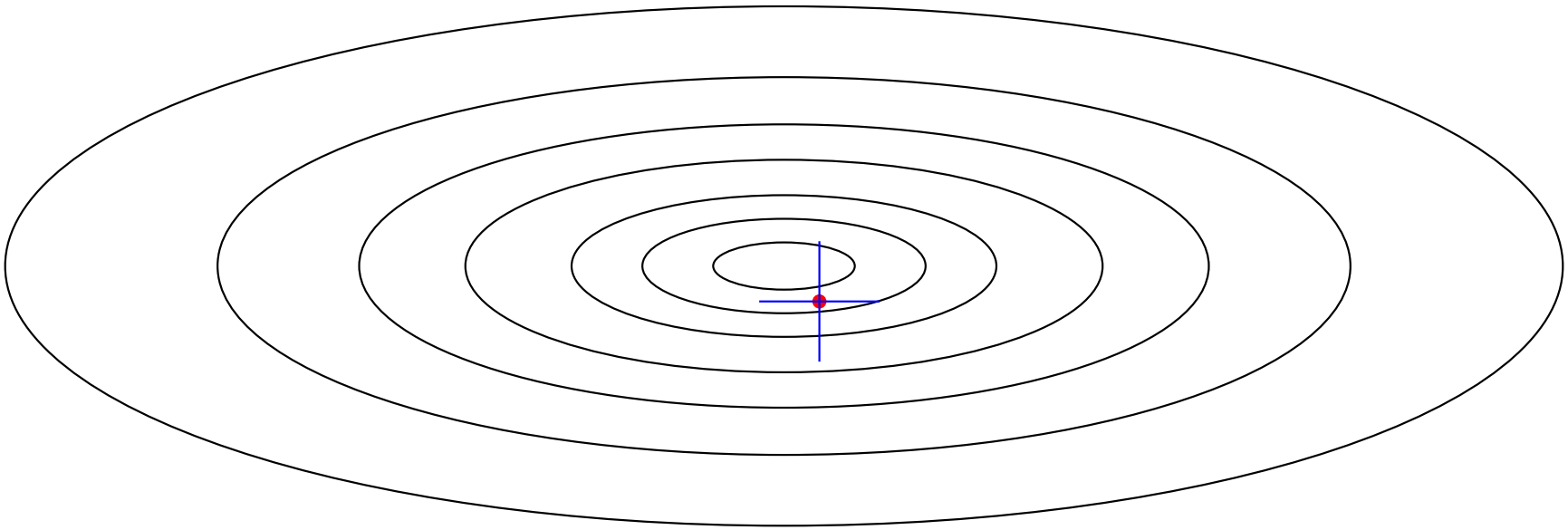
Coordinate search (poll step)



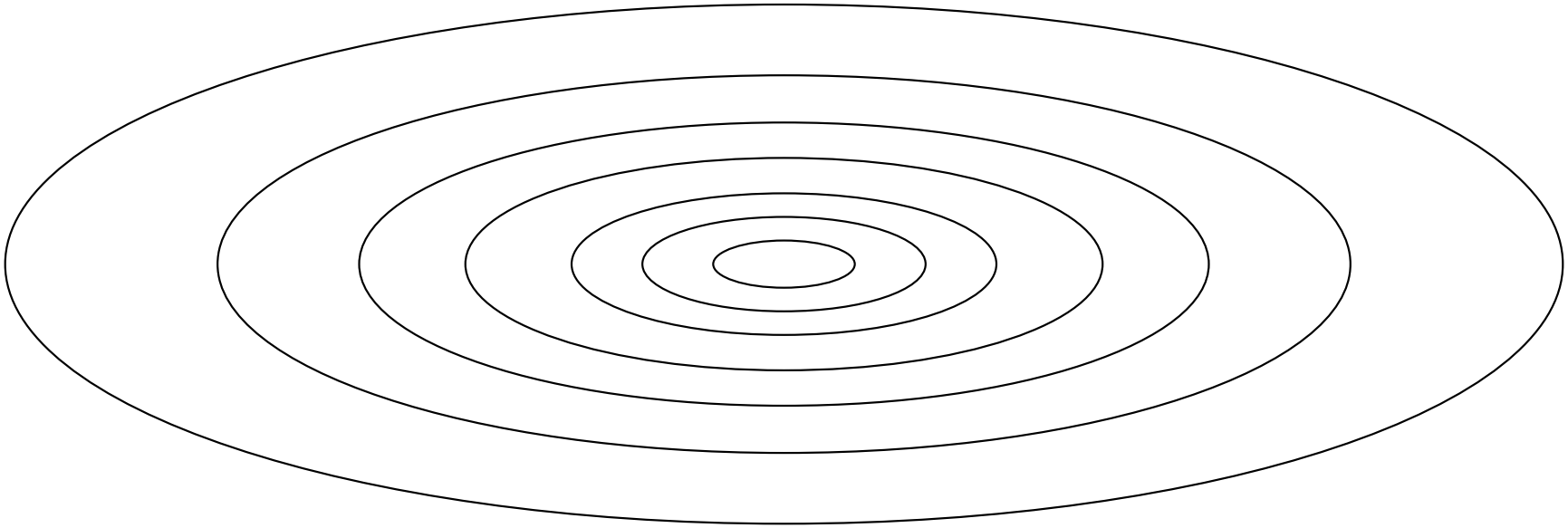
Coordinate search (poll step)



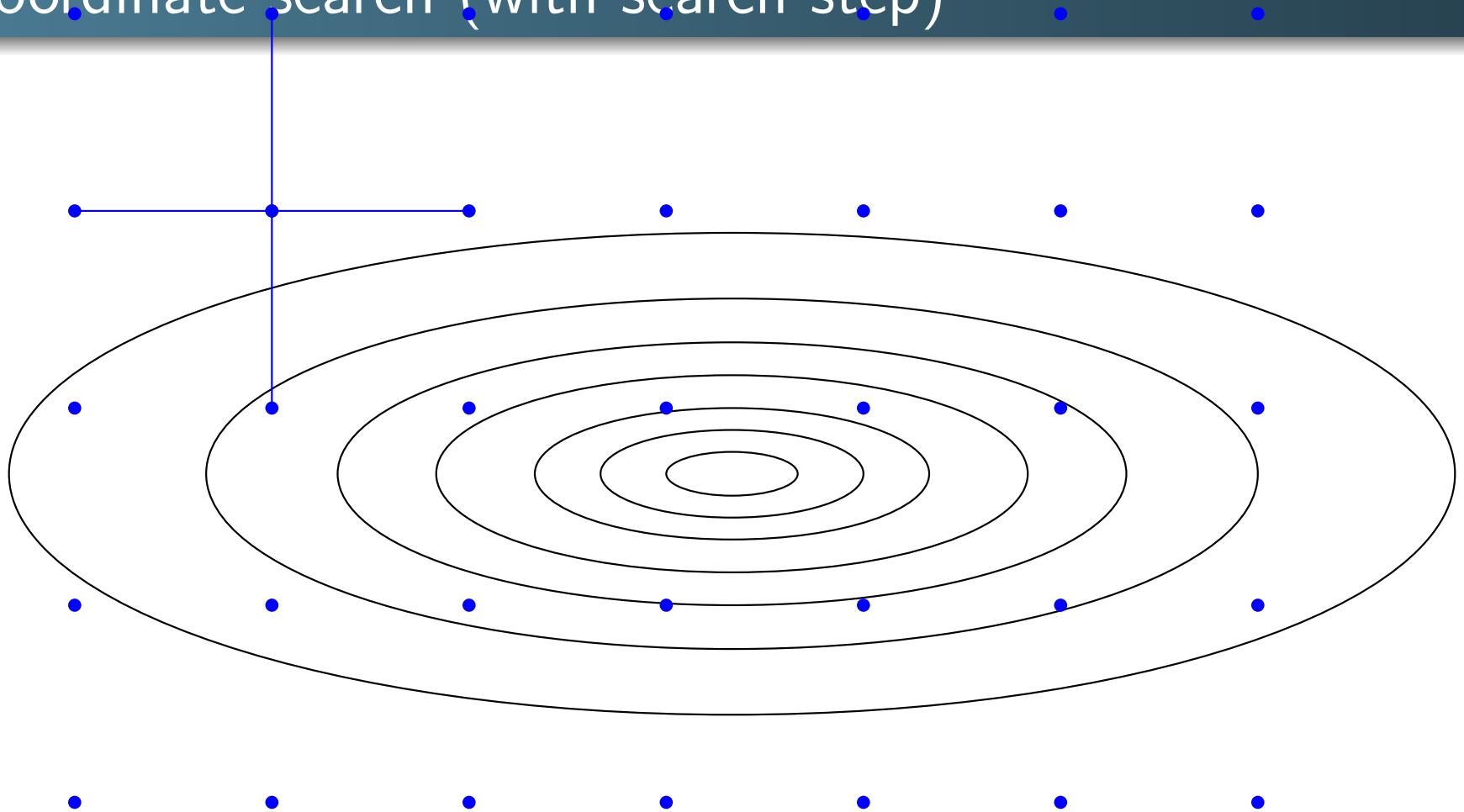
Coordinate search (poll step)



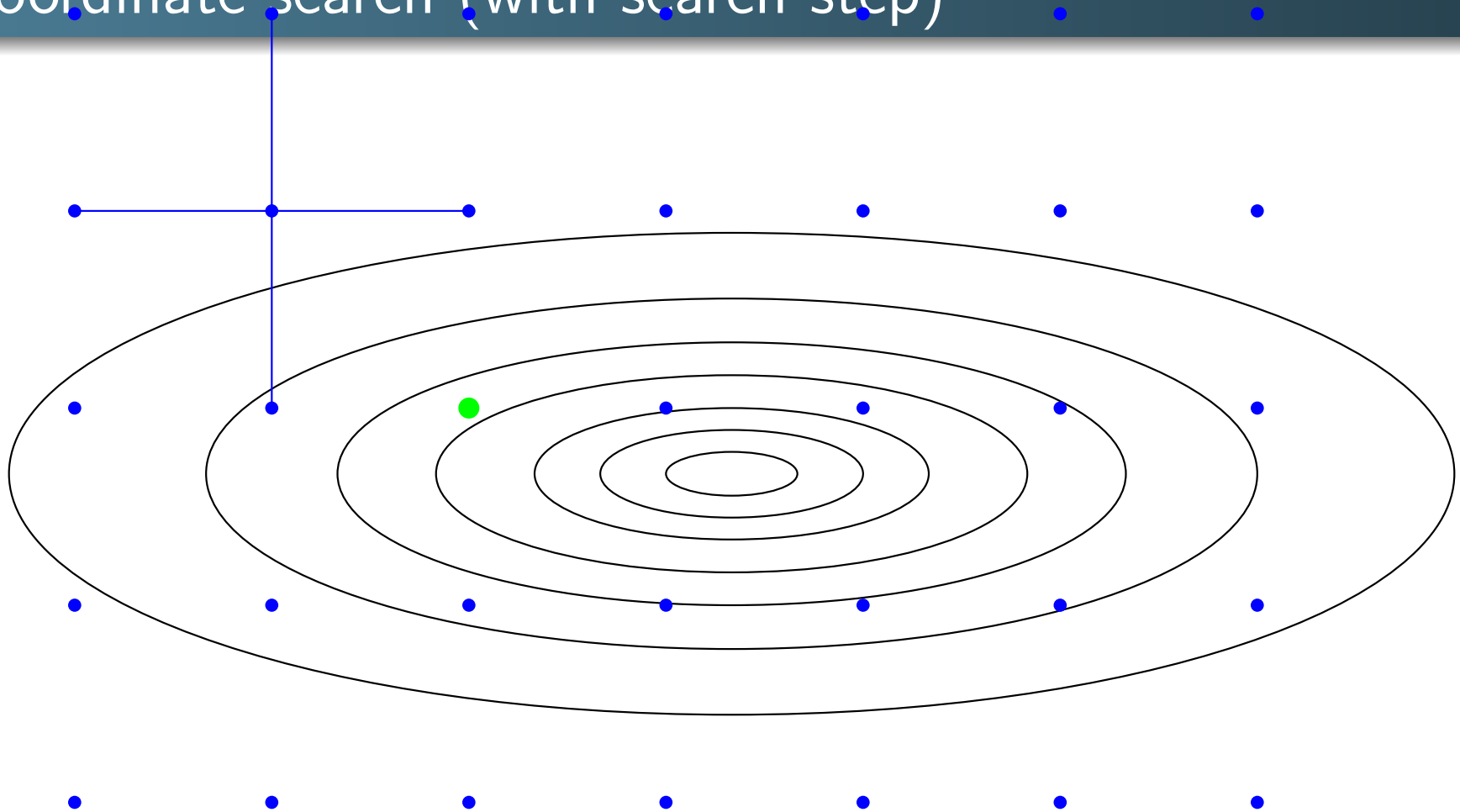
Coordinate search (with search step)



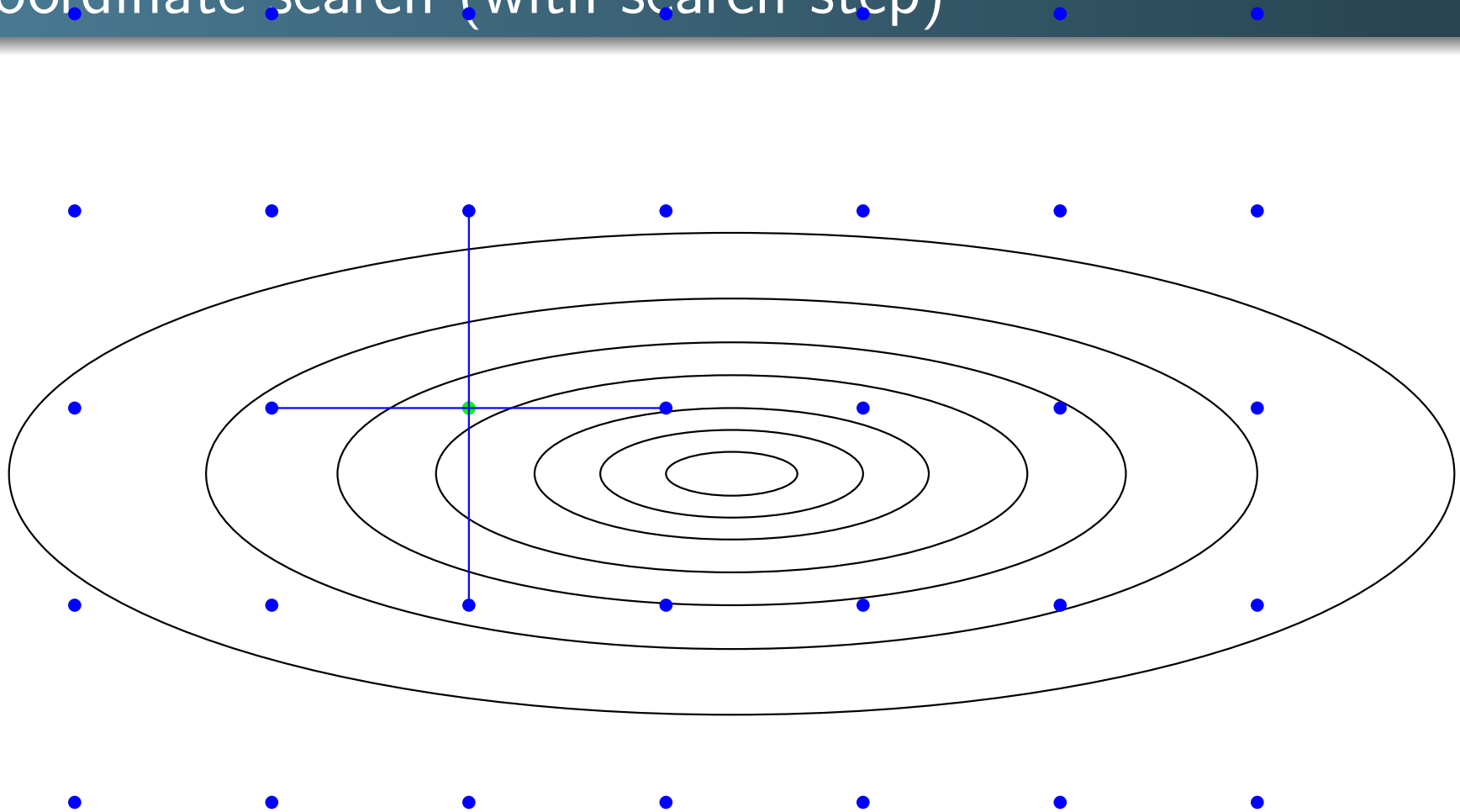
Coordinate search (with search step)



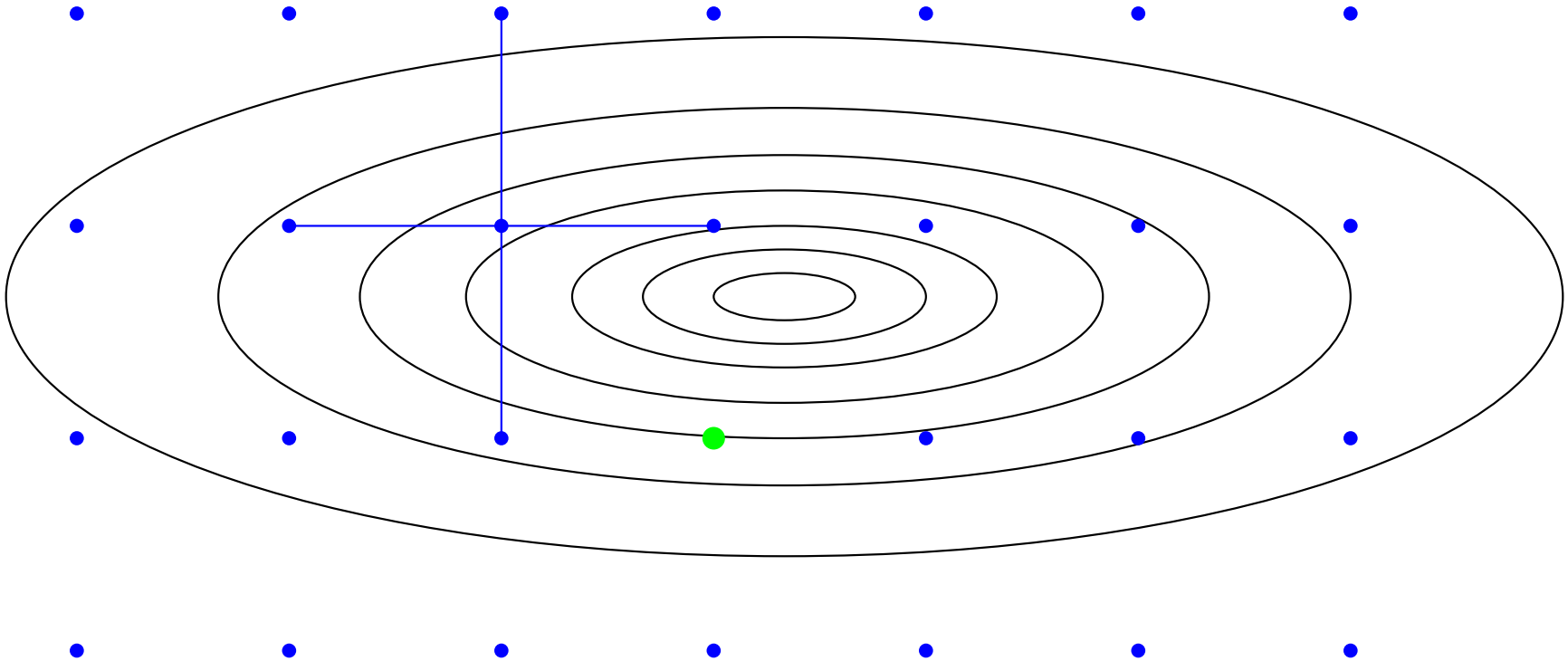
Coordinate search (with search step)



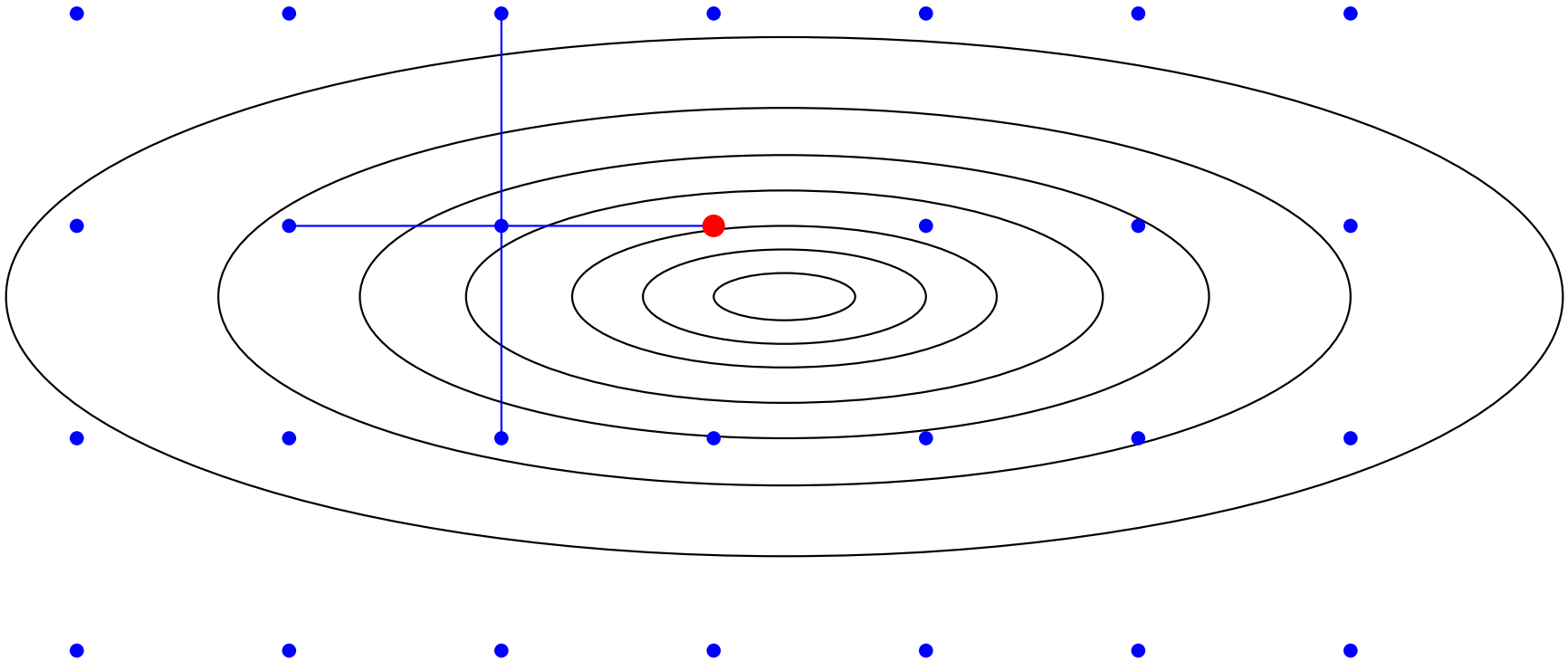
Coordinate search (with search step)



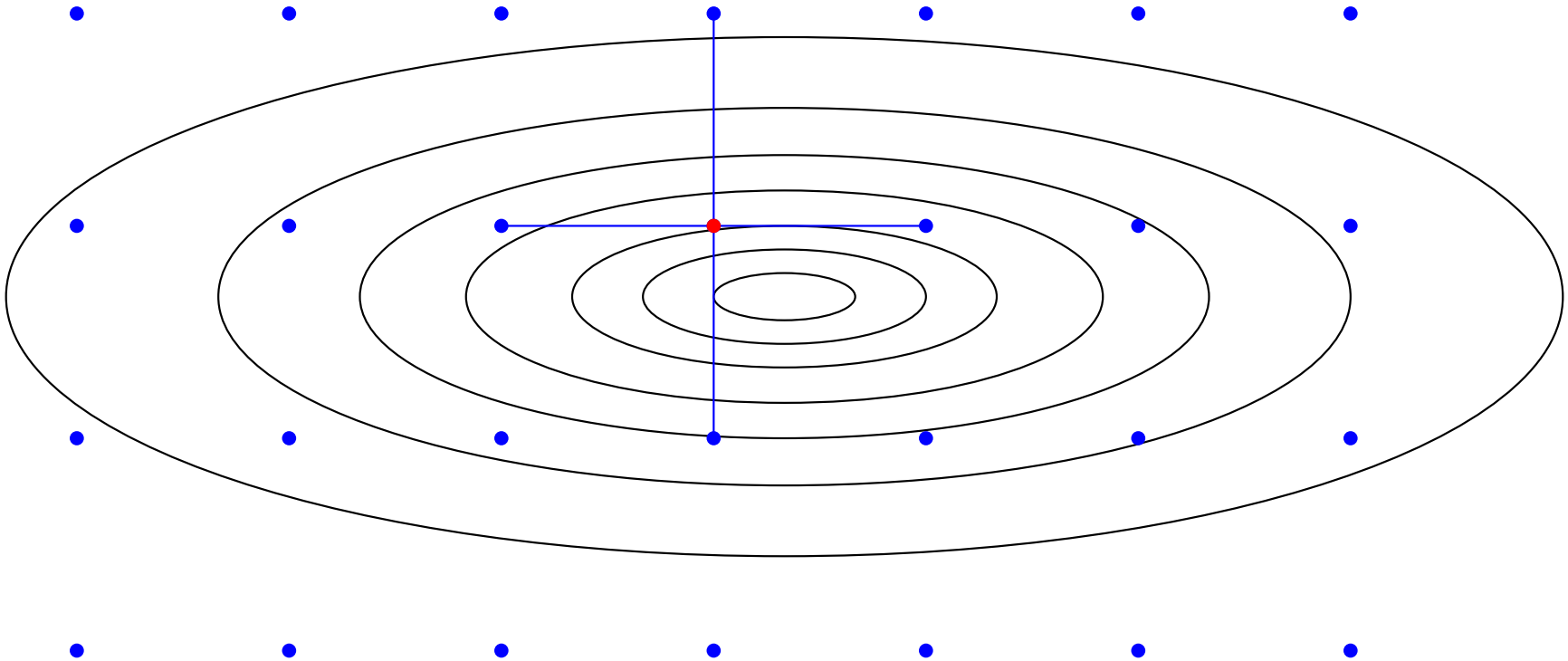
Coordinate search (with search step)



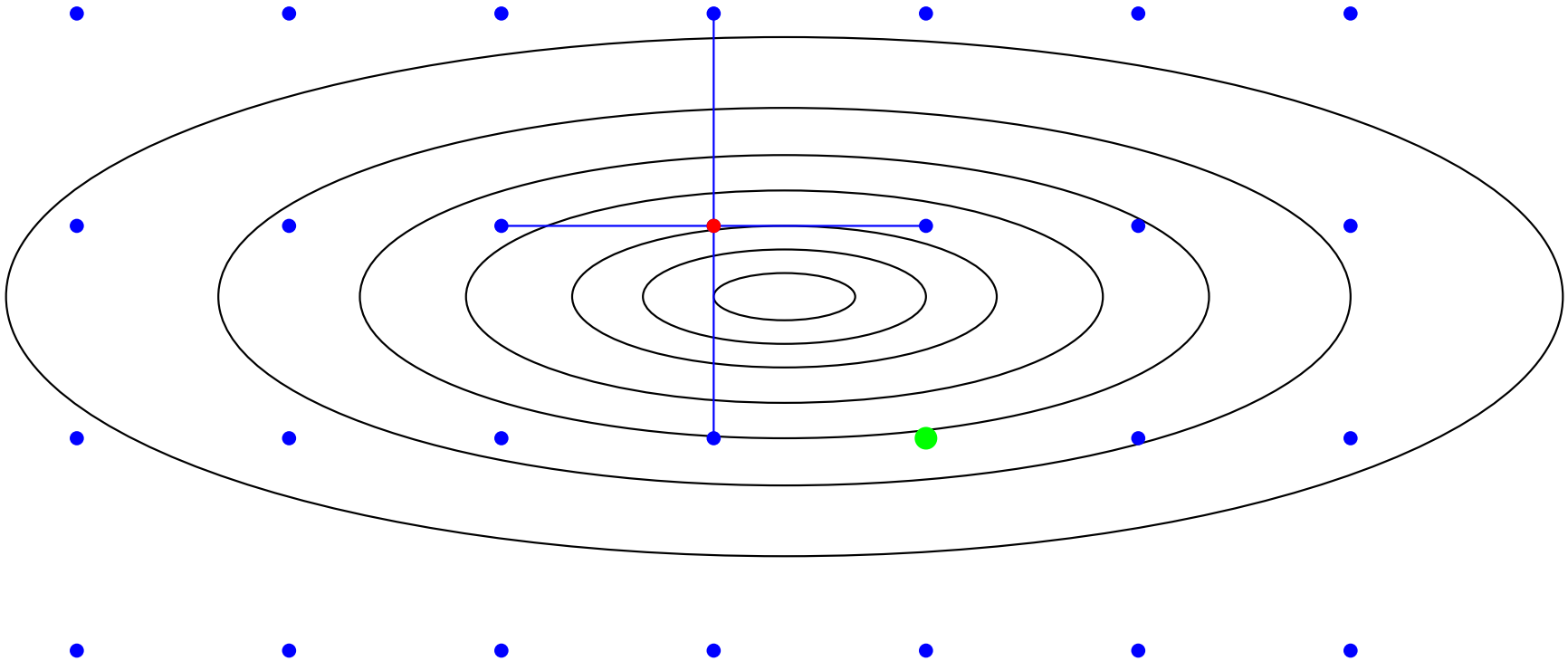
Coordinate search (with search step)



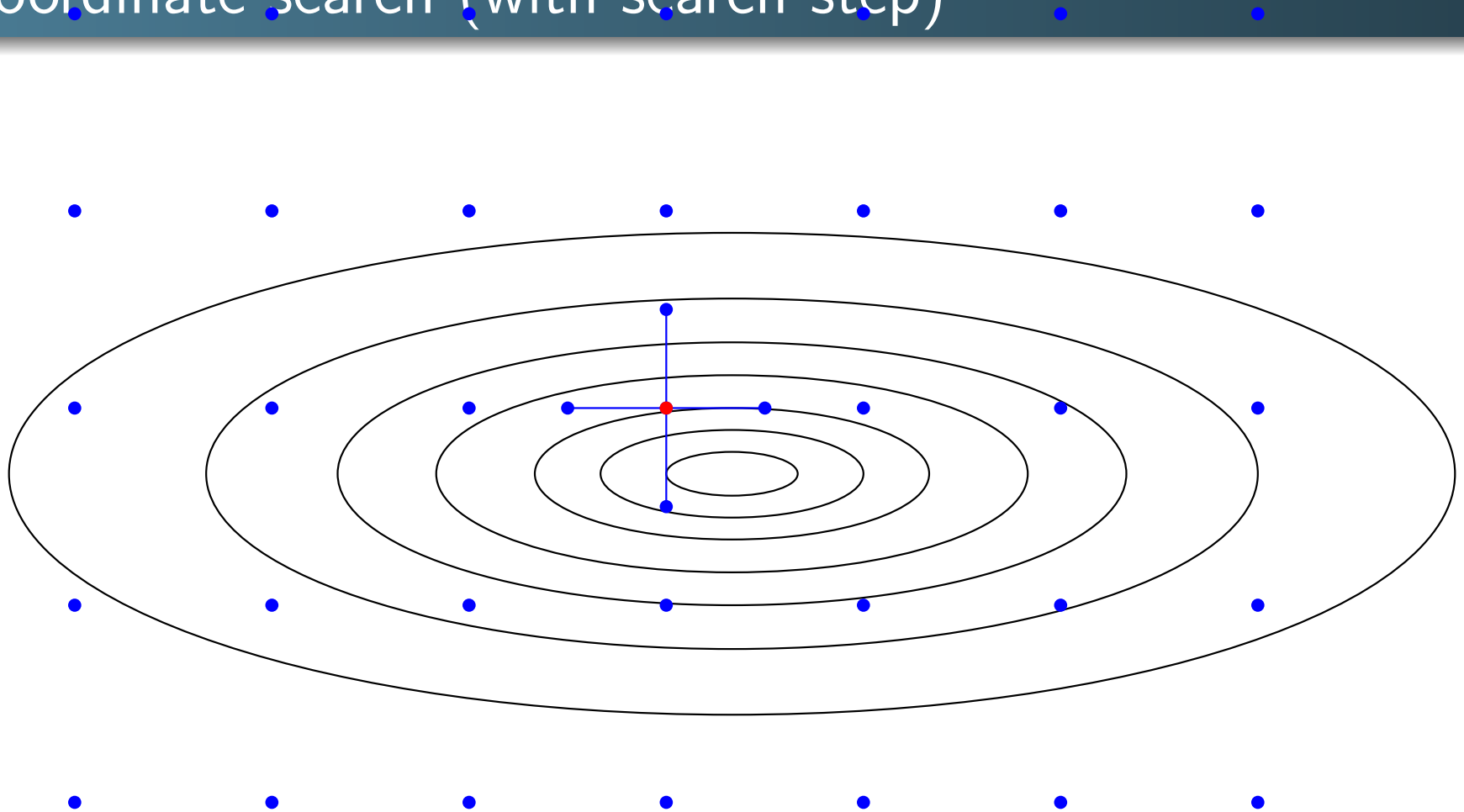
Coordinate search (with search step)



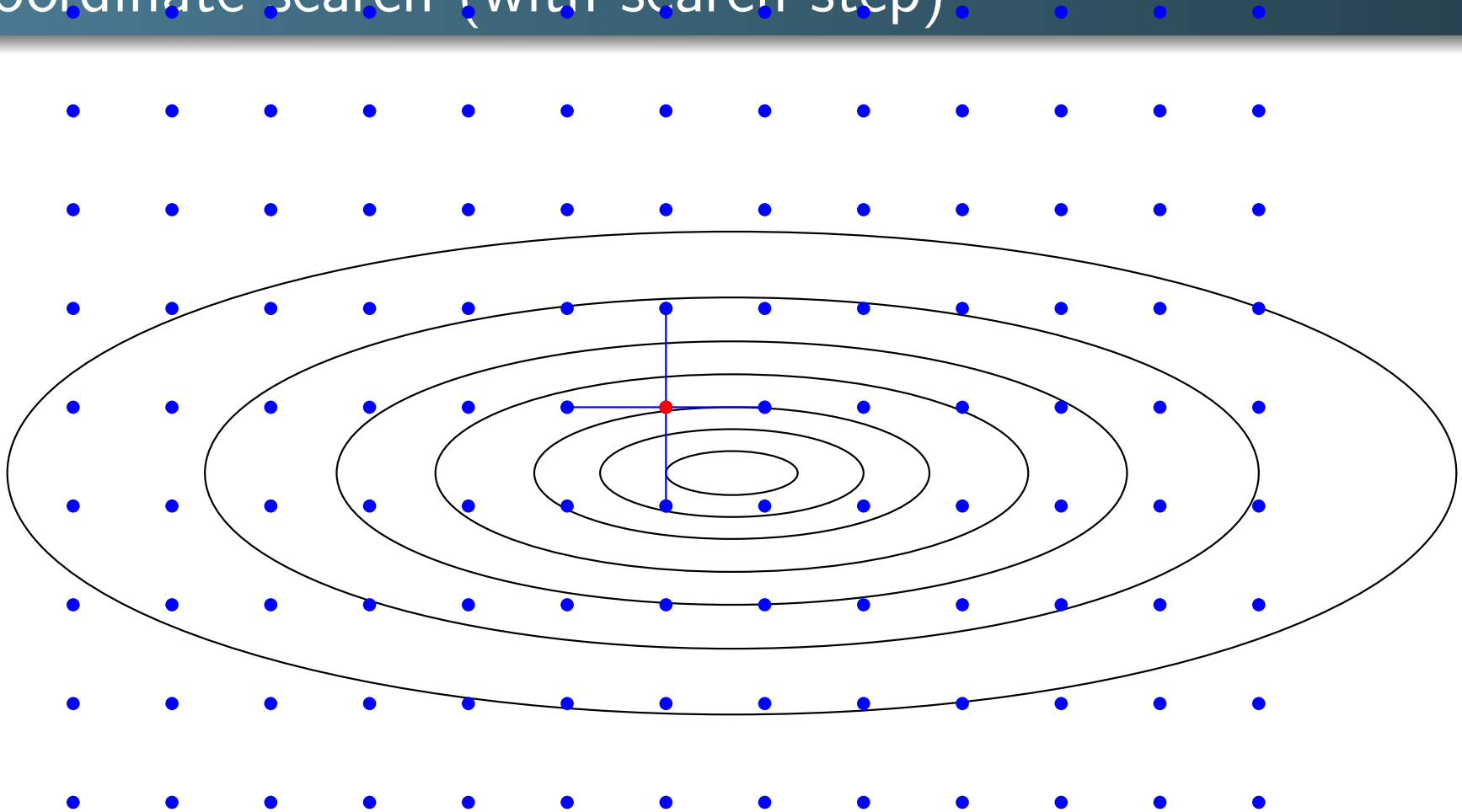
Coordinate search (with search step)



Coordinate search (with search step)



Coordinate search (with search step)



Poll step

The poll step is only performed if the search step has been unsuccessful.

It consists of a local search around the current iterate, exploring a set of points P_k defined by the step size parameter α_k and by a PSS.

The points $x_k + \alpha_k d \in P_k$ are called the poll points and the vectors $d \in D_k$ the poll vectors or poll directions.

The purpose of the poll step is to ensure a decrease of the objective function for a sufficiently small step size parameter α_k.

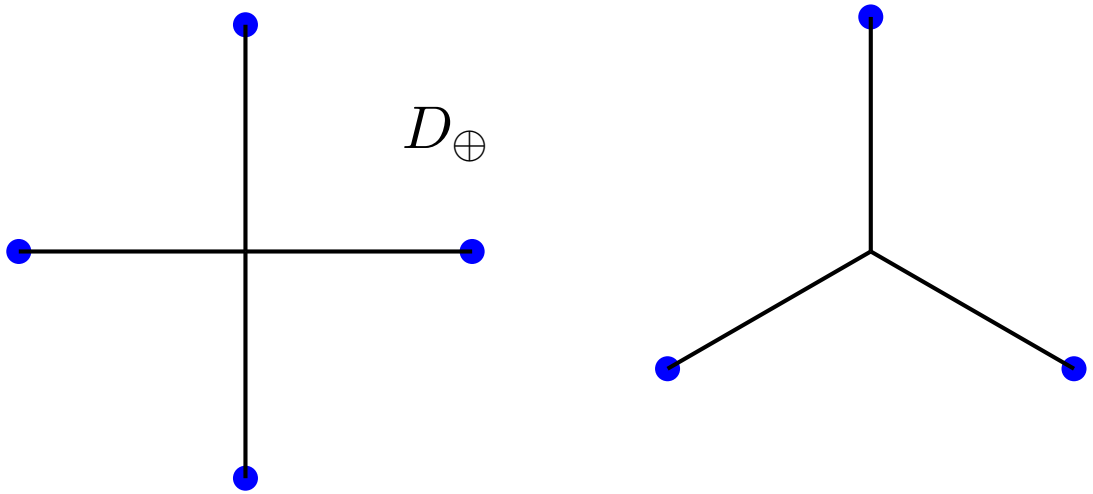
Poll step

The poll step is only performed if the search step has been unsuccessful.

It consists of a local search around the current iterate, exploring a set of points P_k defined by the step size parameter α_k and by a PSS.

The points $x_k + \alpha_k d \in P_k$ are called the poll points and the vectors $d \in D_k$ the poll vectors or poll directions.

The purpose of the poll step is to ensure a decrease of the objective function for a sufficiently small step size parameter α_k.



Poll step

The poll step is only performed if the search step has been unsuccessful.

It consists of a local search around the current iterate, exploring a set of points P_k defined by the step size parameter α_k and by a PSS.

The points $x_k + \alpha_k d \in P_k$ are called the poll points and the vectors $d \in D_k$ the poll vectors or poll directions.

The purpose of the poll step is to ensure a decrease of the objective function for a sufficiently small step size parameter α_k.

Poll step

The poll step and the current iteration are declared successful if a new point $x_{k+1} \in P_k$ is found such that $f(x_{k+1}) < f(x_k) - \rho(\alpha_k)$.

If the poll step fails to produce a point in P_k where the objective function is lower than $f(x_k) - \rho(\alpha_k)$, then both the poll step and the iteration are declared unsuccessful.

In these circumstances the step size parameter α_k is decreased.

Poll step

Polling can be opportunistic, moving to the first encountered better point, or complete in which case all the poll points are evaluated and the best point is taken (if better than the current iterate).

Efficient procedures to order the poll directions include:

- ordering according to the angle proximity to a negative simplex gradient,
- random ordering,
- ordering following the original order (when using the same PSS throughout the iterations) but avoiding restarts at new poll iterations
- and combinations of these strategies.

Complete polling is particularly attractive for running on a parallel environment.

Search step

The search step consists of evaluating the objective function at a finite number of points.

The search step is optional and is not necessary for the convergence properties of the method.

The search step and the current iteration are declared successful if a new point x_{k+1} is found such that $f(x_{k+1}) < f(x_k) - \rho(\alpha_k)$.

The search step can take advantage of the existence of surrogate models for f to improve the efficiency of the direct-search method.

A class of direct-search methods

Step size update: If the iteration was successful then maintain or increase the step size parameter: $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$.

Otherwise decrease the step size parameter: $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$.

The parameters are chosen at initialization: $0 < \beta_1 \leq \beta_2 < 1$, and $\gamma \geq 1$.

A natural stopping criterion (see later...) is to terminate a run when $\alpha_k < \alpha_{tol}$, for a chosen tolerance $\alpha_{tol} > 0$ (for instance $\alpha_{tol} = 10^{-5}$).

Globalization strategies

Using Integer/Rational Lattices (Torczon [1997], Audet and Dennis [2003])

- requires only simple decrease, $\rho = 0$,
- poll directions and step size must satisfy integer requirements,
- search step is restricted to an implicit mesh.

Imposing Sufficient Decrease (Kolda, Lewis, and Torczon [2003])

- Use of a forcing function

  $\rho : (0, +\infty) \to (0, +\infty)$, nondecreasing, satisfying

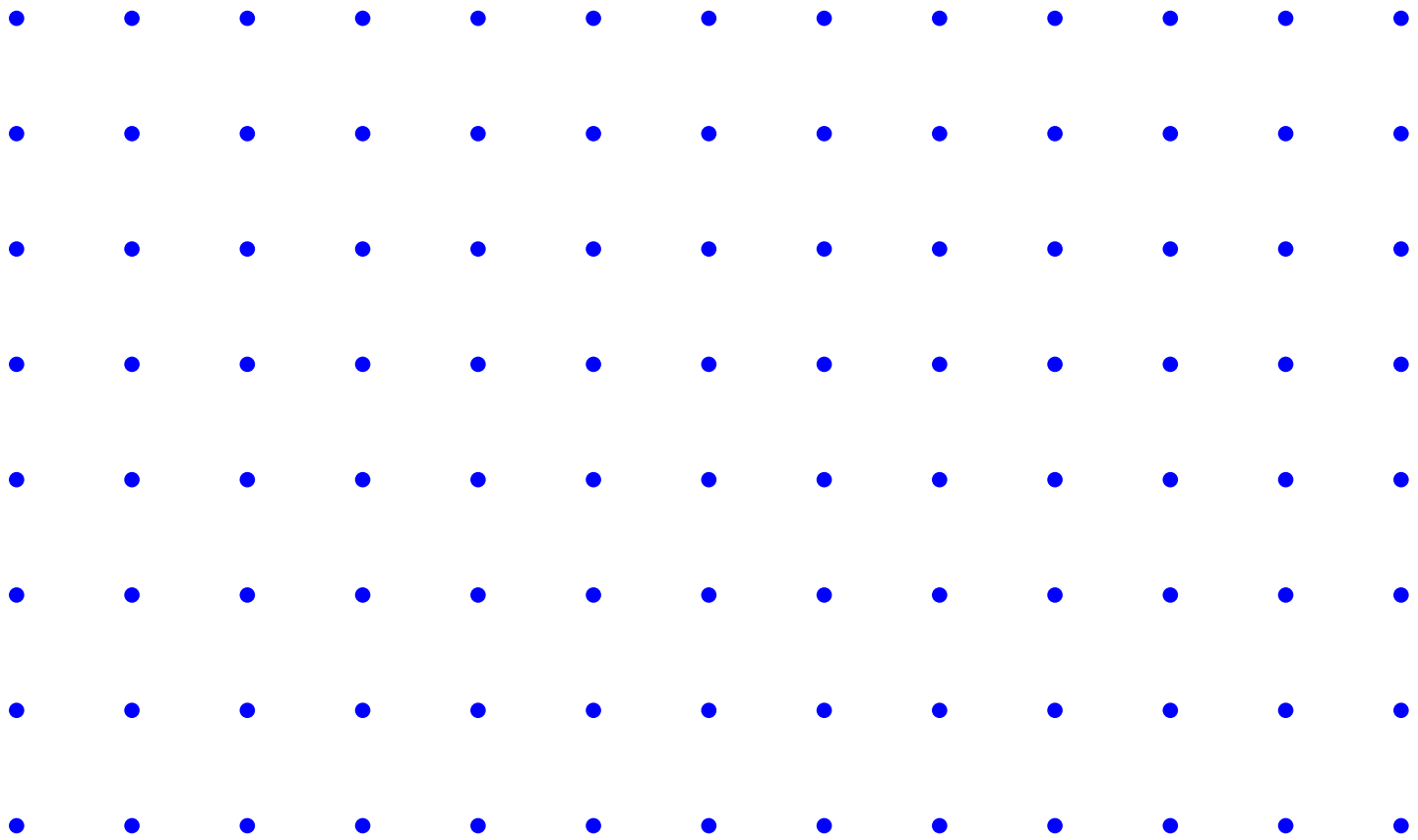
  $$\rho(t)/t \to 0 \quad \text{when} \quad t \downarrow 0.$$

  For the theory, most of the times, one can think of $\rho(\alpha) = \alpha^2$. In practice, imagine $\rho(\alpha) =$ very tiny constant $\times\ \alpha^2$.

Step size behavior — for integer lattices

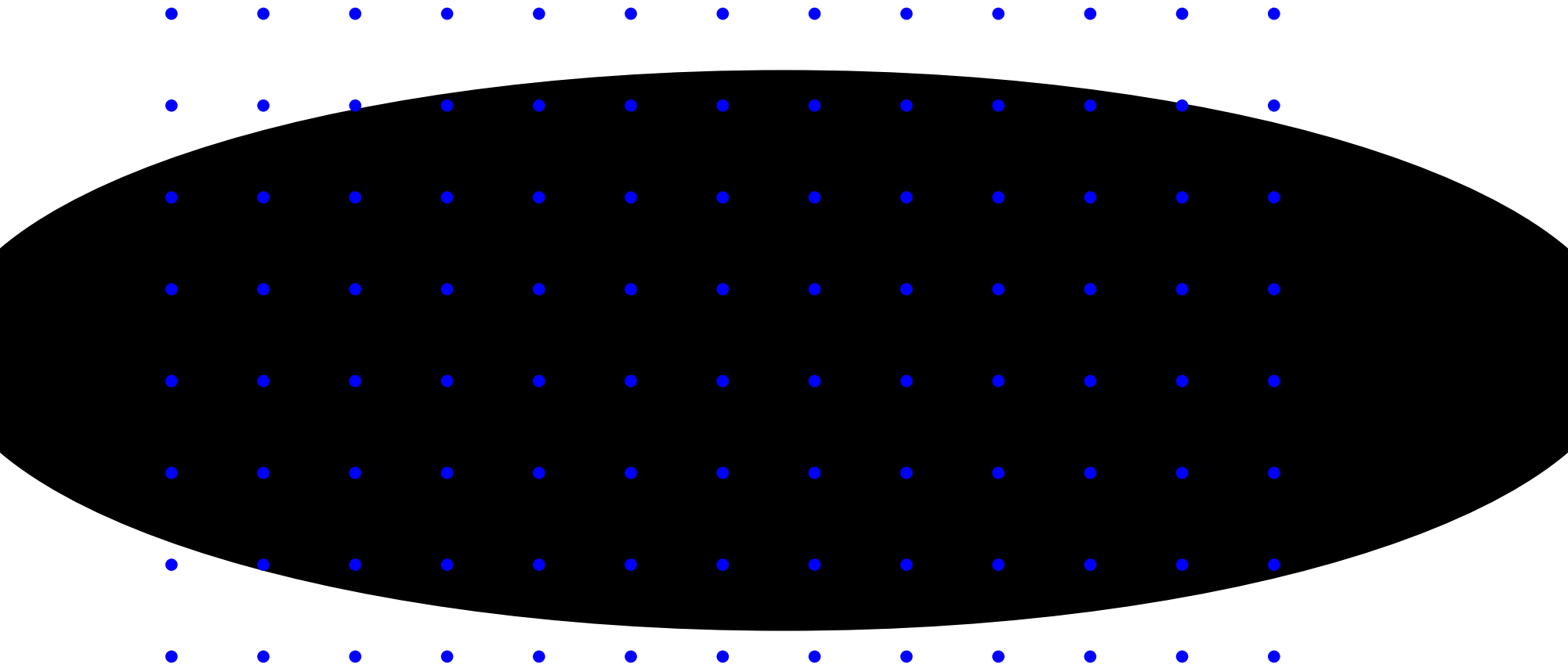
In this globalization scheme, all potential iterates must lie on an integer lattice when the step size α_k is bounded away from zero.

Step size behavior — for integer lattices



Intuitively, if α_k is bounded away from zero, points in this integer lattice would be separated by a finite and positive distance.

Step size behavior — integer lattices



It would therefore be impossible to fit an infinity of iterates inside a bounded level set.

One form of generating those integer lattices

Assumption

All PSS D_k, $\forall k$, are in a finite set D.

The columns of D are of the form $G\bar{z}_j$, $j = 1, \ldots, |D|$, where $G \in \mathbb{R}^{n\times n}$ is a nonsingular matrix and each $\bar{z}_j$ is a vector in $\mathbb{Z}^n$.

The search step only evaluates points in the mesh

$$M_k = \{x_k + \alpha_k D z, \;\; z \text{ vector of positive integers}\}.$$

Poll points are obviously in the mesh, i.e. $P_k \subset M_k$.

The step size parameter is updated as follows: Choose a rational number $\tau > 1$, a nonnegative integer $m^+ \geq 0$, and a negative integer $m^- \leq -1$. If the iteration is successful, $\alpha_{k+1} = \tau^{m_k^+}\alpha_k$, with $m_k^+ \in \{0, \ldots, m^+\}$. Otherwise, $\alpha_{k+1} = \tau^{m_k^-}\alpha_k$, with $m_k^- \in \{m^-, \ldots, -1\}$.

One form of generating those integer lattices

Remember that what is important is to guarantee a separation bounded away from zero for a fixed value of the step size parameter.

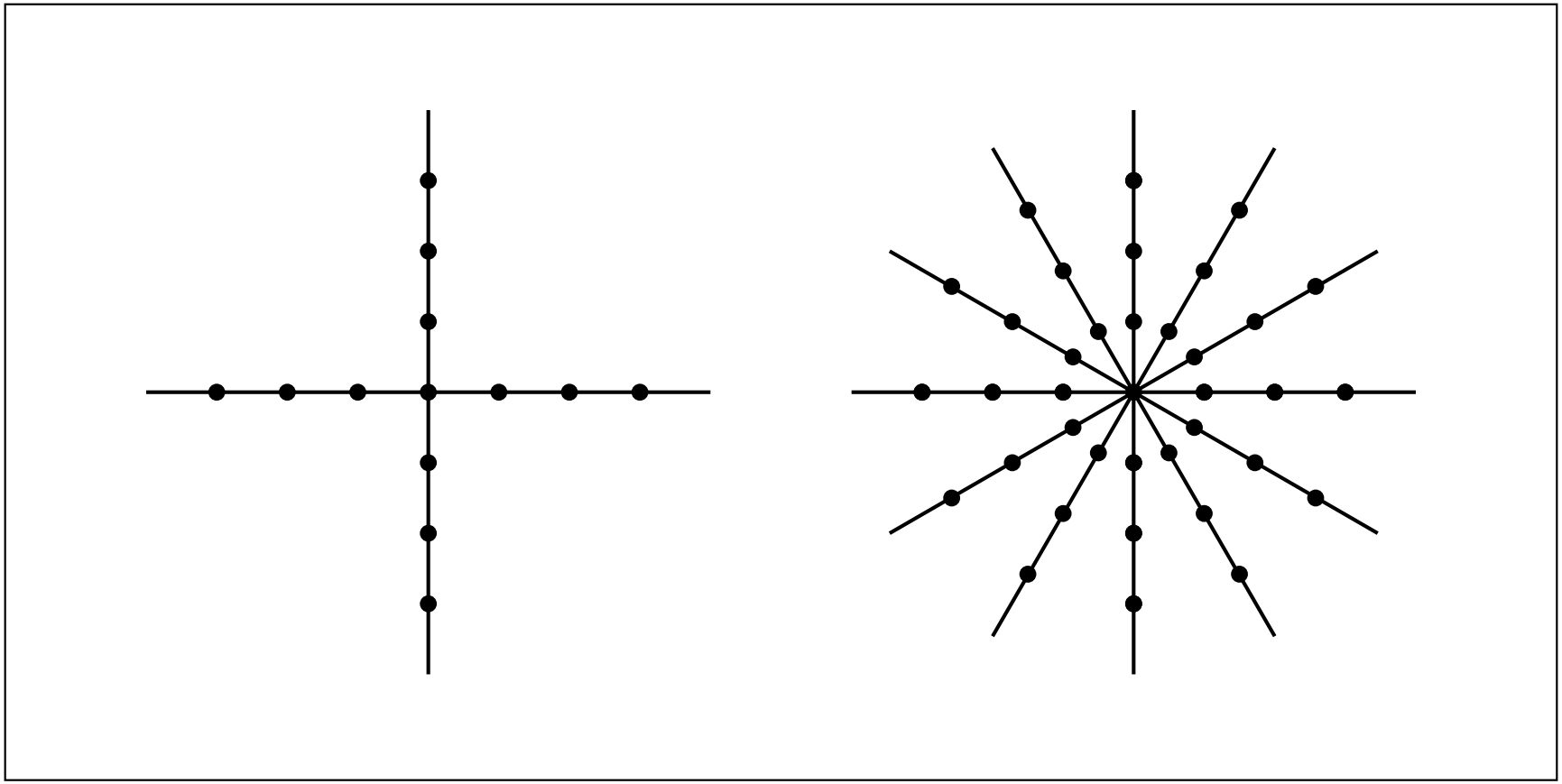
Integrality is a convenient way of guaranteeing that separation:

Lemma (based on the previous slide)

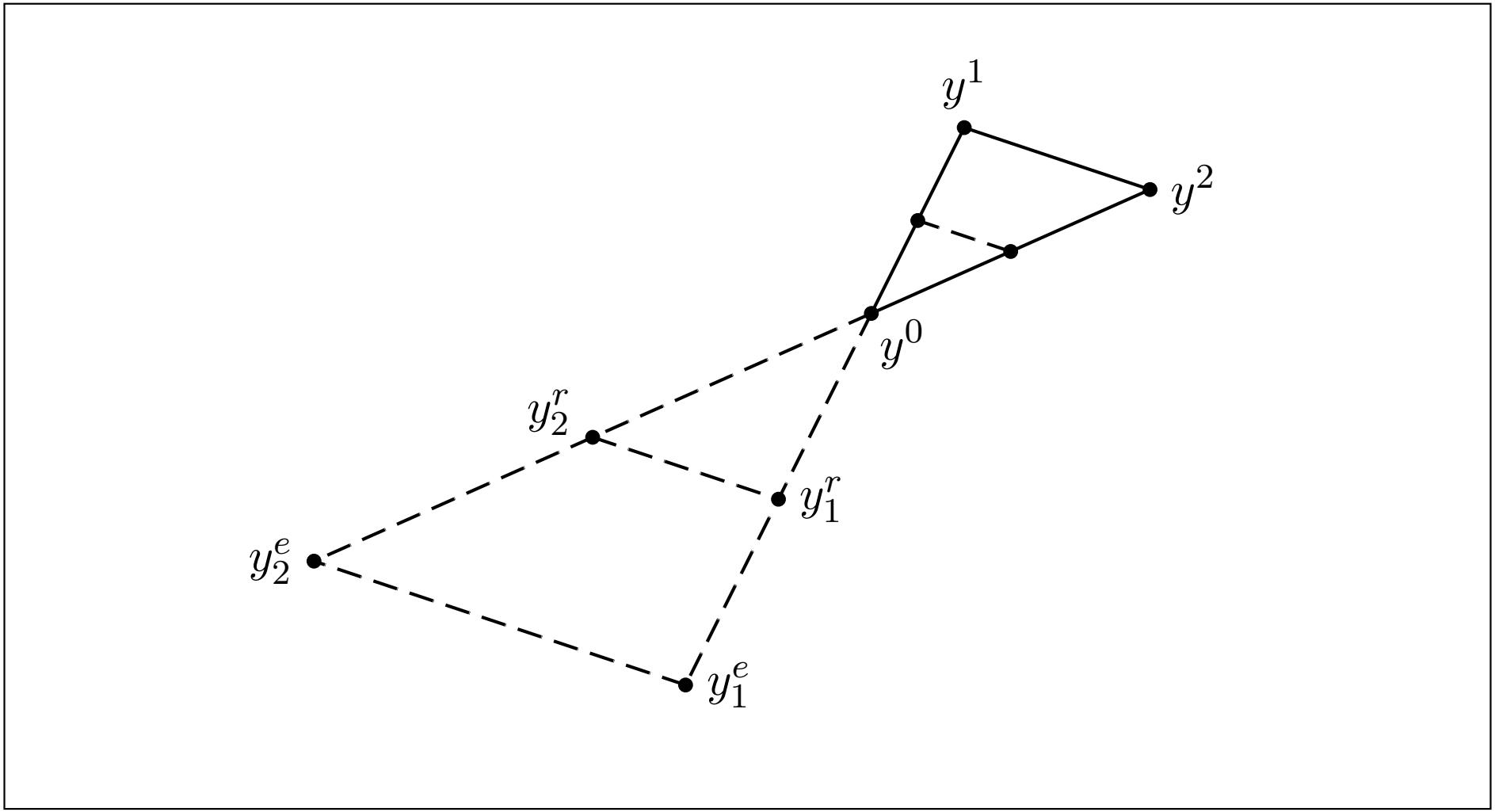
$$\min_{\substack{y,w \in M_k \\ y \neq w}} \|y - w\| \geq \frac{\alpha_k}{\|G^{-1}\|}.$$

As a counter example, all the positive integer combinations of directions in $\{-1, +\pi\}$ are dense in the real line, which does not happen with $\{-1, +1\}$.

Other examples of meshes



Multidirectional search (MDS)



Behavior of the step size — for integer lattices

Lemma

If $L(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is bounded and the integrality requirements hold, then there exists a subsequence K of unsuccessful iterations:

$$\lim_{k \in K} x_k = x_* \quad \textit{and} \quad \lim_{k \in K} \alpha_k = 0.$$

The difficult part (sketched before) is to first show that there is a subsequence of $\{\alpha_k\}$ converging to zero.

Torczon [1997], Audet and Dennis [2003]

See also Custódio, Madeira, Vaz, and Vicente [2010] for multiobjective optimization.

A subsequence K of unsuccessful iterations such that $\lim_{k \in K} \alpha_k = 0$ is called a refining subsequence.

Step size behavior — for sufficient decrease

Acceptance of new iterates is based on (search & poll):

$$f(x_{k+1}) \; < \; f(x_k) - \rho(\alpha_k),$$

where the forcing function $\rho : \mathbb{R}_+ \longrightarrow \mathbb{R}_+$ is nondecreasing and satisfies

$$\lim_{t \longrightarrow 0^+} \frac{\rho(t)}{t} \; = \; 0.$$

Step size behavior — for sufficient decrease

An iteration is successful only if it produces a point that has sufficiently decreased the objective function.

Also, $\rho(\alpha_k)$ is a monotonically nondecreasing function of the step size α_k.

Thus, α_k cannot be bounded away from zero since otherwise the objective function would tend to $-\infty$.

Intuitively, insisting on a sufficient decrease will make it harder to have a successful step and therefore will generate more unsuccessful poll steps.

Step size behavior — for sufficient decrease

If f is bounded below in $L(x_0)$ and sufficient decrease is imposed,

$$\lim_{k \longrightarrow +\infty} \alpha_k = 0.$$

In particular, there are refining subsequences:

Lemma

If f is bounded below in $L(x_0)$ and sufficient decrease is imposed, then there exists a subsequence K of unsuccessful iterations:

$$\lim_{k \in K} \alpha_k = 0.$$

single objective optimization:	Torczon [1997], Audet and Dennis [2003]
multiobjective optimization:	Custódio, Madeira, Vaz, and Vicente [2010]

If we further assume that $L(x_0)$ is bounded, then $\exists$ a convergent refining subsequence, in other words, and w.l.o.g., $\exists x_*$ such that for K above $\lim_{k \in K} x_k = x_*$.

Recall (now including the sufficient decrease term) ...

Theorem (Lewis, Tolda, and Torczon 2003)

Let D_k be a PSS.

Assume ∇f is Lipschitz continuous around iterate x_k (constant ν).

If the iterate k is unsuccessful, i.e.,

$$f(x_k + \alpha_k d) \geq f(x_k) - \rho(\alpha_k), \qquad \text{for all } d \in D_k,$$

then

$$\|\nabla f(x_k)\| \leq \left(\frac{\nu}{2} \operatorname{cm}(D_k)^{-1} \max_{d \in D_k} \|d\| \right) \alpha_k + \frac{\operatorname{cm}(D_k)^{-1}}{\min_{d \in D_k} \|d\|} \frac{\rho(\alpha_k)}{\alpha_k}.$$

Then global convergence is deduced from here: $\|\nabla f(x_k)\| \underset{K}{\longrightarrow} 0$.

Global convergence of DS in the smooth case

Assumption

The directions in D_k are bounded above and away from zero.

The cosine measure of D_k is bounded away from zero.

Theorem

If $L(x_0)$ is bounded for integer lattices (or just if f is bounded below in $L(x_0)$ for sufficient decrease), and ∇f is Lipschitz continuous in $L(x_0)$, then there exists K such that:

$$\lim_{k \in K} \|\nabla f(x_k)\| = 0.$$

Convergence of the whole gradient sequence

Theorem

Under complete polling and if

$$\lim_{k \longrightarrow +\infty} \alpha_k = 0,$$

then

$$\lim_{k \longrightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

NOTE: To have $\alpha_k \longrightarrow 0$ for integer lattices, all we need is not to increase α_k at successful iterations. However, that may lead to inefficiency.

GSS (smooth case as well)

In a SIREV (2003) paper, another framework for globally convergent directional direct-search methods was introduced.

There is no explicit separation in the algorithmic description between the search step and the poll step.

A successful iterate in the so-called Generating Set Search (GSS) framework is of the form $x_k + \alpha_k d_k$, where d_k belongs to a set of directions $G_k \cup H_k$.

In GSS, G_k plays the role of D_k (used in the poll step).

The search step is accommodated by the additional set of directions H_k (which might be empty).

A non-smooth example

Next we depict the contours of the function:

$$f(x) = \frac{1}{2} \max \left\{ \|x - c_1\|^2, \|x - c_2\|^2 \right\},$$

where $c_1 = (1, -1)$ and $c_2 = -c_1$.

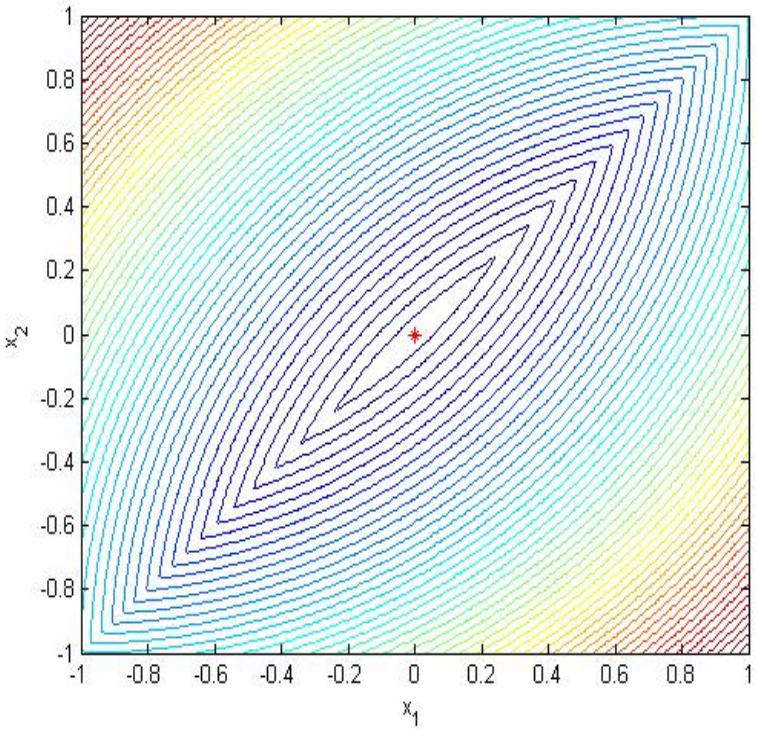
The function is continuous and strictly convex everywhere but its gradient is discontinuous along the line $x_1 = x_2$ (and it can be modified to be strict diff. at $(0, 0)$).

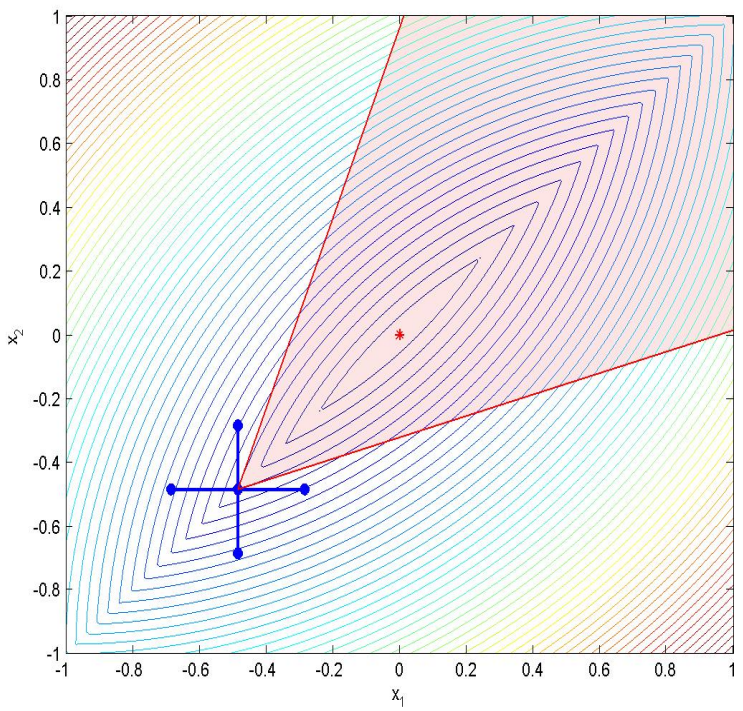
The function has a strict minimizer at $(0, 0)$.

At any point of the form (a, a), with $a \neq 0$, coordinate search generates an infinite number of unsuccessful iterations without any progress.

None of the elements of $D_{\oplus} = [e_1 \ e_2 \ -e_1 \ -e_2]$ are descent directions.

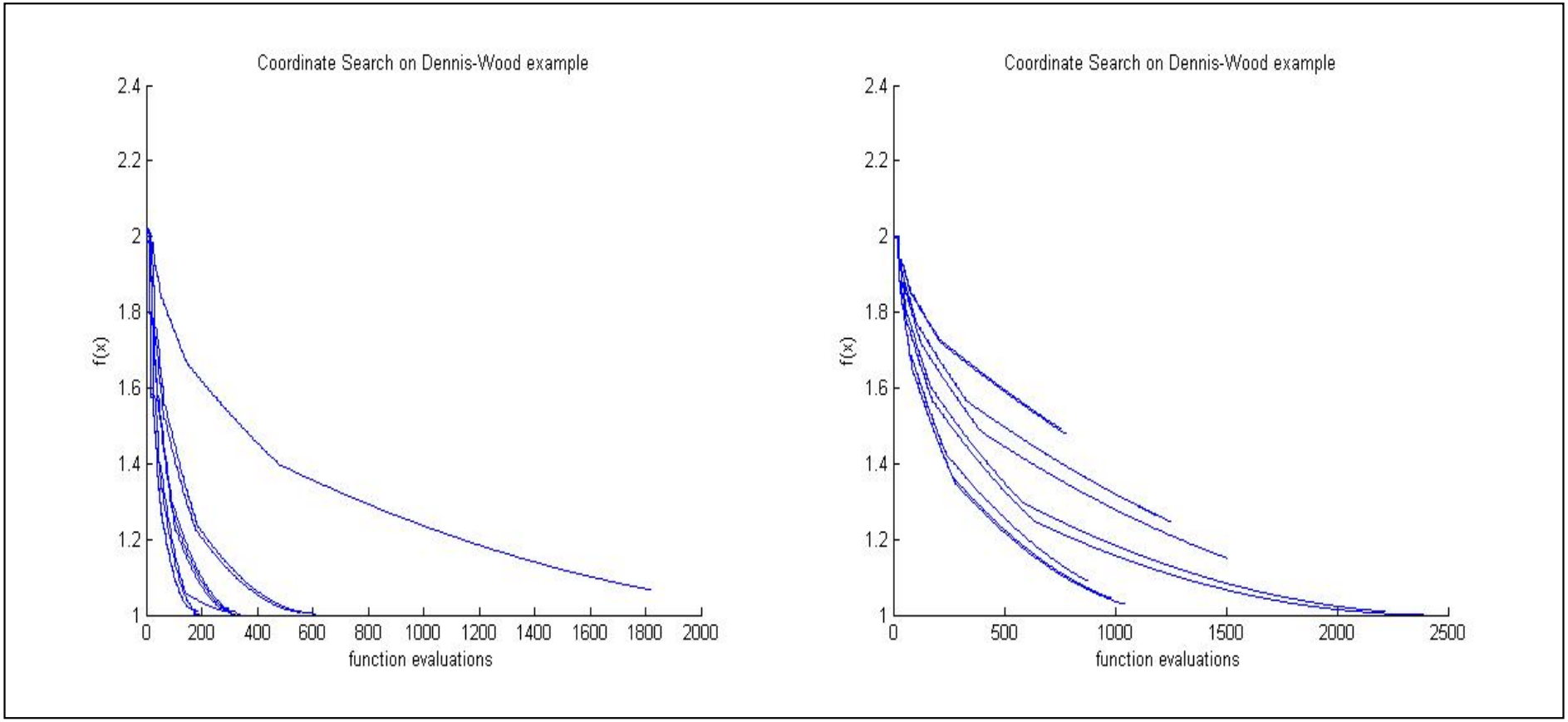
Difficulties in the non-smooth case





Contours of function for $c_1 = (1, -1)$ and $c_2 = -c_1$. The cone of descent directions at the poll center is shaded.

CS in the non-smooth case

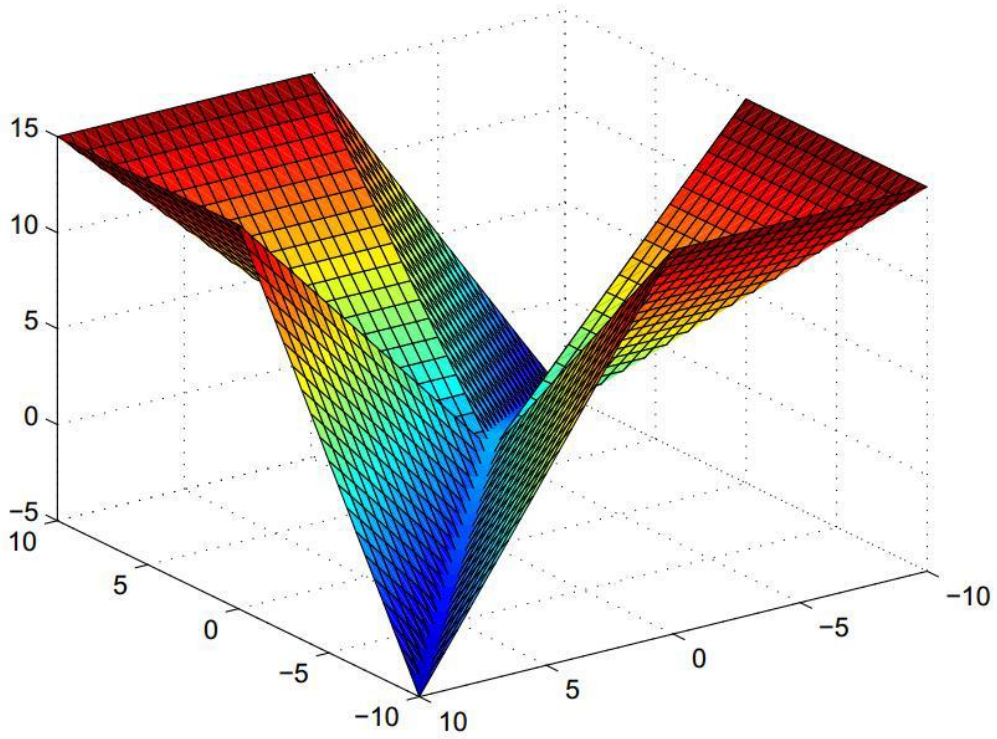


The plots on the left (resp. right) correspond to 10 starting points randomly generated in a box of ℓ_∞ radius 10^{-2} (resp. 10^{-3}) around the point $(1, 1)$. Stagnation only occurs when the starting points are too close to points on the line.

NP-hardness of finding a local minimizer of a non-smooth nonconvex function (Nesterov, 2010)

Let $c \in \mathbb{N}^n$. Consider the piecewise linear function:

$$\phi(x) = \left(1 - \frac{1}{\sum_{i=1}^{n} c_i}\right) \max_{1 \le i \le n} |x_i| - \min_{1 \le i \le n} |x_i| + |\langle c, x \rangle|.$$



NP-hardness of finding a local minimizer of a non-smooth nonconvex function (Nesterov, 2010)

Now,

$$\exists x \in \mathbb{R}^n : \ \phi(x) < 0 \iff \exists \sigma \in \{\pm 1\}^n : \ \langle c, \sigma \rangle = 0.$$

The second problem is NP-hard.

Thus, the first one is also NP-hard $\Longrightarrow$ finding a descent direction for ϕ is NP-hard.

Therefore, finding a local minimum of the non-smooth nonconvex $f(x) = \max\{-1, \phi(x)\}$ is also NP-hard.

One possible fix: Infinite number of directions

One possibility is to use an infinite number of polling directions.

This does not pose a problem to global convergence, which can be guaranteed a.e. in the unit sphere:

- C. Audet and J. E. Dennis Jr., Mesh adaptive direct search algorithms for constrained optimization, SIAM J. Optim., 17 (2006) 188–217.

  LTMADS, ORTHOMADS: ways of dense generation in the unit sphere guaranteeing the integer lattice requirement.

- L. N. Vicente and A. L. Custódio Analysis of direct searches for discontinuous functions, Math. Program., 133 (2012) 299–325.

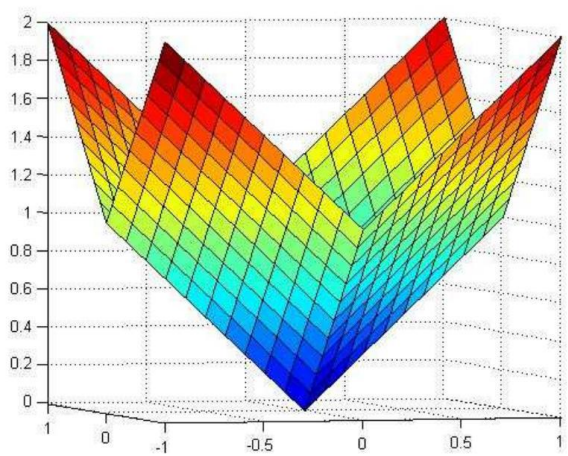
  Dense generation in the unit sphere is waived of rules when imposing sufficient decrease.

Another possible fix: Smoothing functions

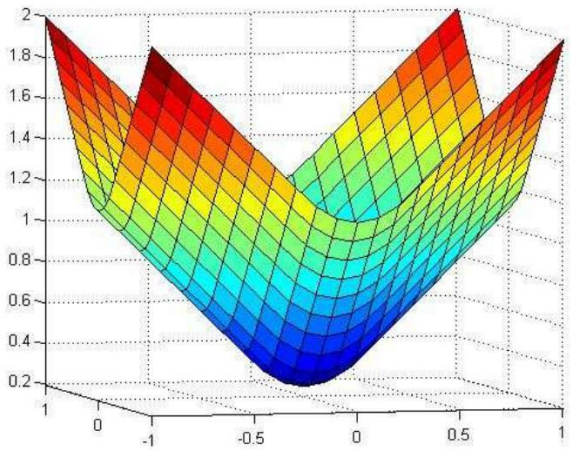
Definition

We call $\tilde{f} : \mathbb{R}^n \times [0, +\infty) \to \mathbb{R}$ *a smoothing function of* f *if,* $\forall \mu \in (0, +\infty)$, $\tilde{f}(\cdot, \mu)$ *is* $\mathcal{C}^1$ *and,* $\forall x \in \mathbb{R}^n$,

$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) = f(x).$$



(a) Original function.



(b) Smoothed function.

Refining directions

Stationarity results for DS consist of nonnegativity of generalized directional derivatives along certain limit directions.

Definition (refining directions)

Let K be a refining subsequence converging to x_*.

Refining directions for x_* are limit points of $\{d_k/\|d_k\|\}_{k\in K}$, where $d_k \in D_k$.

Audet and Dennis [2006]

Definition

For f Lipschitz continuous near x, the Clarke generalized directional derivative is:

$$f^{\circ}(x;d) = \limsup_{y \to x \; t \downarrow 0} \frac{f(y+td) - f(y)}{t}.$$

What does this $\limsup$ exactly mean?

$$\limsup_{y \to x \; t \downarrow 0} \frac{f(y+td) - f(y)}{t} = \lim_{\epsilon \downarrow 0} \sup_{\|y-x\| \leq \epsilon, 0 < t \leq \epsilon} \left\{ \frac{f(y+td) - f(y)}{t} \right\}$$

Clarke's non-smooth calculus

If f is increasing from x, along d, then

$$f^\circ(x;d) \ \geq \ 0.$$

Definition

x is a (Clarke) stationary point if

$$f^\circ(x;d) \ \geq \ 0, \quad \forall d \in \mathbb{R}^n.$$

Global convergence of DS in the non-smooth case

Consider a refining subsequence converging to x_* (and assume that f is Lipschitz continuous near x_*).

Theorem

If d is a refining direction for x_* then:

$$f^\circ(x_*; d) \geq 0.$$

single objective optimization:	Audet and Dennis [2006], Vicente and Custódio [2010]
multiobjective optimization:	Custódio, Madeira, Vaz, and Vicente [2010]

Proof sketch (assuming normalized directions)

Let d be a limit point of $\{d_k\}$ (assume $\|d_k\| = 1$). W.l.o.g. assume $d_k \to d$ in K.

$$\begin{aligned} f^\circ(x_*; d) &= \limsup_{x' \to x_*, t \downarrow 0} \frac{f(x' + td) - f(x')}{t} \\ &\geq \limsup_{k \in K} \left\{ \frac{f(x_k + \alpha_k d_k) - f(x_k)}{\alpha_k} + o(\alpha_k)??? \right\} \\ &= \limsup_{k \in K} \left\{ \frac{f(x_k + \alpha_k d_k) - f(x_k) + \rho(\alpha_k)}{\alpha_k} - \frac{\rho(\alpha_k)}{\alpha_k} \right\}. \end{aligned}$$

Since $\{x_k\}_{k \in K}$ is a refining subsequence, for each $k \in K$,

$$f(x_k + \alpha_k d_k) - f(x_k) + \rho(\alpha_k) \geq 0.$$

Global convergence of DS in the non-smooth case

How is the proof changed if the directions are not normalized and what condition should these satisfy?...

The answer is that $\alpha_k \|d_k\|$ must still go to zero in K.

Global convergence of DS in the non-smooth case

Theorem

If $L(x_0)$ is bounded, then, for both integer lattices or for sufficient decrease, there exists a refining subsequence converging to a point x_*.

Let f be Lipschitz continuous near x_*.

If all refining directions for that refining subsequence are asymptotically dense in the unit sphere, then:

$$f^\circ(x_*; d) \geq 0, \quad \forall d \in \mathbb{R}^n.$$

The proof is based on

$$f^\circ(x_*; d) = \lim_{\bar{d} \to d,\ \bar{d} \text{ refining direction}} f^\circ(x_*; \bar{d}),$$

on the previous theorem, and on the fact that $f^\circ(x_*, \cdot)$ is continuous.

Global convergence of DS in the non-smooth case

When imposing sufficient decrease, there is no need for integrality, and thus we can freely generate the poll directions in the unit sphere with the asymptotic density requirement, i.e., randonly:

generate (q_k) randomly
build Q_k orthogonal from the first column q_k
set $D_k = [\, Q_k \; -Q_k \,]$.

It is not trivial to do it while imposing integer lattices (especially together with a PSS requirement), and that construction is precisely MADS (LTMADS, ORTHOMADS, etc.). In particular normality is given up.

Even more non-smooth (discontinuities)...

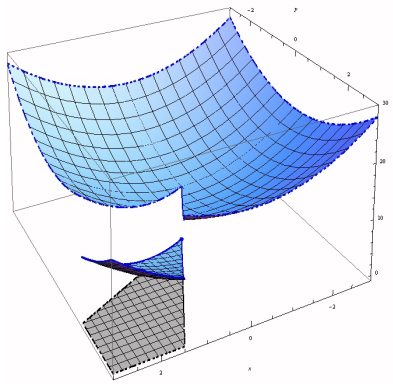
One can derive similar results for discontinuous functions for refining directions in $D_f(x_*)$ where:

Definition

f directionally Lipschitz at x_* with respect to v when

$$\limsup_{x' \to_f x_*, t \downarrow 0} \sup_{v' \to v} \frac{f(x' + tv') - f(x')}{t} < +\infty.$$

$D_f(x_*)$ is the interior of the shaded branch domain.



L. N. Vicente and A. L. Custódio Analysis of direct searches for discontinuous functions, Math. Program., 133 (2012) 299–325.

From non-smooth to smooth

Definition

When f is Lipschitz continuous near x, f is strictly differentiable at x if there exists a vector $w = \nabla f(x)$ — 'the gradient' — such that

$$f^\circ(x; v) \;=\; w^\top v \quad \forall v \in \mathbb{R}^n.$$

If the function f is strictly differentiable at x_*, then in particular

$$\nabla f(x_*)^\top d \;\geq\; 0, \quad \forall d \in D, \qquad \text{for some } D \text{ PSS},$$

and the iff characterization of positive spanning sets implies that

$$\nabla f(x_*) \;=\; 0$$

(which is similar to what we obtained in the continuously diff. case).

REMEMBER: Our problem setting

Unconstrained optimization

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

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

f is at least locally Lipschitz continuous

A forcing function $\rho(\cdot)$ is a positive and monotonically nondecreasing function such that

$$\lim_{\alpha \downarrow 0} \frac{\rho(\alpha)}{\alpha} = 0.$$

We will consider $\rho(\alpha) = \alpha^p$, with $p > 1$.

In most of this presentation, we take $p = 2$: $\rho(\alpha) = \alpha^2$.

REMEMBER: A class of DS methods, f smooth for now

Choose: x_0 and α_0.

For $k = 0, 1, 2, \ldots$ (Until α_k is suff. small)

- **Search step (optional)**
- **Poll step:** Select D_k PSS and find $x_k + \alpha_k d_k$ ($d_k \in D_k$):

$$f(x_k + \alpha_k d_k) < f(x_k) - \rho(\alpha_k).$$

- Update the new iterate x_{k+1} (stay at x_k is unsuccessful).
- Update the step size α_{k+1}.
  Possible increase if iteration is successful. Decrease otherwise.

REMEMBER: Behavior of the step size (sufficient decrease, meaning ρ forcing function)

Assumption

f is bounded below in $L(x_0) = \{x \in \mathbb{R}^n : \ f(x) \leq f(x_0)\}$.

Lemma (Seen before; see also IDFO book or SIAM Review 2003 survey on DS)

$$\lim_{k \to +\infty} \alpha_k \ = \ 0.$$

Assumption

The directions in D_k are bounded above and away from zero.

The cosine measure of D_k is bounded away from zero.

REMEMBER: Behavior of unsuccessful iterations

Theorem (Lewis, Tolda, and Torczon 2003)

Let D_k be a PSS.

Assume $f \in \mathcal{C}^1_\nu$.

If the iterate k is unsuccessful, i.e.,

$$f(x_k + \alpha_k d) \geq f(x_k) - \rho(\alpha_k), \qquad \text{for all } d \in D_k,$$

then

$$\|\nabla f(x_k)\| \leq \frac{C(\nu) \times \alpha_k}{\mathrm{cm}(D_k)} \qquad \ldots \text{ since } \rho(\alpha) = \alpha^2.$$

Remember that global convergence was deduced from here:
$\|\nabla f(x_k)\| \underset{K}{\longrightarrow} 0$.

NOW: The question that interests us (smooth case)

Question

Given $\epsilon \in (0, 1)$, *how many iterations* $\bar{k}$ *are needed to reach*

$$\|\nabla f(x_{\bar{k}})\| \leq \epsilon \quad ?$$

Gradient methods (smooth, non-convex case)

Choose x_0.

For $k = 0, 1, 2, \ldots$ $x_{k+1} = x_k - h_k \nabla f(x_k)$, $h_k > 0$ satisfying Wolfe conditions (SDC+CC or SDC+backtracking)

When $f \in \mathcal{C}^1_\nu$, there is a constant $C(\frac{1}{\nu}) > 0$ such that

$$f(x_k) - f(x_{k+1}) \geq C(\frac{1}{\nu}) \|\nabla f(x_k)\|^2.$$

This inequality leads to an $\mathcal{O}(\epsilon^{-2})$ WCC bound:

$$f(x_0) - f_* \geq f(x_0) - f(x_{k+1}) \geq C(\frac{1}{\nu}) \sum_{l=0}^{k} \|\nabla f(x_l)\|^2 \geq C(\frac{1}{\nu})(k+1)\epsilon^2.$$

- Y. Nesterov, Introductory Lectures on Convex Optimization, Kluwer Academic Publishers, Dordrecht, 2004.

Global rate of DS (tools for proofs)

For an unsuccessful iteration k_u,

$$\mathcal{O}\left(\alpha_{k_u}\right) \;=\; \|\nabla f(x_{k_u})\|.$$

Global rate of DS (tools for proofs)

For an unsuccessful iteration k_u,

$$\mathcal{O}\left(\alpha_{k_u}\right) \;=\; \|\nabla f(x_{k_u})\|.$$

By backtracking on the successful ones,

$$f(x_k) \;<\; f(x_{k-1}) - \alpha_{k-1}^2 \;<\; \cdots \;<\; f(x_{k_u}) - \mathcal{O}\left((k-k_u)\alpha_{k_u}^2\right),$$

Global rate of DS (tools for proofs)

For an unsuccessful iteration k_u,

$$\mathcal{O}\left(\alpha_{k_u}\right) \;=\; \|\nabla f(x_{k_u})\|.$$

By backtracking on the successful ones,

$$f(x_k) \;<\; f(x_{k-1}) - \alpha_{k-1}^2 \;<\; \cdots \;<\; f(x_{k_u}) - \mathcal{O}\left((k-k_u)\alpha_{k_u}^2\right),$$

one can show that

$$|\mathcal{S}(k)| \;=\; \mathcal{O}\left(\frac{1}{\|\nabla f(x_k)\|^2}\right).$$

Global rate of DS (tools for proofs)

For the unsuccessful ones,

$$|\mathcal{U}(k)| \leq \mathcal{O}\left(\log(\gamma)|\mathcal{S}(k)|\right) + \mathcal{O}\left(-\log\|\nabla f(x_k)\|\right).$$

Global rate of DS (tools for proofs)

For the unsuccessful ones,

$$|\mathcal{U}(k)| \leq \mathcal{O}\left(\log(\gamma)|\mathcal{S}(k)|\right) + \mathcal{O}\left(-\log\|\nabla f(x_k)\|\right).$$

The number of successful iterations until the first unsuccessful iteration k_0 is at most

$$\left\lceil \frac{f(x_0) - f_*}{\alpha_0^2} \right\rceil.$$

WCC of DS (smooth case)

Theorem

Any DS method (based on sufficient decrease) generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$\min_{0\leq j\leq k} \|\nabla f(x_j)\| \;=\; \mathcal{O}(1/\sqrt{k})$$

WCC of DS (smooth case)

Theorem

Any DS method (based on sufficient decrease) generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$\min_{0\leq j\leq k} \|\nabla f(x_j)\| \;=\; \mathcal{O}(1/\sqrt{k})$$

and takes at most

$$\mathcal{O}\left(n\,\epsilon^{-2}\right)$$

iterations to reduce the gradient below $\epsilon \in (0,1)$.

WCC of DS (smooth case)

Theorem

Any DS method (based on sufficient decrease) generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$\min_{0\leq j\leq k} \|\nabla f(x_j)\| \;=\; \mathcal{O}(1/\sqrt{k})$$

and takes at most

$$\mathcal{O}\left(n\,\epsilon^{-2}\right)$$

iterations to reduce the gradient below $\epsilon \in (0,1)$.

- The # of fevals must be multiplied by n: $\mathcal{O}\left(n^2\,\epsilon^{-2}\right)$.

WCC of DS (smooth case)

Theorem

Any DS method (based on sufficient decrease) generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$\min_{0\leq j\leq k} \|\nabla f(x_j)\| \;=\; \mathcal{O}(1/\sqrt{k})$$

and takes at most

$$\mathcal{O}\left(n\,\epsilon^{-2}\right)$$

iterations to reduce the gradient below $\epsilon \in (0,1)$.

- The # of fevals must be multiplied by n: $\mathcal{O}\left(n^2\,\epsilon^{-2}\right)$.
- Bounds depend on $\frac{1}{L^2_{\nabla f}}$ (instead of $\frac{1}{L_{\nabla f}}$ as in gradient method).
- LNV, Worst case complexity of direct search, EURO J. on Computational Optimization, 1 (2013) 143–153.

Gradient methods (smooth, convex case)

Theorem (Y. Nesterov)

Let $f \in \mathcal{F}_\nu^1$. Then, for h suff. small, $\|x_m - x_*\| \leq \|x_0 - x_*\|$ and

$$f(x_m) - f_* \leq \frac{2\nu\|x_0 - x_*\|^2}{m+4}.$$

This implies, for $m < k$,

$$C(1/\nu) \sum_{l=m}^{k} \|\nabla f(x_l)\|^2 \leq \frac{2\nu\|x_0 - x_*\|^2}{m+4}$$

By choosing $k = 2m$, the gradient method has a WCC of $\mathcal{O}(\epsilon^{-1})$ iter.

- Y. Nesterov, Introductory Lectures on Convex Optimization, Kluwer Academic Publishers, Dordrecht, 2004.

Assumption in the smooth, convex case

Assumption

The level set $L_f(x) = \{y \in \mathbb{R}^n : f(y) \leq f(x)\}$ *is bounded for some* x *or, if that is not the case, there exists* $R > 0$ *such that*

$$\sup_{y \in L_f(x_0)} \operatorname{dist}(y, X_*^f) \leq R,$$

where X_*^f *is the solution set (assumed non empty).*

Assumption in the smooth, convex case

Assumption

The level set $L_f(x) = \{y \in \mathbb{R}^n : f(y) \leq f(x)\}$ *is bounded for some* x *or, if that is not the case, there exists* $R > 0$ *such that*

$$\sup_{y \in L_f(x_0)} \operatorname{dist}(y, X_*^f) \leq R,$$

where X_*^f *is the solution set (assumed non empty).*

Satisfied under strong conv. of f or boundedness of either X_*^f or $L_f(x_0)$.

Assumption in the smooth, convex case

Assumption

The level set $L_f(x) = \{y \in \mathbb{R}^n : f(y) \leq f(x)\}$ *is bounded for some* x *or, if that is not the case, there exists* $R > 0$ *such that*

$$\sup_{y \in L_f(x_0)} \operatorname{dist}(y, X_*^f) \leq R,$$

where X_*^f *is the solution set (assumed non empty).*

Satisfied under strong conv. of f or boundedness of either X_*^f or $L_f(x_0)$.

One needs this assumption because

$$\|x_k - x_*\| \leq \|x_0 - x_*\|, \qquad \forall k \geq 0,$$

does NOT hold as in the gradient method (because $d_k \neq -\nabla f(x_k)$).

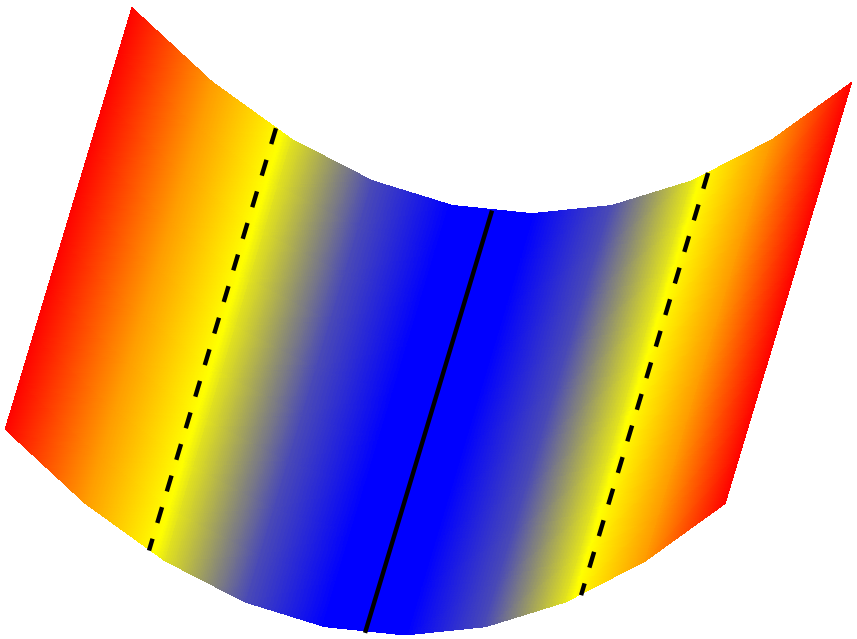
Discussion (Example 1)

There are convex functions f such that R is finite but neither f is strongly convex nor $L_f(x_0)$ is bounded.

Discussion (Example 1)

There are convex functions f such that R is finite but neither f is strongly convex nor $L_f(x_0)$ is bounded.

$$f(x, y) = y^2$$



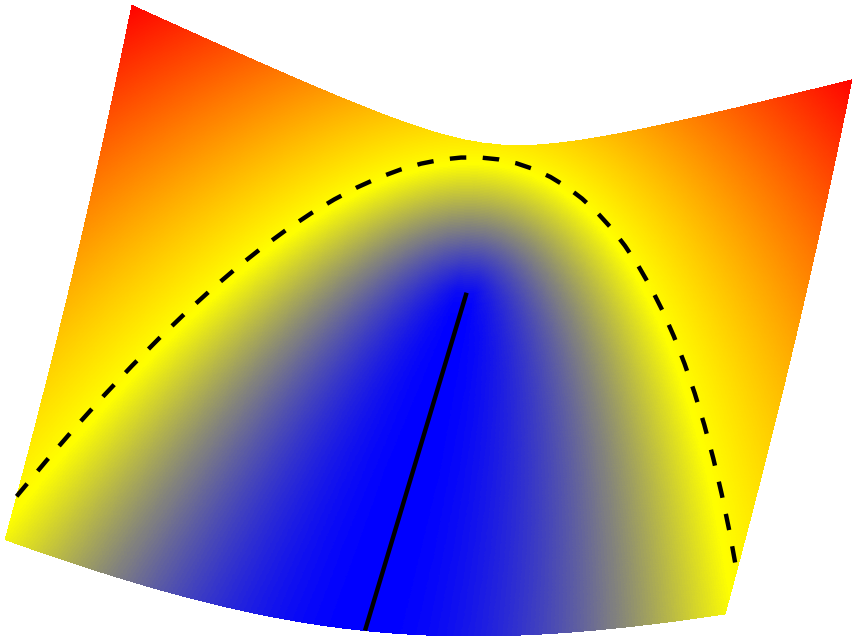
Discussion (Example 2)

There are some rare pathological instances such that the assumption does not hold and X_*^f is unbounded.

Discussion (Example 2)

There are some rare pathological instances such that the assumption does not hold and X_*^f is unbounded.

$$f(x,y) = \sqrt{x^2+y^2} - x$$



Tools for proofs (under convexity)

For an unsuccessful iteration k_u, using convexity and the R assumption

$$\mathcal{O}\left(\alpha_{k_u}\right) \;=\; \|\nabla f(x_{k_u})\| \geq \frac{f(x_{k_u}) - f_*}{\|x_{k_u} - x_*\|} \;\geq\; \frac{f(x_{k_u}) - f_*}{R}.$$

Tools for proofs (under convexity)

For an unsuccessful iteration k_u, using convexity and the R assumption

$$\mathcal{O}\left(\alpha_{k_u}\right) \;=\; \|\nabla f(x_{k_u})\| \geq \frac{f(x_{k_u}) - f_*}{\|x_{k_u} - x_*\|} \;\geq\; \frac{f(x_{k_u}) - f_*}{R}.$$

By backtracking on the successful ones, one can show (under some manipulation...) that

$$|\mathcal{S}(k)| \;=\; \mathcal{O}\left(\frac{1}{f(x_k) - f_*}\right).$$

Tools for proofs (under convexity)

For an unsuccessful iteration k_u, using convexity and the R assumption

$$\mathcal{O}\left(\alpha_{k_u}\right) \;=\; \|\nabla f(x_{k_u})\| \geq \frac{f(x_{k_u}) - f_*}{\|x_{k_u} - x_*\|} \;\geq\; \frac{f(x_{k_u}) - f_*}{R}.$$

By backtracking on the successful ones, one can show (under some manipulation...) that

$$|\mathcal{S}(k)| \;=\; \mathcal{O}\left(\frac{1}{f(x_k) - f_*}\right).$$

For the unsuccessful ones,

$$|\mathcal{U}(k)| \;\leq\; \mathcal{O}\left(\log(\gamma)|\mathcal{S}(k)|\right) + \mathcal{O}\left(-\log\left(f(x_k) - f_*\right)\right).$$

WCC of DS (smooth, convex case)

Theorem (M. Dodangeh and LNV 2014)

Under this assumption, any DS method (based on sufficient decrease) generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$f(x_k) - f_* \;=\; \mathcal{O}(1/k)$$

WCC of DS (smooth, convex case)

Theorem (M. Dodangeh and LNV 2014)

Under this assumption, any DS method (based on sufficient decrease) generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$f(x_k) - f_* \;=\; \mathcal{O}(1/k)$$

and takes at most

$$\mathcal{O}\left(n\,\epsilon^{-1}\right)$$

iterations to reduce the gradient below $\epsilon \in (0,1)$.

WCC of DS (smooth, convex case)

Theorem (M. Dodangeh and LNV 2014)

Under this assumption, any DS method (based on sufficient decrease) generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$f(x_k) - f_* \;=\; \mathcal{O}(1/k)$$

and takes at most

$$\mathcal{O}\left(n\,\epsilon^{-1}\right)$$

iterations to reduce the gradient below $\epsilon \in (0,1)$.

- The # of fevals must be multiplied by n: $\mathcal{O}\left(n^2\,\epsilon^{-1}\right)$.
- M. Dodangeh and LNV, Worst case complexity of direct search under convexity, Math. Program., 155 (2016) 307–332.

Optimal order of the WCC bounds

We have seen that the WCC of DS in #fevals is

$$\mathcal{O}(n^2\epsilon^{-2}) \qquad \mathcal{O}(n^2\epsilon^{-1}) \text{ (convex)}$$

Optimal order of the WCC bounds

We have seen that the WCC of DS in #fevals is

$$\mathcal{O}(n^2\epsilon^{-2}) \qquad \mathcal{O}(n^2\epsilon^{-1}) \text{ (convex)}$$

The n^2 factor came from

$$\frac{|D|}{\mathrm{cm}(D)^2}.$$

Optimal order of the WCC bounds

We have seen that the WCC of DS in #fevals is

$$\mathcal{O}(n^2\epsilon^{-2}) \qquad \mathcal{O}(n^2\epsilon^{-1}) \text{ (convex)}$$

The n^2 factor came from

$$\frac{|D|}{\mathrm{cm}(D)^2}.$$

For $D = D_{\oplus}$ one obtains

$$\frac{2n}{(1/\sqrt{n})^2} = 2n^2.$$

Optimal order of the WCC bounds

We have seen that the WCC of DS in #fevals is

$$\mathcal{O}(n^2\epsilon^{-2}) \qquad \mathcal{O}(n^2\epsilon^{-1}) \text{ (convex)}$$

The n^2 factor came from

$$\frac{|D|}{\mathrm{cm}(D)^2}.$$

For $D = D_{\oplus}$ one obtains

$$\frac{2n}{(1/\sqrt{n})^2} = 2n^2.$$

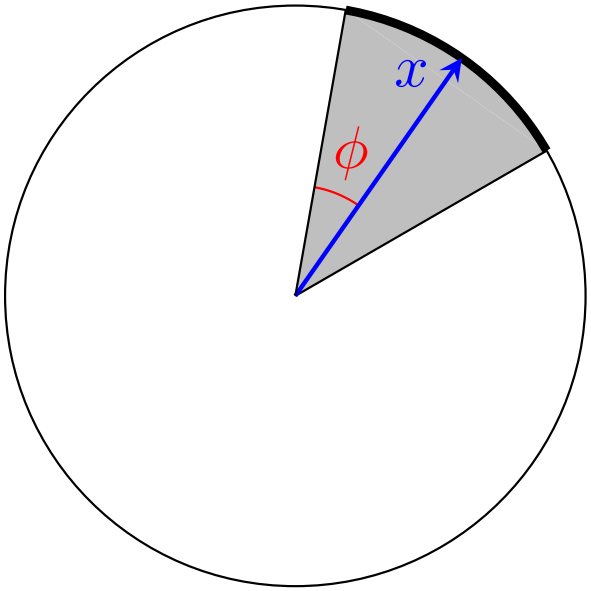
Is this optimal?

PSSs and unit sphere covering

Given a PSS we consider a covering of the unit sphere by sph. caps.

PSSs and unit sphere covering

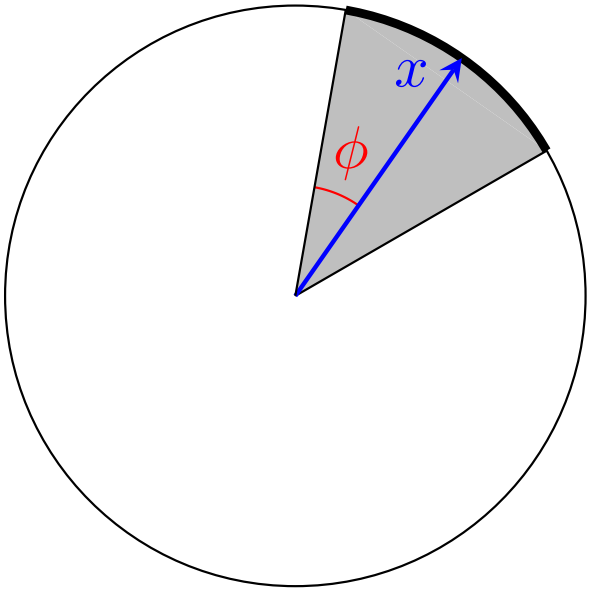
Given a PSS we consider a covering of the unit sphere by sph. caps.



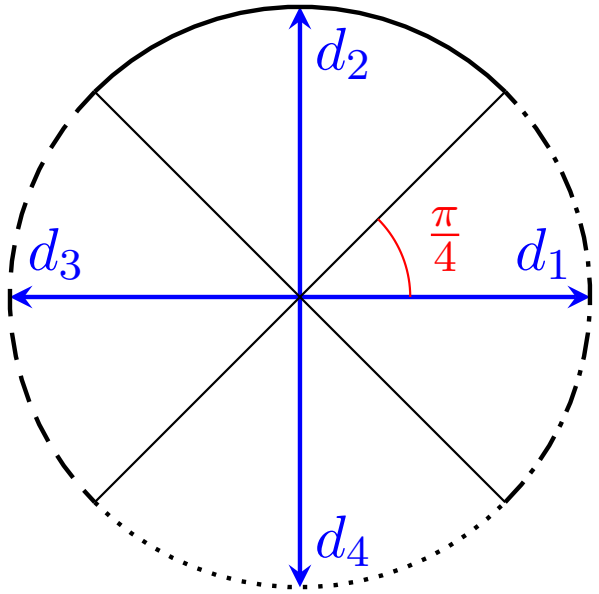
Spherical cap $\mathbb{C}(x, \phi)$

PSSs and unit sphere covering

Given a PSS we consider a covering of the unit sphere by sph. caps.



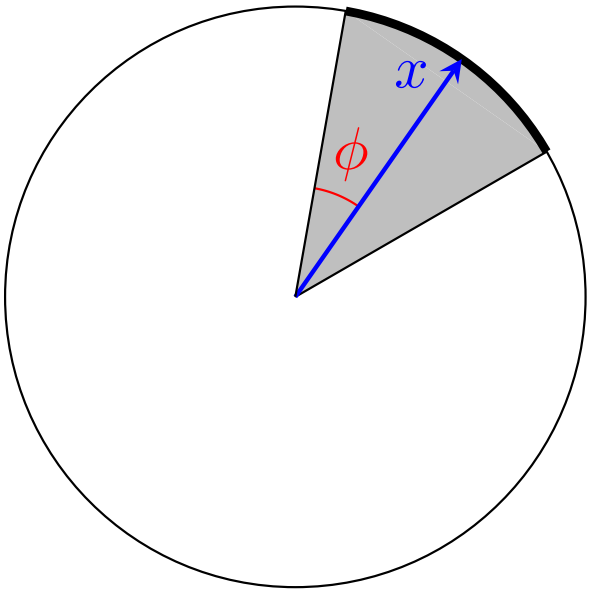
Spherical cap $\mathbb{C}(x, \phi)$



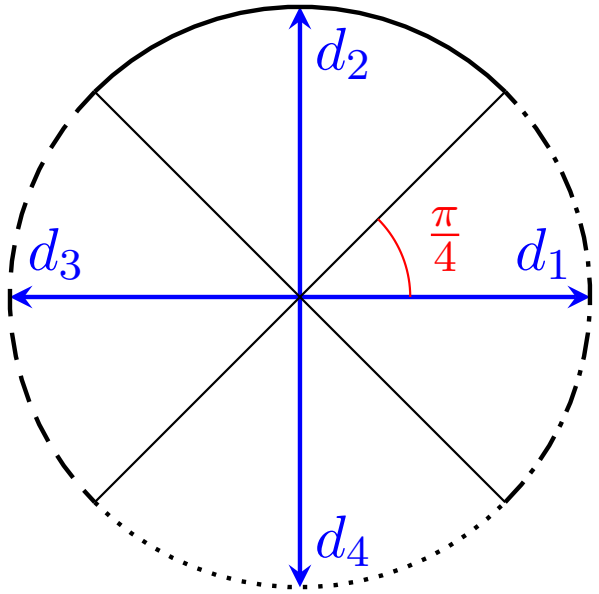
$\mathbb{S}^1 \subseteq \bigcup_{i=1}^{4} \mathbb{C}(d_i, \pi/4)$

PSSs and unit sphere covering

Given a PSS we consider a covering of the unit sphere by sph. caps.



Spherical cap $\mathbb{C}(x, \phi)$



$\mathbb{S}^1 \subseteq \bigcup_{i=1}^{4} \mathbb{C}(d_i, \pi/4)$

Lemma

For any PSS $D = [d_1 \cdots d_m]$ *in* $\mathbb{R}^n$ *with unit vectors*

$$\mathbb{S}^{n-1} \subseteq \bigcup_{i=1}^{m} \mathbb{C}\left(d_i, \arccos(\mathrm{cm}(D))\right).$$

Optimal order of the WCC bounds

Theorem (Tikhomirov 2014)

Any covering of $\mathbb{S}^{n-1}$ *by* $m \geq n+1$ *spherical caps of geodesic radius* ϕ *satisfies*

$$\cos(\phi) \;\leq\; \zeta\sqrt{n^{-1}\log(n^{-1}m)} \;\leq\; \zeta n^{-1}\sqrt{m}.$$

Optimal order of the WCC bounds

Theorem (Tikhomirov 2014)

Any covering of $\mathbb{S}^{n-1}$ *by* $m \geq n+1$ *spherical caps of geodesic radius* ϕ *satisfies*

$$\cos(\phi) \;\leq\; \zeta\sqrt{n^{-1}\log(n^{-1}m)} \;\leq\; \zeta n^{-1}\sqrt{m}.$$

Theorem

Any PSS D *in* $\mathbb{R}^n$ *satisfies*

$$\mathrm{cm}(D) \;\leq\; \zeta n^{-1}\sqrt{|D|}.$$

Optimal order of the WCC bounds

Theorem (Tikhomirov 2014)

Any covering of $\mathbb{S}^{n-1}$ *by* $m \geq n+1$ *spherical caps of geodesic radius* ϕ *satisfies*

$$\cos(\phi) \leq \zeta\sqrt{n^{-1}\log(n^{-1}m)} \leq \zeta n^{-1}\sqrt{m}.$$

Theorem

Any PSS D *in* $\mathbb{R}^n$ *satisfies*

$$\operatorname{cm}(D) \leq \zeta n^{-1}\sqrt{|D|}. \quad \left(\Rightarrow \frac{|D|}{\operatorname{cm}(D)^2} \geq \frac{1}{\zeta^2}n^2\right)$$

Optimal order of the WCC bounds

Theorem (Tikhomirov 2014)

Any covering of $\mathbb{S}^{n-1}$ *by* $m \geq n+1$ *spherical caps of geodesic radius* ϕ *satisfies*

$$\cos(\phi) \leq \zeta\sqrt{n^{-1}\log(n^{-1}m)} \leq \zeta n^{-1}\sqrt{m}.$$

Theorem

Any PSS D *in* $\mathbb{R}^n$ *satisfies*

$$\operatorname{cm}(D) \leq \zeta n^{-1}\sqrt{|D|}. \quad \left(\Rightarrow \frac{|D|}{\operatorname{cm}(D)^2} \geq \frac{1}{\zeta^2}n^2\right)$$

- M. Dodangeh, LNV, and Z. Zhang, On the optimal order of the worst case complexity of direct search, to appear in Optimization Letters.

Illustration of the n^2 order

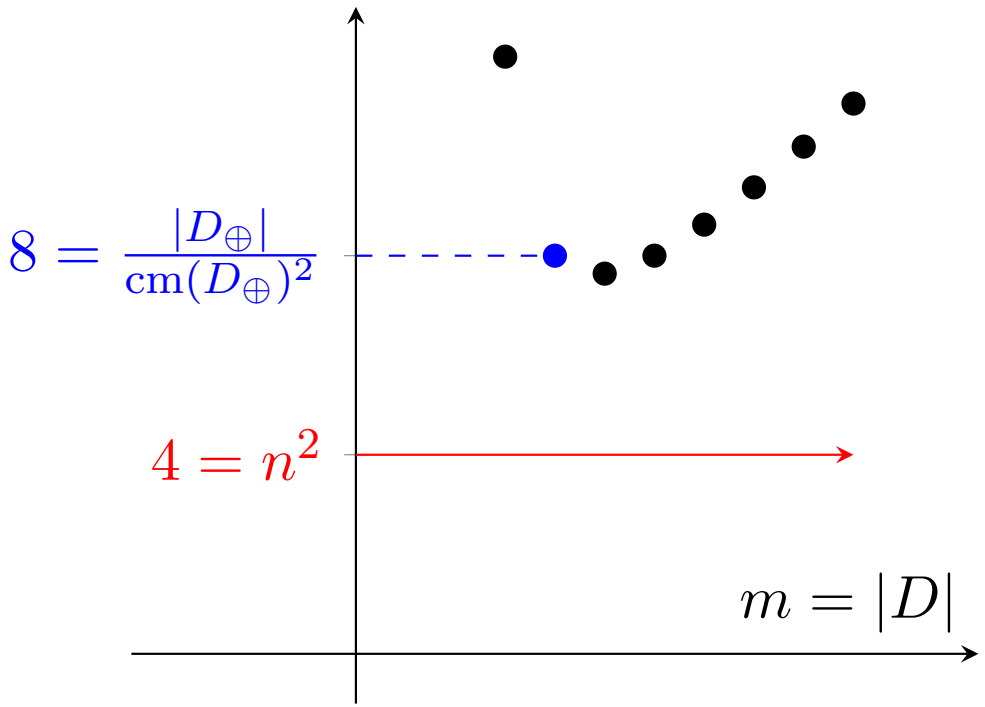


Illustration of

$$\frac{|D|}{\mathrm{cm}(D)^2} \geq \frac{1}{\zeta^2} n^2.$$

We plot the case $n = 2$ and D's with uniform angles (where $\frac{1}{\zeta^2}$ can be replaced by 1).

Global rate of DS (smooth, strongly convex case)

Theorem (M. Dodangeh and LNV 2014)

Any DS method (based on sufficient decrease) generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$f(x_k) - f_* < Cr^k,$$

where $r \in (0,1)$ *and* $C > 0$.

Global rate of DS (smooth, strongly convex case)

Theorem (M. Dodangeh and LNV 2014)

Any DS method (based on sufficient decrease) generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$f(x_k) - f_* \ < \ Cr^k,$$

where $r \in (0,1)$ *and* $C > 0$.

- When f is SC (constant $\mu > 0$), one has (k_0 first unsucc.)

$$\|x_k - x_*\| \ \leq \ \sqrt{L_{\nabla f}/\mu}\, \|x_{k_0} - x_*\|.$$

Global rate of DS (smooth, strongly convex case)

Theorem (M. Dodangeh and LNV 2014)

Any DS method (based on sufficient decrease) generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$f(x_k) - f_* \ < \ Cr^k,$$

where $r \in (0,1)$ *and* $C > 0$.

- When f is SC (constant $\mu > 0$), one has (k_0 first unsucc.)

$$\|x_k - x_*\| \ \leq \ \sqrt{L_{\nabla f}/\mu}\, \|x_{k_0} - x_*\|.$$

- A linear rate for the iterates can be derived from

$$\frac{1}{2}\mu\|x - x_*\|^2 \ \leq \ f(x) - f_*.$$

Rate in the strongly convex case (comparison to the state-of-the-art)

Under stronger assumptions (Dolan, Lewis, and Torczon 2003):

- α_k is monotonically non-increasing.
- x_k is sufficiently close to a point x_*.
- $\nabla^2 f(x)$ is positive definite around x_*.

The r-linear rate was only for unsuccessful iterates.

Rate in the strongly convex case (comparison to the state-of-the-art)

Under stronger assumptions (Dolan, Lewis, and Torczon 2003):

- α_k is monotonically non-increasing.
- x_k is sufficiently close to a point x_*.
- $\nabla^2 f(x)$ is positive definite around x_*.

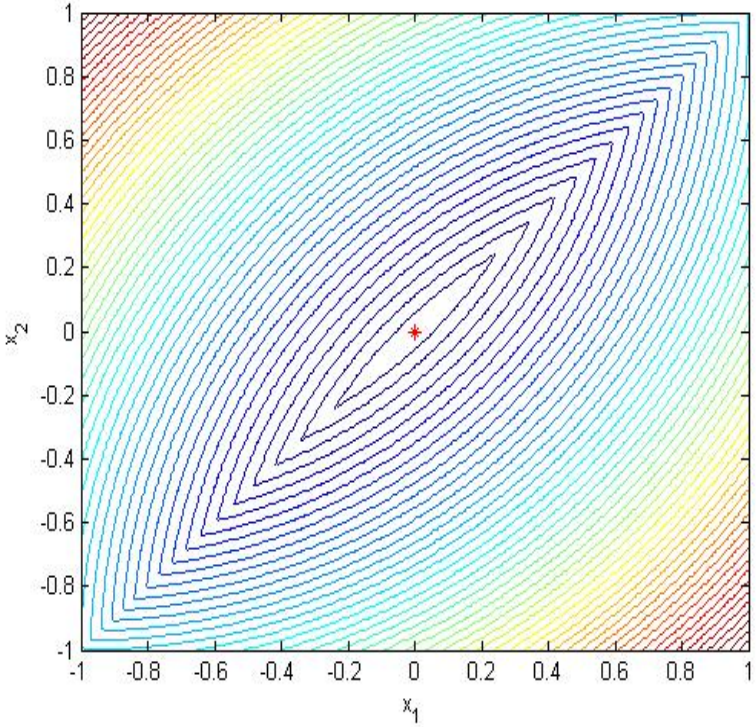
The r-linear rate was only for unsuccessful iterates.

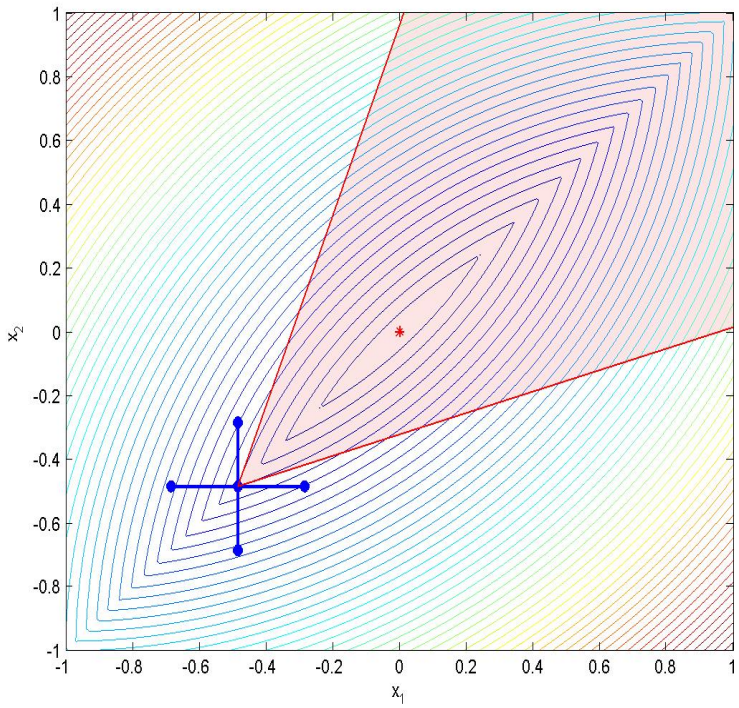
In the previous slide:

- α_k can be increased at successful iterates.
- no assumption on x_k.
- only Lipschitz continuity of ∇f is assumed.

The r-linear rate is over the whole sequence $\|x_k - x_*\|$, whether k is successful or not.

REMEMBER: Difficulties in the nonsmooth case





The cone of descent directions at the poll center is shaded.

WCC in the non-smooth case: Smoothing functions

Definition

We call $\tilde{f} : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}$ *a smoothing function of* f *if,* $\forall \mu \in (0, \infty)$, $\tilde{f}(\cdot, \mu)$ *is* $\mathcal{C}^1$ *and,* $\forall x \in \mathbb{R}^n$,

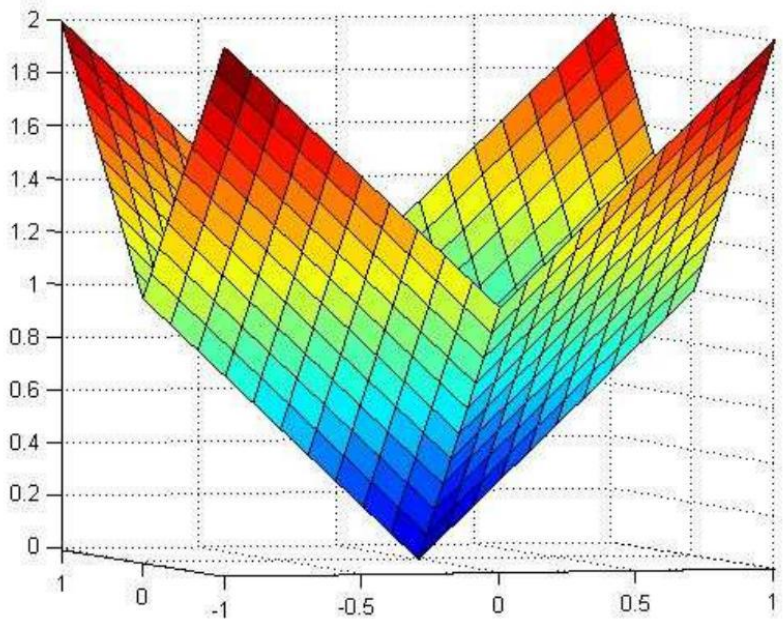
$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) \; = \; f(x).$$

WCC in the non-smooth case: Smoothing functions

Definition

We call $\tilde{f} : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}$ *a smoothing function of* f *if,* $\forall \mu \in (0, \infty)$, $\tilde{f}(\cdot, \mu)$ *is* $\mathcal{C}^1$ *and,* $\forall x \in \mathbb{R}^n$,

$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) \; = \; f(x).$$

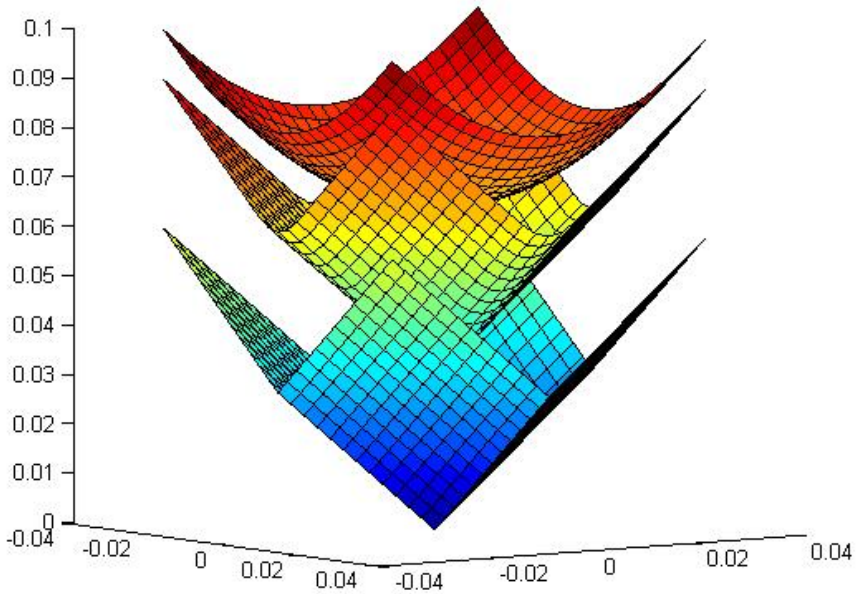


WCC in the non-smooth case: Smoothing functions

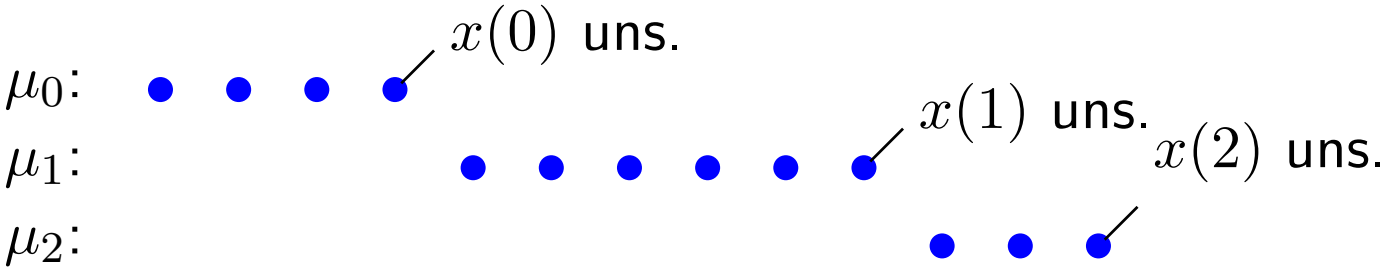
Definition

We call $\tilde{f} : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}$ *a smoothing function of* f *if,* $\forall \mu \in (0, \infty)$, $\tilde{f}(\cdot, \mu)$ *is* $\mathcal{C}^1$ *and,* $\forall x \in \mathbb{R}^n$,

$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) \; = \; f(x).$$



A class of smoothing DS methods



Initialization: Choose a function $r(\cdot)$ such that $\lim_{\mu \downarrow 0} r(\mu) = 0$.

Choose $\mu_0 > 0$, and $\sigma \in (0, 1)$

For $k = 0, 1, 2 \ldots$ (Until μ_k is suff. small)

- Apply DS to $\tilde{f}(\cdot, \mu_k)$ until step size $< r(\mu_k)$.
- Decrease the smoothing parameter: $\mu_{k+1} = \sigma \mu_k$.

Global convergence of smoothing DS (behavior of μ)

Assumption

Smoothing functions and their level sets are bounded for all k.

If we let DS run forever for a given k, then $\alpha \longrightarrow 0$. Thus

Theorem

The smoothing parameter goes to zero: $\lim_{k\to\infty} \mu_k = 0.$

Theorem

1. $\lim_{k\to+\infty} \alpha(k) = 0.$
2. $\exists x_*$ *and a subsequence* $K \subseteq \{(0), (1), \ldots\}$ *of unsucc. DS iterates such that* $x(k) \underset{K}{\longrightarrow} x_*$.

Global convergence of smoothing DS

Now, $\|\nabla \tilde{f}(x(k), \mu_k)\| \leq C(\tilde{\nu}(\mu_k))\, \alpha(k) \leq C(\tilde{\nu}(\mu_k))\, r(\mu_k)$.

Thus, choosing $r(\cdot)$ appropriately, i.e., $r(\mu) = \mu^2$ when $\tilde{\nu}(\mu) = \mathcal{O}\left(\frac{1}{\mu}\right)$:

Theorem

$$\lim_{k \in K} \|\nabla \tilde{f}(x(k), \mu_k)\| = 0$$

and x_* is stationary point associated with the smoothing function $\tilde{f}$.

Definition

We say that x_* is a stationary point associated with the smoothing function $\tilde{f}$ if $0 \in G_{\tilde{f}}(x_*)$, where

$$G_{\tilde{f}}(x_*) = \{\text{all limits of } \nabla \tilde{f}(x, \mu) \text{ when } x \to x_* \text{ and } \mu \to 0\}.$$

Clarke generalized derivative and subdifferential

Does $0 \in G_{\tilde{f}}(x_*)$ mean any form of true stationarity?

Definition

Let f be Lipschitz cont. near x. Remember that the Clarke generalized directional derivative is defined by

$$f^\circ(x;v) \;=\; \limsup_{\bar{x}\to x \;\; t\downarrow 0} \frac{f(\bar{x}+tv)-f(\bar{x})}{t}.$$

The Clarke subdifferential is then given by:

$$\partial f(x) \;=\; \{s \in \mathbb{R}^n : f^\circ(x;v) \geq \langle v, s\rangle,\ \forall v \in \mathbb{R}^n\}.$$

Clarke stationarity

If x_* is a local minimizer, $f^\circ(x_*;v) \geq 0,\ \forall v \in \mathbb{R}^n$ or, equivalently, $0 \in \partial f(x_*)$.

How to construct smoothing functions

There are forms of building smoothing functions $\tilde{f}$ such that

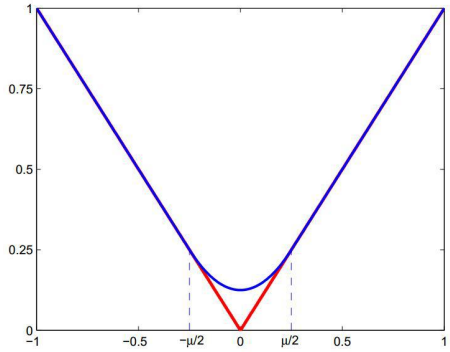
- $\tilde{f}$ satisfies the gradient consistency property

$$\partial f(x_*) \;=\; \text{co}\; G_{\tilde{f}}(x_*).$$

  Thus, if $0 \in G_{\tilde{f}}(x_*) \subset \text{co}\, G_{\tilde{f}}(x_*)$, then $0 \in \partial f(x_*)$.

- $\tilde{\nu}(\mu) \;=\; \mathcal{O}\left(\frac{1}{\mu}\right)$.

Chen and Zhou introduced such a smoothing function $\tilde{s}(t, \mu)$ of $|t|$:



Then we obtain $\tilde{F}(x, \mu) = \sum_{i=1}^{m} \tilde{s}(F_i(x), \mu)$ for $\|F\|_1 = \sum_{i=1}^{m} |F_i|$.

WCC of smoothing DS (to reduce μ)

Theorem

Let $\rho(\alpha) = \alpha^p$ *and* $r(\alpha) = \alpha^q$, *with* $p, q > 1$.

Any smoothing DS (based on sufficient decrease) takes at most

$$\mathcal{O}\left((-\log(\xi))\xi^{-pq}\right)$$

DS inner iterations to reduce μ *below* $\xi \in (0, 1)$.

WCC of smoothing DS (to reduce smoothing gradient)

Corollary

Assume $\tilde{\nu}(\mu) = \mathcal{O}(1/\mu)$.

When μ *becomes lower than* ξ, $\nabla \tilde{f}$ *becomes*

$$\mathcal{O}\left(n^{\frac{1}{2}}(\xi^{q-1} + \xi^{(p-1)q})\right).$$

So, for having $\xi^{q-1} + \xi^{(p-1)q} = \mathcal{O}(\xi)$, one selects

$$p = \frac{3}{2} \quad \text{and} \quad q = 2$$

leading to

$$\mathcal{O}(n^{\frac{1}{2}}\xi).$$

WCC of smoothing DS (function evaluations)

Therefore, the number of iterations needed to reach $\|\nabla \tilde{f}\| \leq \epsilon$ and $\mu \leq \xi = \mathcal{O}(n^{-\frac{1}{2}}\epsilon)$ is

$$\mathcal{O}\left((-\log(\xi))\xi^{-pq}\right) \overset{p=\frac{3}{2},q=2}{=} \mathcal{O}\left(n^{\frac{3}{2}}[-\log(\epsilon)+\log(n)]\epsilon^{-3}\right).$$

In terms of function evaluations:

$$\mathcal{O}\left(n^{\frac{5}{2}}[-\log(\epsilon)+\log(n)]\epsilon^{-3}\right).$$

This compares to $\mathcal{O}\left(n^3\epsilon^{-3}\right)$ using Gaussian densities (Nesterov, 2011).

Reference:

- R. Garmanjani and L. N. Vicente, Smoothing and worst-case complexity for direct-search methods in non-smooth optimization, IMA Journal of Numerical Analysis, 33 (2013) 1008–1028.

Another perspective on WCC of smoothing DS

By choosing a fixed μ, a DS algorithm takes

$$\mathcal{O}\left(\left(\sqrt{n}\tilde{\nu}(\mu)\right)^{\frac{p}{\min(p-1,1)}}\epsilon^{-\frac{p}{\min(p-1,1)}}\right)$$

iterations to bring the norm of the smoothing gradient below ϵ.

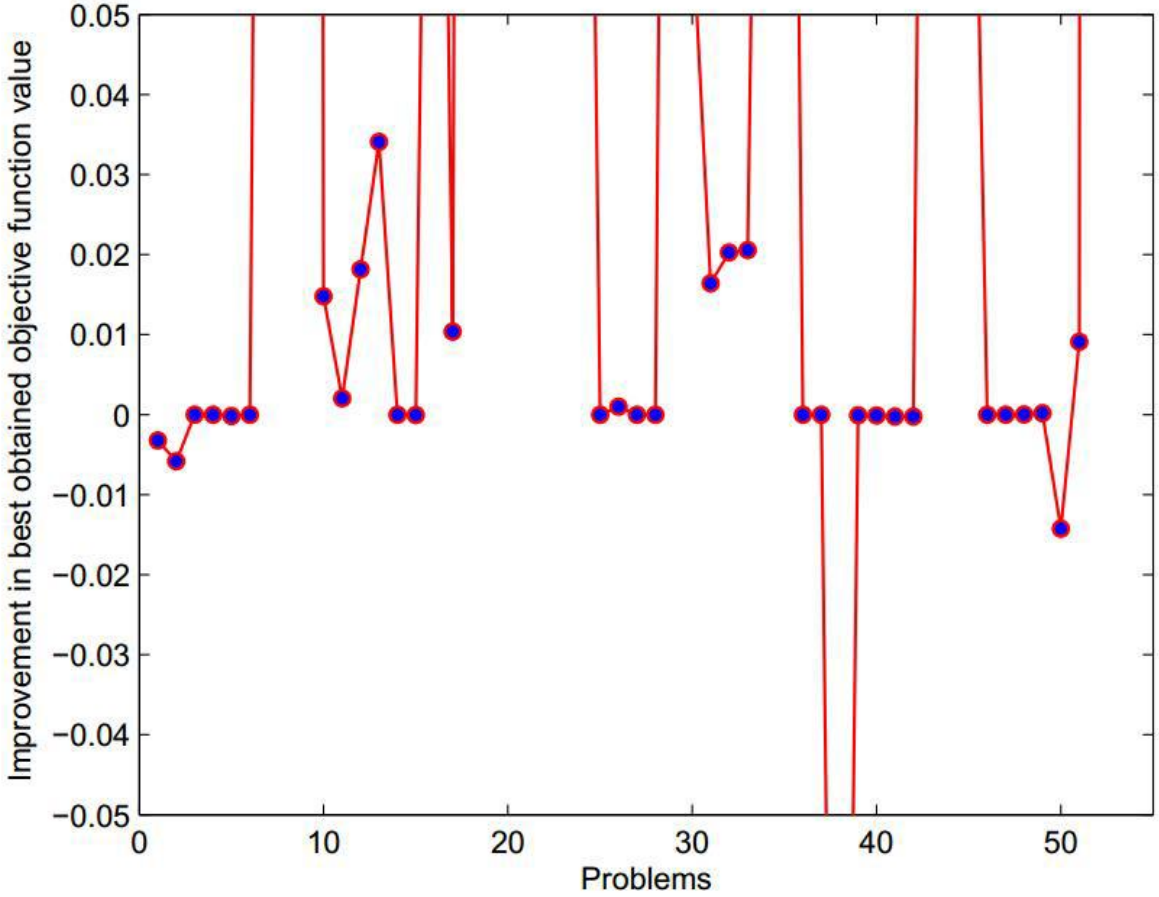
When $\tilde{\nu}(\mu) = \mathcal{O}(1/\mu)$ and $\mu \leq \epsilon$ $(= \xi)$, one then obtains

$$\mathcal{O}\left(\left(\sqrt{n}\right)^{\frac{p}{\min(p-1,1)}}\epsilon^{-\frac{2p}{\min(p-1,1)}}\right),$$

leading to the optimal choice $p = 2$ and a WCC of $\mathcal{O}(n\epsilon^{-4})$ direct-search iterations, and thus $\mathcal{O}(n^2\epsilon^{-4})$ in function evaluations.

Thus, not recommended.

Comparing final function values



DS for fixed $\mu = 10^{-4}$ vs Smoothing DS

Both strategies were run for a fixed budget.

A smoothing function for $|\cdot|$

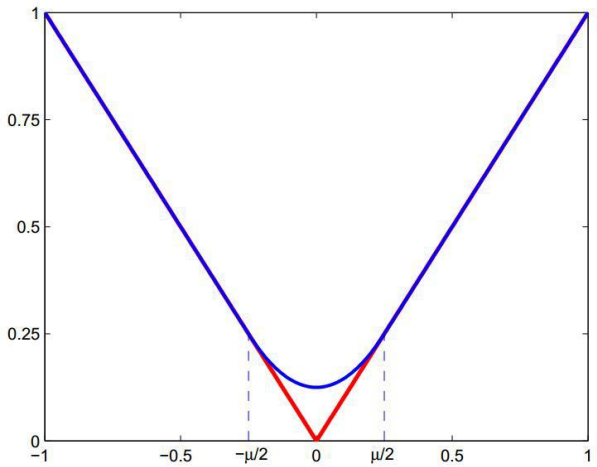
Chen and Zhou have introduced the following smoothing function of $|t|$:

$$\tilde{s}(t,\mu) = \int_{-\infty}^{+\infty} |t-\mu\tau|\varrho(\tau)d\tau.$$

Proposition (Chen and Zhou 2010)

(i) $\tilde{s}$ *satisfies the gradient consistent property.*

(ii) $\tilde{s}'(t,\mu)$ *is Lipschitz continuous with constant* $L_{\tilde{s}'}(\mu) = \mathcal{O}\left(\frac{1}{\mu}\right)$.



A smoothing function for $\|F(\cdot)\|_1$

Now, let $f : \mathbb{R}^n \to \mathbb{R}$ be C^1. Then $\tilde{s}(f)$ is a smoothing function of $|f|$.

Let $F = (F_1, \ldots, F_m) : \mathbb{R}^n \to \mathbb{R}^m$, where each ∇F_i is Lips. continuous.

Theorem

(i) $\tilde{F}(x, \mu) = \sum_{i=1}^m \tilde{s}(F_i(x), \mu)$ *is a smooth. func. of* $\|F\|_1 = \sum_{i=1}^m |F_i|$.

(ii) $\tilde{F}(\cdot, \mu)$ *satisfies the gradient consistent property*

$$G_{\tilde{F}}(x_*) \;=\; \left\{ \lim_{x \to x_*, \mu \downarrow 0} \nabla \tilde{F}(x, \mu) \right\} \quad = \quad \partial \|F\|_1(x_*).$$

(iii) For each μ, $\nabla \tilde{F}(\cdot, \mu)$ *is Lips. cont. with constant* $L_{\nabla \tilde{F}}(\mu) = \mathcal{O}\left(\frac{1}{\mu}\right)$.

Numerical testing

We have tested the smoothing direct-search approach on the MATLAB direct-search `sid-psm` code:

- A. L. Custódio and L. N. Vicente, SIOPT, 18 (2007), 537-555.
- A. L. Custódio, H. Rocha, and L. N. Vicente, COAP, 46 (2010) 265–278.

We tested the piecewise-linear problems ($\min \|F(\cdot)\|_1$) from:

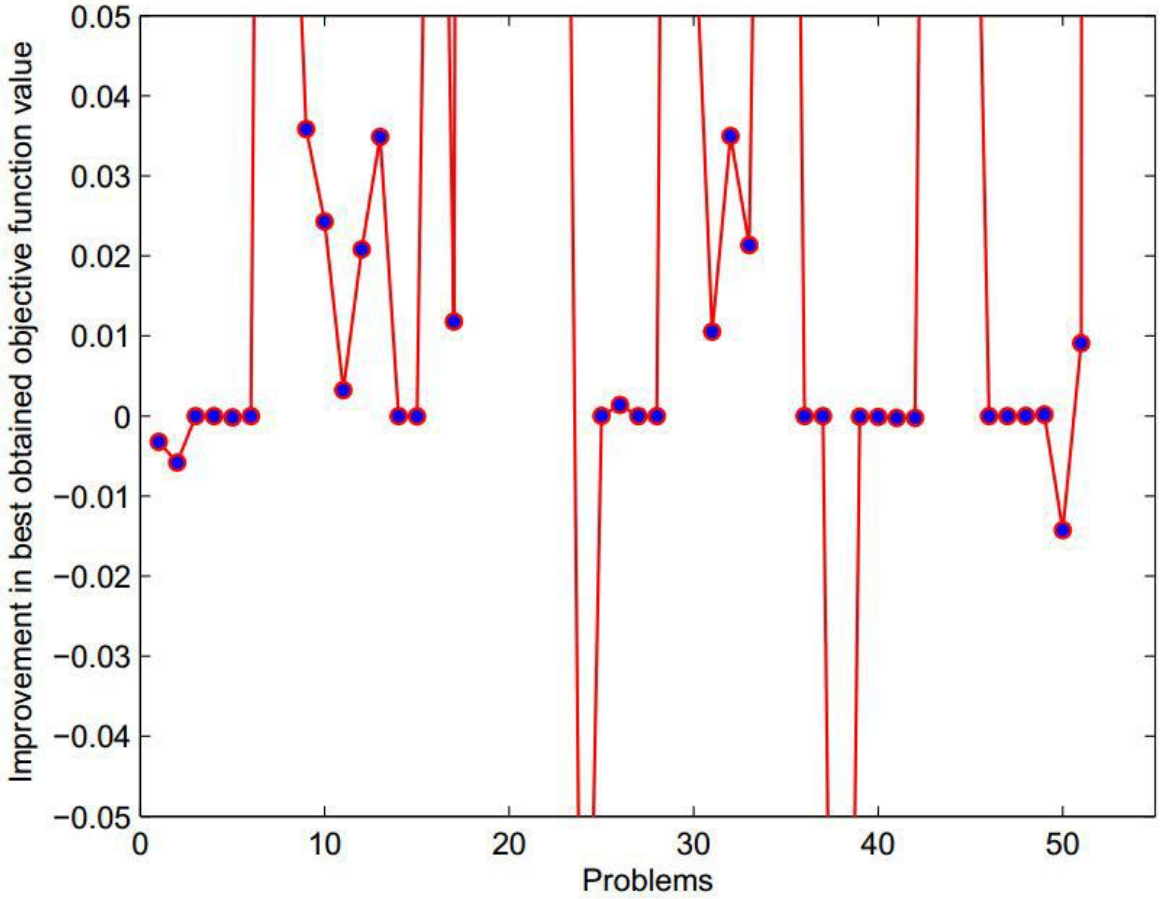
- J. J. Moré and S. M. Wild, SIOPT, 20 (2009), 172–191.

With the following initial parameters, functions, and budget:

$$\mu_0 = 10^{-2},\ r(\mu) = \max(10^{-5}, \mu^2), \text{ and } \mu_{k+1} = \mu_k/10$$

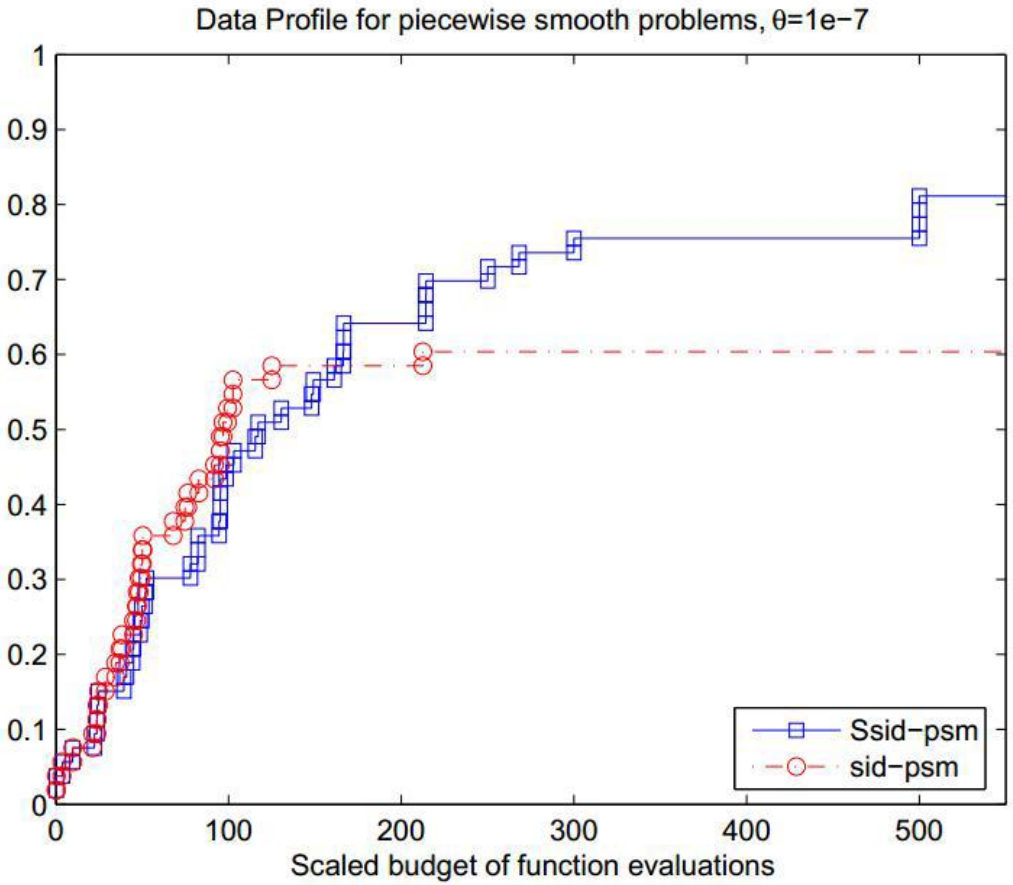
Maximum number of function evaluations = 1500

Comparing final function values



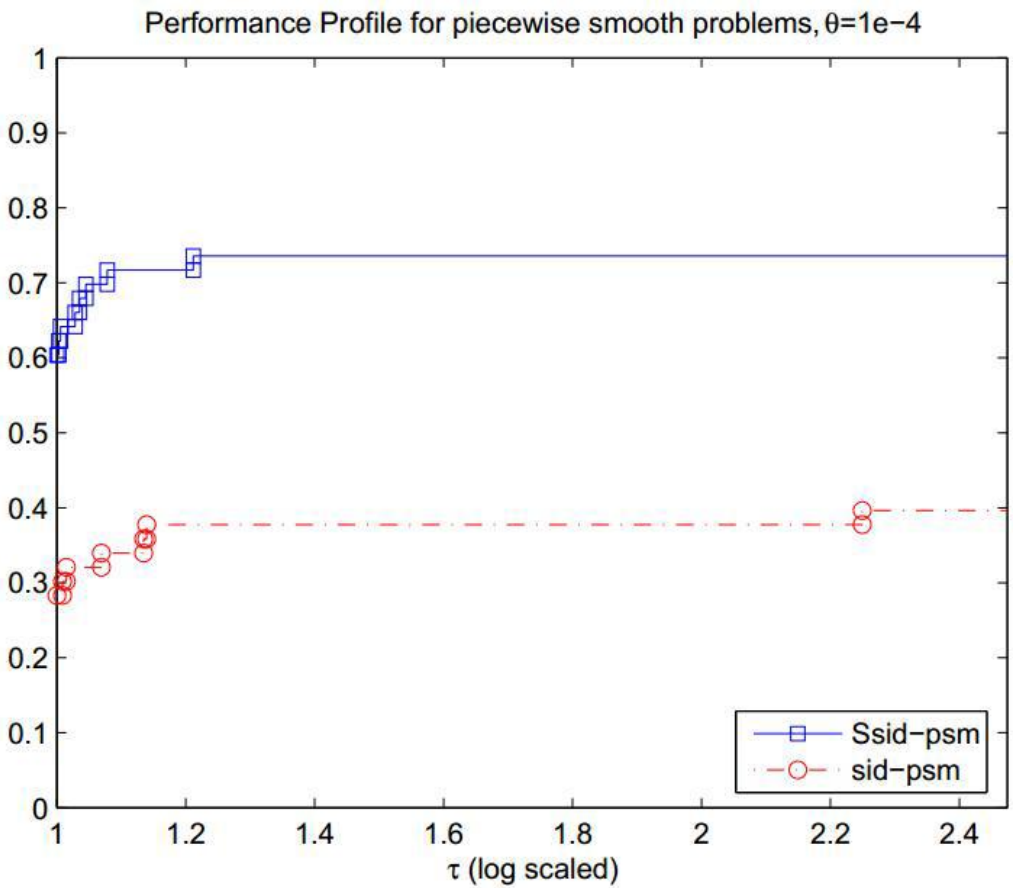
Smoothing DS vs DS
(search step and simplex ordering).

Comparing progress for a given budget



Data profile with the accuracy of 10^{-7} (search step and simplex ordering).

Comparing efficiency/robustness



Performance profile with the accuracy of 10^{-4} (search step and simplex ordering).

Summary of DS global rates

Imposing sufficient decrease to accept new iterates:

Summary of DS global rates

Imposing sufficient decrease to accept new iterates:

- $\mathcal{O}(\epsilon^{-1})$ smooth, convex — global rate $1/k$ for f and ∇f.

Summary of DS global rates

Imposing sufficient decrease to accept new iterates:

- $\mathcal{O}(\epsilon^{-1})$ smooth, convex — global rate $1/k$ for f and ∇f.
- $\mathcal{O}(\epsilon^{-2})$ smooth, non-convex — global rate $1/\sqrt{k}$ for ∇f.

Summary of DS global rates

Imposing sufficient decrease to accept new iterates:

- $\mathcal{O}(\epsilon^{-1})$ smooth, convex — global rate $1/k$ for f and ∇f.
- $\mathcal{O}(\epsilon^{-2})$ smooth, non-convex — global rate $1/\sqrt{k}$ for ∇f.
- In terms of function evaluations: $\mathcal{O}(n^2\epsilon^{-1})$, $\mathcal{O}(n^2\epsilon^{-2})$. The factor n^2 is proved approximately optimal.

Summary of DS global rates

Imposing sufficient decrease to accept new iterates:

- $\mathcal{O}(\epsilon^{-1})$ smooth, convex — global rate $1/k$ for f and ∇f.
- $\mathcal{O}(\epsilon^{-2})$ smooth, non-convex — global rate $1/\sqrt{k}$ for ∇f.
- In terms of function evaluations: $\mathcal{O}(n^2\epsilon^{-1})$, $\mathcal{O}(n^2\epsilon^{-2})$. The factor n^2 is proved approximately optimal.
- $\mathcal{O}(\epsilon^{-3})$ non-smooth, non-convex — (using smoothing techniques).

Presentation outline

In model-based trust-region methods...

- One typically minimizes a model m in a trust region $B(x;\Delta)$:

Trust-region subproblem

$$\min_{y \in B(x;\Delta)} m(y)$$

In derivative-based optimization, one could use:

1st order Taylor:

$$m(y) = f(x) + \nabla f(x)^\top (y - x)$$

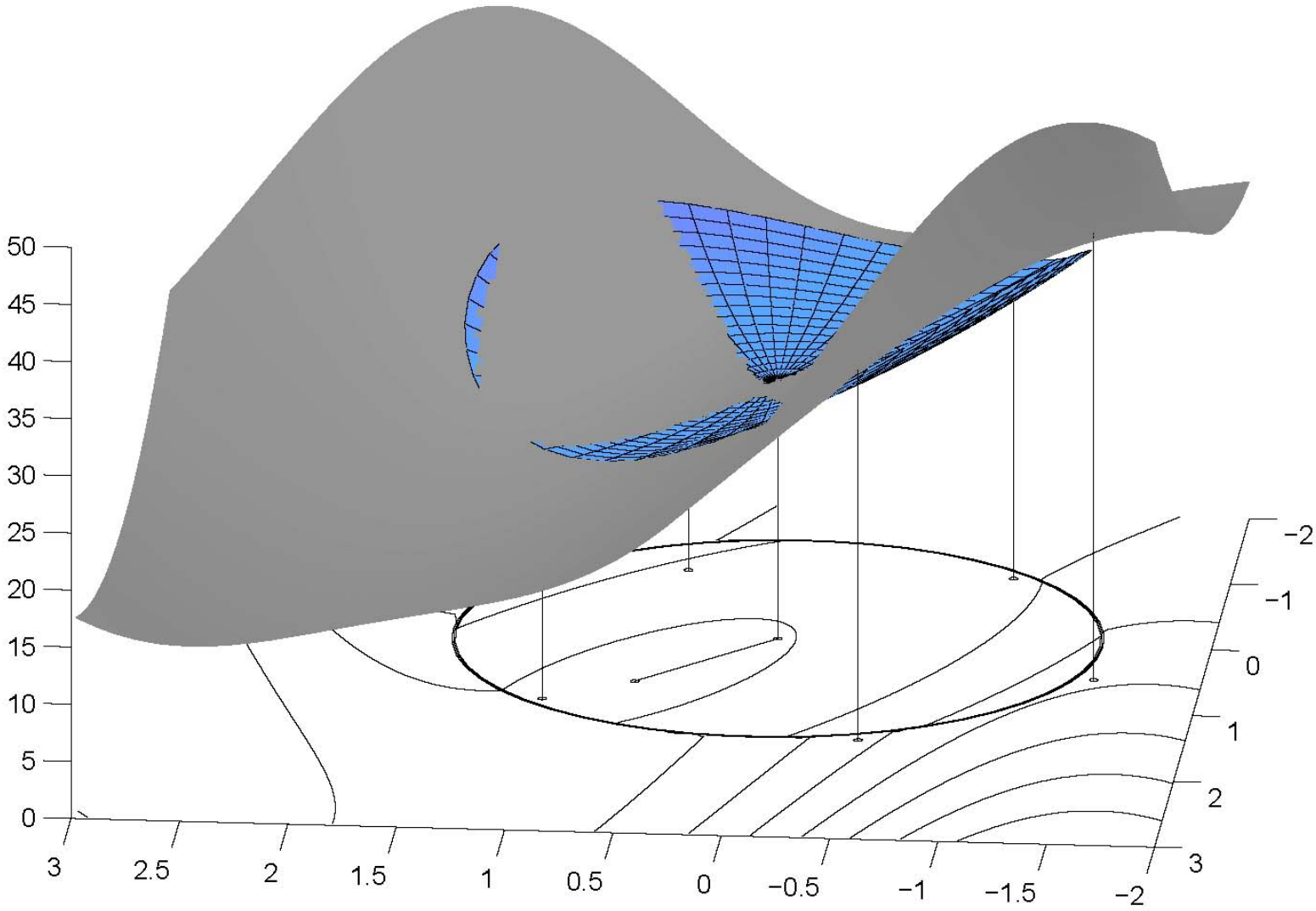
1st order Taylor:

$$m(y) = f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2}(y - x)^\top H (y - x)$$

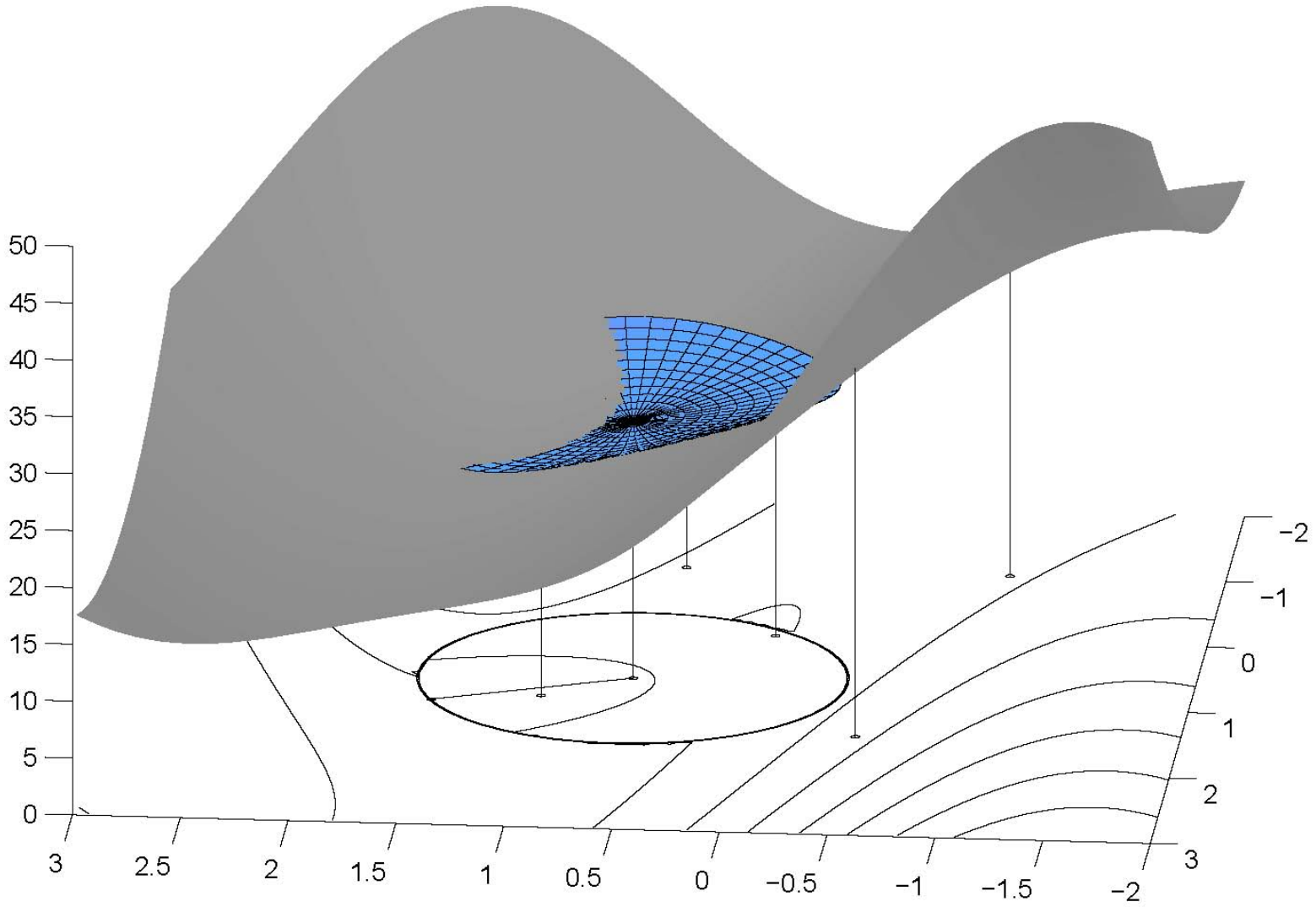
2nd order Taylor:

$$m(y) = f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2}(y - x)^\top \nabla^2 f(x)(y - x)$$

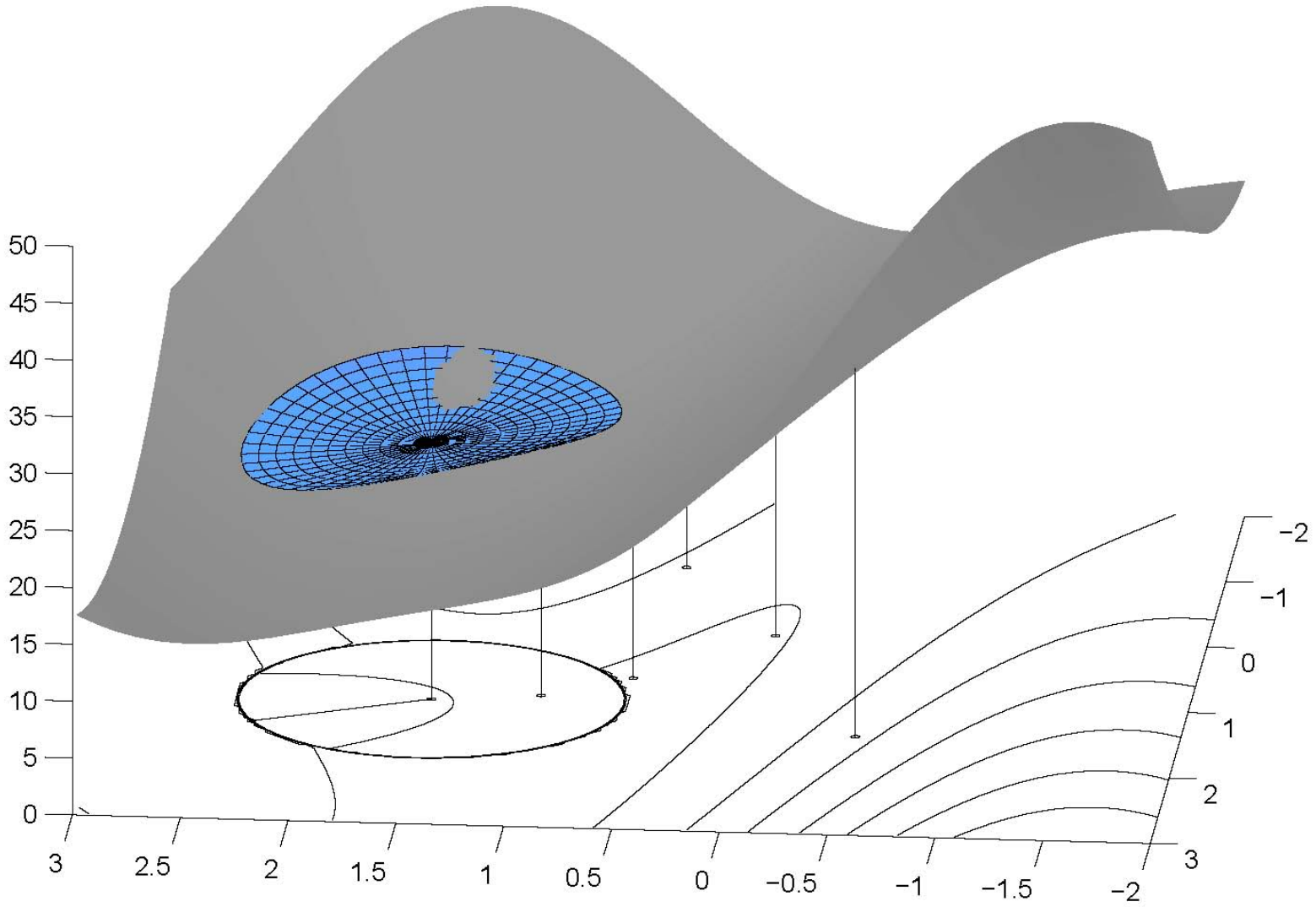
When using interpolation (example of ill poisedness)



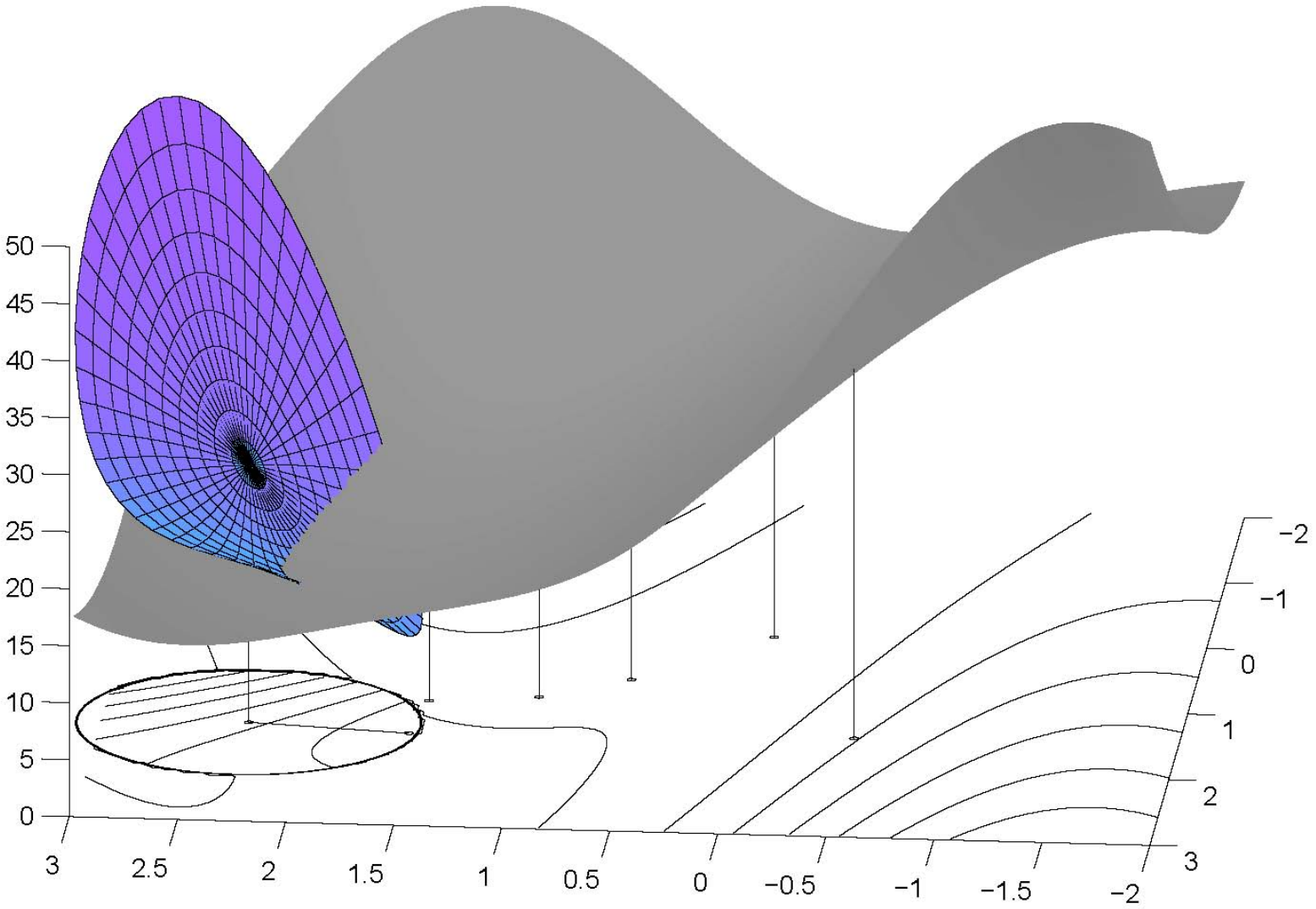
When using interpolation (example of ill poisedness)



When using interpolation (example of ill poisedness)



When using interpolation (example of ill poisedness)



Fully linear models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully linear if

- It is $\mathcal{C}^1$ with Lipschitz continuous gradient.
- The following error bounds hold:

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\,\Delta \qquad \forall y \in B(x;\Delta)$$

$$|f(y) - m(y)| \leq \kappa_{ef}\,\Delta^2 \qquad \forall y \in B(x;\Delta).$$

For a class of fully linear models, the (unknown) constants $\kappa_{ef}, \kappa_{eg} > 0$ must be independent of x and Δ.

Fully linear models can be quadratic (or even nonlinear).

Fully linear models

For a class of fully linear models, one must also guarantee that:

- There exists a model-improvement algorithm, that in a finite, uniformly bounded (with respect to x and Δ) number of steps can:

  — certificate that a given model is fully linear on $B(x; \Delta)$,

  — or (if the above fails), find a model that is fully linear on $B(x; \Delta)$.

Fully quadratic models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully quadratic if

- It is $\mathcal{C}^2$ with Lipschitz continuous Hessian.
- The following error bounds hold:

$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \leq \kappa_{eh}\,\Delta \qquad \forall y \in B(x;\Delta)$$

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\,\Delta^2 \qquad \forall y \in B(x;\Delta)$$

$$|f(y) - m(y)| \leq \kappa_{ef}\,\Delta^3 \qquad \forall y \in B(x;\Delta).$$

For a class of fully quadratic models, the (unknown) constants $\kappa_{ef}, \kappa_{eg}, \kappa_{eh} > 0$ must be independent of x and Δ.

Fully quadratic models are only necessary for global convergence to 2nd order stationary points.

Fully quadratic models

For a class of fully quadratic models, one must also guarantee that:

- There exists a model-improvement algorithm, that in a finite, uniformly bounded (with respect to x and Δ) number of steps can:

  — certificate that a given model is fully quadratic on $B(x; \Delta)$,

  — or (if the above fails), find a model that is fully quadratic on $B(x; \Delta)$.

Polynomial models

Given a sample set $Y = \{y^0, y^1, \ldots, y^p\}$, a polynomial basis ϕ for $\mathcal{P}_n^d$, and a polynomial model $m(y) = \alpha^\top \phi(y)$, the interpolating conditions are the following system of linear equations:

$$M(\phi, Y)\alpha \;=\; f(Y),$$

where

$$M(\phi, Y) = \begin{bmatrix} \phi_0(y^0) & \phi_1(y^0) & \cdots & \phi_p(y^0) \\ \phi_0(y^1) & \phi_1(y^1) & \cdots & \phi_p(y^1) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_0(y^p) & \phi_1(y^p) & \cdots & \phi_p(y^p) \end{bmatrix} \quad f(Y) = \begin{bmatrix} f(y^0) \\ f(y^1) \\ \vdots \\ f(y^p) \end{bmatrix}.$$

Example of the interpolation matrix

For instance, when $n = d = 2$, $p = 5$, and

$$\phi \;=\; \{1, x_1, x_2, x_1^2/2, x_2^2/2, x_1 x_2\},$$

the matrix $M(\phi, Y)$ becomes

$$\begin{bmatrix} 1 & y_1^0 & y_2^0 & (y_1^0)^2/2 & y_1^0 y_2^0 & (y_2^0)^2/2 \\ 1 & y_1^1 & y_2^1 & (y_1^1)^2/2 & y_1^1 y_2^1 & (y_2^1)^2/2 \\ 1 & y_1^2 & y_2^2 & (y_1^2)^2/2 & y_1^2 y_2^2 & (y_2^2)^2/2 \\ 1 & y_1^3 & y_2^3 & (y_1^3)^2/2 & y_1^3 y_2^3 & (y_2^3)^2/2 \\ 1 & y_1^4 & y_2^4 & (y_1^4)^2/2 & y_1^4 y_2^4 & (y_2^4)^2/2 \\ 1 & y_1^5 & y_2^5 & (y_1^5)^2/2 & y_1^5 y_2^5 & (y_2^5)^2/2 \end{bmatrix}.$$

Polynomial models

The system

$$M(\phi, Y)\alpha \;=\; f(Y)$$

can be

- Determined when (# points) $=$ (# basis components).
- Undetermined when (# points) $<$ (# basis components).

  $\longrightarrow$ minimum-norm solution

  $\longrightarrow$ minimum Frobenius norm ('Hessian components ℓ_2-norm') solution

  $\longrightarrow$ minimum 'Hessian components ℓ_1-norm' solution
- Overdetermined when (# points) $>$ (# basis components).

  $\longrightarrow$ least-squares regression solution.

Polynomial models

Linear interpolation: $d = 1$
In the determined case: # points $= n + 1$.
Natural basis of monomials ($n = 2$): $\bar{\phi} = \{1, x_1, x_2\}$.

Quadratic interpolation: $d = 2$
In the determined case: # points $p = 1 + n + n(n+1)/2$.
Natural basis of monomials ($n = 2$): $\bar{\phi} = \{1, x_1, x_2, x_1^2/2, x_2^2/2, x_1 x_2\}$.

Natural basis

The natural basis is the basis of polynomials as they appear in the Taylor expansion.

For instance, assuming the appropriate smoothness, the Taylor model of order $d = 2$ in $\mathbb{R}^3$, centered at the point y, is the following polynomial in z_1, z_2, and z_3:

$$f(y)\,[1] + \frac{\partial f}{\partial x_1}(y)[z_1] + \frac{\partial f}{\partial x_2}(y)[z_2] + \frac{\partial f}{\partial x_3}(y)[z_3]$$
$$+ \frac{\partial^2 f}{\partial x_1^2}(y)[z_1^2/2] + \frac{\partial^2 f}{\partial x_1 x_2}(y)[z_1 z_2] + \frac{\partial^2 f}{\partial x_2^2}(y)[z_2^2/2]$$
$$+ \frac{\partial^2 f}{\partial x_1 x_3}(y)[z_1 z_3] + \frac{\partial^2 f}{\partial x_2 x_3}(y)[z_2 z_3] + \frac{\partial^2 f}{\partial x_3^2}(y)[z_3^2/2].$$

Poisedness for interpolation

Definition

The set $Y = \{y^0, y^1, \ldots, y^p\}$ is poised for polynomial interpolation in $\mathbb{R}^n$ if the corresponding matrix $M(\phi, Y)$ is nonsingular for some basis ϕ in $\mathcal{P}_n^d$.

Simple facts:

If $M(\phi, Y)$ is nonsingular for some basis ϕ then it is nonsingular for any basis of $\mathcal{P}_n^d$.

Under these circumstances, the interpolating polynomial $m(x)$ exists and is unique.

Poisedness for interpolation

The matrix $M(\phi, Y)$ is singular if and only if there exists $\gamma \in \mathbb{R}^{p+1}$ such that $\gamma \neq 0$ and $M(\phi, Y)\gamma = 0$ and that implies that there exists a polynomial, of degree at most d, expressed as

$$m(x) = \sum_{j=0}^{p} \gamma_j \phi_j(x),$$

such that $m(y) = 0$ for all $y \in Y$.

In other words, $M(\phi, Y)$ is singular if and only if the points of Y lie on a polynomial manifold of degree d or less.

For instance, 6 points on a second-order curve in $\mathbb{R}^2$, such as a curve, form a non-poised set for quadratic interpolation.

The conditioning of the interpolation matrix

Since nonsingularity of $M(\phi, Y)$ is the indicator of poisedness, will the condition number be an indicator of well poisedness?

The answer, in general, is 'no', since the condition number of $M(\phi, Y)$ depends on the choice of ϕ.

Moreover, for any given poised interpolation set Y, one can choose the basis ϕ so that the condition number of $M(\phi, Y)$ can equal any number between 1 and $+\infty$.

The conditioning of the interpolation matrix

Also, for any fixed choice of ϕ, the condition number of $M(\phi, Y)$ depends on the scaling of Y.

Hence, the condition number of $M(\phi, Y)$ is not considered to be a good characterization of the level of poisedness of a set of points.

However, we will show that for a specific choice of ϕ, namely for the natural basis $\bar{\phi}$, and for Y_{scaled}, a scaled version of Y, the condition number of $M(\bar{\phi}, Y_{scaled})$ is a meaningful measure of well poisedness.

Lagrange polynomials

A fundamental tool in interpolation are Lagrange polynomials:

Definition

Given a set of interpolation points $Y = \{y^0, y^1, \ldots, y^p\}$, *a basis of* $p+1$ *polynomials* $\ell_j(x)$, $j = 0, \ldots, p$, *in* $\mathcal{P}_n^d$, *is called a basis of Lagrange polynomials if*

$$\ell_j(y^i) \;=\; \delta_{ij} \;=\; \begin{cases} 1 & \text{if } \; i = j, \\ 0 & \text{if } \; i \neq j. \end{cases}$$

If Y is poised, Lagrange polynomials exist, are unique, and have a number of useful properties.

Lagrange polynomials

The existence of Lagrange polynomials, in turn, implies poisedness of the sampling set.

$m(x)$ in $\mathcal{P}_n^d$ is a linear combination of Lagrange polynomials whose coefficients are the interpolation values at the points in Y:

Theorem

For any function $f : \mathbb{R}^n \to \mathbb{R}$ and any poised set $Y = \{y^0, y^1, \ldots, y^p\} \subset \mathbb{R}^n$, the unique polynomial $m(x)$ that interpolates $f(x)$ on Y can be expressed as

$$m(x) = \sum_{i=0}^{p} f(y^i)\ell_i(x),$$

where $\{\ell_i(x), i = 0, \ldots, p\}$ is the basis of Lagrange polynomials for Y.

Example of Lagrange polynomials

To illustrate LPs in $\mathbb{R}^2$, consider interpolating the cubic function

$$f(x_1, x_2) \;=\; x_1 + x_2 + 2x_1^2 + 3x_2^3$$

at the six interpolating points $y^0 = (0,0)$, $y^1 = (1,0)$, $y^2 = (0,1)$, $y^3 = (2,0)$, $y^4 = (1,1)$, and $y^5 = (0,2)$. One has $f(y^0) = 0, f(y^1) = 3, f(y^2) = 4$, $f(y^3) = 10$, $f(y^4) = 7$, and $f(y^5) = 26$.

The Lagrange polynomials $\ell_j(x_1, x_2), j = 0, \ldots, 5$, are given by

$$\begin{aligned}
\ell_0(x_1,x_2) &= 1 - \tfrac{3}{2}x_1 - \tfrac{3}{2}x_2 + \tfrac{1}{2}x_1^2 + \tfrac{1}{2}x_2^2 + x_1x_2, \\
\ell_1(x_1,x_2) &= 2x_1 - x_1^2 - x_1x_2, \\
\ell_2(x_1,x_2) &= 2x_2 - x_2^2 - x_1x_2, \qquad \ell_3(x_1,x_2) \;=\; -\tfrac{1}{2}x_1 + \tfrac{1}{2}x_1^2, \\
\ell_4(x_1,x_2) &= x_1x_2, \qquad \ell_5(x_1,x_2) \;=\; -\tfrac{1}{2}x_2 + \tfrac{1}{2}x_2^2.
\end{aligned}$$

Then:

$$\begin{aligned}
m(x_1,x_2) \;=\;& 0\ell_0(x_1,x_2) + 3\ell_1(x_1,x_2) + 4\ell_2(x_1,x_2) + 10\ell_3(x_1,x_2) \\
&+ 7\ell_4(x_1,x_2) + 26\ell_5(x_1,x_2).
\end{aligned}$$

Second definition of LPs

The Lagrange polynomials are the solution of:

$$\sum_{i=0}^{p} \ell_i(x)\phi(y^i) \;=\; \phi(x),$$

or, in matrix form, of

$$M(\phi, Y)^\top \ell(x) \;=\; \phi(x), \quad \text{where } \ell(x) \;=\; [\ell_0(x), \ldots, \ell_p(x)]^\top.$$

Consider a mapping $x \to \phi(x)$. Let $\phi(Y)$ and $\phi(B)$ be the images of Y and B under this mapping.

If $\phi(Y)$ spans $\phi(B)$ well, then any vector in $\phi(B)$ can be expressed as a linear combination of vectors in $\phi(Y)$ with reasonably small coefficients.

Third definition of LPs

Given the set Y and a point x consider the set $Y_i(x) = Y \setminus \{y^i\} \cup \{x\}$, $i = 0, \ldots, p$.

From applying Cramer's rule, we see

$$\ell_i(x) \;=\; \frac{\det(M(\phi, Y_i(x)))}{\det(M(\phi, Y))}.$$

It follows that the Lagrange polynomials do not depend on the choice of ϕ, as long as the polynomial space $\mathcal{P}_n^d$ is fixed.

Third definition of LPs

Consider a set $\phi(Y) = \{\phi(y^i), i = 0, \ldots, p\}$ in $\mathbb{R}^{p+1}$. Let the volume of the simplex of vertices in $\phi(Y)$ be

$$\text{vol}(\phi(Y)) \;=\; \frac{|\det(M(\phi, Y))|}{(p+1)!}.$$

Such a simplex is the $(p+1)$-dimensional convex hull of $\phi(Y)$. Then

$$|\ell_i(x)| \;=\; \frac{\text{vol}(\phi(Y_i(x)))}{\text{vol}(\phi(Y))}.$$

In other words, the absolute value of the i-th Lagrange polynomial at a given point x is the change in the volume in this simplex when y^i is replaced by x.

Λ–poisedness

Definition

Let $\Lambda > 0$ and a set $B \in \mathbb{R}^n$ be given. Let ϕ be a basis in $\mathcal{P}_n^d$.

A poised set $Y = \{y^0, y^1, \ldots, y^p\}$ is said to be Λ–poised in B (in the interpolation sense) if and only if

1. $$\Lambda \ \geq \ \max_{0 \leq i \leq p} \max_{x \in B} |\ell_i(x)|, \quad \text{or, equivalently,}$$

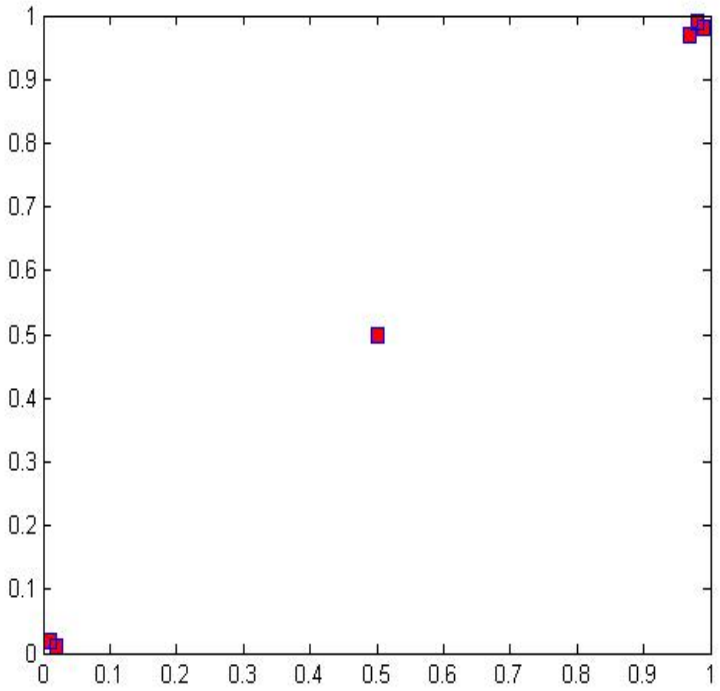
2. *for any $x \in B$ there exists $\ell(x) \in \mathbb{R}^{p+1}$ such that*

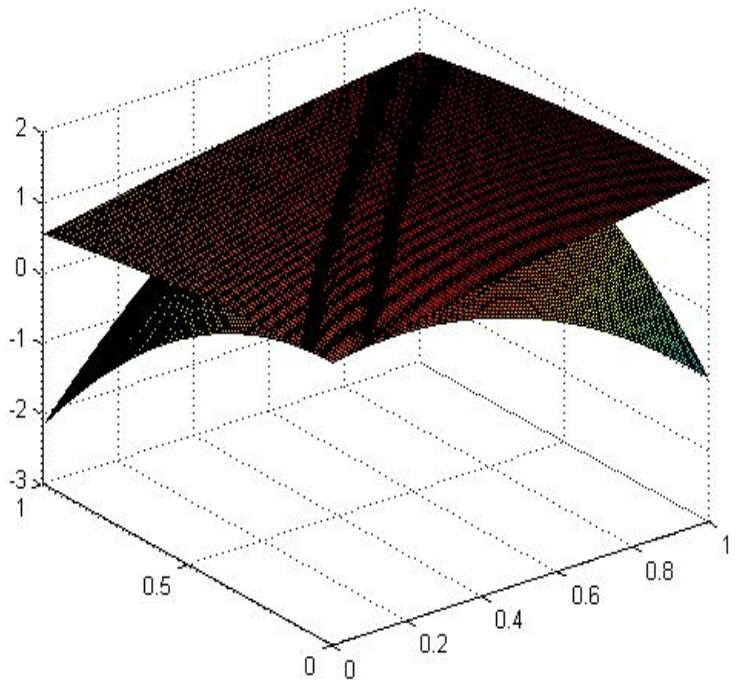
$$\sum_{i=0}^{p} \ell_i(x)\phi(y^i) \ = \ \phi(x) \qquad \text{with} \qquad \|\ell(x)\|_\infty \ \leq \ \Lambda,$$

   or, equivalently,

3. *replacing any point in Y by any $x \in B$ can increase the volume of the simplex of vertices $\{\phi(y^0), \phi(y^1), \ldots, \phi(y^p)\}$ at most by a factor Λ.*

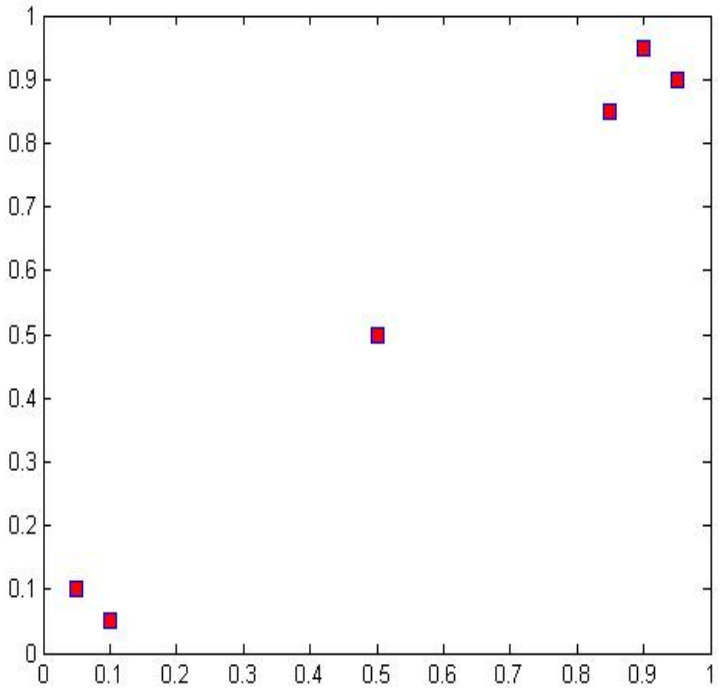
A badly poised set

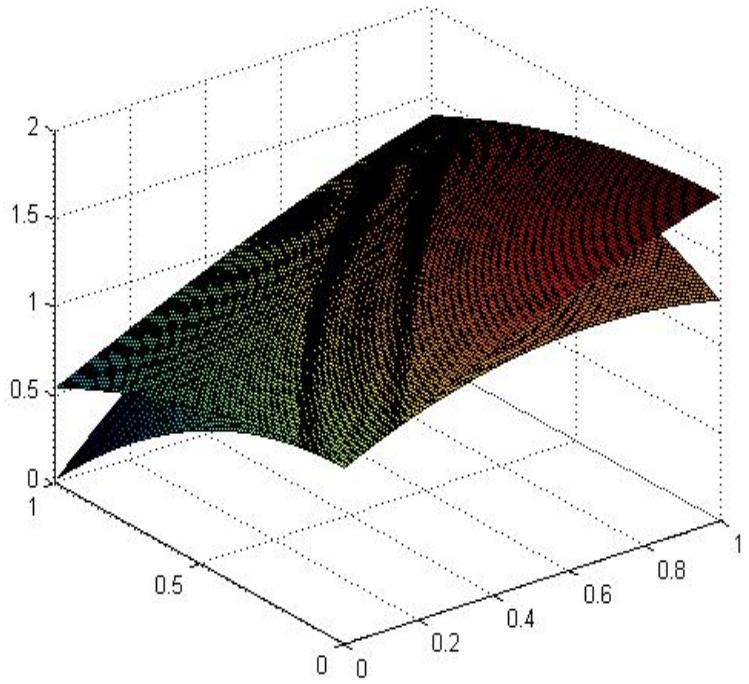




$$\Lambda = 5324.$$

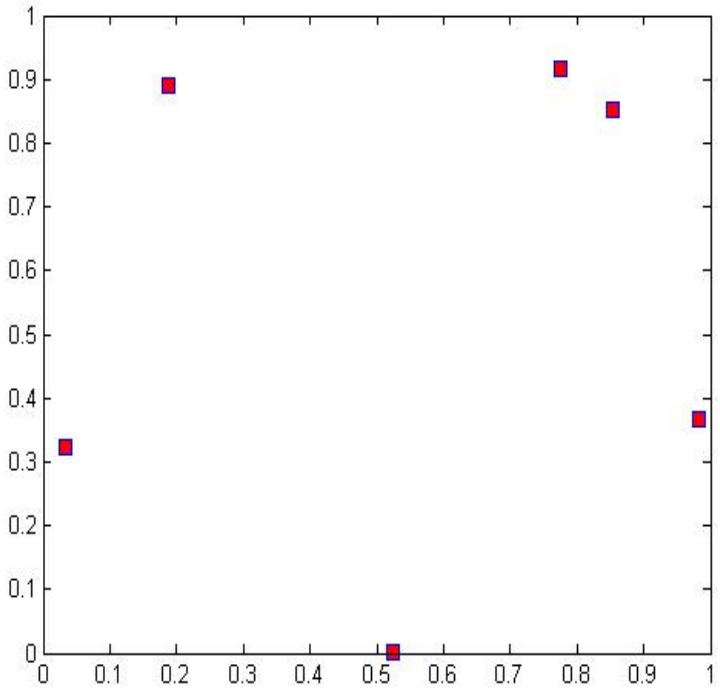
A not so badly poised set

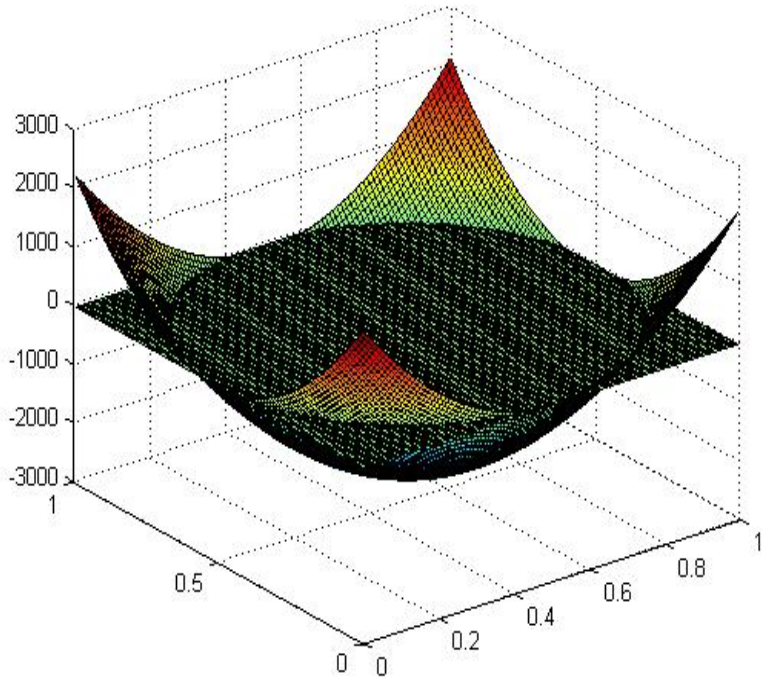




$$\Lambda = 294.$$

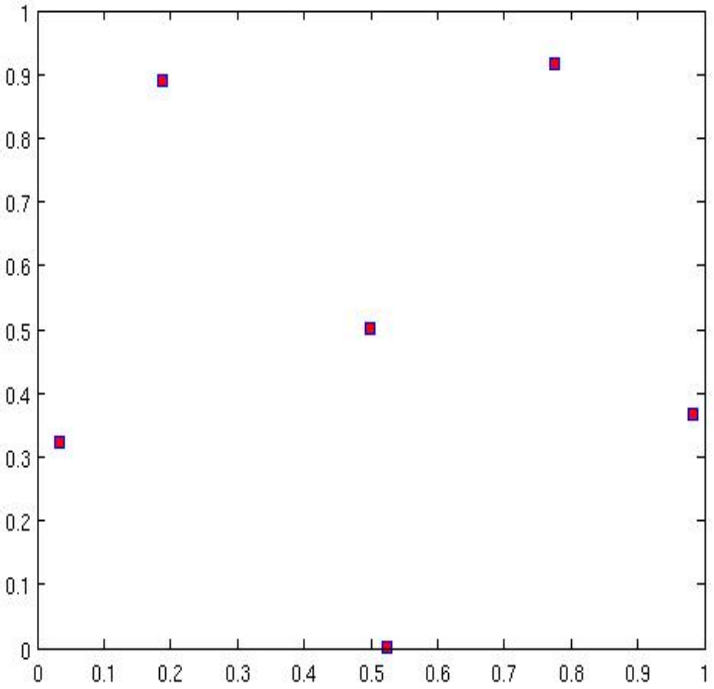
Another badly poised set

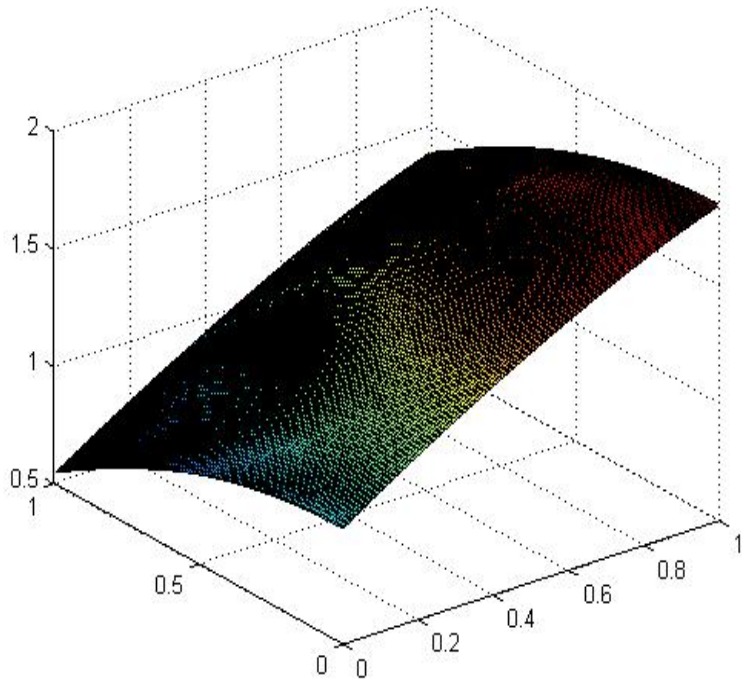




$$\Lambda = 492624.$$

An ideal set





$$\Lambda = 1.$$

Invariance of LPs

Lagrange polynomials and therefore the poisedness constant Λ do not depend on the scaling of the sample set.

Lagrange polynomials and therefore the poisedness constant Λ are invariant with respect to a shift of coordinates.

Shifting and scaling

Given any sample set written as

$$Y = \{y^0, y^1, \ldots, y^p\},$$

one can shift it by $-y^0$ to center the new set at the origin:

$$\{0, y^1 - y^0, \ldots, y^p - y^0\}.$$

Then, one can consider

$$\Delta = \Delta(Y) = \max_{1 \le i \le p} \|y^i - y^0\|$$

and scale the set by Δ:

$$\{0, \hat{y}^1, \ldots, \hat{y}^p\} = \{0, (y^1 - y^0)/\Delta, \ldots, (y^p - y^0)/\Delta\} \subset B(0; 1).$$

The resulting sample set Y_{scaled} is contained in $B(0; 1)$ and has at least one point on the ball boundary.

The algorithms which factorize $M(\phi, Y)$ to achieve well poisedness work on the shifted, scaled sets.

Connecting the two notions of well-poisedness

Note that by the invariance properties of LPs, the set Y is Λ–poised in the $B(y^0; \Delta)$ if and only if the set Y_{scaled} is Λ–poised in the unit ball $B(0; 1)$.

And then we also have:

Theorem

If $M(\bar{\phi}, Y_{scaled})$ is nonsingular and $\|M(\bar{\phi}, Y_{scaled})^{-1}\| \leq \Lambda$, then the set Y_{scaled} is $\sqrt{p+1}\Lambda$–poised in the unit ball $B(0; 1)$.

Conversely, if the set Y_{scaled} is Λ–poised in the unit ball $B(0; 1)$, then

$$\|M(\bar{\phi}, Y_{scaled})^{-1}\| \leq \theta(p+1)^{\frac{1}{2}}\Lambda,$$

where $\theta > 0$ is dependent on n and d but independent of Y_{scaled} and Λ.

Error bounds for quadratic interpolation

Assumption

We assume that $Y = \{y^0, y^1, \ldots, y^p\} \subset \mathbb{R}^n$, *with* $p + 1 = (n + 1)(n + 2)/2$, *is a poised set of sample points (in the quadratic interpolation sense,* $d = 2$*) contained in the ball* $B(y^0; \Delta(Y))$.

Further, we assume that the function f *is twice continuously differentiable in an open domain* Ω *containing* $B(y^0; \Delta)$ *and* $\nabla^2 f$ *is Lipschitz continuous in* Ω *with constant* $\nu_2 > 0$.

Error bounds for quadratic interpolation

Theorem

For all points y in $B(y^0; \Delta(Y))$, we have that

- *the error between the Hessian of the quadratic interpolation model and the Hessian of the function satisfies*

$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \leq \kappa_{eh}\, \Delta,$$

- *the error between the gradient of the quadratic interpolation model and the gradient of the function satisfies*

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\, \Delta^2,$$

- *the error between the quadratic interpolation model and the function satisfies*

$$|f(y) - m(y)| \leq \kappa_{ef}\, \Delta^3,$$

where ...

Theorem

... where κ_{eh}, κ_{eg}, and κ_{ef} are given by:

$$\begin{aligned}
\kappa_{eh} &= 3\sqrt{2}p^{\frac{1}{2}}\nu_2\|Q_{scaled}^{-1}\|/2, \\
\kappa_{eg} &= 3(1+\sqrt{2})p^{\frac{1}{2}}\nu_2\|Q_{scaled}^{-1}\|/2, \\
\kappa_{ef} &= (6+9\sqrt{2})p^{\frac{1}{2}}\nu_2\|Q_{scaled}^{-1}\|/4 + \nu_2/6.
\end{aligned}$$

Q_{scaled} is matrix obtained from $M(\bar{\phi}, Y_{scaled})$ by deleting the first row and column.

Model improvement (Lagrange polynomials)

Choose $\Lambda > 1$. Let Y be a poised set.

Each iteration of a model-improvement algorithm consists of:

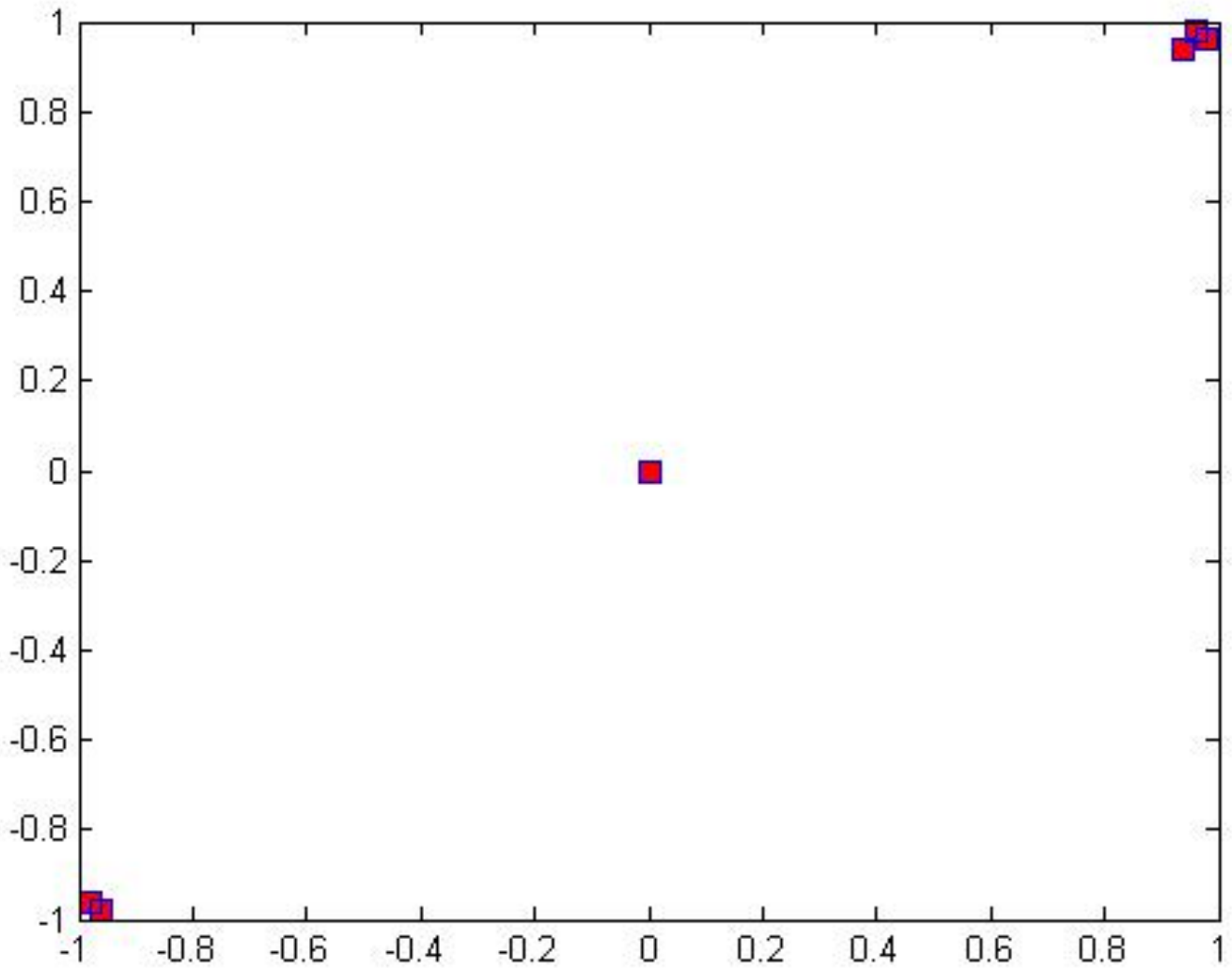
- Estimate
$$C \;=\; \max_{y \in Y} \max_{z \in B} |\ell_y(z)|.$$
- If $C > \Lambda$ then let y^{out} correspond to the polynomial where the maximum was attained. Let
$$y^{in} \;\in\; \operatorname{argmax}_{z \in B} |\ell_{y^{out}}(z)|.$$
Update Y (and the Lagrange polynomials):
$$Y \;\leftarrow\; Y \cup \{y^{in}\} \setminus \{y^{out}\}.$$
- Otherwise (i.e., $C \leq \Lambda$), Y is Λ–poised and stop.

Model improvement (Lagrange polynomials)

Theorem

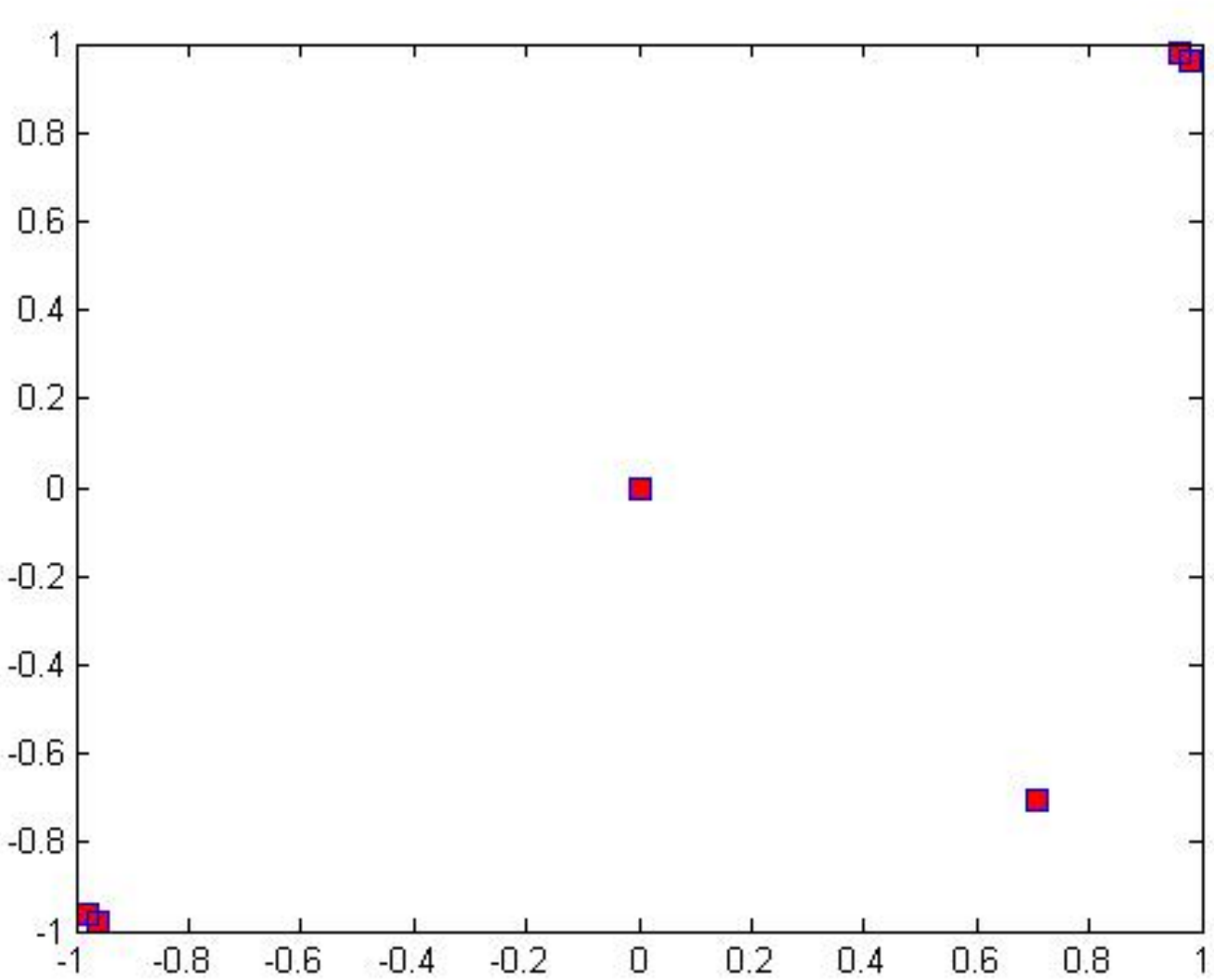
For any given $\Lambda > 1$ and a closed ball B, the previous model-improvement algorithm terminates with a Λ–poised set Y after at most $N = N(\Lambda)$ iterations.

Example (model improvement based on LP)



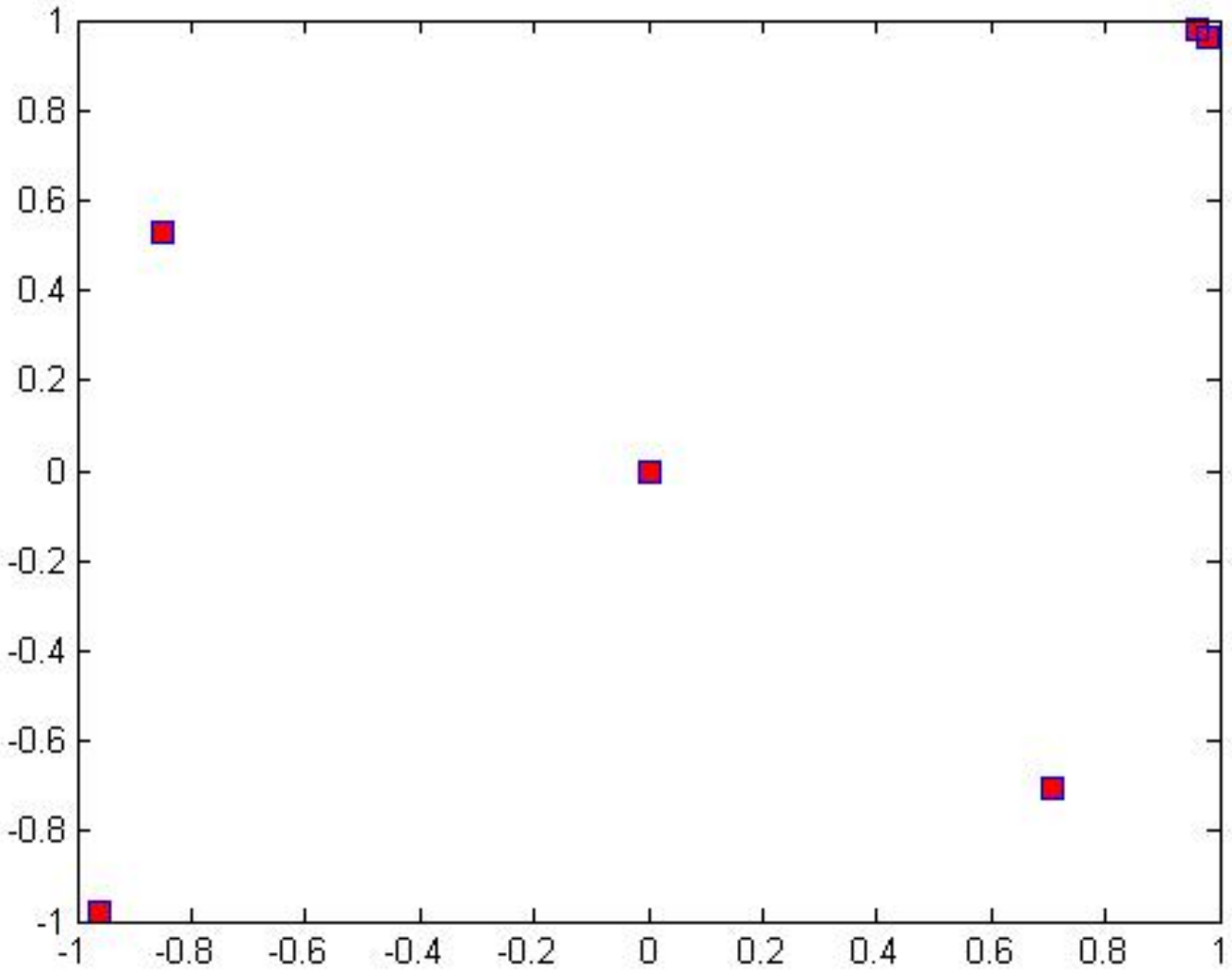
$C = 5324$

Example (model improvement based on LP)



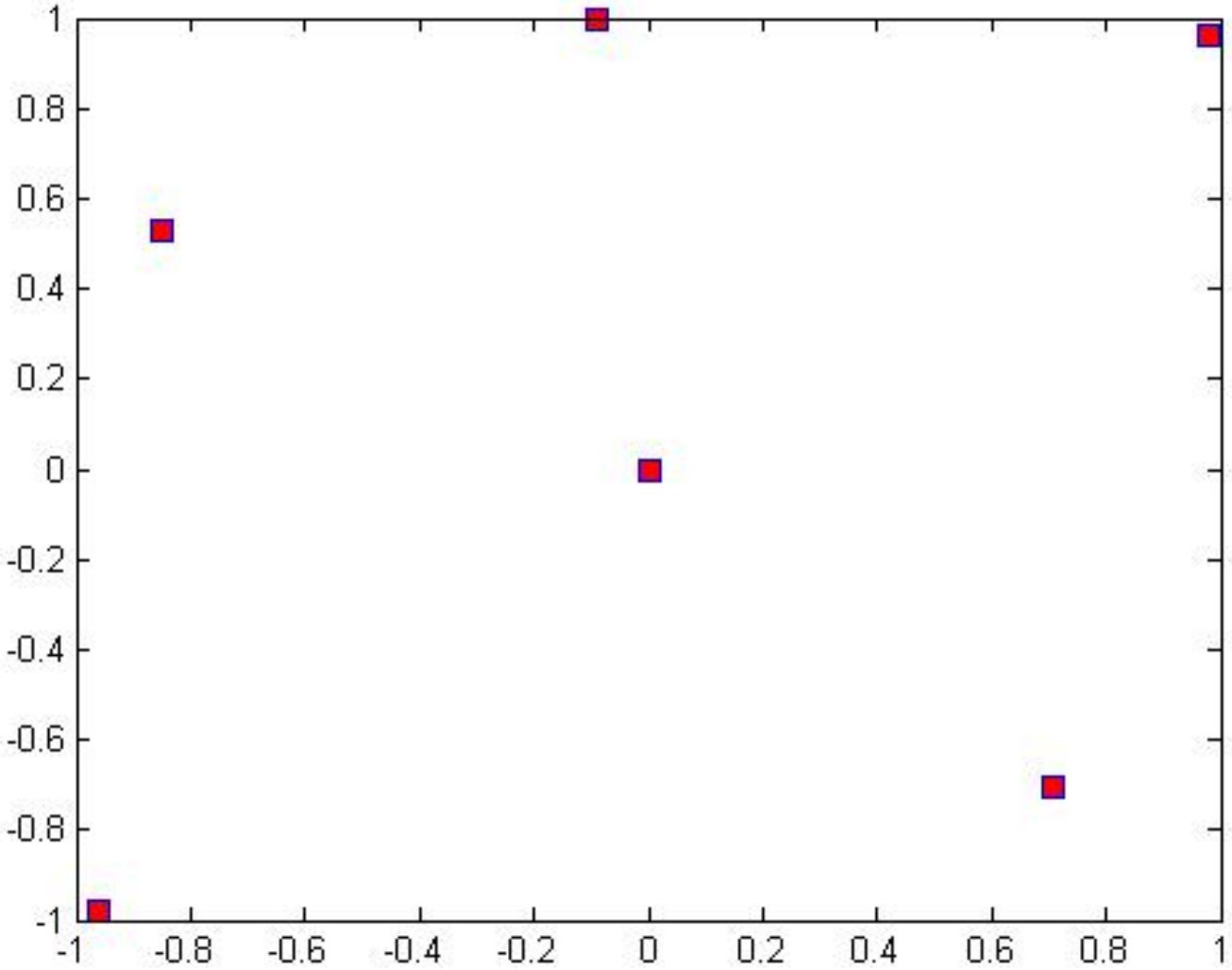
$C = 36.88$

Example (model improvement based on LP)



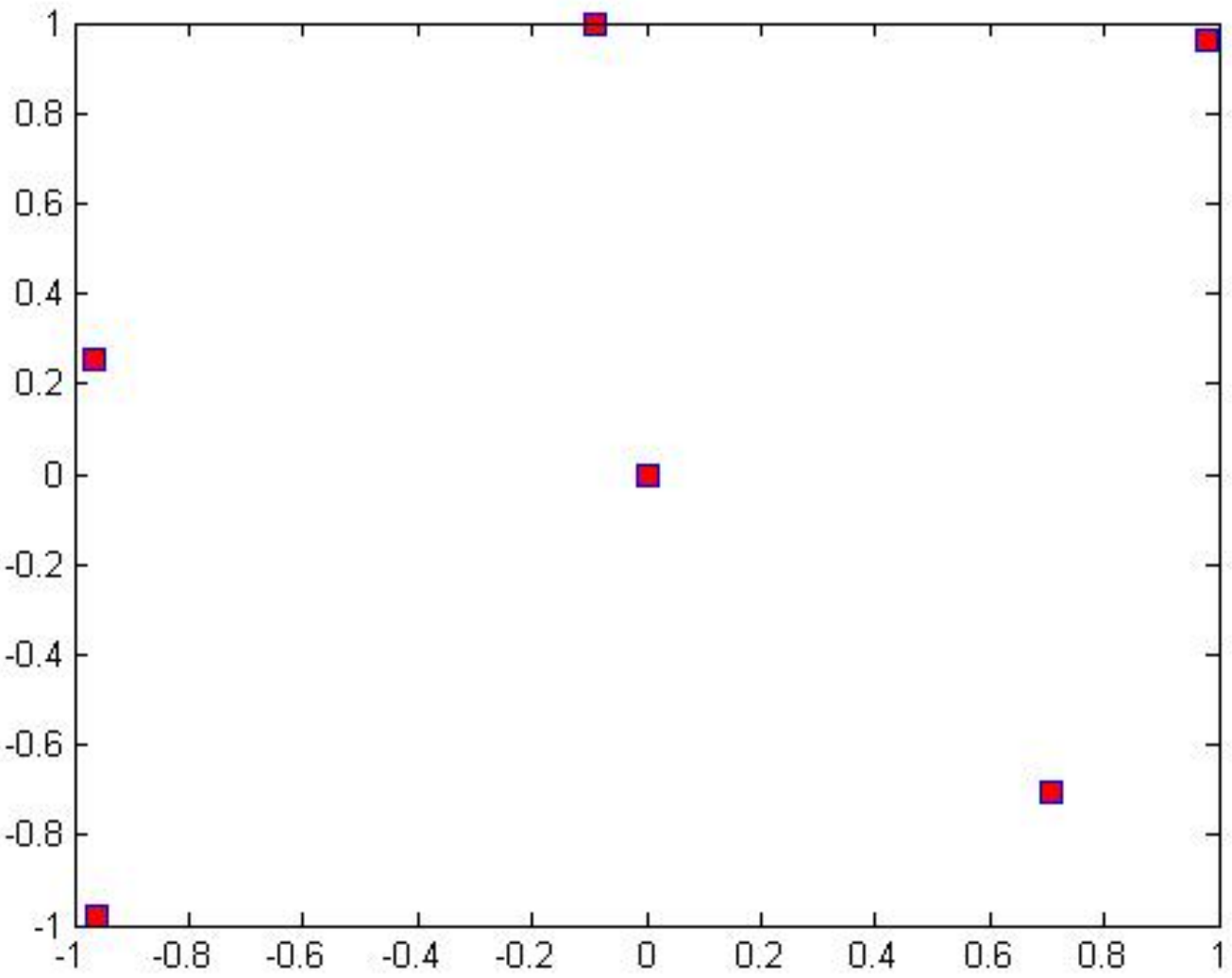
$C = 15.66$

Example (model improvement based on LP)



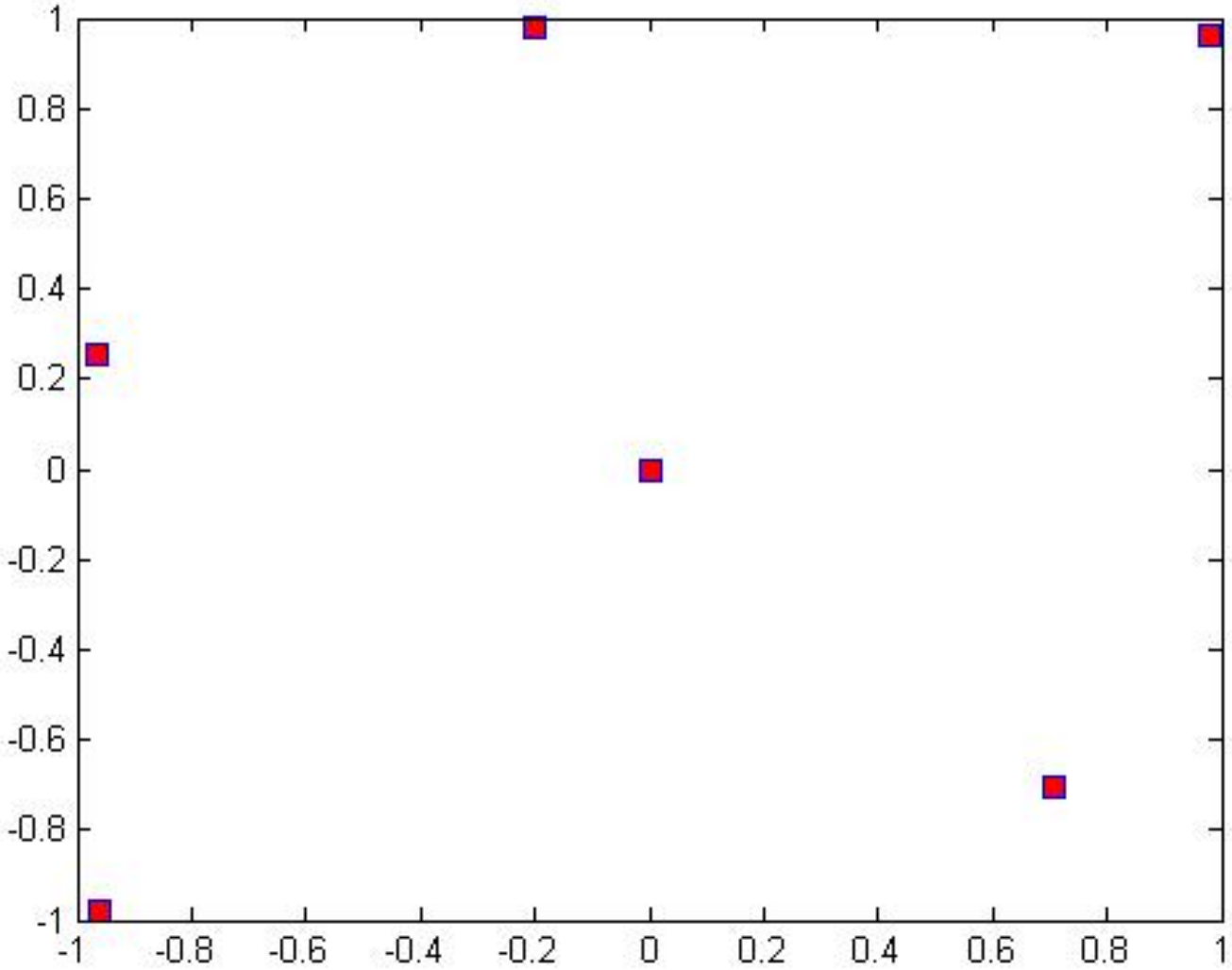
$C = 1.11$

Example (model improvement based on LP)



$C = 1.01$

Example (model improvement based on LP)



$C = 1.001$

Model improvement (condition number)

The well-poisedness/geometry constant can also be given by

$$\|M(\bar{\phi}, Y_{scaled})^{-1}\|$$

$\longrightarrow$ leading to model-improvement algorithms based on pivotal factorizations (LU/QR or Newton polynomials).

These algorithms yield:

$$\|M(\bar{\phi}, Y_{scaled})^{-1}\| \leq C_n \frac{\varepsilon_{growth}}{\xi}$$

where

- ε_{growth} is the growth factor of the factorization.
- $\xi > 0$ is a (imposed) lower bound on the absolute value of the pivots.

  $\longrightarrow$ one knows that $\xi \leq 1/4$ for quadratics and $\xi \leq 1$ for linears.

Recapitulation

The error bounds for linear interpolation or regression or quadratic interpolation obey:

$$\|\nabla f(y) - \nabla m(y)\| \leq [C_n C_f \Lambda] \Delta \qquad \forall y \in B(x; \Delta)$$

where $Y \subset B(x; \Delta)$ and

- C_n is a small constant depending on n.
- C_f measures the smoothness of f (in this case the Lipschitz constant of ∇f).
- and...

Recapitulation

- Λ is a Λ–poisedness constant related to the geometry of Y.

The original definition of Λ–poisedness says that the maximum absolute value of the Lagrange polynomials in $B(x;\Delta)$ is bounded by Λ.

An equivalent characterization of Λ–poisedness is

$$\|M(\bar{\phi}, Y_{scaled})^{-1}\| \leq \Lambda,$$

with Y_{scaled} obtained from Y such that $Y_{scaled} \subset B(0;1)$.

Presentation outline

Underdetermined interpolation

We now consider the case when the number $p+1$ of interpolation points in Y is smaller than the number $q+1$ of elements in the polynomial basis ϕ.

In this case, the matrix $M(\phi, Y)$ defining the interpolating conditions has more columns than rows and the interpolation polynomials defined are no longer unique.

This situation is extremely frequent in model-based DFO methods. Most of the derivative-free applications are based on objective functions that are costly to compute.

A simple way to conserve the cost of building a sample set is to use linear models, however, it is well known that convergence slows down significantly when no curvature is exploited.

Sub-basis sense

The simplest approach is to remove, from the system, the last $q - p$ columns of $M(\phi, Y)$.

This causes the last $q - p$ elements of the solution α to be zero.

Such an approach approximates some elements of α, while it sets others to zero based solely on the order of the elements in the basis ϕ.

Clearly this approach is not very desirable without any knowledge of, for instance, the sparsity structure of the gradient and the Hessian of the function f.

There is also a more fundamental drawback: the first $p + 1$ columns of $M(\phi, Y)$ may be linearly dependent.

Sub-bases example

Let $\phi = \{1, x_1, x_2, \frac{1}{2}x_1^2, x_1x_2, \frac{1}{2}x_2^2\}$, $Y = \{y^0, y^1, y^2, y^3\}$, and $y^0 = (0, 0)$, $y^1 = (0, 1)$, $y^2 = (0, -1)$, and $y^3 = (1, 0)$.

The matrix $M(\phi, Y)$ is given by

$$M(\phi, Y) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0.5 \\ 1 & 0 & -1 & 0 & 0 & 0.5 \\ 1 & 1 & 0 & 0.5 & 0 & 0 \end{bmatrix}.$$

If we select the first four columns of $M(\phi, Y)$ then the system is still not well defined, since the matrix is singular.

Hence the set Y is not poised with respect to the sub-basis $\tilde{\phi} = \{1, x_1, x_2, \frac{1}{2}x_1^2\}$.

Sub-bases example

If another sub-basis was selected, for instance, $\tilde{\phi} = \{1, x_1, x_2, \frac{1}{2}x_2^2\}$, then the set Y is well poised and the matrix consisting of the first, the second, the third, and the sixth columns of $M(\phi, Y)$ is well conditioned and a unique solution exists.

If the Hessian of f happens to look like

$$\begin{bmatrix} 0 & 0 \\ 0 & \frac{\partial^2 f}{\partial x_2^2}(x) \end{bmatrix}$$

then this reduced system actually produces the complete quadratic model of f.

Best sub-basis sense

If the sparsity structure of the derivatives of f is known in advance then this can be exploited by deleting appropriate columns from the system.

A more sophisticated version of this idea is exploited for group partial separable functions and works well in practice when there is a known structure of the Hessian and gradient elements.

If no such structure is known, then there is no reason to select one set of columns over another except for geometry considerations.

Hence it makes sense to select those columns that produce the best geometry (selecting the sub-basis $\tilde{\phi}$ so that the poisedness constant Λ is minimized).

Best sub-basis sense (disadvantages)

Let us consider the purely linear case in $\mathbb{R}^3$ for simplicity. An example for a quadratic case can be constructed in a similar manner.

Consider $\phi = \{1, x_1, x_2, x_3\}$ and $Y = \{y^0, y^1, y^2\}$, where, as usual, $y^0 = (0, 0, 0)$, and where $y^1 = (1, 0, 0)$, and $y^2 = (0, 1, 1 - \epsilon)$. Assume $f(Y) = (0, b_1, b_2)$. The system then becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1-\epsilon \end{bmatrix} \alpha = \begin{bmatrix} 0 \\ b_1 \\ b_2 \end{bmatrix}.$$

The best sub-basis for Y is then $\tilde{\phi} = \{1, x_1, x_2\}$. If we select the appropriate columns of $M(\phi, Y)$ and solve the reduced system, we obtain the following solution for the coefficients of $m(x)$

$$\alpha = \begin{bmatrix} 0 \\ b_1 \\ b_2 \\ 0 \end{bmatrix}.$$

Best sub-basis sense (disadvantages)

Now, if we consider $y^2 = (0, 1 - \epsilon, 1)$, then the best sub-basis is $\tilde{\phi} = \{1, x_1, x_3\}$ and the solution that we will find with this approach is

$$\alpha = \begin{bmatrix} 0 \\ b_1 \\ 0 \\ b_2 \end{bmatrix}.$$

Best sub-basis sense (disadvantages)

The two possible solutions are very different from each other, yet as ϵ goes to zero the two sets of sample points converge pointwise to each other.

Hence, we see that the sub-basis approach suffers from a lack of robustness with respect to small perturbations in the sample set.

We also notice that in the first (second) case the fourth (third) element of the coefficient vector is set to zero and the third (fourth) element is set to b_2 (b_2).

Hence, each solution is biased towards one of the basis components (x_2 or x_3) without using any actual information about the structure of f.

Minimum-norm sense (example)

A more suitable approach would be to treat all such components equally in some sense. This can be achieved by the minimum-norm solution. In the first case:

$$\alpha^{mn} = M(\phi, Y)^\top [M(\phi, Y) M(\phi, Y)^\top]^{-1} f(Y) = \begin{bmatrix} 0 \\ b_1 \\ \frac{b_2}{2-2\epsilon+\epsilon^2} \\ \frac{(1-\epsilon)b_2}{2-2\epsilon+\epsilon^2} \end{bmatrix}$$

and in the second case is

$$\alpha^{mn} = \begin{bmatrix} 0 \\ b_1 \\ \frac{(1-\epsilon)b_2}{2-2\epsilon+\epsilon^2} \\ \frac{b_2}{2-2\epsilon+\epsilon^2} \end{bmatrix}.$$

Minimum-norm sense (example)

These two solutions converge to $(0, b_1, b_2/2, b_2/2)$ as ϵ converges to zero.

Hence, not only is the minimum-norm solution robust with respect to small perturbations of the data, but also it evens out the elements of the gradient over the x_2 and x_3 basis components.

Minimum-norm sense (disadvantages)

For the reasons described above it is beneficial to consider the minimum-norm solution of the system. The minimum-norm solution is expressed as

$$M(\phi, Y)^\top [M(\phi, Y) M(\phi, Y)^\top]^{-1} f(Y),$$

and can be computed via the QR factorization or the reduced singular value decomposition of $M(\phi, Y)$.

It is well know that a minimum-norm solution of an underdetermined system of linear equations is not invariant under linear transformations.

In our case, this fact means that the minimum-norm solution and corresponding polynomial depend on the choice of ϕ.

Underdetermined polynomial models

Consider a underdetermined quadratic polynomial model

$$m(y) \;=\; c + g^\top y + \frac{1}{2} y^\top H y$$

built with less than $(n+1)(n+2)/2$ points.

Assumption

We assume that $Y = \{y^0, y^1, \ldots, y^p\} \subset \mathbb{R}^n$ is a set of sample points poised in the linear interpolation sense (or in the linear regression sense if $p+1 > n+1$) contained in the ball $B(y^0; \Delta(Y))$.

Further, we assume that the function f is continuously differentiable in an open domain Ω containing $B(y^0; \Delta)$ and ∇f is Lipschitz continuous in Ω with constant $\nu > 0$.

Error bounds for underdetermined interpolation

Theorem

If Y is Λ_L–poised for linear interpolation or regression then

$$\|\nabla f(y) - \nabla m(y)\| \;\leq\; \Lambda_L \left[C_f + \|H\|\right] \Delta \qquad \forall y \in B(x;\Delta).$$

$\longrightarrow$ One has $\|M(\phi_L, Y_{scaled})^\dagger\| \leq \Lambda_L$ ('equivalent' to the notion of linear Λ_L–poisedness), where

$$M(\phi, Y) \;=\; \left[\; M(\phi_L, Y) \quad M(\phi_Q, Y) \;\right].$$

Minimum Frobenius norm models

Again,

$$\|\nabla f(y) - \nabla m(y)\| \leq \Lambda_L \left[C_f + \|H\|\right] \Delta \qquad \forall y \in B(x; \Delta).$$

Q: What should we do?

A: One should build models by minimizing the norm of H.

Minimum Frobenius norm models

It will be useful to explore the partition in linear and quadratic terms

$$M(\phi, Y) \;=\; \left[\; M(\phi_L, Y) \quad M(\phi_Q, Y) \;\right].$$

corresponding in 2D to

$$\phi_L = \{1, x_1, x_2\} \quad \text{and} \quad \phi_Q = \{x_1^2/2, x_2^2/2, x_1 x_2\}.$$

Minimum Frobenius norm models

Recall the sample set $Y = \{y^0, y^1, \ldots, y^p\}$ and the quadratic model

$$m(y) \;=\; c + g^\top y + \frac{1}{2} y^\top H y \;=\; \alpha_L^\top \phi_L(x) + \alpha_Q^\top \phi_Q(x).$$

The models can be built by minimizing the entries of the Hessian (in the Frobenius norm) subject to the interpolation conditions:

$$\begin{aligned} \min \quad & \tfrac{1}{4}\|H\|_F^2 \\ \text{s.t.} \quad & c + g^\top(y^i) + \tfrac{1}{2}(y^i)^\top H(y^i) \;=\; f(y^i), \quad i = 0, \ldots, p, \end{aligned}$$

or, 'equivalently',

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|\alpha_Q\|^2 \\ \text{s.t.} \quad & M(\phi, Y)\alpha \;=\; f(Y). \end{aligned}$$

Minimum Frobenius norm models

The solution of this QP problem requires a linear solve with:

$$F(\phi, Y) \;=\; \begin{bmatrix} M(\phi_Q, Y)M(\phi_Q, Y)^\top & M(\phi_L, Y) \\ M(\phi_L, Y)^\top & 0 \end{bmatrix},$$

where

$$M(\phi, Y) \;=\; \begin{bmatrix} M(\phi_L, Y) & M(\phi_Q, Y) \end{bmatrix}.$$

Minimum Frobenius norm sense

It is possible to define minimum Frobenius norm Lagrange polynomials (replacing f by the various indicator functions in the QP).

Then (AS IN THE DETERMINED CASE):

- The MFN polynomial and the MFN Lagrange polynomials enjoy the same properties as before.
- Λ–poisedness has the same two equivalent redefinitions as before.
- Λ_F–poisedness (defined using MFN Lagrange polynomials) is equivalent to:

$$\|F(\bar{\phi}, Y_{scaled})^{-1}\| \leq \Lambda_F.$$

Minimum Frobenius norm models

Such minimum Frobenius norm models are used in the DFO code (implements an interpolation based trust-region method).

Theoretical support:

Theorem

If Y is Λ_F–poised in the minimum Frobenius norm sense then

$$\|H\| \leq C_n C_f \Lambda_F,$$

where H is, again, the Hessian of the model.

Minimum Frobenius norm models

Putting the two theorems together yield:

$$\|\nabla f(y) - \nabla m(y)\| \leq \Lambda_L \left[C_f + C_n C_f \Lambda_F\right] \Delta \quad \forall y \in B(x; \Delta).$$

$\longrightarrow$ MFN models are 'fully linear'.

$\longrightarrow$ The poisedness constant appears 'squared'...

Minimum Frobenius norm models (least updating)

These models can be built, alternatively, by minimizing the difference between the current and previous Hessians (in the Frobenius norm):

$$\begin{aligned} \min \quad & \tfrac{1}{4}\|H - H^{old}\|_F^2 \\ \text{s.t.} \quad & c + g^\top(y^i) + \tfrac{1}{2}(y^i)^\top H(y^i) \;=\; f(y^i), \quad i = 0, \ldots, p. \end{aligned}$$

$\longrightarrow$ Used in `NEWUOA` interpolation based trust-region solver.

Theoretical support:

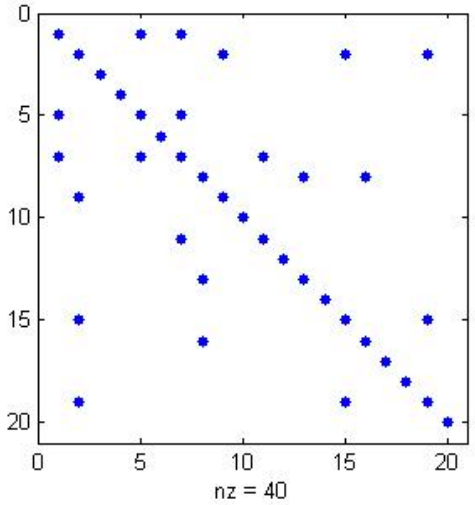
Theorem

If f is itself a quadratic then:

$$\|H - \nabla^2 f\| \;\leq\; \|H^{old} - \nabla^2 f\|.$$

Sparsity on the Hessian

- In many problems, pairs of variables have no 'correlation', leading to zero second order partial derivatives in f:



- Thus, the Hessian $H = \nabla^2 m$ of the model should be sparse,

  i.e., the vector α_Q in the basis ϕ should be sparse.

A relevant question

Question

Is it possible to build fully quadratic models by quadratic underdetermined interpolation (i.e., using less than $\mathcal{O}(n^2)$ points) in the SPARSE case?

An answer will be given by building the models using instead the ℓ_1-norm and relaxing the interpolating conditions for noisy recovery

$$\begin{aligned} \min \quad & \|\alpha_Q\|_1 \\ \text{s.t.} \quad & \|M(\psi, Y)\alpha - f(Y)\|_2 \ \leq \ \eta \end{aligned}$$

with ψ orthonormal in some sense (and not very different from the natural basis $\bar{\phi}$).

Sparse Hessian recovery

Using compressed sensing / sparse recovery results (Rauhut 2010):

Theorem

If

- *the Hessian of* f *at* x *has at most* h *non-zeros.*
- Y *is a random sample set chosen w.r.t. the uniform measure on* $B_\infty(x;\Delta)$.
- $\frac{p}{\log p} \geq 9c\,(h+n+1)\log^2(h+n+1)\log \mathcal{O}(n^2)$.

Then, with high probability, the quadratic

$$q^* = \sum \alpha_\iota^* \psi_\iota$$

obtained by solving the noisy and partial ℓ_1*-minimization problem is a fully quadratic model for* f *(with error constants not depending on* x, Δ*).*

An answer to the question

- For instance, when the number of non-zeros of the Hessian is $h = \mathcal{O}(n)$, we are able to construct fully quadratic models with

$$\mathcal{O}(n \log^4 n) \quad \text{points.}$$

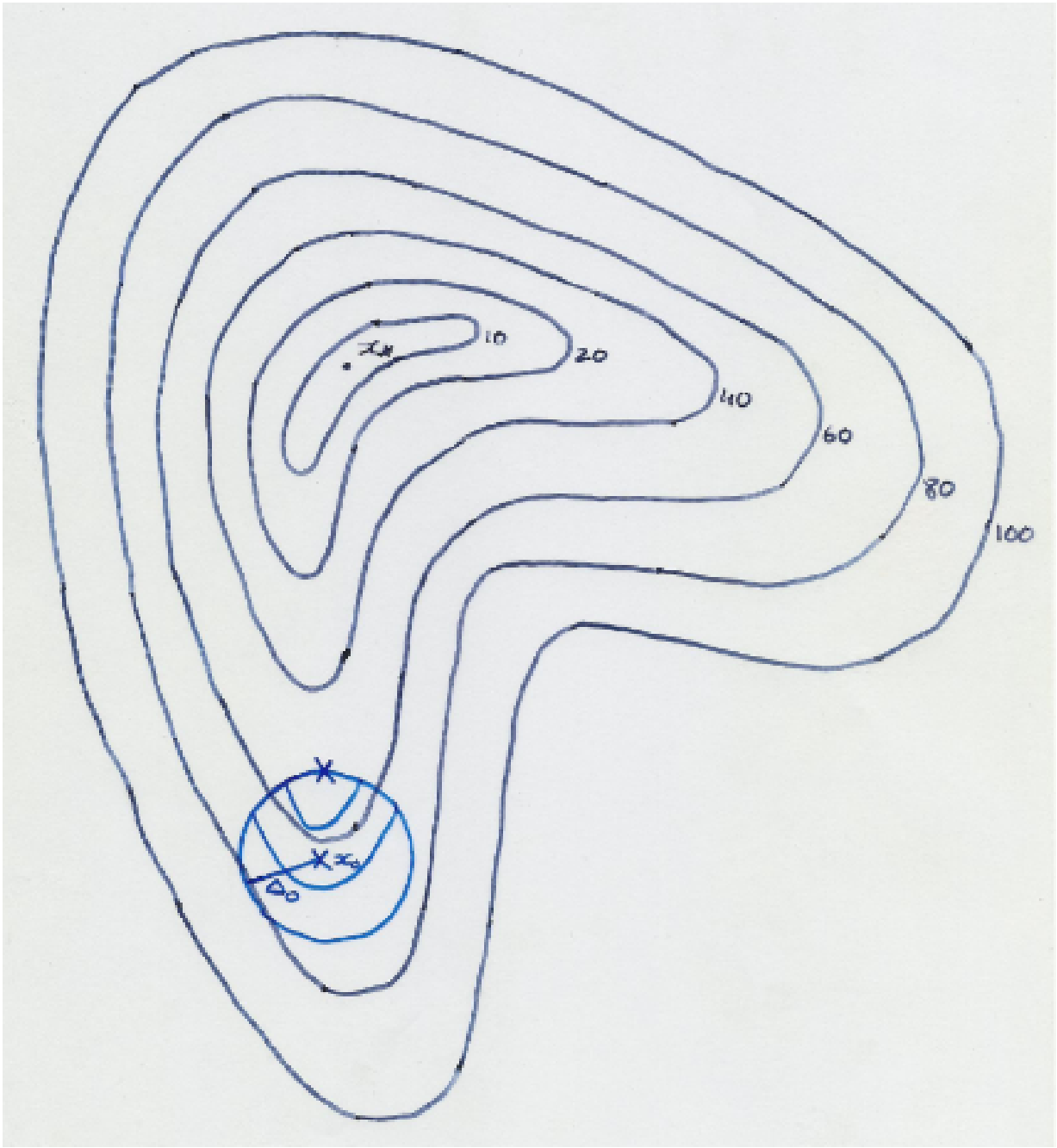
- Also, we recover both the function and its sparsity structure.

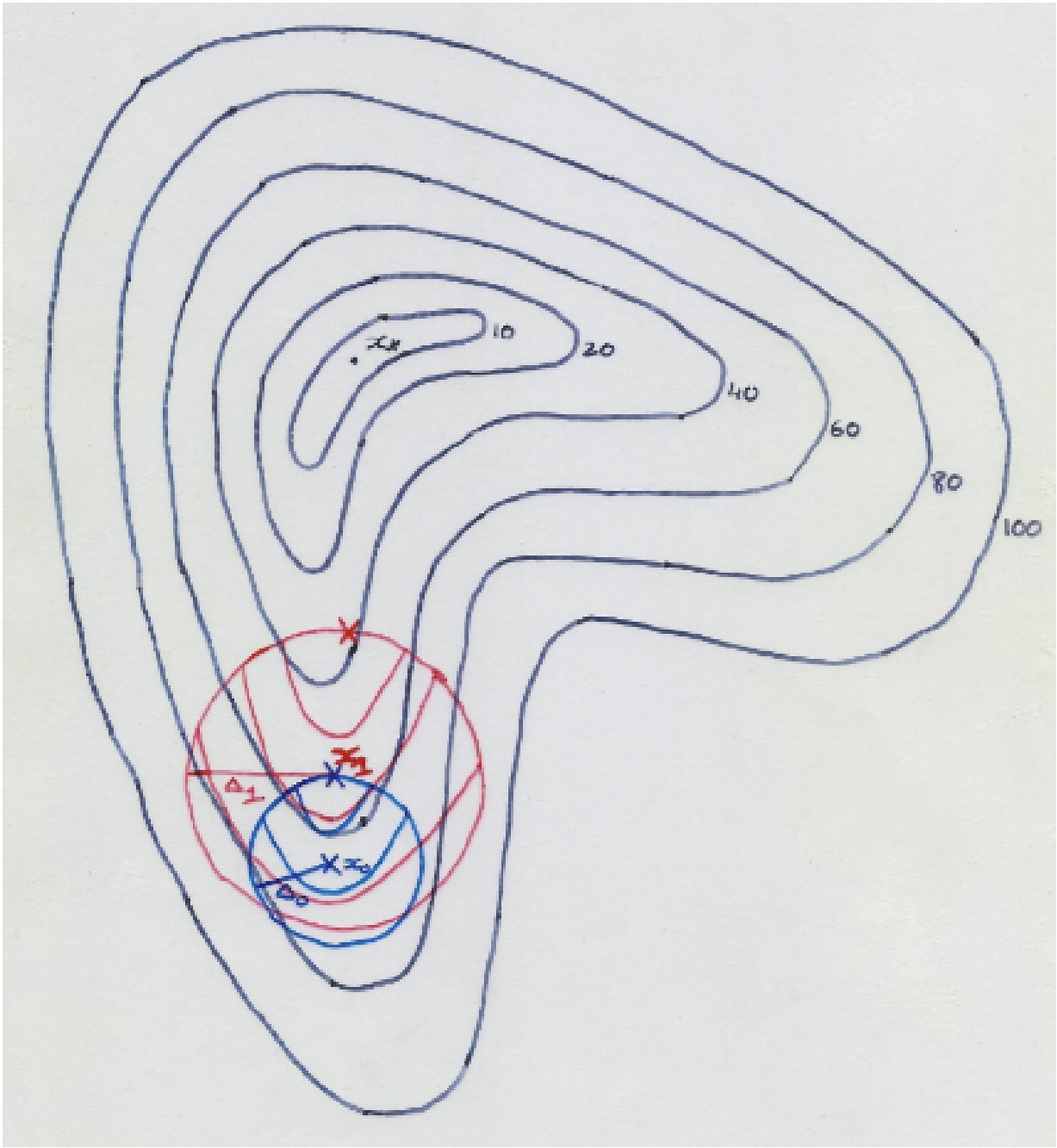
Reference:

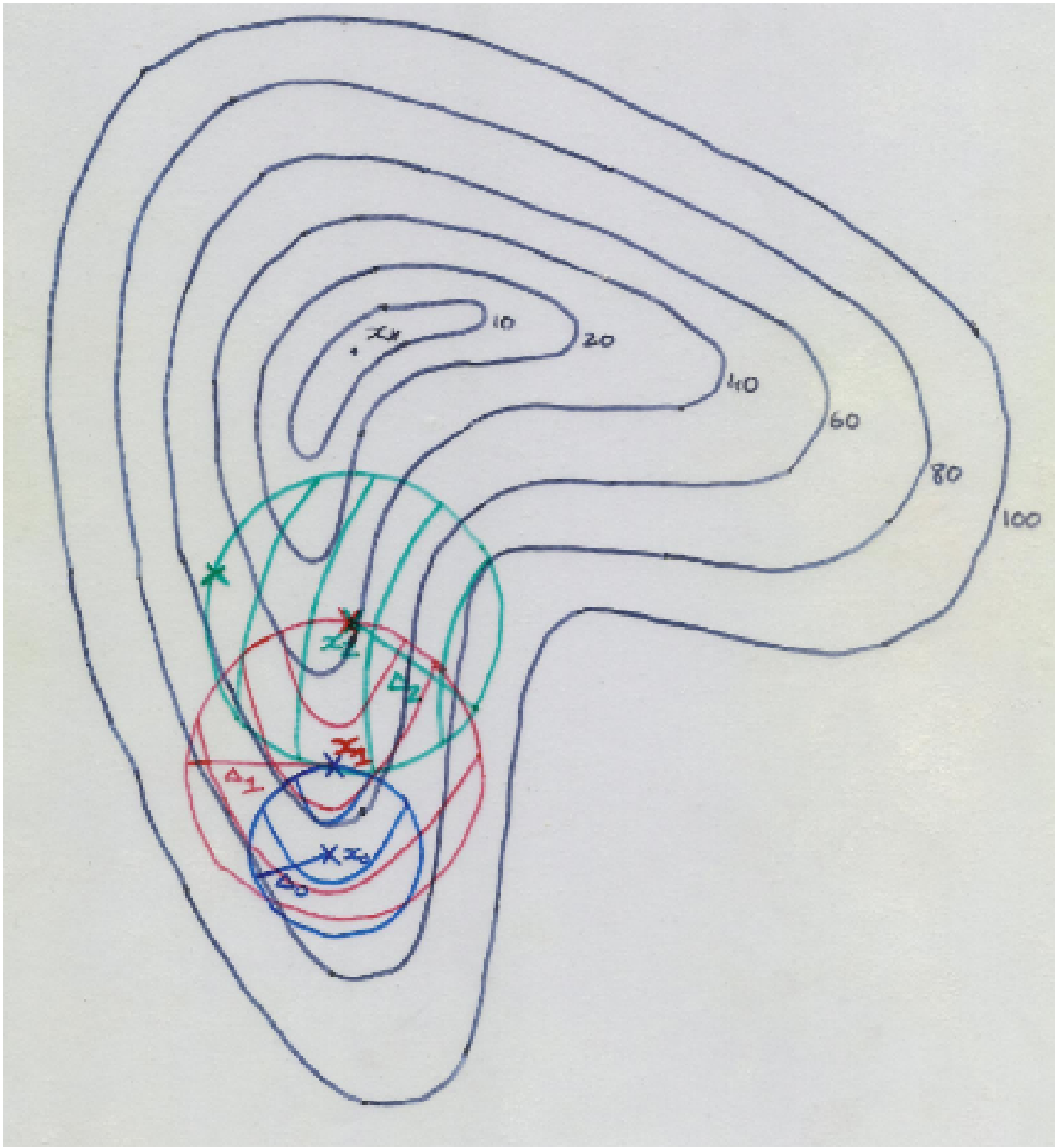
- A. S. Bandeira, K. Scheinberg, and L. N. Vicente, Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization, Math. Program., 134 (2012) 223–257.

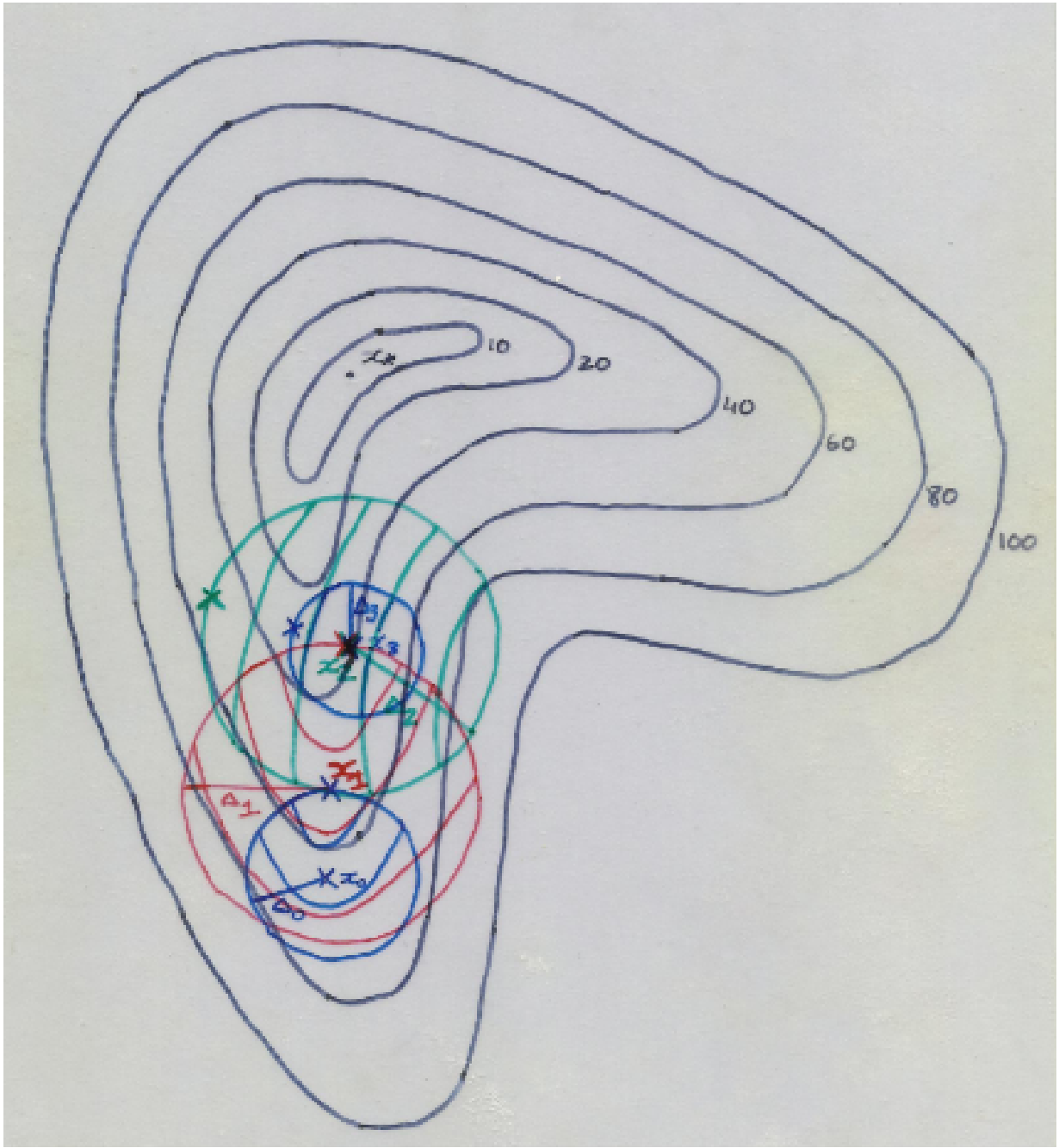
Presentation outline


x_k
10
20
40
60
80
100


10
20
40
60
80
100


10
20
40
60
80
100
x_k
x_{k+1}
x_{k+2}
Δ_1
Δ_2


10
20
40
60
80
100
x_4
x
x_2
x_3
x_1
Δ_2
Δ_1
Δ_0


10
20
40
60
80
100

Model-based trust-region methods

Trust-region methods for DFO typically:

- Attempt to form quadratic models (by interpolation/regression and using polynomials or radial basis functions)

$$m_k(x_k + s) \;=\; f_k + g_k^\top s + \frac{1}{2} s^\top H_k s$$

  based on (well poised) sample sets.

$\longrightarrow$ Well poisedness ensures fully linear or fully quadratic models.

Model-based trust-region methods

Trust-region methods for DFO typically:

- Attempt to form quadratic models (by interpolation/regression and using polynomials or radial basis functions)

$$m_k(x_k + s) \;=\; f_k + g_k^\top s + \frac{1}{2} s^\top H_k s$$

  based on (well poised) sample sets.

$\longrightarrow$ Well poisedness ensures fully linear or fully quadratic models.

- Calculate a step s_k by approximately solving the trust-region subproblem

$$\min_{s \in B_2(x_k;\Delta_k)} \quad m_k(x_k + s).$$

Model-based trust-region methods

- Set x_{k+1} to $x_k + s_k$ (success) or to x_k (unsuccess) and update Δ_k depending on the value of

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

Model-based trust-region methods

- Set x_{k+1} to $x_k + s_k$ (success) or to x_k (unsuccess) and update Δ_k depending on the value of

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

- Reduce Δ_k only if ρ_k is small and the model is FL/FQ — unsuccessful iterations.

Model-based trust-region methods

- Set x_{k+1} to $x_k + s_k$ (success) or to x_k (unsuccess) and update Δ_k depending on the value of

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

- Reduce Δ_k only if ρ_k is small and the model is FL/FQ — unsuccessful iterations.

- Allow for model-improving iterations (when ρ_k is not large enough and the model is not certifiably FL/FQ).

  $\longrightarrow$ Do not reduce Δ_k.

Model-based trust-region methods

- Accept new iterates based on simple decrease, i.e., if

$$\rho_k > 0 \quad \Longleftrightarrow \quad f(x_k + s_k) < f(x_k),$$

as long as the model is FL/FQ — acceptable iterations.

Model-based trust-region methods

- Accept new iterates based on simple decrease, i.e., if

$$\rho_k > 0 \quad \Longleftrightarrow \quad f(x_k + s_k) < f(x_k),$$

  as long as the model is FL/FQ — acceptable iterations.

- Incorporate a criticality step (1st or 2nd order) when the 'stationarity' of the model is small.

  $\longrightarrow$ Internal cycle of reductions of Δ_k — until model is well poised in $B(x_k; \|g_k\|)$.

  Scheinberg and Toint (2010) showed that a criticality step is indeed necessary (see later).

Behavior of the trust-region radius (DFO case)

Due to the criticality step, one has for successful iterations:

$$f(x_k) - f(x_{k+1}) \;\geq\; \mathcal{O}(\|g_k\| \min\{\|g_k\|, \Delta_k\}) \geq\; \mathcal{O}(\Delta_k^2).$$

Thus:

Theorem (Conn, Scheinberg, and Vicente, 2009)

The trust-region radius converges to zero:

$$\lim_{k \to +\infty} \Delta_k \;=\; 0.$$

$\longrightarrow$ Similar to direct-search methods where $\liminf_{k \to +\infty} \alpha_k = 0$.

Analysis of model-based TR methods (1st order)

Using fully linear models:

Theorem (Conn, Scheinberg, and Vicente, 2009)

If ∇f is Lips. continuous and f is bounded below on $L(x_0)$ then

$$\lim_{k \to +\infty} \|\nabla f(x_k)\| = 0.$$

$\longrightarrow$ Valid also for simple decrease (acceptable iterations).

$\longrightarrow$ Requires steps satisfying a fraction of Cauchy decrease.

Fraction of Cauchy decrease (FCD)

The Cauchy step s_k^C is the minimizer of $m_k(s)$, along $-g_k$, in $B(x_k; \Delta_k)$.

Definition

Let $\kappa_{fcd} \in (0, 1]$. The step s_k provides a fraction of Cauchy decrease (FCD) if

$$m_k(0) - m_k(s_k) \;\geq\; \kappa_{fcd} \left[m_k(0) - m_k(s_k^C) \right].$$

The optimal solution of the TRS satisfies the FCD trivially.

Analysis of model-based TR methods (2nd order)

Using fully quadratic models:

Theorem (Conn, Scheinberg, and Vicente, 2009)

If $\nabla^2 f$ is Lips. continuous and f is bounded below on $L(x_0)$ then

$$\lim_{k \to +\infty} \max \left\{ \|\nabla f(x_k)\|, -\lambda_{min}[\nabla^2 f(x_k)] \right\} = 0.$$

$\longrightarrow$ Valid also for simple decrease (acceptable iterations).

$\longrightarrow$ Requires steps satisfying a fraction of eigenstep decrease.

$\longrightarrow$ Going from $\liminf$ to $\lim$ requires changing the update of Δ_k.

Fraction of eigenstep decrease (FED)

Suppose the model has negative curvature.

The eigenstep s_k^{E} is the eigenvector of H_k corresponding to the most negative eigenvalue, so that $(s_k^{\mathsf{E}})^\top (g_k) \leq 0$ and $\|s_k^{\mathsf{E}}\| = \Delta_k$.

Definition

Let $\kappa_{fed} \in (0, 1]$. The step s_k provides a fraction of eigenstep decrease (FED) if

$$m_k(0) - m_k(s_k) \;\geq\; \kappa_{fed} \left[m_k(0) - m_k(s_k^E) \right].$$

The optimal solution of the TRS satisfies the FED trivially.

Sample set management

Recently, Fasano, Morales, and Nocedal (2009) suggested an one-point exchange:

- In successful iterations:

$$Y_{k+1} \;=\; Y_k \cup \{x_k + s_k\} \setminus \{y_{out}\}.$$

  where $y_{out} = \operatorname{argmax} \|y - x_k\|_2$.

- In the unsuccessful case:

$$Y_{k+1} \;=\; Y_k \cup \{x_k + s_k\} \setminus \{y_{out}\} \quad \text{if} \quad \|y_{out} - x_k\| \geq \|s_k\|.$$

- Do not perform model-improving iterations.

They observed sample sets not badly poised!

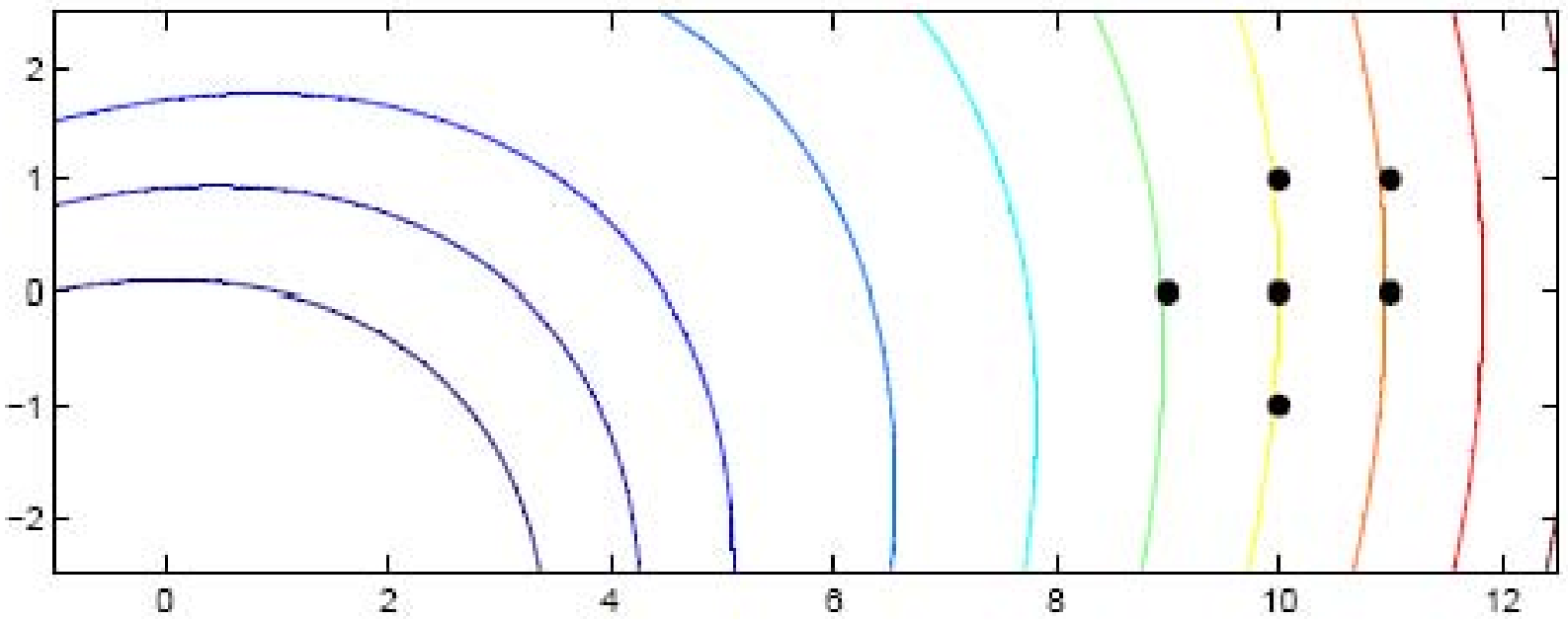
Self-correcting geometry

Later, Scheinberg and Toint (2009) proposed:

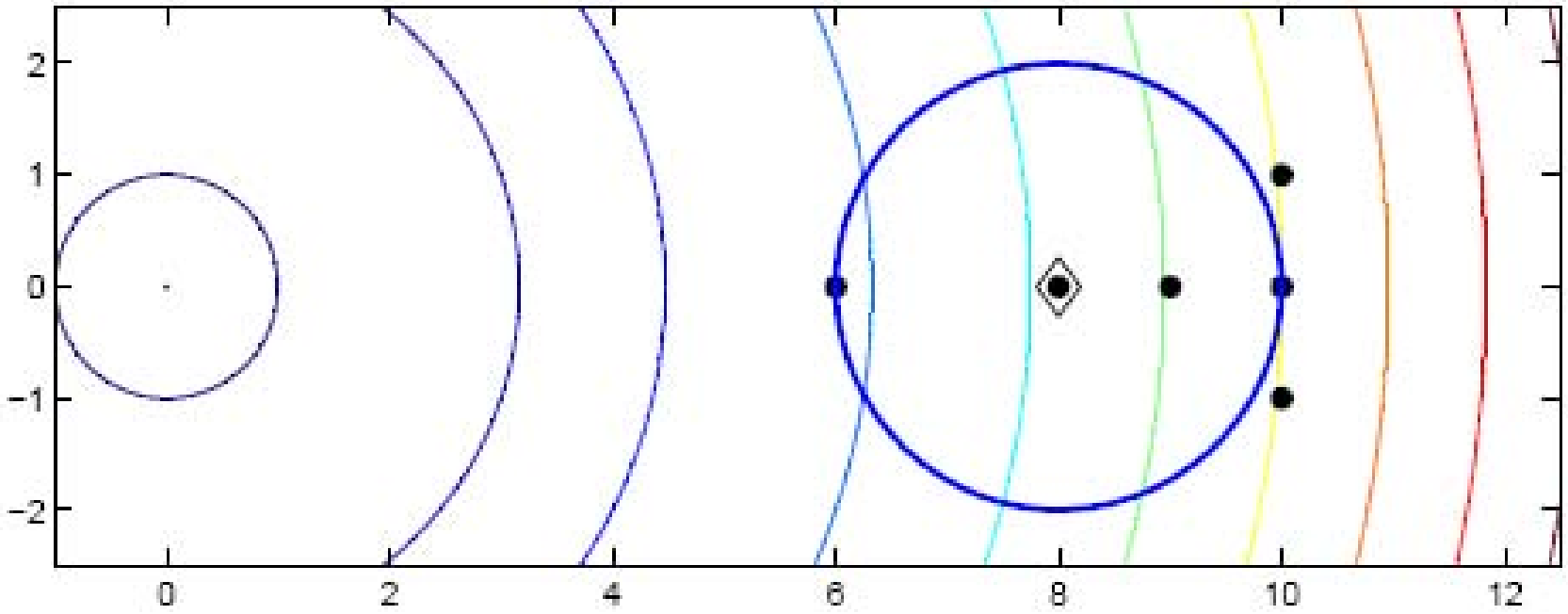
- A self-correcting geometry approach based on one-point exchanges, globally convergent to first-order stationary points ($\liminf$).
- In the unsuccessful case, y_{out} is not only based on $\|y - x_k\|_2$, but also on the values of the Lagrange polynomials at $x_k + s_k$.
- They showed that, if Δ_k is small compared to $\|g_k\|$, then the step either improves the function or the geometry/poisedness of the model.
- In their approach, model-improving iterations are not needed.

They showed, however, that the criticality step is indeed necessary.

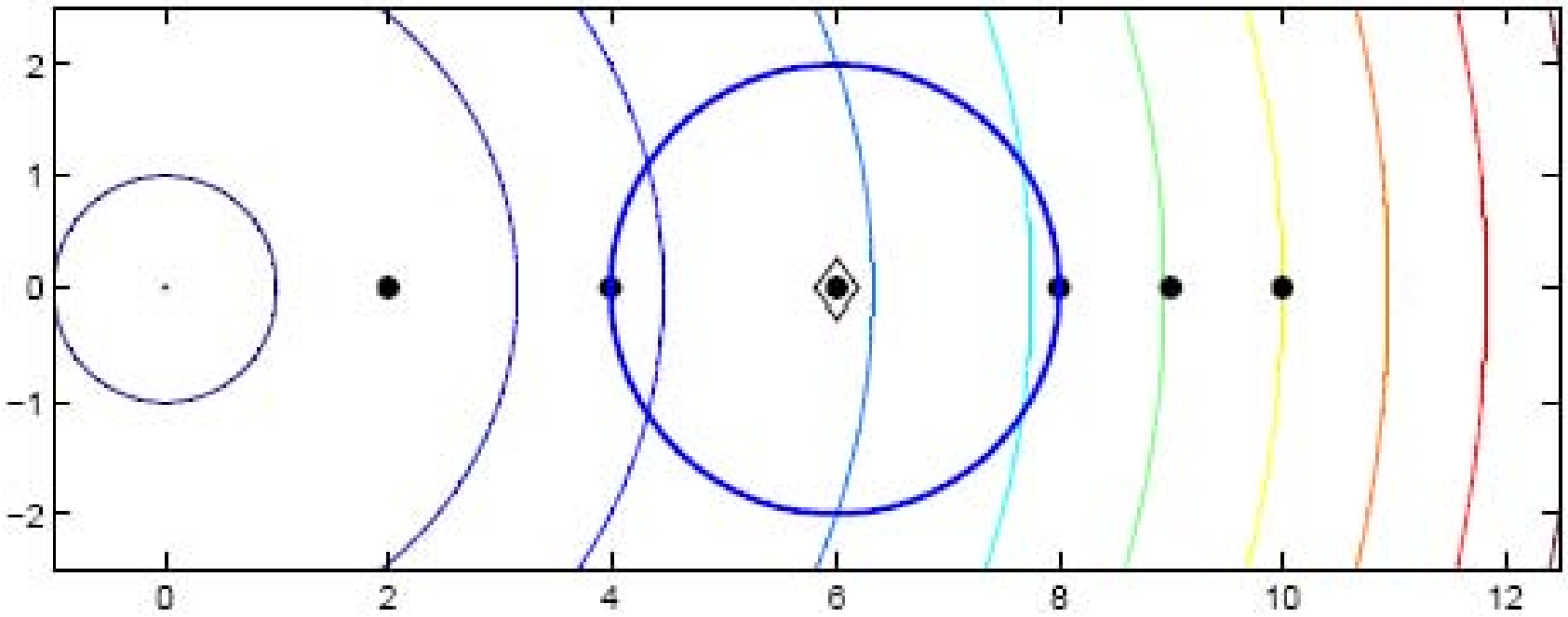
Without criticality step... (Scheinberg and Toint)



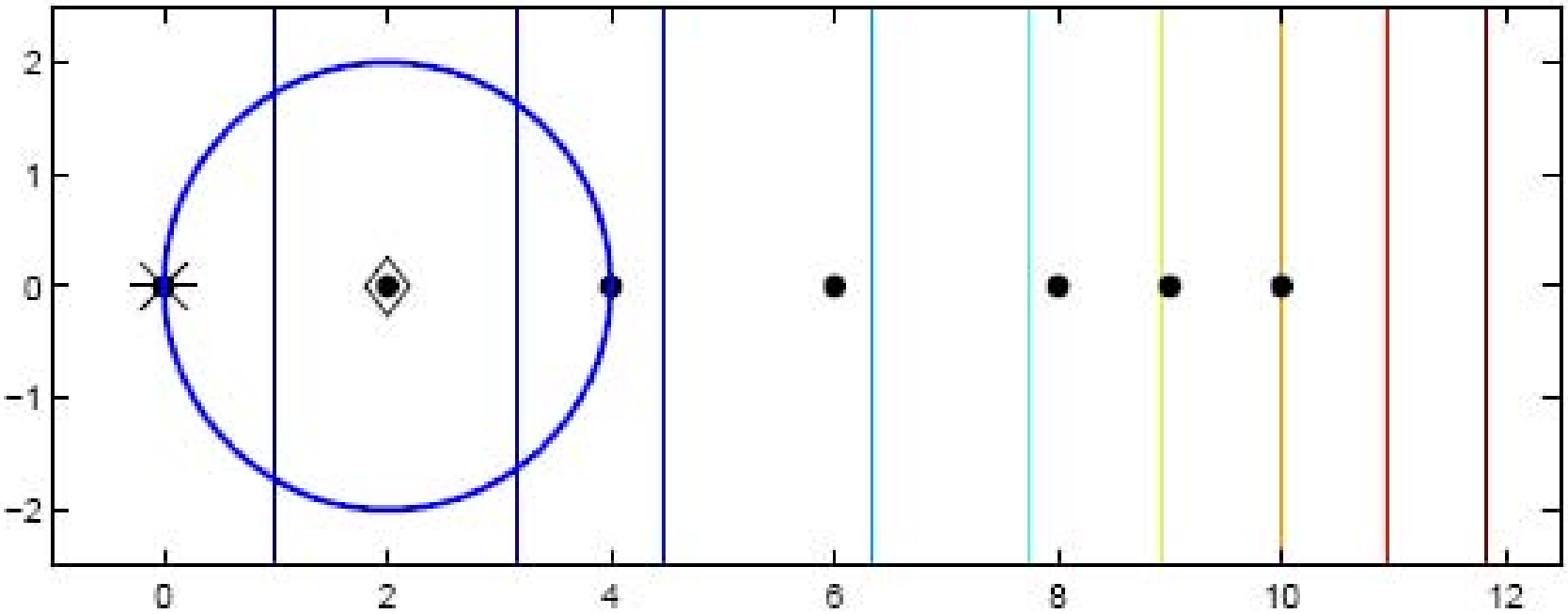
Without criticality step... (Scheinberg and Toint)



Without criticality step... (Scheinberg and Toint)



Without criticality step... (Scheinberg and Toint)



A practical interpolation-based trust-region method

Model building:

- If $|Y_k| = N = (n+1)(n+2)/2$, use determined quadratic interpolation.
- Otherwise use ℓ_1 $(p = 1)$ or Frobenius $(p = 2)$ minimum norm quadratic interpolation:

$$\begin{aligned} \min \quad & \tfrac{1}{p}\|\alpha_Q\|_p^p \\ \text{s. t.} \quad & M(\bar{\phi}_L, Y_{scaled})\alpha_L + M(\bar{\phi}_Q, Y_{scaled})\alpha_Q = f(Y_{scaled}). \end{aligned}$$

Sample set update — one starts with $|Y_0| = \mathcal{O}(n)$:

- If $|Y_k| < N = (n+1)(n+2)/2$, set $Y_{k+1} = Y_k \cup \{x_k + s_k\}$.
- Otherwise as in Fasano et al., but with $y_{out} = \operatorname{argmax} \|y - x_{k+1}\|_2$.

'Criticality step': If Δ_k is very small, discard points far away from the trust region.

Performance profiles (accuracy of 10^{-4} in function values)

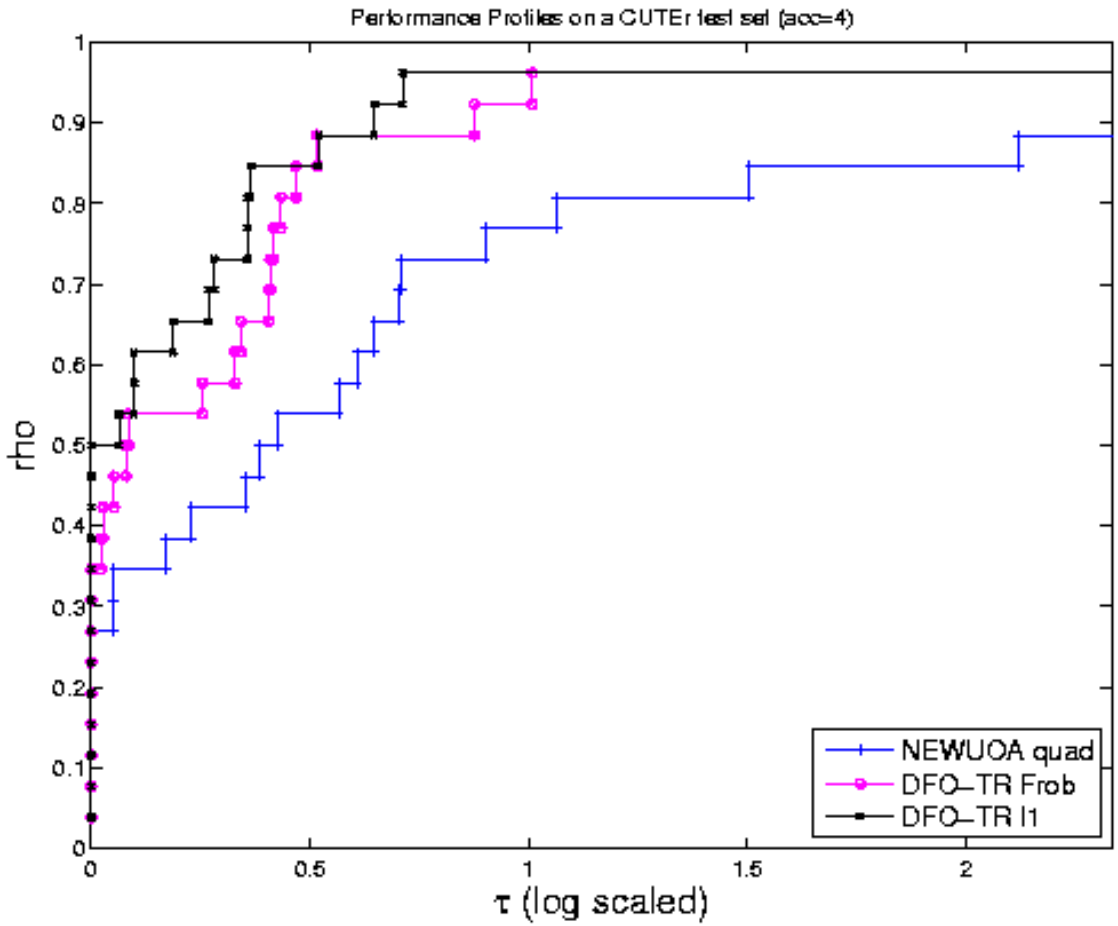


Figure: Performance profiles comparing `DFO-TR` (ℓ_1 and Frobenius) and `NEWUOA` (Powell) in a test set from `CUTEr` (Fasano et al.).

Performance profiles (accuracy of 10^{-6} in function values)

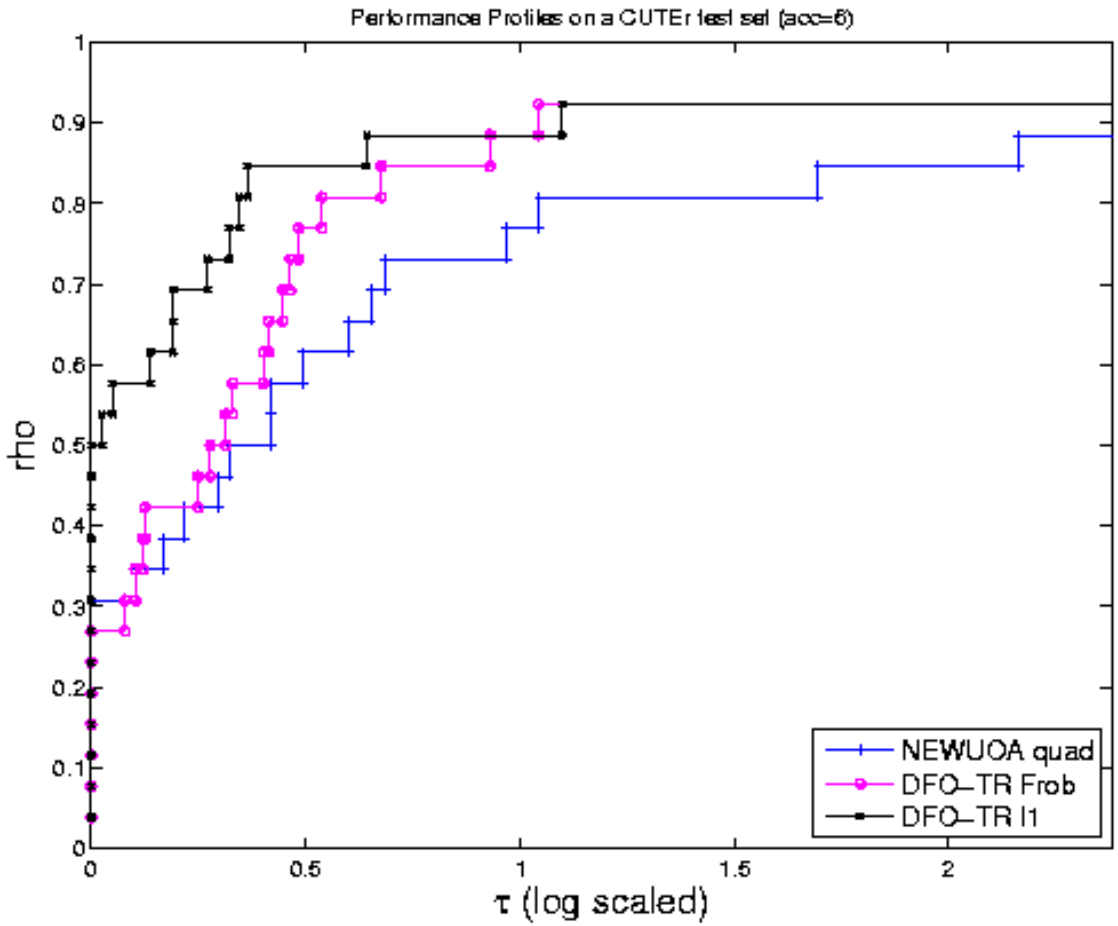


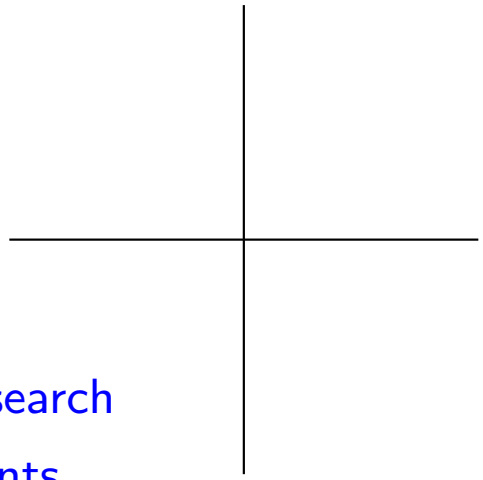
Figure: Performance profiles comparing `DFO-TR` (ℓ_1 and Frobenius) and `NEWUOA` (Powell) in a test set from `CUTEr` (Fasano et al.).

References

- A. S. Bandeira, K. Scheinberg, and L. N. Vicente, Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization, Math. Program., 134 (2012) 223–257.
- A. R. Conn, K. Scheinberg, and L. N. Vicente, Global convergence of general derivative-free trust-region algorithms to first and second order critical points, SIAM J. Optim., 20 (2009) 387–415.
- G. Fasano, J. L. Morales, and J. Nocedal, On the geometry phase in model-based algorithms for derivative-free optimization, Optim. Methods Softw., 24 (2009) 145–154.
- K. Scheinberg and Ph. L. Toint, Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization, SIAM J. Optim., 20 (2010) 3512–3532.

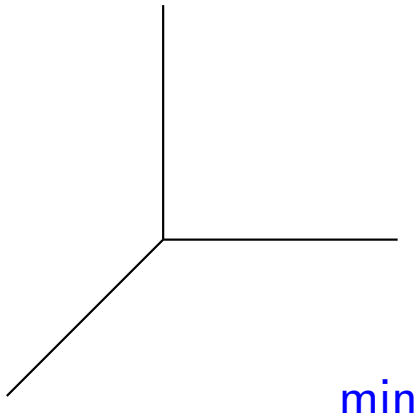
Presentation outline

Positive spanning sets



coordinate search

$2n$ elements

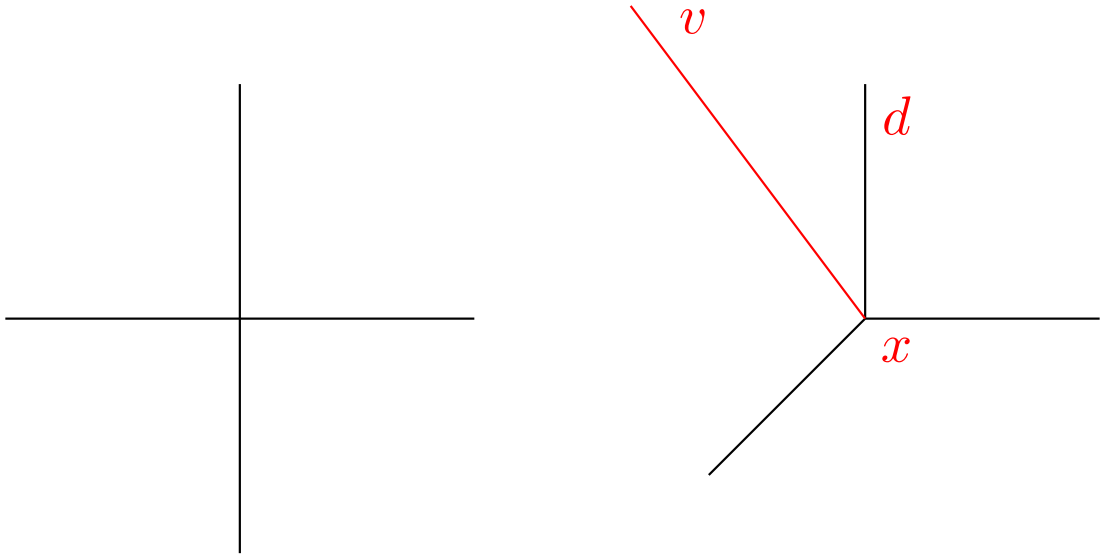


minimal case

$n+1$ elements

$$\mathrm{cm}(D) \;=\; \min_{0\neq v\in\mathbb{R}^n} \max_{d\in D} \frac{v^\top d}{\|v\|\|d\|}$$

Positive spanning sets



If $v = -\nabla f(x)$ then d is a descent direction.

Descent condition and successful iterations

Assume the polling directions are normalized.

Lemma

If

$$\operatorname{cm}\left(D_k, -\nabla f(x_k)\right) \geq \kappa \quad \text{and} \quad \alpha_k < \frac{2\kappa\|\nabla f(x_k)\|}{L_{\nabla f}+1},$$

the k-th iteration is successful.

Descent condition and successful iterations

Assume the polling directions are normalized.

Lemma

If

$$\mathrm{cm}\left(D_k, -\nabla f(x_k)\right) \geq \kappa \quad \textit{and} \quad \alpha_k < \frac{2\kappa\|\nabla f(x_k)\|}{L_{\nabla f}+1},$$

the k-th iteration is successful.

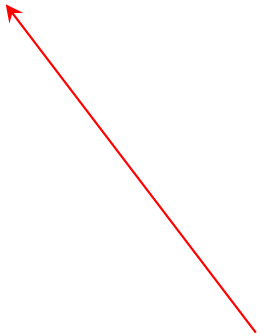
where $\mathrm{cm}(D,v)$ is the cosine measure of D given v, defined by

$$\mathrm{cm}(D,v) = \max_{d\in D} \frac{d^\top v}{\|d\|\|v\|}$$

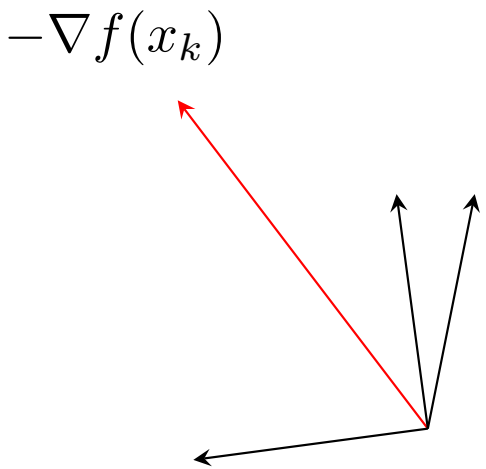
and $L_{\nabla f}$ is a Lipschitz constant of ∇f.

Randomly generating ‘positive spanning sets’ ...

$-\nabla f(x_k)$



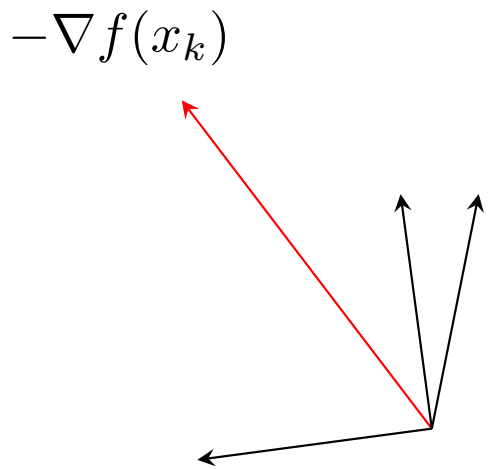
Randomly generating 'positive spanning sets' ...



$n+1$ random polling directions

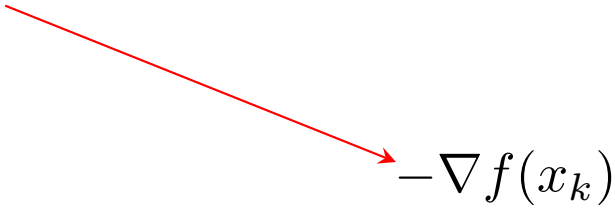
in this case not a PSS

Randomly generating 'positive spanning sets' ...

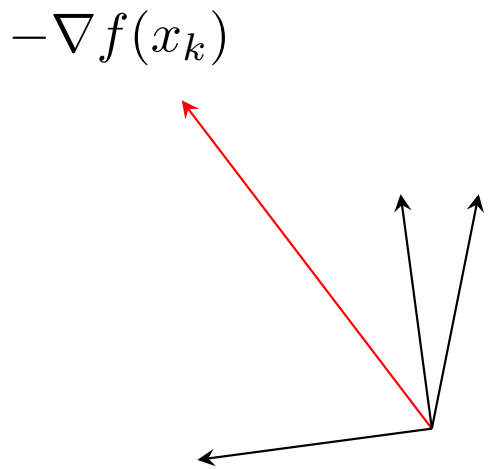


$n+1$ random polling directions

in this case not a PSS

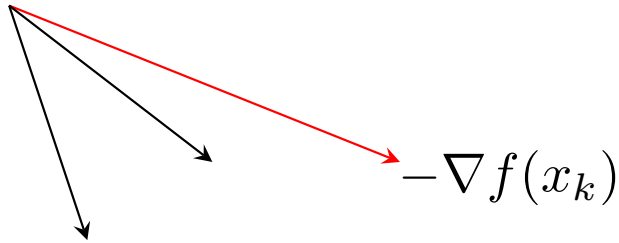


Randomly generating 'positive spanning sets' ...



$n+1$ random polling directions

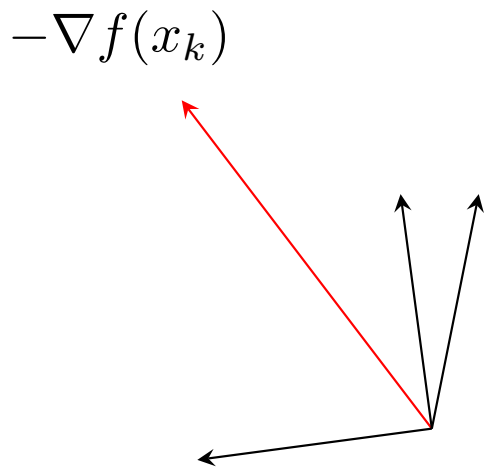
in this case not a PSS



$\leq n$ random polling directions

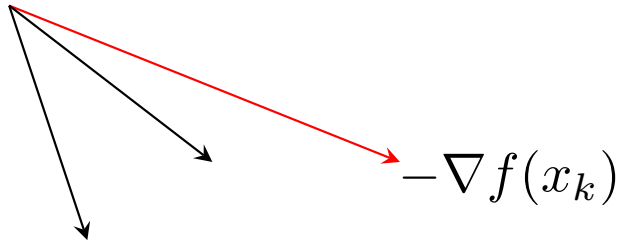
certainly not a PSS ...

Randomly generating 'positive spanning sets' ...



$n+1$ random polling directions

in this case not a PSS



$\leq n$ random polling directions

certainly not a PSS ...

$\mathrm{cm}\big(D_k, -\nabla f(x_k)\big) \geq \kappa$ can be satisfied 'probabilistically' ...

Numerical illustration

Relative performance for different sets of polling directions ($n = 40$).

	$[I \ -I]$	$[Q \ -Q]$	$2n$	$n+1$	$n/4$	2	1
arglina	3.42	8.44	10.30	6.01	1.88	1.00	–
arglinb	20.50	10.35	7.38	2.81	1.85	1.00	2.04
broydn3d	4.33	6.55	6.54	3.59	1.28	1.00	–
dqrtic	7.16	9.37	9.10	4.56	1.70	1.00	–
engval1	10.53	20.89	11.90	6.48	2.08	1.00	2.08
freuroth	56.00	6.33	1.00	1.67	1.67	1.00	4.00
integreq	16.04	16.29	12.44	6.76	2.04	1.00	–
nondquar	6.90	30.23	7.56	4.23	1.87	1.00	–
sinquad	–	–	1.65	2.01	1.00	1.55	–
vardim	1.00	3.80	1.80	2.40	1.80	1.80	4.30

Solution accuracy was 10^{-3}. Averages were taken over 10 independent runs.

Probabilistic descent

From now on, we suppose that the polling directions are not defined deterministically but generated randomly.

Probabilistic descent

From now on, we suppose that the polling directions are not defined deterministically but generated randomly.

	Iterate	Direction set
Random variables	X_k	$\mathfrak{D}_k$
Realizations	x_k	D_k

Probabilistic descent

From now on, we suppose that the polling directions are not defined deterministically but generated randomly.

	Iterate	Direction set
Random variables	X_k	$\mathfrak{D}_k$
Realizations	x_k	D_k

Definition

The sequence $\{\mathfrak{D}_k\}$ *is* p*-probabilistically* κ*-descent if, for each* $k \geq 0$,

$$\Pr\left(\operatorname{cm}(\mathfrak{D}_k, -\nabla f(X_k)) \geq \kappa \mid \mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}\right) \geq p.$$

Getting ready for global convergence

Let Z_k be the indicator function of $\left\{\mathrm{cm}\left(\mathfrak{D}_k, -\nabla f(X_k)\right) \geq \kappa\right\}$.

Getting ready for global convergence

Let Z_k be the indicator function of $\left\{\mathrm{cm}\left(\mathfrak{D}_k, -\nabla f(X_k)\right) \geq \kappa\right\}$.

Let

$$p_0 = \frac{\ln \beta}{\ln(\gamma^{-1}\beta)} = \frac{1}{2} \quad \text{when} \quad \beta = 1/2,\ \gamma = 2.$$

Getting ready for global convergence

Let Z_k be the indicator function of $\left\{\mathrm{cm}\left(\mathfrak{D}_k, -\nabla f(X_k)\right) \geq \kappa\right\}$.

Let

$$p_0 \;=\; \frac{\ln \beta}{\ln(\gamma^{-1}\beta)} \;=\; \frac{1}{2} \quad \text{when} \quad \beta = 1/2,\ \gamma = 2.$$

If $\{\mathfrak{D}_k\}$ is p_0-probabilistically κ-descent, then (due to a submartingale argument)

$$P\left[\sum_{\ell=0}^{\infty} (Z_\ell - p_0) \;=\; -\infty\right] \;=\; 0.$$

Global convergence

This then implies:

Theorem

Let $\{X_k\}$ be a sequence of random iterates generated by the algorithm.

Suppose that $\{\mathfrak{D}_k\}$ is p_0-probabilistically κ-descent for some $\kappa > 0$. Then,

$$P\left[\liminf_{k\to\infty} \|\nabla f(X_k)\| = 0\right] = 1.$$

S. Gratton, C. W. Royer, LNV, and Z. Zhang, Direct search based on probabilistic descent, SIAM J. Optim., 25 (2015) 1249–1716.

Global rate: What is desirable?

For each realization of the DS algorithm, define

- $\tilde{g}_k$: the gradient with minimum norm among $\nabla f(x_0), \ldots, \nabla f(x_k)$,

Global rate: What is desirable?

For each realization of the DS algorithm, define

- $\tilde{g}_k$: the gradient with minimum norm among $\nabla f(x_0), \ldots, \nabla f(x_k)$,
- k_ϵ: the smallest integer such that $\|\nabla f(x_k)\| \leq \epsilon$.

Global rate: What is desirable?

For each realization of the DS algorithm, define

- $\tilde{g}_k$: the gradient with minimum norm among $\nabla f(x_0), \ldots, \nabla f(x_k)$,
- k_ϵ: the smallest integer such that $\|\nabla f(x_k)\| \leq \epsilon$.

Denote the corresponding random variables by $\tilde{G}_k$ and K_ϵ.

Global rate: What is desirable?

For each realization of the DS algorithm, define

- $\tilde{g}_k$: the gradient with minimum norm among $\nabla f(x_0), \ldots, \nabla f(x_k)$,
- k_ϵ: the smallest integer such that $\|\nabla f(x_k)\| \leq \epsilon$.

Denote the corresponding random variables by $\tilde{G}_k$ and K_ϵ.

We are interested in the probabilities

Global rate

$$\Pr\left(\|\tilde{G}_k\| \leq \mathcal{O}(1/\sqrt{k})\right)$$

and

Global rate: What is desirable?

For each realization of the DS algorithm, define

- $\tilde{g}_k$: the gradient with minimum norm among $\nabla f(x_0), \ldots, \nabla f(x_k)$,
- k_ϵ: the smallest integer such that $\|\nabla f(x_k)\| \leq \epsilon$.

Denote the corresponding random variables by $\tilde{G}_k$ and K_ϵ.

We are interested in the probabilities

Global rate

$$\Pr\left(\|\tilde{G}_k\| \leq \mathcal{O}(1/\sqrt{k})\right)$$

and

Worst case complexity

$$\Pr\left(K_\epsilon \leq \mathcal{O}(\epsilon^{-2})\right).$$

Global rate: Counting descent

Let z_ℓ denote the realization of Z_ℓ $(\ell \geq 0)$.

Intuition: If $\|\tilde{g}_k\|$ is 'big', then $\sum_{\ell=0}^{k-1} z_\ell$ is probably 'small'.

Global rate: Counting descent

Let z_ℓ denote the realization of Z_ℓ ($\ell \geq 0$).

Intuition: If $\|\tilde{g}_k\|$ is 'big', then $\sum_{\ell=0}^{k-1} z_\ell$ is probably 'small'.

In fact, one can prove

$$\sum_{\ell=0}^{k-1} z_\ell \leq \mathcal{O}\left(\frac{1}{\|\tilde{g}_k\|^2}\right) + p_0 k.$$

Global rate: Counting descent

Let z_ℓ denote the realization of Z_ℓ $(\ell \geq 0)$.

Intuition: If $\|\tilde{g}_k\|$ is 'big', then $\sum_{\ell=0}^{k-1} z_\ell$ is probably 'small'.

In fact, one can prove

$$\sum_{\ell=0}^{k-1} z_\ell \;\leq\; \mathcal{O}\left(\frac{1}{\|\tilde{g}_k\|^2}\right) + p_0 k.$$

It then results,

$$\left\{\|\tilde{G}_k\| > \epsilon\right\} \;\subset\; \left\{\sum_{\ell=0}^{k-1} Z_\ell \leq \left[\mathcal{O}\left(\frac{1}{k\epsilon^2}\right) + p_0\right] k\right\}.$$

Global rate: Counting descent

Let z_ℓ denote the realization of Z_ℓ $(\ell \geq 0)$.

Intuition: If $\|\tilde{g}_k\|$ is 'big', then $\sum_{\ell=0}^{k-1} z_\ell$ is probably 'small'.

In fact, one can prove

$$\sum_{\ell=0}^{k-1} z_\ell \leq \mathcal{O}\left(\frac{1}{\|\tilde{g}_k\|^2}\right) + p_0 k.$$

It then results,

$$\left\{\|\tilde{G}_k\| > \epsilon\right\} \subset \left\{\sum_{\ell=0}^{k-1} Z_\ell \leq \underbrace{\left[\mathcal{O}\left(\frac{1}{k\epsilon^2}\right) + p_0\right]}_{\lambda} k\right\}.$$

A universal result

Hence $\Pr\left(\|\tilde{G}_k\| \leq \epsilon\right) \;=\; 1 - \Pr\left(\|\tilde{G}_k\| > \epsilon\right) \;\geq\; 1 - \Pr\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right)$.

A universal result

Hence $\Pr\left(\|\tilde{G}_k\| \leq \epsilon\right) \;=\; 1 - \Pr\left(\|\tilde{G}_k\| > \epsilon\right) \;\geq\; 1 - \Pr\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right)$.

Denote

$$\pi_k(\lambda) = \Pr\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right).$$

A universal result

Hence $\Pr\left(\|\tilde{G}_k\| \leq \epsilon\right) \;=\; 1 - \Pr\left(\|\tilde{G}_k\| > \epsilon\right) \;\geq\; 1 - \Pr\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right)$.

Denote

$$\pi_k(\lambda) = \Pr\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right).$$

If $\{\mathfrak{D}_k\}$ is probabilistic descent, then π_k obeys a Chernoff type bound:

Lemma

Suppose that $\{\mathfrak{D}_k\}$ is p-probabilistically κ-descent and $\lambda \in (0, p)$. Then

$$\pi_k(\lambda) \;\leq\; \exp\left[-\frac{(p-\lambda)^2}{2p} k\right].$$

Global rate

Now we plug the Chernoff type bound into

$$\Pr\left(\|\tilde{G}_k\| \leq \epsilon\right) \geq 1 - \Pr\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right).$$

Global rate

Now we plug the Chernoff type bound into

$$\Pr\left(\|\tilde{G}_k\| \leq \epsilon\right) \;\geq\; 1 - \Pr\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right).$$

Theorem

Suppose that $\{\mathfrak{D}_k\}$ *is* p*-probabilistically* κ*-descent with* $p > p_0$ *and*

$$k \;\geq\; \mathcal{O}\left(\frac{1}{\kappa^2\epsilon^2}\right).$$

Global rate

Now we plug the Chernoff type bound into

$$\Pr\left(\|\tilde{G}_k\| \leq \epsilon\right) \geq 1 - \Pr\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right).$$

Theorem

Suppose that $\{\mathfrak{D}_k\}$ *is* p*-probabilistically* κ*-descent with* $p > p_0$ *and*

$$k \geq \mathcal{O}\left(\frac{1}{\kappa^2 \epsilon^2}\right).$$

Then

$$\Pr\left(\|\tilde{G}_k\| \leq \epsilon\right) \geq 1 - \exp\left[-\mathcal{O}(k)\right].$$

Global rate

Theorem

Suppose that $\{\mathfrak{D}_k\}$ *is* p*-probabilistically* κ*-descent with* $p > p_0$*. Then*

$$\Pr\left(\|\tilde{G}_k\| \leq \frac{1}{\kappa\sqrt{k}}\right) \geq 1 - \exp\left[-\mathcal{O}(k)\right].$$

Global rate

Theorem

Suppose that $\{\mathfrak{D}_k\}$ *is* p*-probabilistically* κ*-descent with* $p > p_0$*. Then*

$$\Pr\left(\|\tilde{G}_k\| \leq \frac{1}{\kappa\sqrt{k}}\right) \geq 1 - \exp\left[-\mathcal{O}(k)\right].$$

$\longrightarrow$ $\mathcal{O}(1/\sqrt{k})$ decaying sublinear rate for gradient holds with overwhelmingly high probability, matching the deterministic case.

Worst case complexity

Since $\Pr(K_\epsilon \leq k) = \Pr(\|\tilde{G}_k\| \leq \epsilon)$, we also get:

Theorem

Suppose that $\{\mathfrak{D}_k\}$ is p-probabilistically κ-descent with $p > p_0$. Then

$$\Pr\left(K_\epsilon \leq \left\lceil \mathcal{O}\left(\frac{\epsilon^{-2}}{\kappa^2}\right)\right\rceil\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right].$$

Worst case complexity

Since $\Pr(K_\epsilon \leq k) = \Pr(\|\tilde{G}_k\| \leq \epsilon)$, we also get:

Theorem

Suppose that $\{\mathfrak{D}_k\}$ is p-probabilistically κ-descent with $p > p_0$. Then

$$\Pr\left(K_\epsilon \leq \left\lceil \mathcal{O}\left(\frac{\epsilon^{-2}}{\kappa^2}\right)\right\rceil\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right].$$

$\longrightarrow$ $\mathcal{O}(\epsilon^{-2})$ complexity bound for # of iterations holds with overwhelmingly high probability, matching the deterministic case.

Practical probabilistic descent sets

Let m be the $\#$ of polling directions. For each $k \geq 0$,

- $\mathfrak{D}_k$ is independent of the previous iterations,

Practical probabilistic descent sets

Let m be the # of polling directions. For each $k \geq 0$,

- $\mathfrak{D}_k$ is independent of the previous iterations,
- $\mathfrak{D}_k$ is a set $\{\mathfrak{d}_1, \ldots, \mathfrak{d}_m\}$ of independent random vectors.
  - $\mathfrak{d}_i$ is uniformly distributed on the unit sphere,

Practical probabilistic descent sets

Let m be the # of polling directions. For each $k \geq 0$,

- $\mathfrak{D}_k$ is independent of the previous iterations,
- $\mathfrak{D}_k$ is a set $\{\mathfrak{d}_1, \ldots, \mathfrak{d}_m\}$ of independent random vectors.
  - $\mathfrak{d}_i$ is uniformly distributed on the unit sphere,
  - $\mathfrak{d}_i$ can be obtained by normalizing a vector from standard normal distribution.

Worst case complexity: Dependence on the dimension

Then, when $m > 1$, $\{\mathfrak{D}_k\}$ is p-probabilistically $(\tau/\sqrt{n})$-descent for some $p > p_0 = 1/2$ and $\tau > 0$ independent of n.

Worst case complexity: Dependence on the dimension

Then, when $m > 1$, $\{\mathfrak{D}_k\}$ is p-probabilistically $(\tau/\sqrt{n})$-descent for some $p > p_0 = 1/2$ and $\tau > 0$ independent of n.

Plugging $\kappa = \tau/\sqrt{n}$ into the WCC bound, one obtains

Worst case complexity: Dependence on the dimension

Then, when $m > 1$, $\{\mathfrak{D}_k\}$ is p-probabilistically $(\tau/\sqrt{n})$-descent for some $p > p_0 = 1/2$ and $\tau > 0$ independent of n.

Plugging $\kappa = \tau/\sqrt{n}$ into the WCC bound, one obtains

WCC (number of function evaluations)

$$\Pr\left(K_\epsilon^f \leq \left\lceil \mathcal{O}\left(n\epsilon^{-2}\right)\right\rceil m\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right].$$

Worst case complexity: Dependence on the dimension

Then, when $m > 1$, $\{\mathfrak{D}_k\}$ is p-probabilistically $(\tau/\sqrt{n})$-descent for some $p > p_0 = 1/2$ and $\tau > 0$ independent of n.

Plugging $\kappa = \tau/\sqrt{n}$ into the WCC bound, one obtains

WCC (number of function evaluations)

$$\Pr\left(K_\epsilon^f \leq \left\lceil \mathcal{O}\left(n\epsilon^{-2}\right)\right\rceil m\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right].$$

The WCC bound is $\mathcal{O}(mn\epsilon^{-2})$, better than when $\mathcal{O}(n^2\epsilon^{-2})$ when $m \ll n$.

Worst case complexity: Dependence on the dimension

Then, when $m > 1$, $\{\mathfrak{D}_k\}$ is p-probabilistically $(\tau/\sqrt{n})$-descent for some $p > p_0 = 1/2$ and $\tau > 0$ independent of n.

Plugging $\kappa = \tau/\sqrt{n}$ into the WCC bound, one obtains

WCC (number of function evaluations)

$$\Pr\left(K_\epsilon^f \leq \left\lceil \mathcal{O}\left(n\epsilon^{-2}\right)\right\rceil m\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right].$$

The WCC bound is $\mathcal{O}(mn\epsilon^{-2})$, better than when $\mathcal{O}(n^2\epsilon^{-2})$ when $m \ll n$.

Theory admits $m = 2$, leading to

$$\mathcal{O}(n\epsilon^{-2}) \quad !!!$$

A more detailed look at the numerical experiments

Relative performance for different sets of polling directions ($n = 40$).

	$[I \ -I]$	$[Q \ -Q]$	2 ($\gamma = 2$)	4 ($\gamma = 1.1$)
arglina	1.00	3.17	5.86	6.73
arglinb	34.12	5.34	1.00	2.02
broydn3d	1.00	1.91	2.04	3.47
dqrtic	1.18	1.36	1.00	1.48
engval1	1.05	1.00	2.29	2.89
freuroth	17.74	7.39	1.35	1.00
integreq	1.54	1.49	1.00	1.34
nondquar	1.00	2.82	1.37	1.73
sinquad	–	1.26	1.00	–
vardim	20.31	11.02	1.00	1.84

Now $\gamma = 1$ for $[I \ -I]$ and $[Q \ -Q]$.

A more detailed look at the numerical experiments

Relative performance for different sets of polling directions ($n = 100$).

	$[I \ -I]$	$[Q \ -Q]$	2 ($\gamma = 2$)	4 ($\gamma = 1.1$)
arglina	1.00	3.86	5.86	7.58
arglinb	138.28	107.32	1.00	1.99
broydn3d	1.00	2.57	1.92	3.21
dqrtic	3.01	3.25	1.00	1.46
engval1	1.04	1.00	2.06	2.84
freuroth	31.94	17.72	1.36	1.00
integreq	1.83	1.66	1.00	1.22
nondquar	1.18	2.83	1.00	1.17
sinquad	–	–	–	–
vardim	112.22	19.72	1.00	2.36

Now $\gamma = 1$ for $[I \ -I]$ and $[Q \ -Q]$.

So, a new proof technique

A new proof technique for establishing global rates and WCC bounds for randomized algorithms for which

So, a new proof technique

A new proof technique for establishing global rates and WCC bounds for randomized algorithms for which

- the new iterate depends on some object (directions, models),
- the quality of the object is favorable with a certain probability.

So, a new proof technique

A new proof technique for establishing global rates and WCC bounds for randomized algorithms for which

- the new iterate depends on some object (directions, models),
- the quality of the object is favorable with a certain probability.

The technique is based on:

- counting the number of iterations for which the quality is favorable,
- examining the probabilistic behavior of this number.

So, a new proof technique

A new proof technique for establishing global rates and WCC bounds for randomized algorithms for which

- the new iterate depends on some object (directions, models),
- the quality of the object is favorable with a certain probability.

The technique is based on:

- counting the number of iterations for which the quality is favorable,
- examining the probabilistic behavior of this number.

It is thus possible to obtain a rate of $\mathcal{O}(1/\sqrt{k})$, with overwhelmingly high probability, also for trust-region methods based on probabilistic models (see next slides).

TR based on probabilistic models

Random models (random sample sets) may maintain a higher quality by using fewer sample points.
$\longrightarrow$ Hessian sparse but sparsity structure unknown.

TR based on probabilistic models

Random models (random sample sets) may maintain a higher quality by using fewer sample points.
$\longrightarrow$ Hessian sparse but sparsity structure unknown.

Random models may give an advantage in a parallel environment without full synchronization.
$\longrightarrow$ Time to compute function values is random and a budget is imposed.

TR based on probabilistic models

Random models (random sample sets) may maintain a higher quality by using fewer sample points.
$\longrightarrow$ Hessian sparse but sparsity structure unknown.

Random models may give an advantage in a parallel environment without full synchronization.
$\longrightarrow$ Time to compute function values is random and a budget is imposed.

So, now, models are built iteratively in some random fashion.

$\longrightarrow$ M_k for random models, and $m_k = M_k(\omega_k)$ for their realizations.

The key assumption for convergence will be then that these models exhibit good accuracy with sufficiently high probability.

TR based on probabilistic models

Fix three positive parameters $\eta_1, \eta_2, \gamma, \beta$, with $\beta < 1 < \gamma$.

Algorithm (Iteration k)

Approximate the function f in $B(x_k; \delta_k)$ with m_k.

Compute a step s_k by solving $\min_{s \in B(0;\delta_k)} m_k(x_k + s)$.

Let

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

If $\rho_k \geq \eta_1$, set $x_{k+1} = x_k + s_k$ and

$$\delta_{k+1} = \begin{cases} \gamma\delta_k & \text{if } \|g_k\| \geq \eta_2\delta_k, \\ \beta\delta_k & \text{otherwise.} \end{cases}$$

Otherwise, set $x_{k+1} = x_k$ and $\delta_{k+1} = \beta\delta_k$.

Probabilistic fully linear/quadratic models

Assumption

We say that a sequence of random models $\{M_k\}$ *is* p*-probabilistically* $(\kappa_{eg}, \kappa_{ef})$*-fully linear for a corresponding sequence* $\{B(X_k, \Delta_k)\}$ *if the events*

$$S_k \;=\; \{M_k \text{ is a } (\kappa_{eg}, \kappa_{ef})\text{-fully linear model of } f \text{ on } B(X_k, \Delta_k)\}$$

satisfy the following submartingale-like condition

$$P(S_k | \sigma(M_0, \ldots, M_{k-1})) \;\geq\; p.$$

Probabilistic fully linear/quadratic models

Assumption

We say that a sequence of random models $\{M_k\}$ *is* p*-probabilistically* $(\kappa_{eg}, \kappa_{ef})$*-fully linear for a corresponding sequence* $\{B(X_k, \Delta_k)\}$ *if the events*

$$S_k \;=\; \{M_k \text{ is a } (\kappa_{eg}, \kappa_{ef})\text{-fully linear model of } f \text{ on } B(X_k, \Delta_k)\}$$

satisfy the following submartingale-like condition

$$P(S_k | \sigma(M_0, \ldots, M_{k-1})) \;\geq\; p.$$

p-probabilistically $(\kappa_{eh}, \kappa_{eg}, \kappa_{ef})$-fully quadratic are defined accordingly.

Global convergence of TR based on probabilistic models

Theorem

Let $\{X_k\}$ be a sequence of random iterates generated by the algorithm.

Suppose that the model sequence $\{M_k\}$ is p_0-probabilistically $(\kappa_{eg}, \kappa_{ef})$-fully linear for some $\kappa_{eg}, \kappa_{ef} > 0$. Then,

$$P\left[\lim_{k\to\infty} \|\nabla f(X_k)\| = 0\right] = 1.$$

Global convergence of TR based on probabilistic models

Theorem

Let $\{X_k\}$ be a sequence of random iterates generated by the algorithm.

Suppose that the model sequence $\{M_k\}$ is p_0-probabilistically $(\kappa_{eg}, \kappa_{ef})$-fully linear for some $\kappa_{eg}, \kappa_{ef} > 0$. Then,

$$P\left[\lim_{k\to\infty} \|\nabla f(X_k)\| = 0\right] = 1.$$

Suppose that the model sequence $\{M_k\}$ is p_0-probabilistically $(\kappa_{eh}, \kappa_{eg}, \kappa_{ef})$-fully quadratic for some $\kappa_{eh}, \kappa_{eg}, \kappa_{ef} > 0$. Then,

$$P\left[\liminf_{k\to\infty} \max\left\{\|\nabla f(x_k)\|, -\lambda_{\min}[\nabla^2 f(x_k)]\right\} = 0\right] = 1.$$

References for TR based on probabilistic models

Global convergence:

- A. S. Bandeira, K. Scheinberg, and L. N. Vicente, Convergence of trust-region methods based on probabilistic models, SIAM Journal on Optimization, 24 (2014) 1238-1264.

Global rate ($\mathcal{O}(1/\sqrt{k})$ with overwhelmingly high probability):

- S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang, Complexity and global rates of trust-region methods based on probabilistic models, preprint 17-09, Dept. Mathematics, Univ. Coimbra.

Stochastic optimization

What is observable is $\tilde{f}(x, \varepsilon(\omega))$, where ε is a random variable. The objective function $f(x)$ may be given by $E(\tilde{f}(x, \varepsilon))$.

Stochastic optimization

What is observable is $\tilde{f}(x, \varepsilon(\omega))$, where ε is a random variable. The objective function $f(x)$ may be given by $E(\tilde{f}(x, \varepsilon))$.

One possible approach is to extend the **CSV framework** using SAA (Sample-Average Approximation).

Stochastic optimization

What is observable is $\tilde{f}(x, \varepsilon(\omega))$, where ε is a random variable. The objective function $f(x)$ may be given by $E(\tilde{f}(x, \varepsilon))$.

One possible approach is to extend the **CSV framework** using SAA (Sample-Average Approximation).

First-order global convergence wp1 was derived in:

- ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free simulation optimization, S. Shashaani, F. S. Hashemi, and R. Pasuparhy, 2015.

Stochastic optimization

What is observable is $\tilde{f}(x, \varepsilon(\omega))$, where ε is a random variable. The objective function $f(x)$ may be given by $E(\tilde{f}(x, \varepsilon))$.

One possible approach is to extend the **CSV framework** using SAA (Sample-Average Approximation).

First-order global convergence wp1 was derived in:

- ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free simulation optimization, S. Shashaani, F. S. Hashemi, and R. Pasuparhy, 2015.

The number of observations in each Monte Carlo oracle may be up to $\mathcal{O}(\delta^{-4})$.

Stochastic optimization

What is observable is $\tilde{f}(x, \varepsilon(\omega))$, where ε is a random variable. The objective function $f(x)$ may be given by $E(\tilde{f}(x, \varepsilon))$.

One possible approach is to extend the **CSV framework** using SAA (Sample-Average Approximation).

First-order global convergence wp1 was derived in:

- ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free simulation optimization, S. Shashaani, F. S. Hashemi, and R. Pasuparhy, 2015.

The number of observations in each Monte Carlo oracle may be up to $\mathcal{O}(\delta^{-4})$.

The proof seems correct but... for algorithmic parameters that depend on unknown constants. To follow...

Stochastic optimization

Another avenue is to extend the **TR based on probabilistic models** to cover also probabilistic estimates of the obj. function:

- Stochastic optimization using a trust-region method and random models, R. Chen, M. Menickelly, and K. Scheinberg, 2015.

Stochastic optimization

Another avenue is to extend the **TR based on probabilistic models** to cover also probabilistic estimates of the obj. function:

- Stochastic optimization using a trust-region method and random models, R. Chen, M. Menickelly, and K. Scheinberg, 2015.

First-order global convergence wp1 has also been derived, but also for algorithmic parameters that depend on unknown constants. To follow...

Stochastic optimization

Another avenue is to extend the **TR based on probabilistic models** to cover also probabilistic estimates of the obj. function:

- Stochastic optimization using a trust-region method and random models, R. Chen, M. Menickelly, and K. Scheinberg, 2015.

First-order global convergence wp1 has also been derived, but also for algorithmic parameters that depend on unknown constants. To follow...

In the non biased case $f(x) = E(\tilde{f}(x, \varepsilon))$, the probabilistic assumptions can be ensured by SAA (with $\mathcal{O}(\delta^{-4})$ observations).

Stochastic optimization

Another avenue is to extend the **TR based on probabilistic models** to cover also probabilistic estimates of the obj. function:

- Stochastic optimization using a trust-region method and random models, R. Chen, M. Menickelly, and K. Scheinberg, 2015.

First-order global convergence wp1 has also been derived, but also for algorithmic parameters that depend on unknown constants. To follow...

In the non biased case $f(x) = E(\tilde{f}(x, \varepsilon))$, the probabilistic assumptions can be ensured by SAA (with $\mathcal{O}(\delta^{-4})$ observations).

This approach can handle biased cases like failures in function evaluations or even processor failures (thus accommodating gradient failures when using f.d.).

Presentation outline

Simplicial direct-search methods

The Nelder-Mead method:

- Considers a simplex of $n+1$ vertices at each iteration, trying to replace the worst vertex by a new one.
- For that it performs one of the following simplex operations: reflexion, expansion, outside contraction, inside contraction.

  $\longrightarrow$ Costs 1 or 2 function evaluations (per iteration).
- If they all the above fail the simplex is shrunk.

  $\longrightarrow$ Additional n function evaluations (per iteration).

Nelder-Mead method

Every iteration in $\mathbb{R}^n$ is based on a simplex of $n+1$ vertices $Y = \{y^0, y^1, \ldots, y^n\}$ ordered by increasing values of f.

The most common Nelder-Mead iterations perform a reflection, an expansion, or a contraction (inside or outside the simplex).

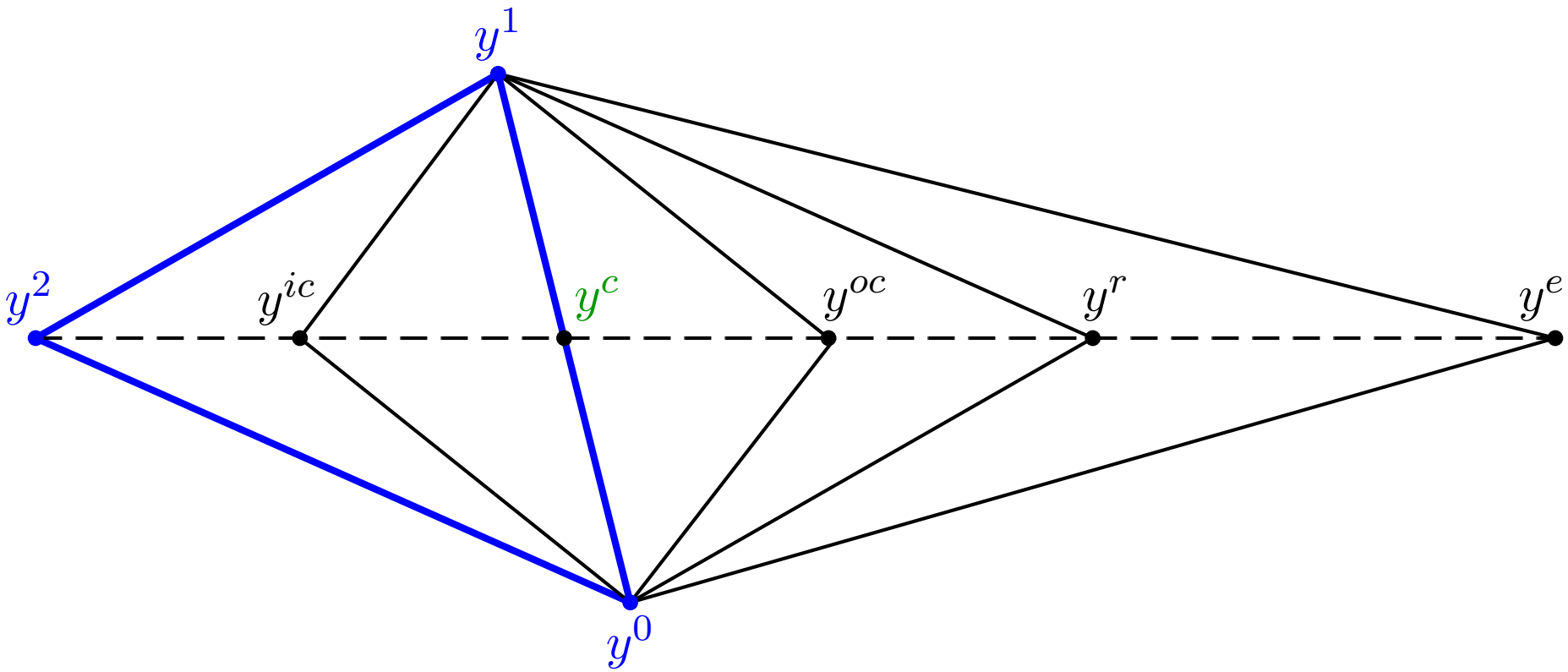
In such iterations the worst vertex y^n is replaced by a point in the line that connects y^n and y^c,

$$y \;=\; y^c + \delta(y^c - y^n), \quad \delta \in \mathbb{R},$$

where $y^c = \sum_{i=0}^{n-1} y^i/n$ is the centroid of the best n vertices.

When $\delta = 1$ we have a (genuine or isometric) reflection, when $\delta = 2$ an expansion, when $\delta = 1/2$ an outside contraction, and when $\delta = -1/2$ an inside contraction.

Nelder Mead simplex operations (reflections, expansions, outside contractions, inside contractions)



y^c is the centroid of the face opposed to the worse vertex y^2.

Nelder-Mead method

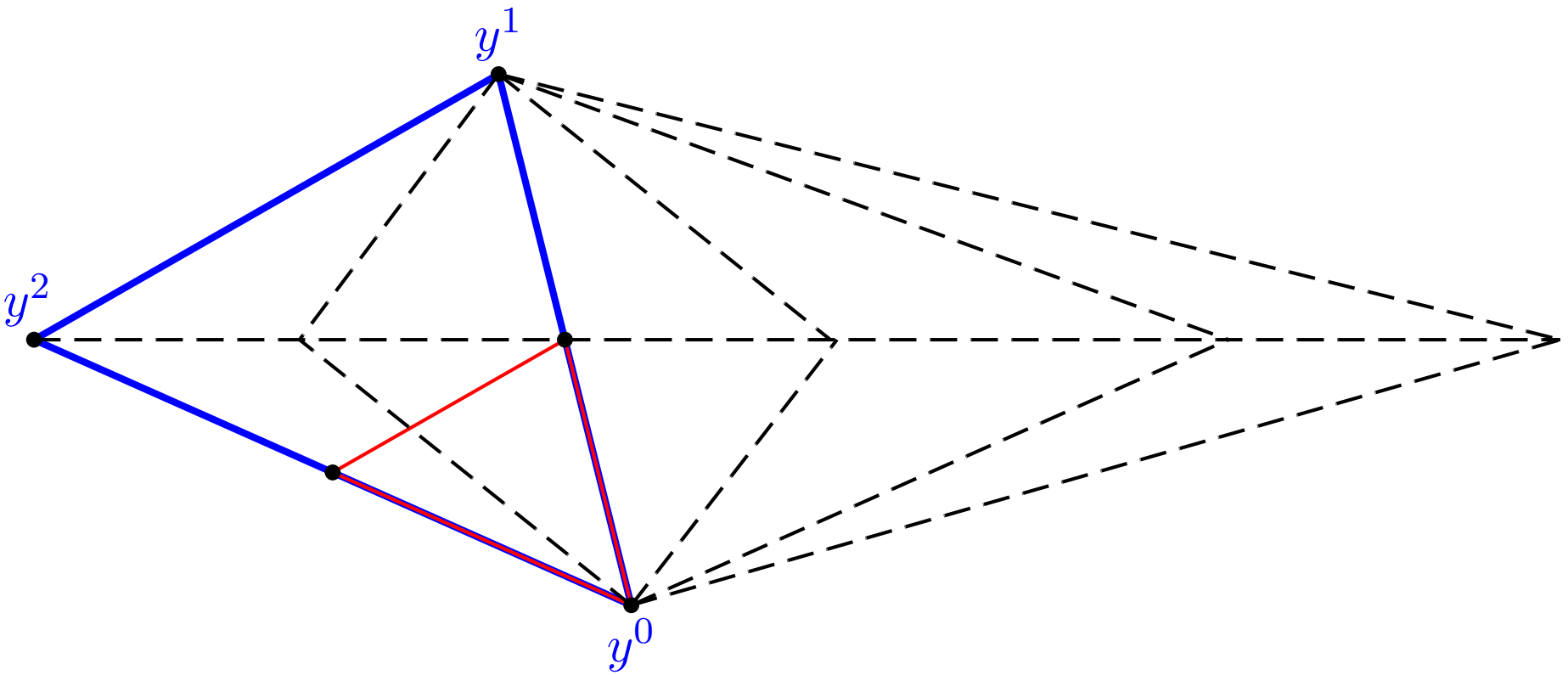
A Nelder-Mead iteration can also perform a simplex shrink, which rarely occurs in practice.

When a shrink is performed all the vertices in Y are thrown away except the best one y^0.

Then, n new vertices are computed by shrinking the simplex at y^0, i.e., by computing, for instance, $y^0 + 1/2(y^i - y^0)$, $i = 1, \ldots, n$.

We note that the shape of the resulting simplices can change by being stretched or contracted, unless a shrink occurs.

Nelder Mead simplex operations (shrinks)



Nelder-Mead method

Note that, except for shrinks, the emphasis is on replacing the worse vertex rather than improving the best.

The Nelder-Mead method does not parallelize well since the sampling procedure is necessarily sequential (except at a shrink).

Nelder-Mead method

Choose an initial simplex of vertices $Y_0 = \{y_0^0, y_0^1 \ldots, y_0^n\}$. Evaluate f at the points in Y_0. Choose constants:

$$0 < \gamma^s < 1, \qquad -1 < \delta^{ic} < 0 < \delta^{oc} < \delta^r < \delta^e.$$

The standard choices for the coefficients used are

$$\gamma^s = \frac{1}{2}, \quad \delta^{ic} = -\frac{1}{2}, \quad \delta^{oc} = \frac{1}{2}, \quad \delta^r = 1, \quad \text{and} \quad \delta^e = 2.$$

Set $Y = Y_k$. **Order** the $n+1$ vertices of $Y = \{y^0, y^1 \ldots, y^n\}$ so that

$$f^0 = f(y^0) \leq f^1 = f(y^1) \leq \cdots \leq f^n = f(y^n).$$

Nelder-Mead method

Reflect: Reflect the worst vertex y^n over the centroid $y^c = \sum_{i=0}^{n-1} y^i/n$ of the remaining n vertices:

$$y^r = y^c + \delta^r(y^c - y^n).$$

Evaluate $f^r = f(y^r)$. If $f^0 \leq f^r < f^{n-1}$ then replace y^n by the reflected point y^r and terminate the iteration: $Y_{k+1} = \{y^0, y^1 \ldots, y^{n-1}, y^r\}$.

Expand: If $f^r < f^0$ then calculate the expansion point:

$$y^e = y^c + \delta^e(y^c - y^n)$$

and evaluate $f^e = f(y^e)$. If $f^e \leq f^r$ replace y^n by the expansion point y^e and terminate the iteration: $Y_{k+1} = \{y^0, y^1 \ldots, y^{n-1}, y^e\}$.

Otherwise replace y^n by the reflected point y^r and terminate the iteration: $Y_{k+1} = \{y^0, y^1 \ldots, y^{n-1}, y^r\}$.

Nelder-Mead method

Contract: If $f^r \geq f^{n-1}$ then a contraction is performed between the best of y^r and y^n.

Outside contraction: If $f^r < f^n$ perform an outside contraction:

$$y^{oc} = y^c + \delta^{oc}(y^c - y^n)$$

and evaluate $f^{oc} = f(y^{oc})$. If $f^{oc} \leq f^r$ then replace y^n by the outside contraction point y_k^{oc} and terminate the iteration:
$Y_{k+1} = \{y^0, y^1 \ldots, y^{n-1}, y^{oc}\}$. Otherwise perform a shrink.

Inside contraction: If $f^r \geq f^n$ perform an inside contraction:

$$y^{ic} = y^c + \delta^{ic}(y^c - y^n)$$

and evaluate $f^{ic} = f(y^{ic})$. If $f^{ic} < f^n$ then replace y^n by the inside contraction point y^{ic} and terminate the iteration:
$Y_{k+1} = \{y^0, y^1 \ldots, y^{n-1}, y^{ic}\}$. Otherwise perform a shrink.

Nelder-Mead method

Shrink: Evaluate f at the n points $y^0 + \gamma^s(y^i - y^0)$, $i = 1, \ldots, n$, and replace $y^1, \ldots, y^n$ by these points, terminating the iteration: $Y_{k+1} = \{y^0 + \gamma^s(y^i - y^0),\ i = 0, \ldots, n\}$.

Nelder-Mead method

A stopping criterion could consist in terminating the run when the diameter of the simplex becomes smaller than a chosen tolerance $\Delta_{tol} > 0$ (for instance $\Delta_{tol} = 10^{-5}$).

The Nelder-Mead algorithm performs the following number of function evaluations per iteration:

1	if the iteration is a reflection,
2	if the iteration is an expansion or contraction,
$n+2$	if the iteration is a shrink.

Nelder-Mead properties

The worst vertex function value might not necessarily decrease after a non-shrink iteration.

For instance, suppose that $n = 4$ and that the vertex function values are $(f_k^0, f_k^1, f_k^2, f_k^3, f_k^4) = (1, 2, 2, 3, 3)$ at the non-shrink iteration k. Suppose also that the new vertex has function value 2.

Then, the vertex function values at iteration $k + 1$ are $(f_{k+1}^0, f_{k+1}^1, f_{k+1}^2, f_{k+1}^3, f_{k+1}^4) = (1, 2, 2, 2, 3)$. It is clear from this example that the worst vertex function has not improved.

However, one can easily see that the worst function value will necessarily decrease after at most $n + 1$ consecutive non-shrink iterations, unless an optimal value has already been attained.

Nelder-Mead properties

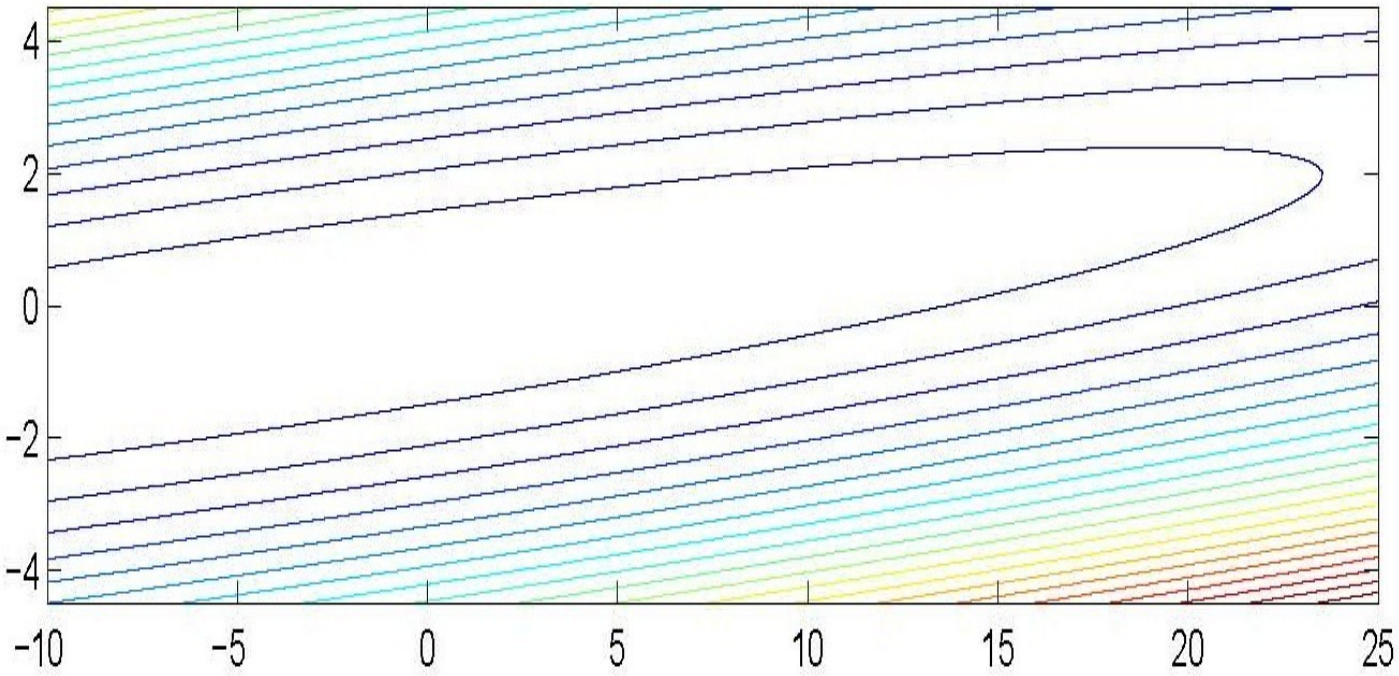
The Nelder-Mead algorithm was designed with the idea that the simplices would adapt themselves to the local landscape.

The moves allow any simplex shape to be approximated. The good practical performance of the algorithm, when it works, is directly related to this capability of fitting well the curvature of the function.

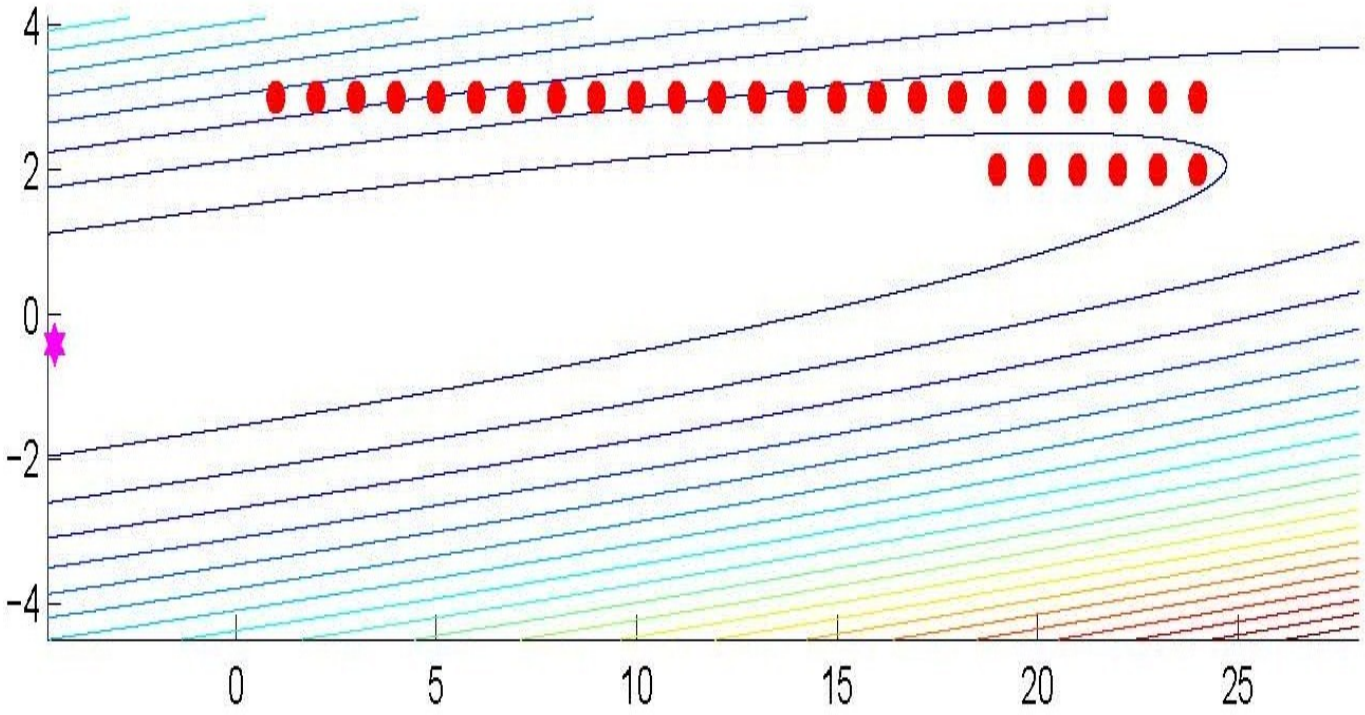
However, the simplices can become arbitrarily flat or needle-shape, which is the reason why it is not possible to establish global convergence to stationary points for the Nelder-Mead algorithm.

A common procedure used by today's practitioners is to restart Nelder-Mead whenever the geometry of the simplex vertices deteriorates.

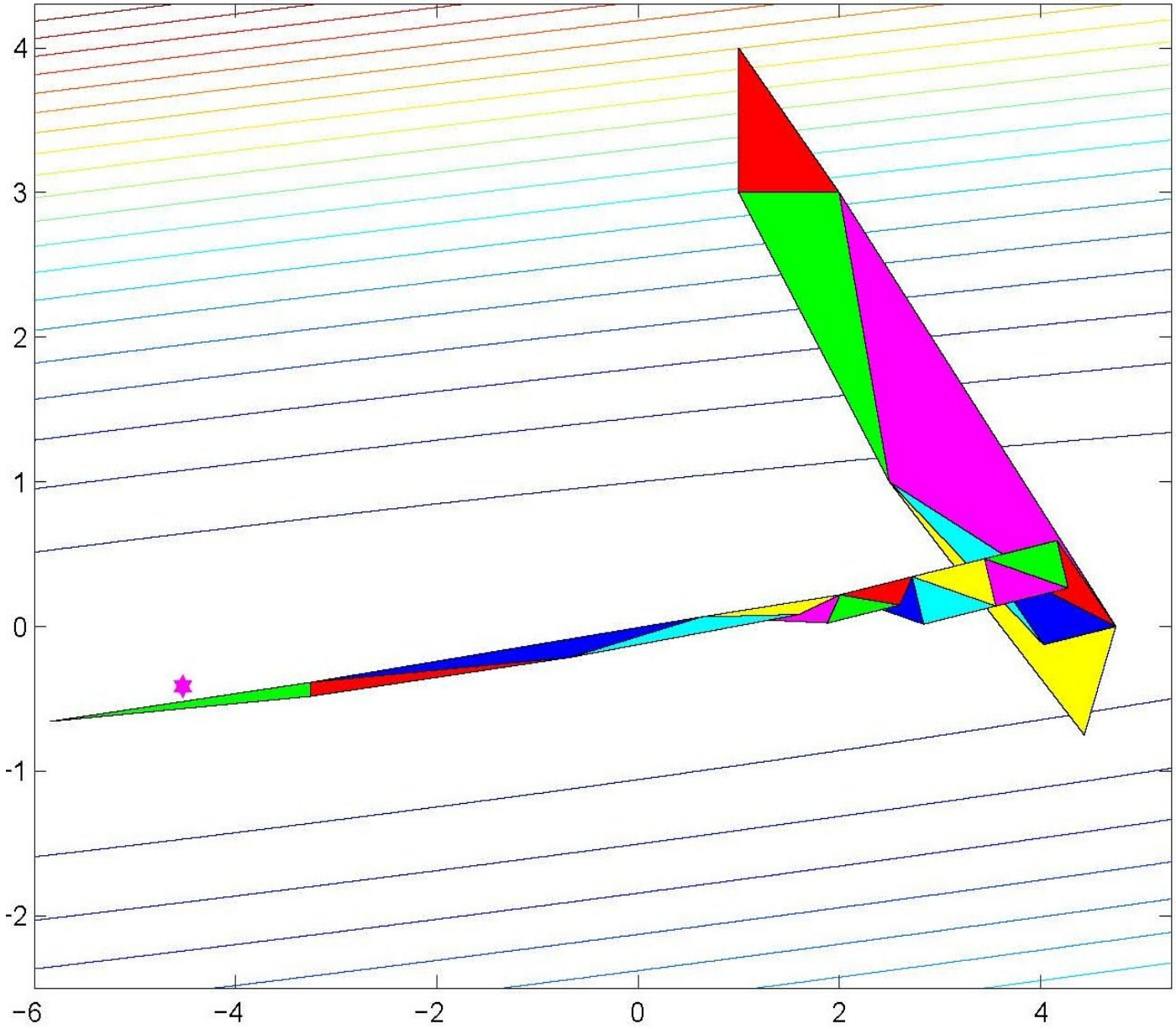
Example



CS on this example



Nelder-Mead on this example



Nelder-Mead properties

Theorem

- *If iteration k performs a non-shrink step (reflection, expansion, or contraction) then*

$$\operatorname{vol}(Y_{k+1}) \;=\; |\delta| \operatorname{vol}(Y_k).$$

- *If iteration k performs a shrink step then*

$$\operatorname{vol}(Y_{k+1}) \;=\; (\gamma^s)^n \operatorname{vol}(Y_k).$$

A simple consequence is that all iterations generate simplices, i.e., $\operatorname{vol}(Y_k) > 0$, for all k (provided that the vertices of Y_0 form a simplex).

Also, isometric reflections ($\delta = 1$) preserve the volume of the simplices, contractions and shrinks are volume decreasing, and expansions are volume increasing.

Nelder-Mead convergence properties

Theorem

Consider the application of the Nelder-Mead method to a function f which is bounded below on $\mathbb{R}^n$.

1. *The sequence $\{f_k^0\}$ is convergent.*
2. *If only a finite number of shrinks occur, then all the $n+1$ sequences $\{f_k^i\}$, $i = 0, \ldots, n$, converge and their limits satisfy $f_*^0 \leq f_*^1 \leq \ldots \leq f_*^n$.*
3. *If only a finite number of non-shrinks occur, then all the simplex vertices converge to a single point.*

The fact that $\{f_k^0\}$ converges does not mean that it converges to the value of f at a stationary point.

Nelder-Mead convergence properties

Theorem

No shrink steps are performed when the Nelder-Mead method is applied to a strictly convex function f.

Lagarias, Reeds, Wright, and Wright [1998]

Theorem

The Nelder-Mead method is globally convergent when $n = 1$.

The proof can be done by reduction to (directional) direct search globalized by simple decrease and integer lattices.

A direct proof is given in: J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, Convergence properties of the Nelder-Mead simplex method in low dimensions, SIAM J. Optim., 9 (1998) 112–147.

McKinnon counter-example

The Nelder-Mead method can fail for $n > 1$ (e.g. due to repeated inside contractions).

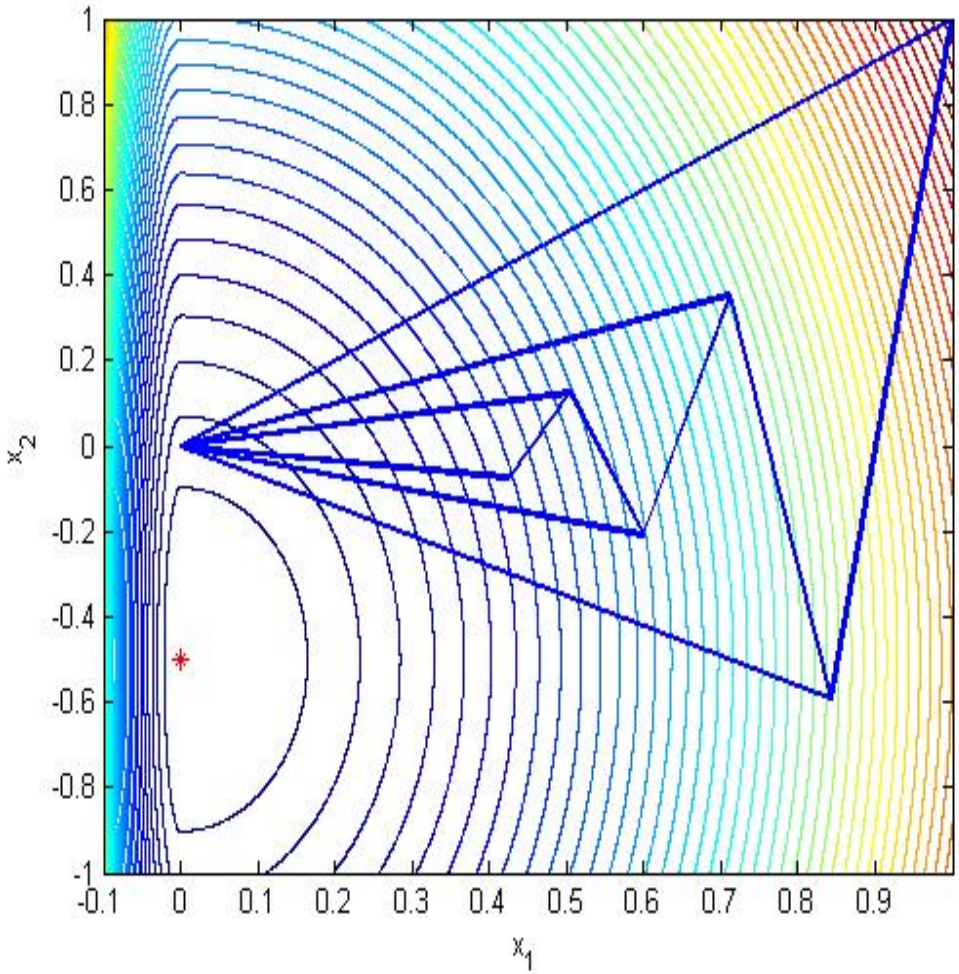
McKinnon counter-example:

$$f(x_1, x_2) = \begin{cases} \theta\phi|x_1|^\tau + x_2 + x_2^2 & \text{if } x_1 \leq 0, \\ \theta x_1^\tau + x_2 + x_2^2 & \text{if } x_1 > 0. \end{cases}$$

The function is strictly convex if $\tau > 1$. It has continuous first derivatives if $\tau > 1$, continuous second derivatives if $\tau > 2$, and continuous third derivatives if $\tau > 3$.

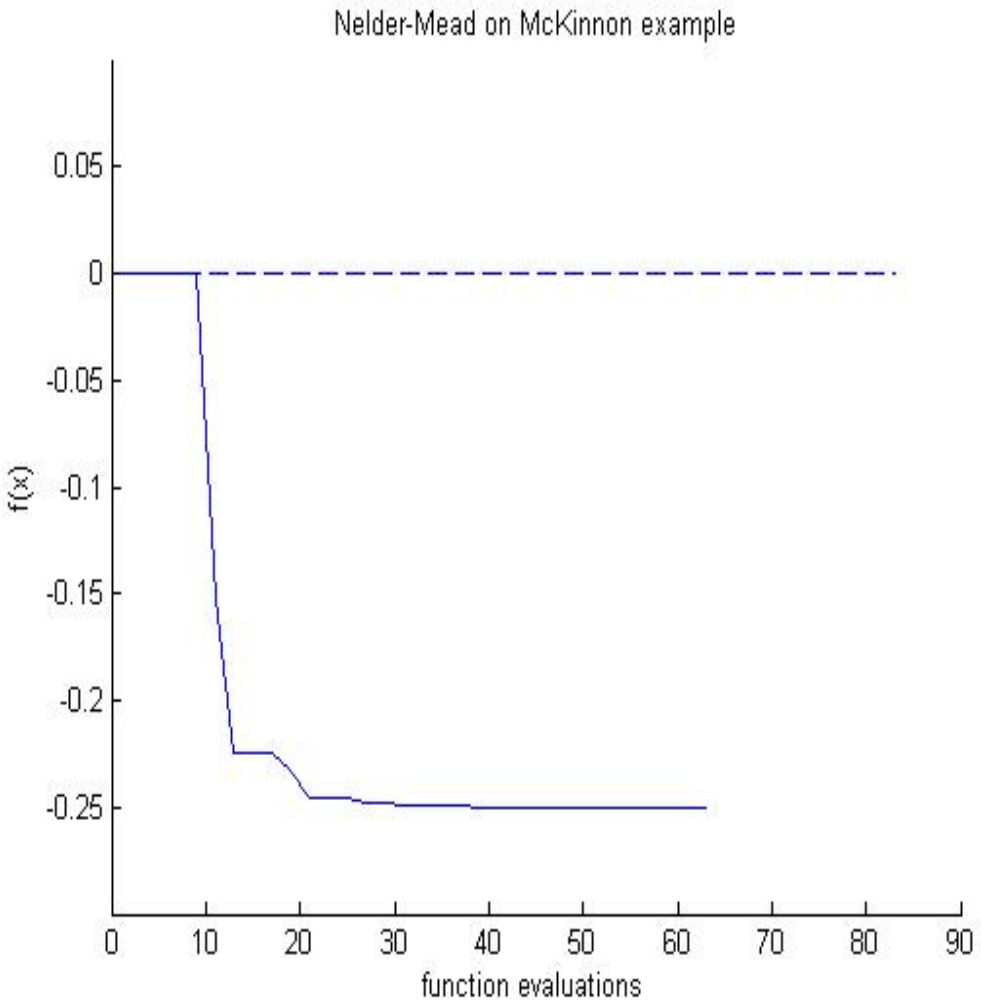
$(0, -1)$ is a descent direction from the origin.

McKinnon counter-example



Contours of the McKinnon function for $\tau = 2$, $\theta = 6$ and repeated focused inside contraction (RFIC).

Nelder-Mead on McKinnon example



Application of the Nelder-Mead method to this function from two different initial simplices.

Nelder-Mead convergence properties

Theorem

The restricted Nelder-Mead method (no expansion steps) is globally convergent for $\mathcal{C}^2$ functions of two variables ($n = 2$) with positive definite Hessian matrices everywhere in their domain.

A proof using symbolic computing was given by: J. C. Lagarias, B. Poonen, and M. H. Wright, Convergence of the restricted Nelder–Mead algorithm in two dimensions, SIAM J. Optim., 22 (2012) 501–532.

Note that the McKinnon counter-example exhibits a point where the Hessian is singular.

Nelder-Mead properties

It is also important to understand how these operations affect the normalized volume of the simplices. When a shrink step occurs one has:

$$\text{von}(Y_{k+1}) \;=\; \text{von}(Y_k).$$

This is also true for isometric reflections ($\delta = 1$) when $n = 2$ or when n is arbitrary but the simplices are equilateral.

Although counterintuitive, isometric reflections do not preserve the normalized volume in general when $n > 2$.

Consider a simplex of vertices $y^0 = (0, 0, 0)$, $y^1 = (1, 1, 0)$, $y^2 = (0, 1, 0)$, and $y^3 = (0, 0, 1)$. The diameter increases from 1.7321 to 1.7951, and the normalized volume decreases from 0.0321 to 0.0288.

Nelder-Mead

We conducted a simple experiment in Matlab to see how often the normalized volume can change.

Number of times where the diameter of a simplex increased and the normalized volume decreased by isometric reflection:

error (power k)	0	2	4	6	8
$\operatorname{diam}(Y^r) > \operatorname{diam}(Y) + 10^{-k}$	0%	24%	26%	26%	26%
$\operatorname{von}(Y^r) < \operatorname{von}(Y) - 10^{-k}$	0%	1%	23%	26%	26%

Experiments made on 10^5 simplices in $\mathbb{R}^3$ with $y^0 = 0$ and remaining vertex components randomly generated in $[-1, 1]$, using Matlab. The notation used is such that $Y = \{y^0, y^1, y^2, y^3\}$ and $Y^r = \{y^0, y^1, y^2, y^r\}$.

Modified Nelder-Mead methods

For Nelder-Mead to globally converge one must:

- Control the internal angles (normalized volume) in all simplex operations but shrinks.

  CAUTION, recall (very counterintuitive): Isometric reflections only preserve internal angles when $n = 2$ or the simplices are equilateral.

  $\longrightarrow$ Need for a back-up polling (e.g., by a safeguard rotation which keeps the normalized volume).

- Impose sufficient decrease instead of simple decrease for accepting new iterates:

$$f(\text{new point}) \leq f(\text{previous point}) - \rho(\text{simplex diameter}).$$

Modified Nelder-Mead method (reflection and safeguard)

Reflect:
Calculate an isometric reflected point y^r (as before). IF

$$\begin{aligned} \operatorname{diam}\left(\{y^0, y^1 \ldots, y^{n-1}\} \cup \{y^r\}\right) &\leq \gamma^e \Delta, \\ \operatorname{von}\left(\{y^0, y^1 \ldots, y^{n-1}\} \cup \{y^r\}\right) &\geq \xi, \end{aligned}$$

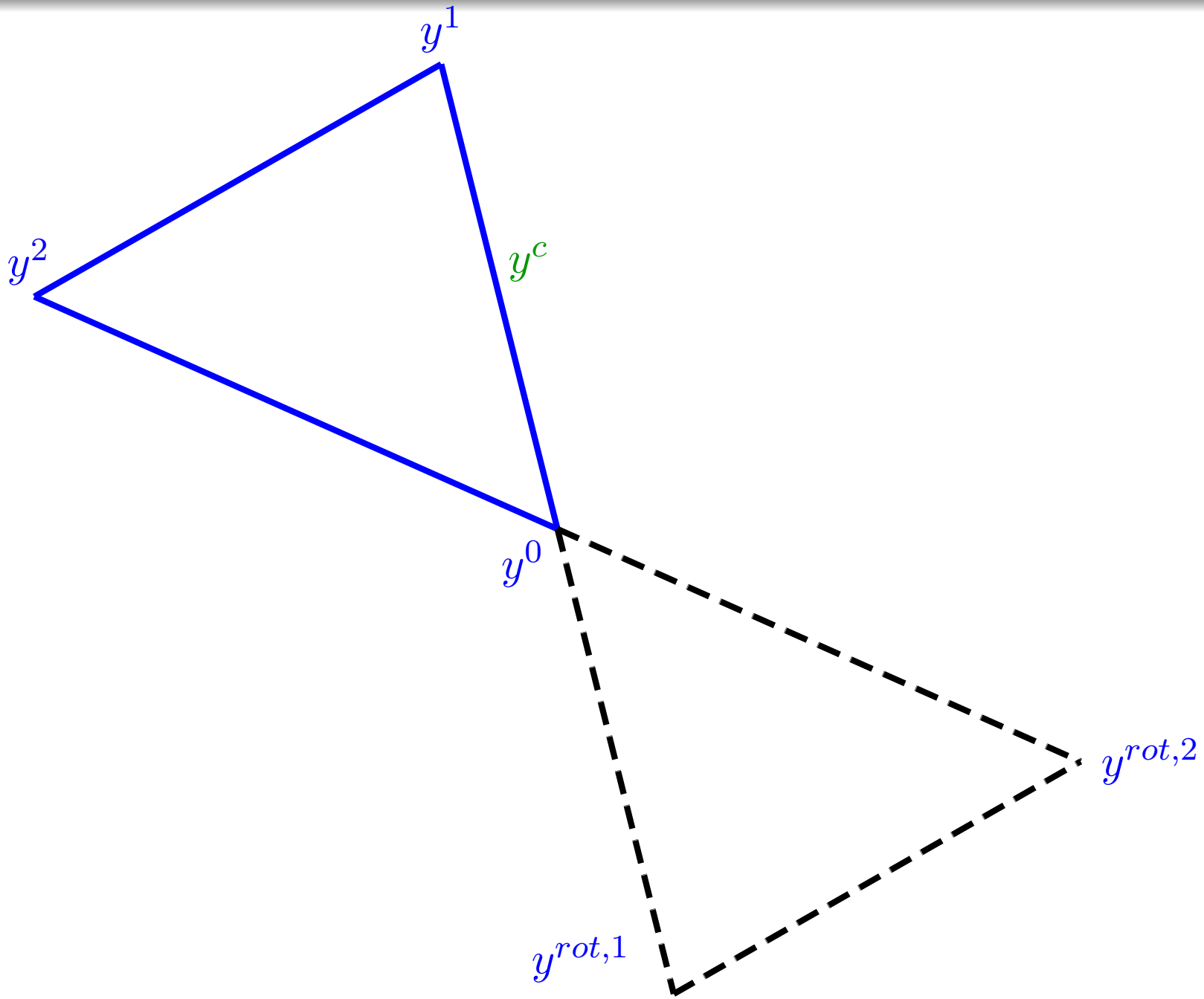
is true then evaluate $f^r = f(y^r)$. If $f^r \leq f^{n-1} - \rho(\Delta)$ then attempt an expansion (and then accept either the reflected or the expanded point). Otherwise attempt a contraction.

Safeguard rotation: OTHERWISE, then rotate the simplex around the best vertex y^0:

$$y^{rot,i} = y^0 - (y^i - y^0), \quad i = 1, \ldots, n.$$

Evaluate $f(y^{rot,i})$, $i = 1, \ldots, n$, and set $f^{rot} = \min\{f(y^{rot,i}) : i = 1, \ldots, n\}$. If $f^{rot} \leq f^0 - \rho(\Delta)$ then terminate the iteration and take the rotated simplex: $Y_{k+1} = \{y^0, y^{rot,1}, \ldots, y^{rot,n}\}$. Otherwise attempt a contraction.

Safeguard rotation



Analysis of modified Nelder-Mead methods

Let $\{Y_k\}$ be the sequence of simplices generated.
Let f be bounded from below and uniformly continuous in $\mathbb{R}^n$.

Theorem

The diameters of the simplices converge to zero:

$$\lim_{k \longrightarrow +\infty} \mathit{diam}(Y_k) \; = \; 0.$$

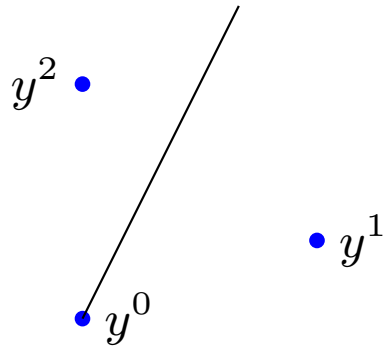
Theorem

If f is continuously differentiable in $\mathbb{R}^n$ and $\{Y_k\}$ lies in a compact set then $\{Y_k\}$ has at least one stationary limit point x_*.

Under additional modifications and conditions, all limit points are stationary.

Presentation outline

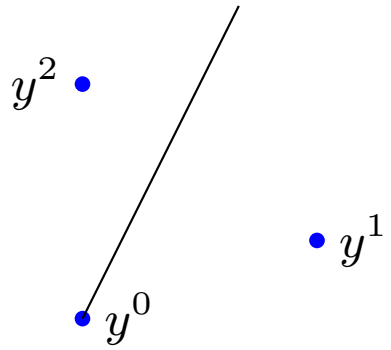
Recapitulation



It is possible to build a simplex gradient:

$$\nabla_s f(y^0) = \begin{bmatrix} y^1 - y^0 & y^2 - y^0 \end{bmatrix}^{-\top} \begin{bmatrix} f(y^1) - f(y^0) \\ f(y^2) - f(y^0) \end{bmatrix}.$$

Recapitulation



It is possible to build a simplex gradient:

$$\nabla_s f(y^0) \;=\; L^{-\top} \delta f(Y).$$

Line-search methods

For instance, the implicit filtering method:

- Computes a simplex gradient (per iteration).
- Improves the negative simplex gradient by applying a quasi-Newton/secant update.
- Performs a line search along the computed direction d_k, to determine $\alpha_k > 0$ such that

$$f(x_k) - f(x_k + \alpha_k d_k) \;\geq\; -\eta \left(\nabla_s f(x_k)\right)^\top (\alpha_k d_k)\,,$$

where $\eta \in (0, 1)$ and, e.g., $d_k = -\nabla_s f(x_k)$.

Implicit filtering method

The function evaluations can be computed in parallel.

It can use regression with more than $n+1$ points for the simplex gradient. For instance:

$$Y_k = \{x_k,\ x_k + \Delta_k d, d \in D_{\oplus}\}.$$

The noise is filtered:

- by the simplex gradient calculation (especially when using regression);
- by not performing an accurate line search.

Presentation outline

Polynomial regression

The most natural approach to handling noisy data is to replace the interpolation of the objective function by least-squares regression. In this case, the interpolation conditions

$$M(\phi, Y)\alpha \;=\; f(Y),$$

are solved in the least-squares sense, meaning that a solution α is found such that $\|M(\phi, Y)\alpha - f(Y)\|^2$ is minimized.

If function evaluations are relatively cheap, but still noisy, then it may be effective to sample the objective function at more local sample points (i.e., at points closer to the center of the model) than it would be necessary for complete interpolation.

In that case, the number of rows in the matrix $M(\phi, Y)$ is larger than the number of columns and the interpolation system is overdetermined:

Stochastic polynomial regression

As the number of sample points increases, the least-squares regression solution to the noisy problem converges to the least-squares regression of the underlying true function.

Specifically, let the noisy function

$$f(x) = f_{smooth}(x) + \varepsilon,$$

where $f_{smooth}(x)$ is the true, mostly likely smooth function which we are trying to optimize and ε is a random variable, independent of x and drawn from some distribution with zero mean.

Stochastic polynomial regression

Assume a polynomial basis ϕ is fixed and consider a sample set Y^p whose size $|Y^p| = p + 1$ is variable.

Let the random vector α^p be the least-squares solution to the system

$$M(\phi, Y^p)\alpha \;=\; f(Y^p)$$

and the real vector α^p_{smooth} be the least-squares solution to the system

$$M(\phi, Y^p)\alpha \;=\; f_{smooth}(Y^p).$$

Consistency result for regression

Now, assume that the size of the sample set Y^p is going to infinity and that the following condition holds:

$$\liminf_{p \to +\infty} \lambda_{\min} \left(\frac{1}{p+1} M(\phi, Y^p)^\top M(\phi, Y^p) \right) \; > \; 0,$$

This means that the data is sampled in a uniformly well-poised manner.

Theorem

If this condition holds and the sequence $\{Y^p\}$ is bounded then

$$\mathbf{E}(\alpha^p) \; = \; \alpha^p_{smooth}, \quad \forall p \geq q,$$

and

$$\lim_{p \to +\infty} \mathbf{Var}(\alpha^p - \alpha^p_{smooth}) \; = \; 0.$$

Deterministic polynomial regression

For a fixed number p of points, larger than q (AS IN THE DETERMINED CASE):

- The least-squares regression polynomial and the least-squares Lagrange polynomials enjoy the same properties as before.
- Λ–poisedness has the same FIRST equivalent redefinition as before.

However, the SECOND alternative definition which defines the value of Lagrange polynomials via a ratio of volumes of the sets Y and $Y_i(x) = Y \setminus \{y^i\} \cup \{x\}$ does not seem to extend in an straightforward manner to the regression case.

Λ–poisedness might not suffice

So, we could define Λ–poisedness as in the determined case, but that when p is allowed to grow arbitrarily large, we need to have a uniform bound on $p\Lambda$:

- for the error bounds to be useful (as we will see later),
- for the consistency of the least-squares regression model to hold (so that the smallest eigenvalue of the matrix $M(\phi, Y^p)^\top M(\phi, Y^p)$ increases with a rate of at least p).

Thus, the definition of Λ–poisedness needs to be strengthened to take the number of sample points into consideration when it becomes too large.

Strong Λ–poisedness

Definition

The definition of strong Λ–poisedness basically says that as $p \to \infty$ the Λ–poisedness constant should decrease with the rate of $1/p$.

Theorem

To have a strongly Λ–poised set of p points it is sufficient to construct such a set by combining p/q Λ–poised interpolation sets of q points.

Error bounds for polynomial regression

The errors bounds for least-squares polynomial regression are as in the determined case, but where κ_{eh}, κ_{eg}, and κ_{ef} are given by:

$$\begin{aligned}
\kappa_{eh} &= \nu_2 + \sqrt{2}\bar{p}^{\frac{1}{2}}\nu_2/2\|\hat{\Sigma}^{-1}\|, \\
\kappa_{eg} &= \nu_2 + (n^{\frac{1}{2}} + \sqrt{2}\bar{p}^{\frac{1}{2}})/2\nu_2\|\hat{\Sigma}^{-1}\|, \\
\kappa_{ef} &= \nu_2/2 + (1/2 + n^{\frac{1}{2}}/2 + \sqrt{2}\bar{p}^{\frac{1}{2}}/4)\nu_2\|\hat{\Sigma}^{-1}\|,
\end{aligned}$$

where $\hat{\Sigma}$ stores the singular values of $M(\bar{\phi}, Y_{scaled})$.

Note that if the set Y is strongly Λ–poised, then $p^{\frac{1}{2}}\|\hat{\Sigma}^{-1}\|$ is uniformly bounded, independently of p and Y.

There is an ERROR in the statement of the errors bounds for quadratic regression in the IDFO book. See a correct derivation at `http://www.mat.uc.pt/~lnv/idfo`

Regularized regression

It may be useful to use regularized regression instead of interpolation or regression to introduce smoothness.

Assume that we are considering complete quadratic models with $p \geq q$.

Let us pick a regularization parameter ρ and consider the following problem

$$\min_{\alpha} \quad \rho \|\alpha_Q\|^2 + \|M(\phi, Y)\alpha - f(Y)\|^2.$$

We know that if $\rho \to 0$ then we recover the least-squares regression model (fully quadratic).

If $\rho \to +\infty$ then, in this case, we recover linear least-squares regression (fully linear).

Regularized regression

To preserve the second-order approximation we need to limit the size of ρ.

In fact, it is possible to derive from the Taylor-like error bounds for least-squares regression models that, for small enough values of ρ, the solutions to the regularized regression problems also provide fully quadratic models.

Presentation outline

Surrogate models

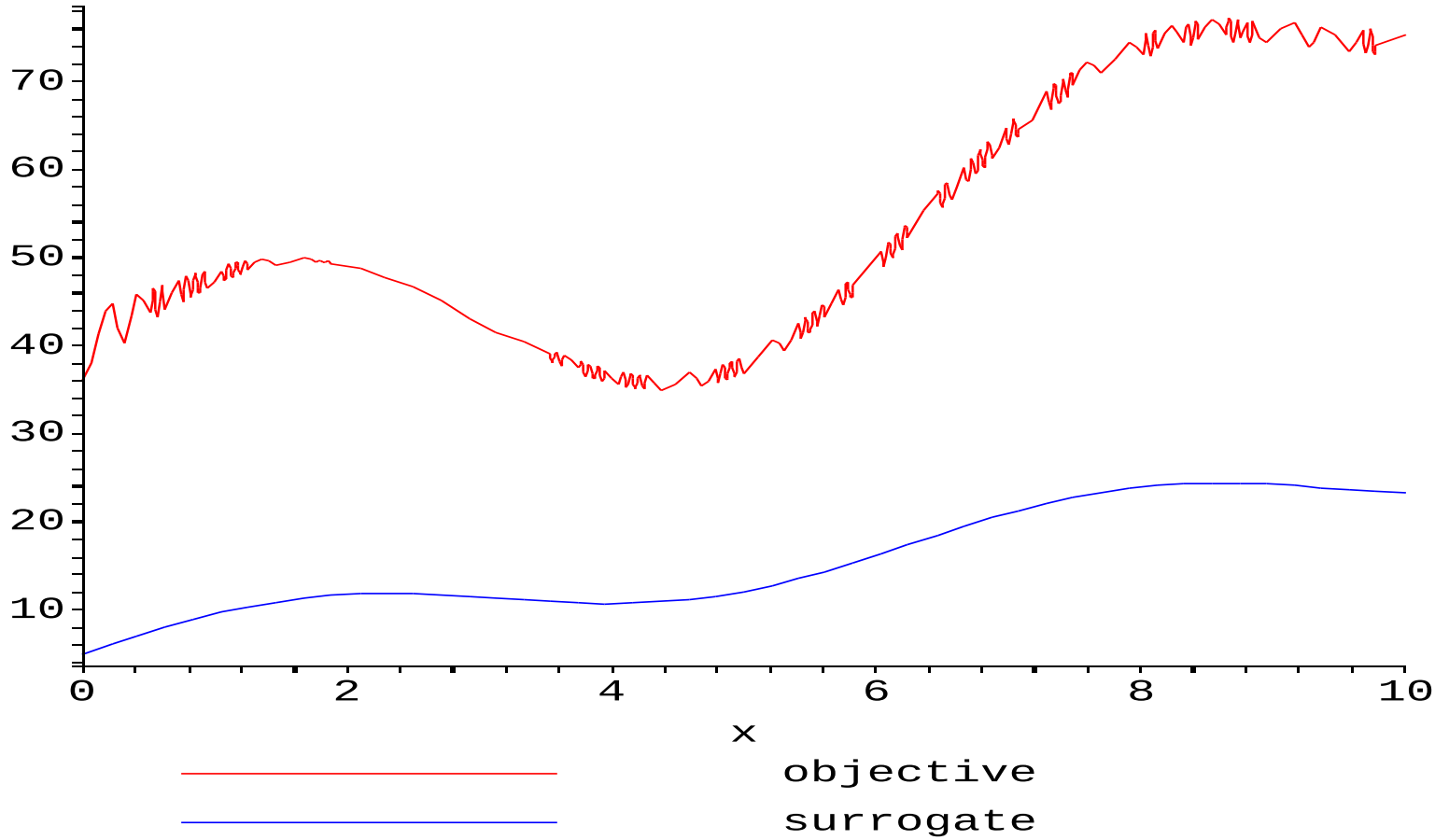
In engineering modeling it is frequently the case that the function to be optimized is expensive to evaluate.

The problem to be (approximately) solved may require extensive simulation of systems of differential equations, possibly associated with different disciplines, or may involve other time consuming numerical computations.

Engineers frequently consider models of the true function or true model, and then consider the model of the model to be a surrogate model.

An example in $\mathbb{R}^1$ of an excellent surrogate

A surrogate s_f is not necessarily a good approximation of the truth f.



Surrogate models

A surrogate model can be used only for modeling and analysis, in order to gain insight about problem features and behavior without many expensive evaluations.

More interestingly for our context, a surrogate model can take the place of the true function for purposes of optimization.

A surrogate model is typically less accurate or has less quality than the true function, but is cheaper to evaluate or consumes fewer computing resources.

Several evaluations of the surrogate model can still be less expensive than one evaluation of the true function (true model).

Types of surrogate models

We will classify surrogate models as either functional or physical.

Physical surrogate models

Definition

By physical surrogate models we mean surrogate models built from a physical or numerical simplification of the true problem functions.

One can think, for instance, of a coarser mesh discretization in numerical PDEs or of a linearization of term sources or equations, as ways of obtaining physical surrogate models.

Physical surrogate models are in many circumstances based on some knowledge of the physical system or phenomena being modeled, and thus any particular such model is difficult to exploit across different problems.

Physical surrogate models

There are other procedures not directly based on simplified physics to build physical surrogate models from true functions with physical meaning.

These procedures typically involve some form of correction, scaling or alignment of the available surrogate model using information (function values or gradients) of the true functions.

An example is the space-mapping method.

If derivatives or approximations to derivatives of the true function are not used in physical surrogate models these might not exhibit the trends of the true function.

In such cases, one may have to resort to optimizing the possibly expensive true function without derivatives (which may be computationally problematic due to the course of dimensionality).

Functional surrogate models

Definition

Functional surrogate models are algebraic representations of the true problem functions.

One can say that functional models are typically based on the following components: a class of basis functions, a procedure for sampling the true functions, a regression or fitting criterion, and some deterministic or stochastic mathematical technique to combine them all.

Functional surrogate models have a mathematical nature different from the true, original functions. The knowledge of truth is revealed implicitly in the values of their coefficients.

Functional surrogate models are not (at least entirely) specific to a class of problems.

Functional surrogate models

Among the most common classes of basis functions used to build functional surrogate models are: low-order polynomials (seen before!!!), radial basis functions, splines, and wavelets.

The most widely used criterion for fitting the data is based on least-squares regression.

A rigorous optimization framework to handle surrogates

A simple idea is explore the flexibility of the search step of directional direct-search algorithms.

Suppose that besides having ways of building such initial surrogate model $sm(x) = sm_0(x)$, one also has ways of possibly improving or re-calibrating the surrogate model along the course of the optimization process.

One can think of a process where a sequence of surrogate models $\{sm_k(x)\}$ is built by successive re-calibrations.

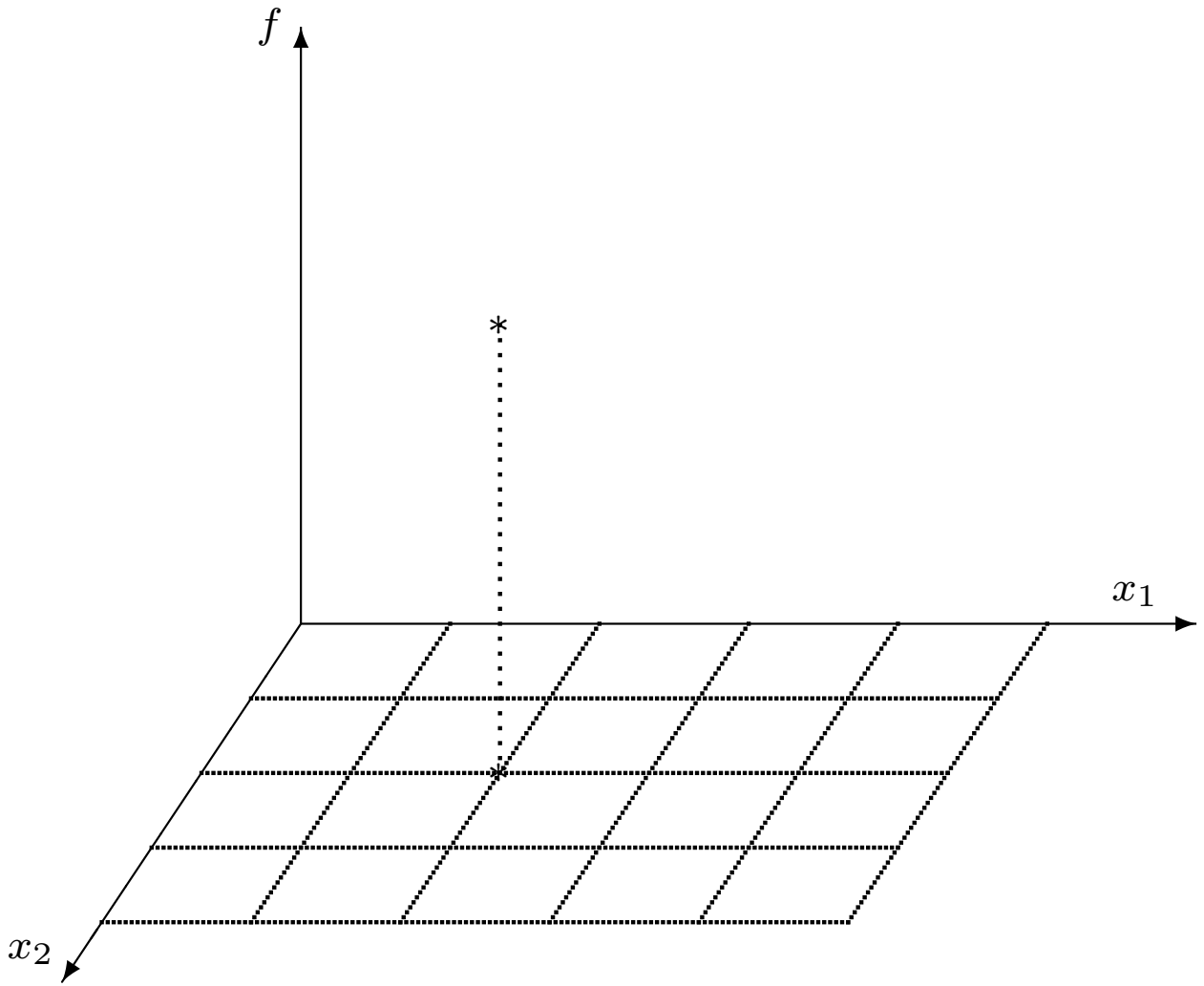
Every time f is evaluated at new points, those values can be used to re-calibrate and hence improve the quality of the surrogate models.

A rigorous optimization framework to handle surrogates

Search step: Try to compute a point x with $f(x) < f(x_k)$ (or with $f(x) < f(x_k) - \rho(\alpha_k)$) by evaluating the function f a finite number of times, by means, for instance, of one of the two alternatives:

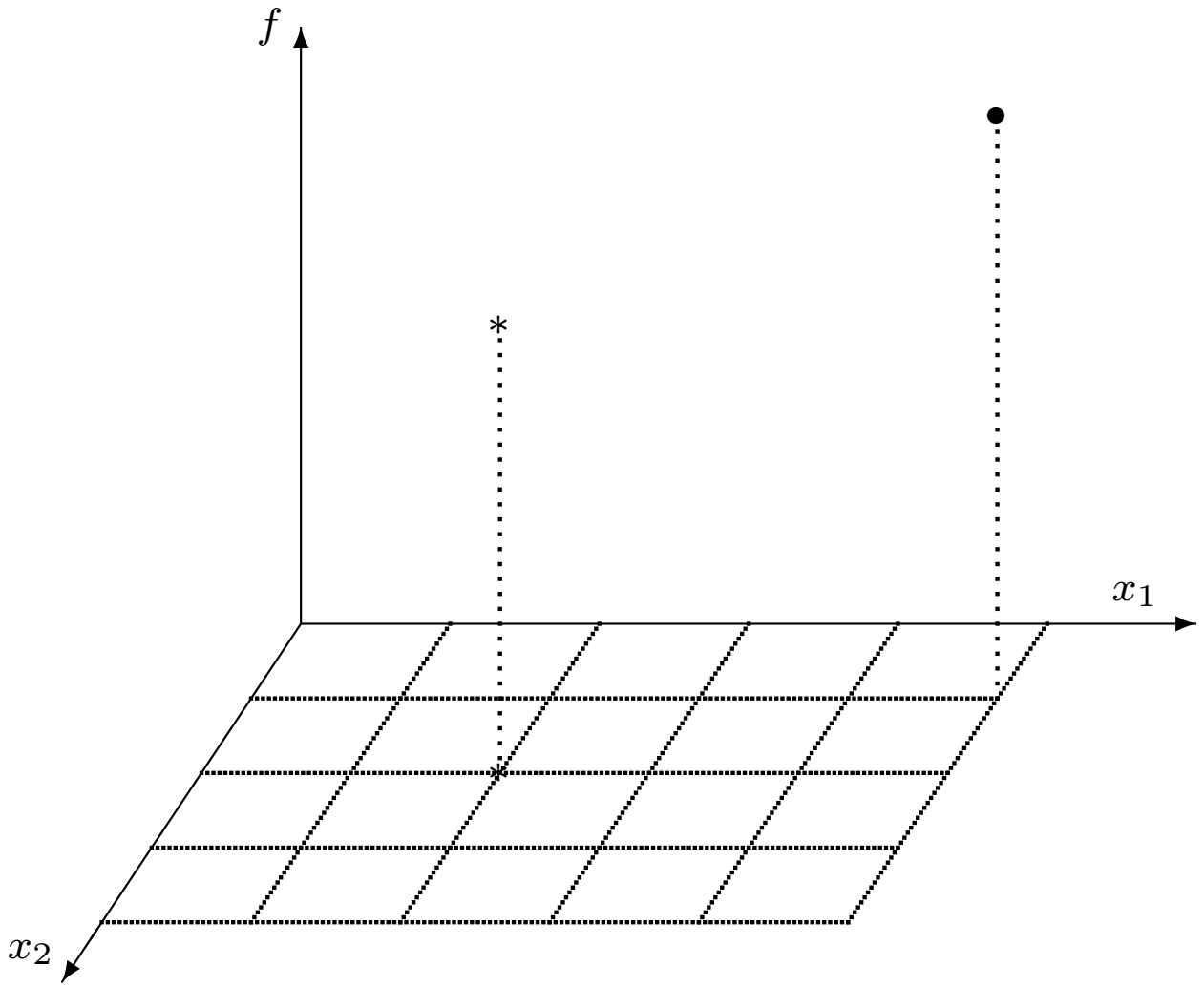
1. Evaluate $sm_k(\cdot)$ on a set of points $Y_k = \{y_k^1, \ldots, y_k^{p_k}\}$. Order Y_k by increasing values: $sm(y_k^1) \leq \cdots \leq sm(y_k^{p_k})$. Start evaluating f in Y_k along this order.
2. Apply some finite optimization process to minimize $sm_k(\cdot)$ possibly in some feasible set. Let y_k be the approximated minimizer. Evaluate $f(y_k)$.

Example (surrogate management)



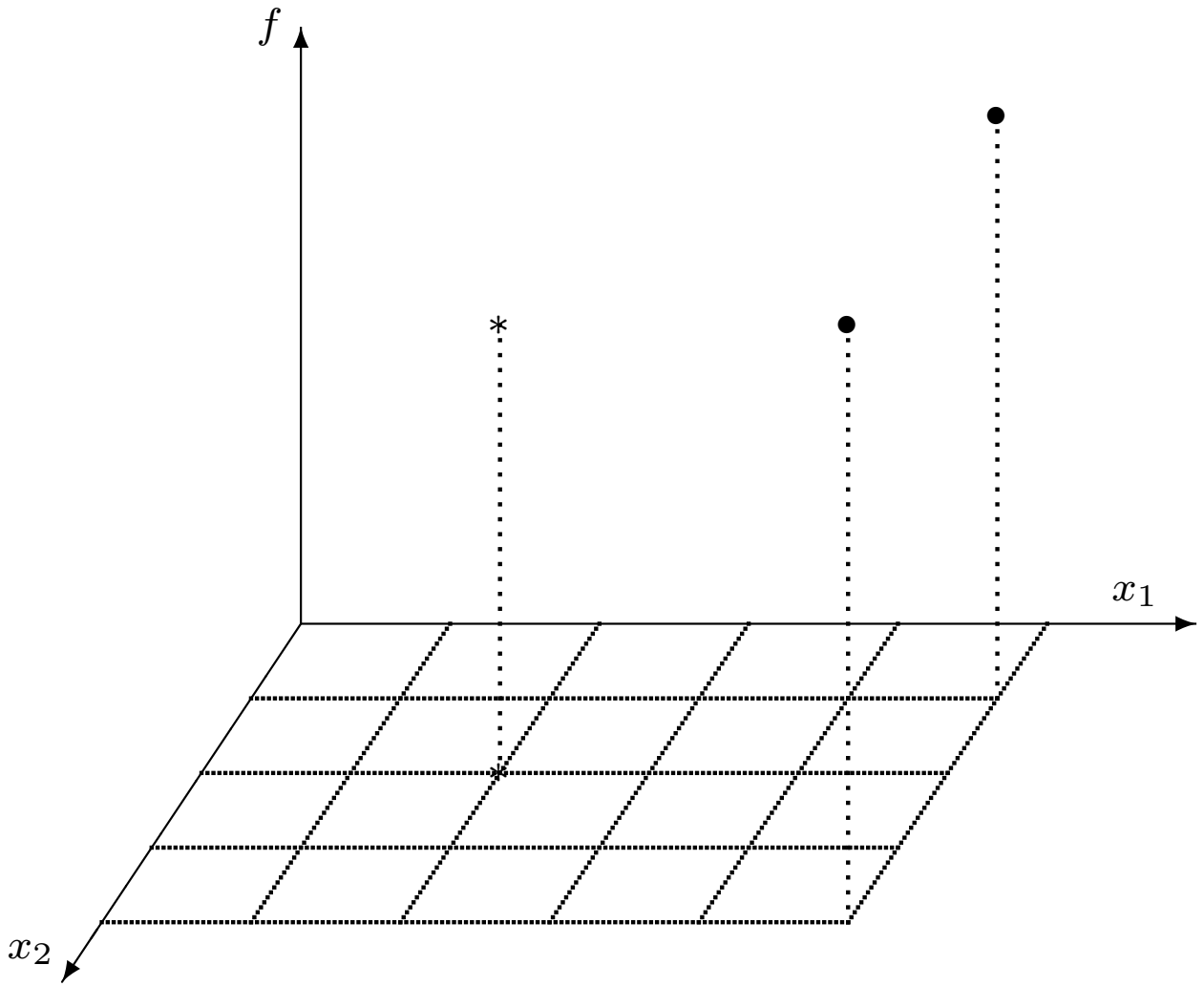
$*$ is the current iterate

Example (surrogate management)



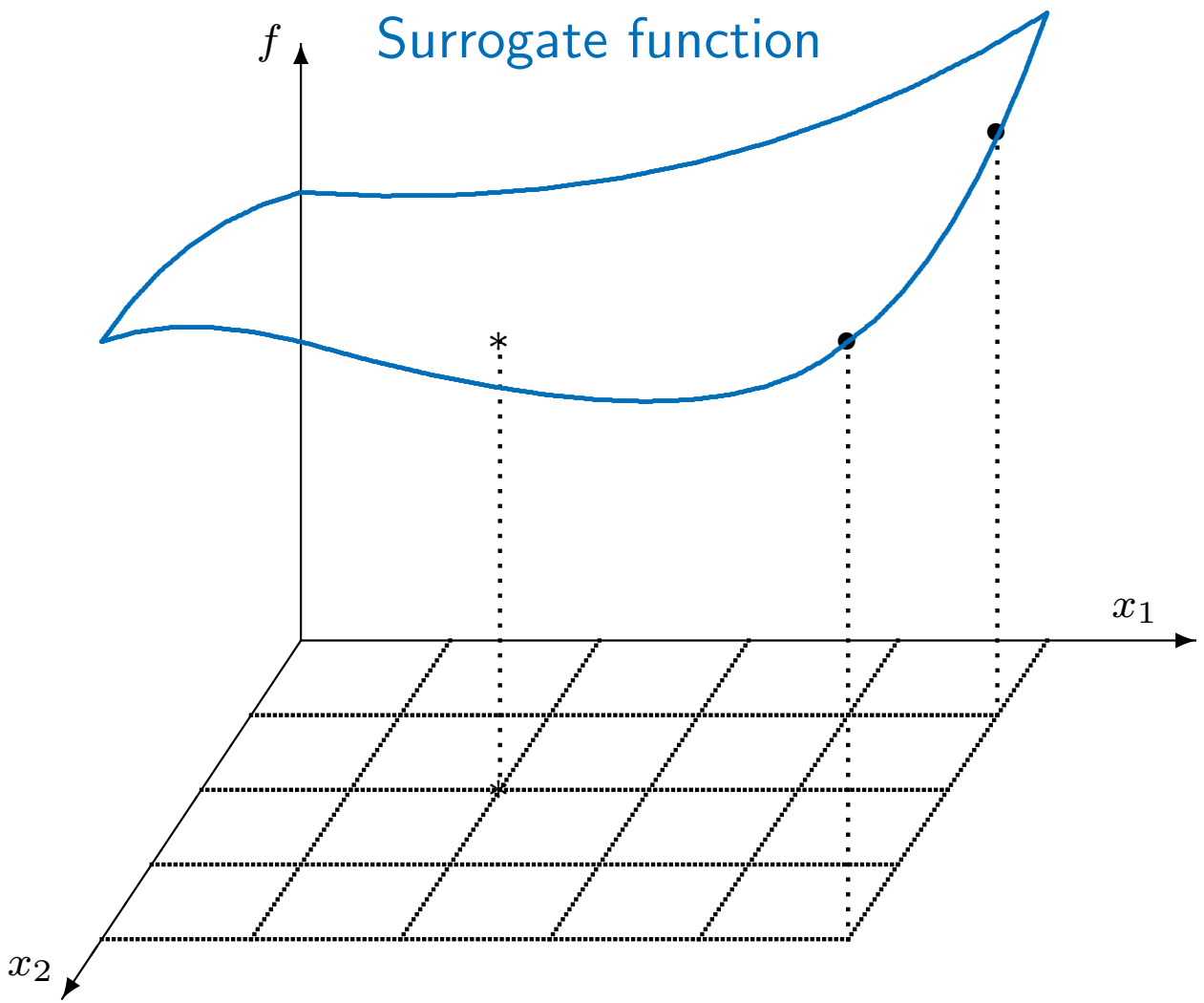
$*$ is the current iterate

Example (surrogate management)



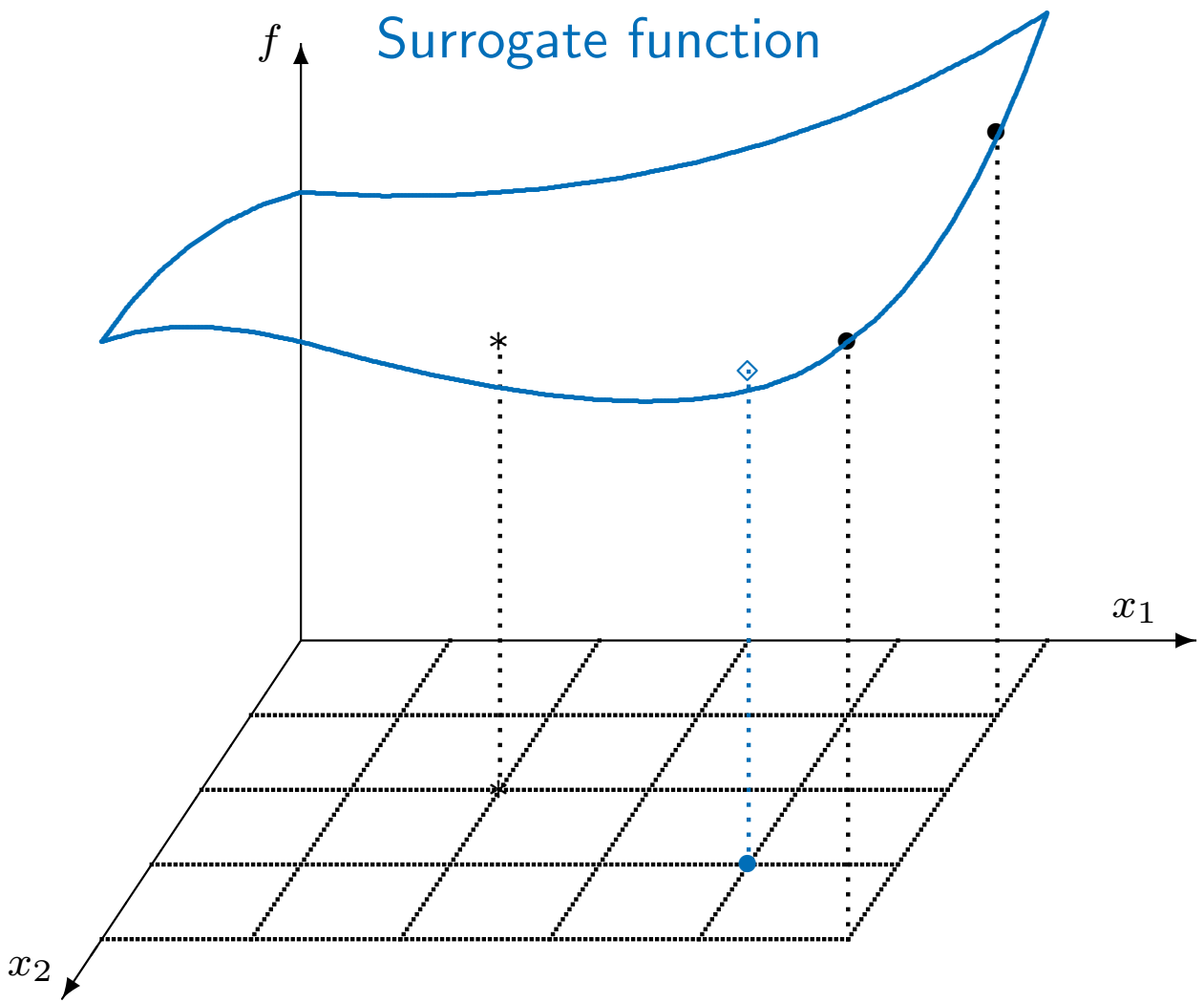
$*$ is the current iterate

Example (surrogate management)



$*$ is the current iterate

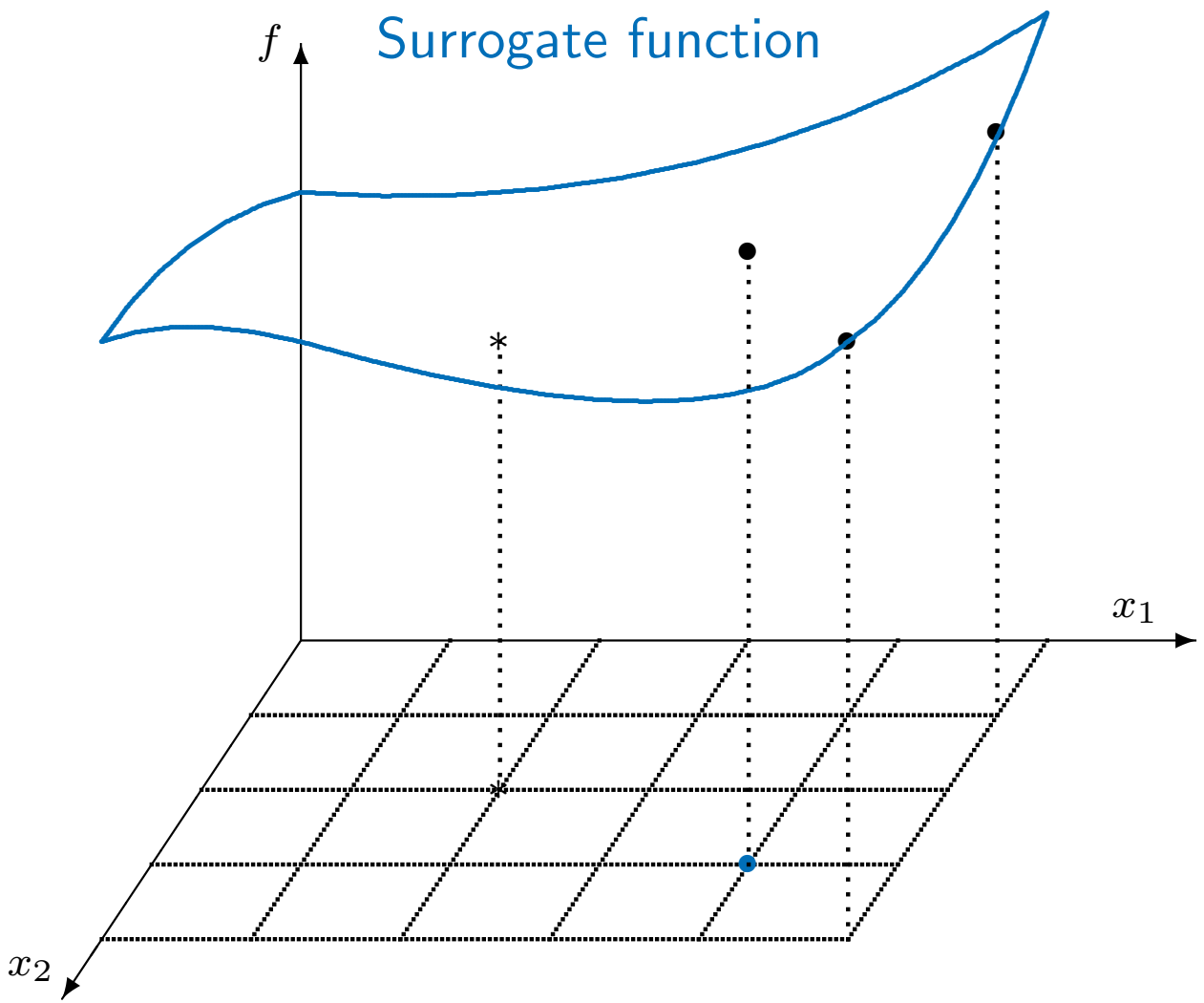
Example (surrogate management)



$*$ is the current iterate

Trial point produced using the surrogate

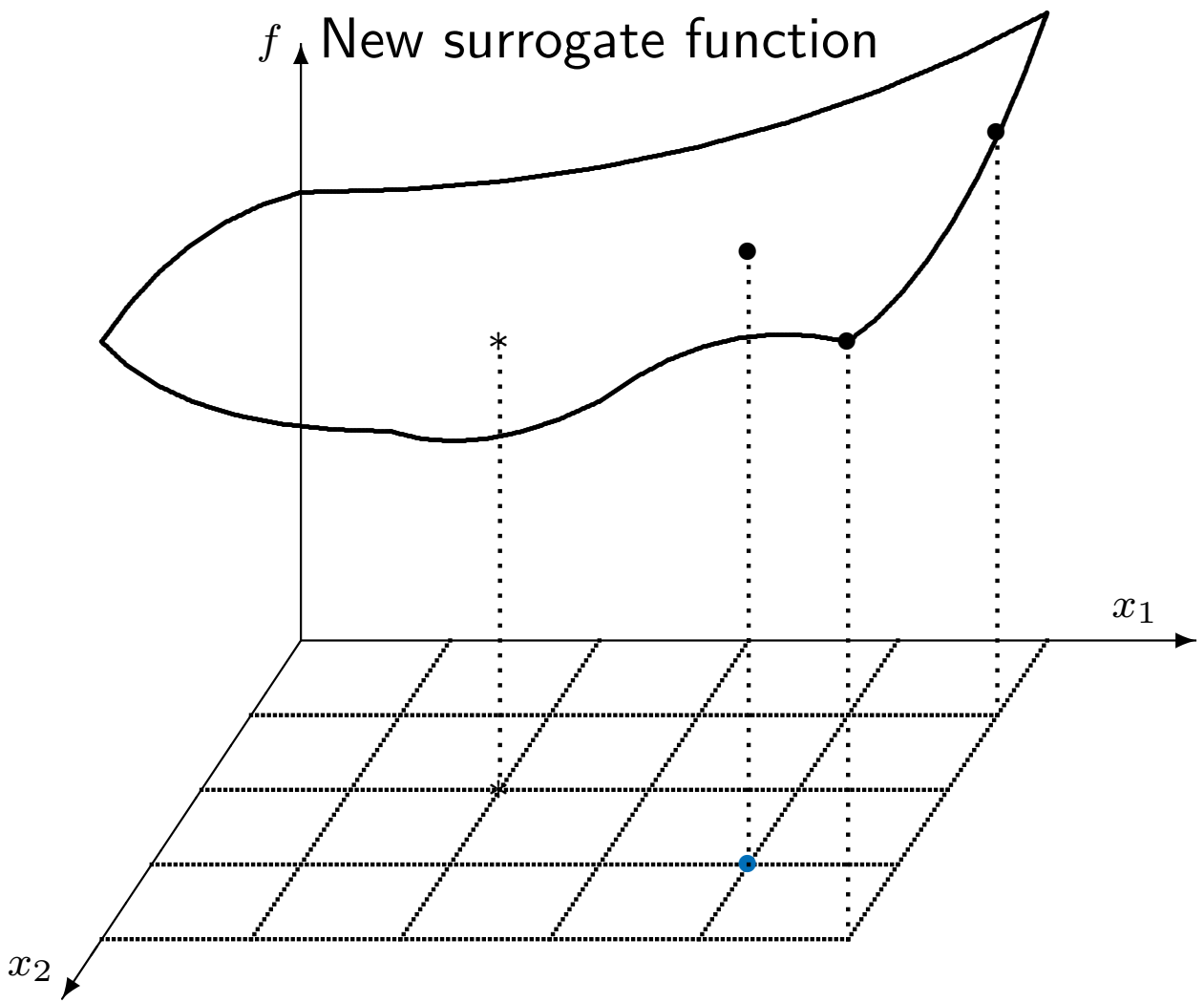
Example (surrogate management)



$*$ is the current iterate

f is evaluated at the trial point

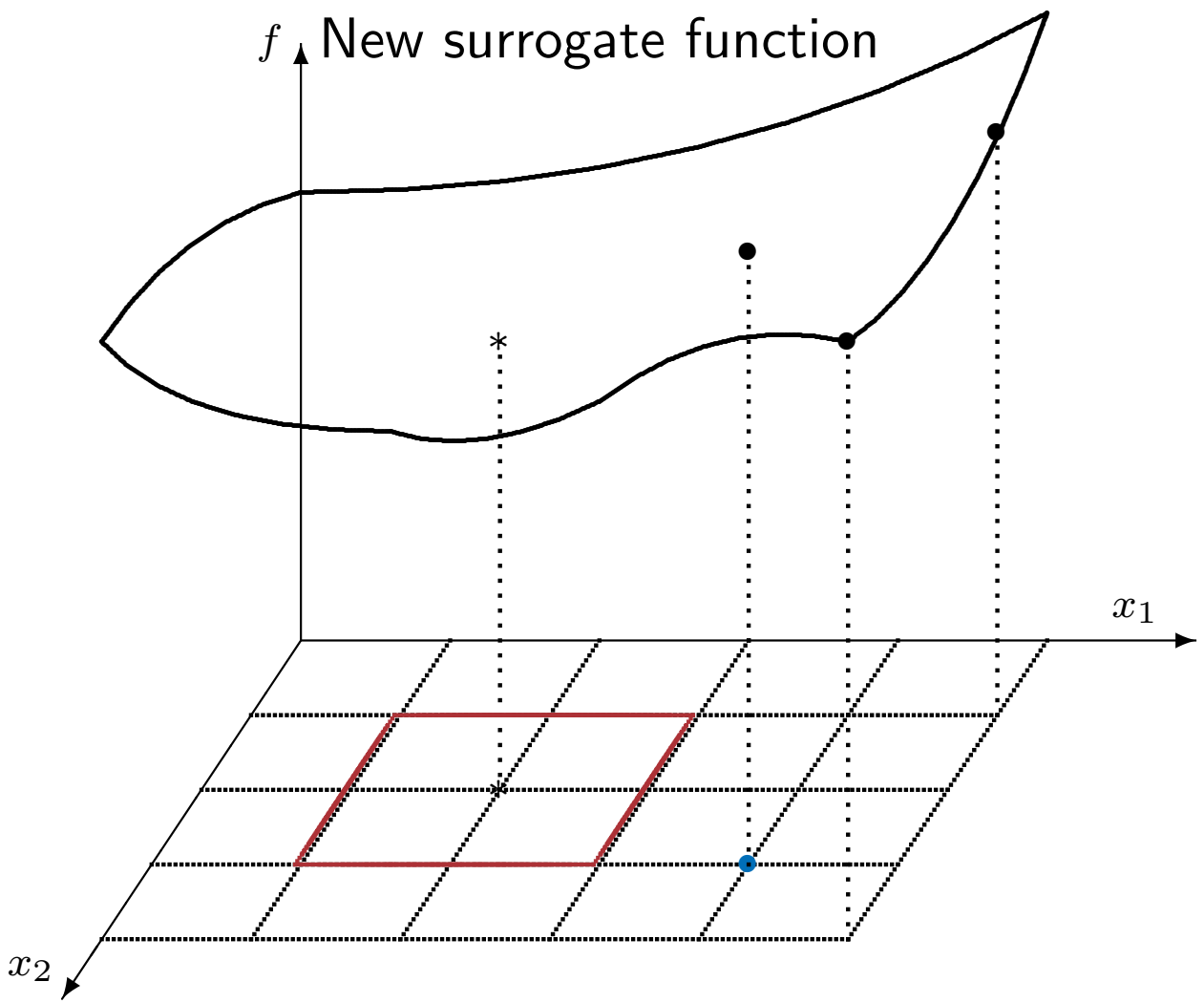
Example (surrogate management)



$*$ is the current iterate

the surrogate function is updated

Example (surrogate management)

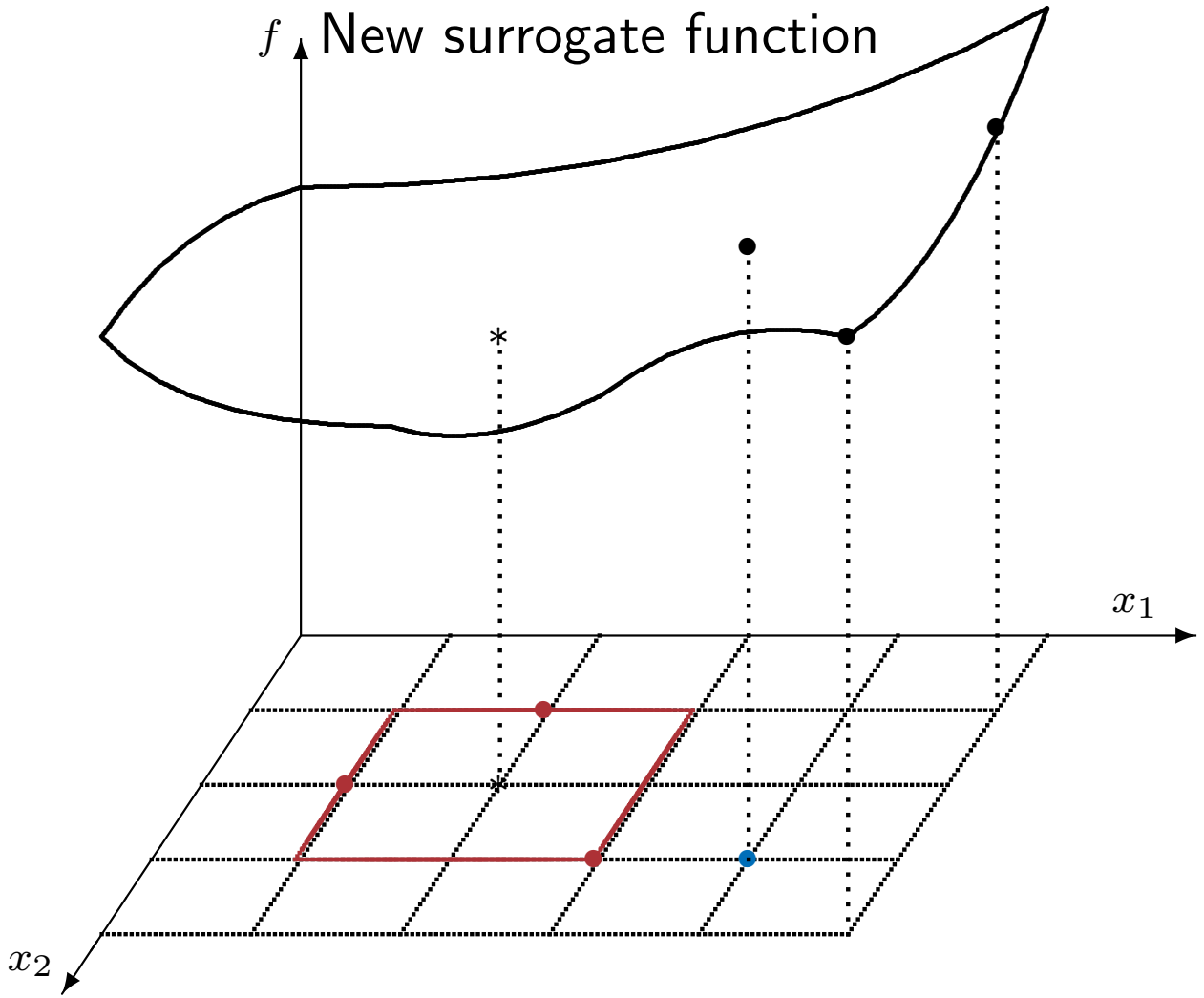


$*$ is the current iterate

the surrogate function is updated

Poll around (order based on surrogate)

Example (surrogate management)

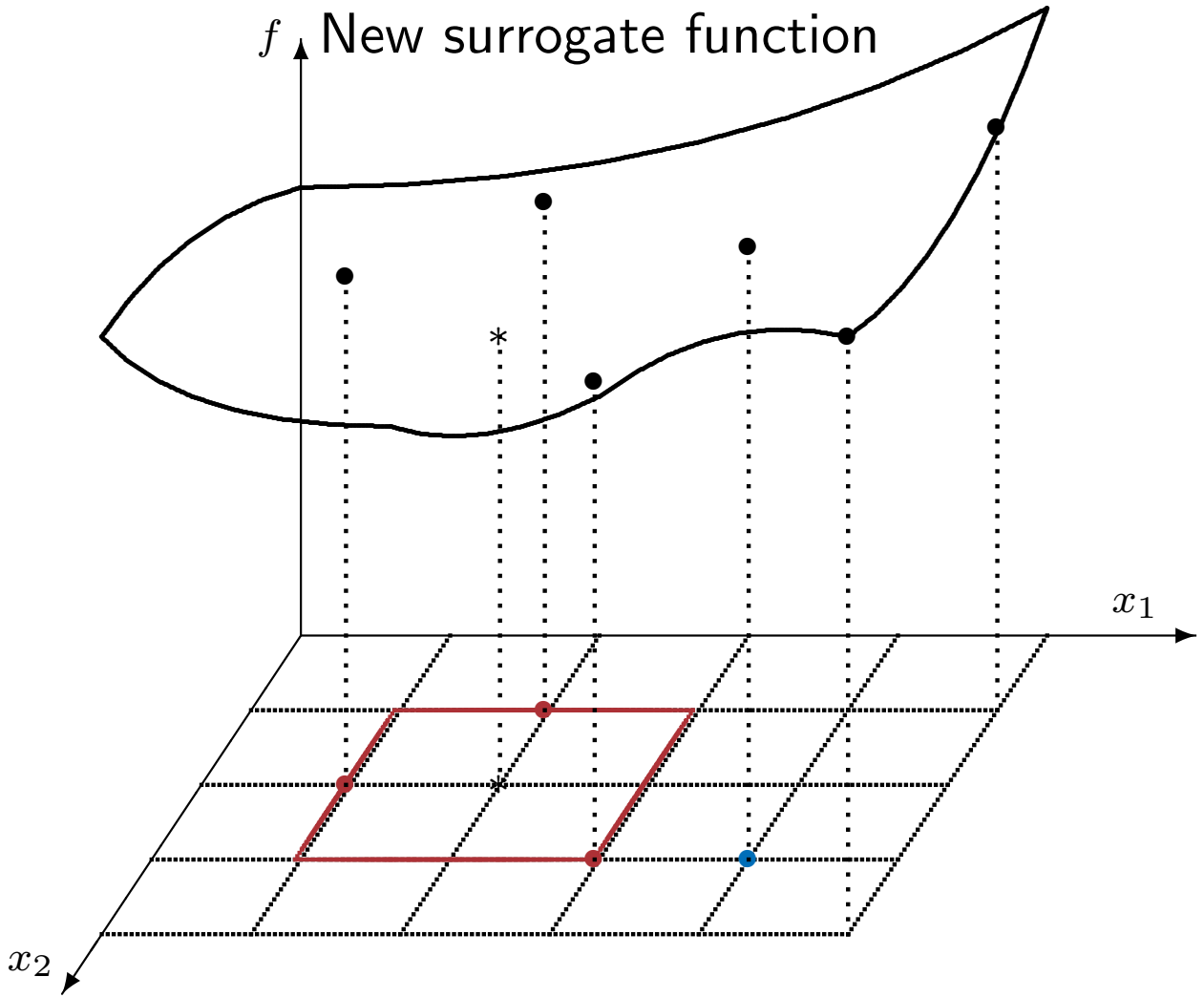


$*$ is the current iterate

the surrogate function is updated

Poll around (order based on surrogate)

Example (surrogate management)



$*$ is the current iterate

the surrogate function is updated

Poll around (order based on surrogate)

Analysis of the surrogate framework

(A) Under simple decrease ($\rho = 0$), the search step must evaluate points only in the mesh M_k.

In the second possibility, we could project y_k onto the mesh M_k. Such projection could be computationally expensive for some choices of positive bases but it is a trivial task for others.

It is trivial to project onto M_k if we choose $D = \{D_\oplus\}$.

Note that there is no guarantee that the projected point, say x, provides a decrease of the form $f(x) < f(x_k)$ even when y_k does.

If $f(y_k) < f(x_k) \leq f(x)$ then either the iteration is considered unsuccessful or y_k is taken and the geometrical considerations needed for convergence are ignored.

Analysis of the surrogate framework

(B) Under sufficient decrease, the search step would only accept a point x when $f(x) < f(x_k) - \rho(\alpha_k)$, where $\rho(\cdot)$ is a forcing function.

Handling nonlinear models in trust-region methods

A natural question that arises is what happens when the trust-region model is not quadratic.

Fraction of Cauchy decrease (FCD) is a very mild condition, but still it could happen that an approximated solution of a trust-region subproblem defined by a nonlinear model does not satisfy it, when $\kappa_{fcd} \in (0, 1)$ is fixed across all iterations.

One way to achieve this is to use a backtracking algorithm along the model steepest descent direction, where the backtracking is from the boundary of the trust region.

Finally, we point out that something similar can be done in the computation of a step satisfying a fraction of the eigenstep decrease (FED) for non-quadratic trust-region models.

Radial basis functions (RBFs)

One of the attractive features of RBFs is their ability to model well the curvature of the underlying function.

Another key feature is that the coefficient matrix defined by the interpolation conditions is nonsingular under relatively weak conditions.

Radial basis functions (RBFs)

In order to interpolate a function f whose values on a set $Y = \{y^0, \ldots, y^p\} \subset \mathbb{R}^n$ are known, one can use a radial basis functional surrogate model of the form

$$sm(x) = \sum_{i=0}^{p} \lambda_i \phi(\|x - y^i\|),$$

where $\phi : \mathbb{R}_+ \to \mathbb{R}$ and $\lambda_0, \ldots, \lambda_p \in \mathbb{R}$.

The term radial basis stems from the fact that $\phi(\|x\|)$ is constant on any sphere centered at the origin in $\mathbb{R}^n$.

For $sm(x)$ to be twice continuously diff., $\phi(x)$ must be both twice continuously diff. and have a derivative that vanishes at the origin.

Radial basis functions (examples)

Some of the most popular (twice continuously diff.) RBFs are the following:

- cubic $\phi(r) = r^3$,
- Gaussian $\phi(r) = e^{-\frac{r^2}{\rho^2}}$,
- multiquadric of the form $\phi(r) = (r^2 + \rho^2)^{\frac{3}{2}}$,
- inverse multiquadric of the form $\phi(r) = (r^2 + \rho^2)^{-\frac{1}{2}}$,

where ρ^2 is any positive constant.

Radial basis functions (RBFs)

In many applications it is desirable that the linear space spanned by the basis functions include constant functions.

Similarly, if one wants to include linear functions (and/or other suitable low-order polynomial functions) one adds a low-order polynomial tail of degree $d-1$ that one can express as $\sum_{j=0}^{q} \gamma_j p_j(x)$, where $p_j \in \mathcal{P}_n^{d-1}$, $j = 0, \ldots, q$, are the basis functions for the polynomial and $\gamma_0, \ldots, \gamma_q \in \mathbb{R}$.

The new surrogate model is now of the form

$$sm(x) \;=\; \sum_{i=0}^{p} \lambda_i \phi(\|x - y^i\|) + \sum_{j=0}^{q} \gamma_j p_j(x).$$

Furthermore, the coefficients λ's are required to satisfy

$$\sum_{i=0}^{p} \lambda_i p_j(y^i) \;=\; 0, \quad j = 0, \ldots, q.$$

RBFs (properties)

These conditions, plus the interpolation conditions $sm(y^i) = f(y^i)$, $i = 0, \ldots, p$, give the linear system

$$\begin{bmatrix} \Phi & P \\ P^\top & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \gamma \end{bmatrix} = \begin{bmatrix} f(Y) \\ 0 \end{bmatrix},$$

where $\Phi_{ij} = \phi(\|y^i - y^j\|)$ and $P_{ij} = p_j(y^i)$ for $i \in \{0, \ldots, p\}$ and $j \in \{0, \ldots, q\}$, and $f(Y)$ is the vector formed by the values $f(y^0), \ldots, f(y^p)$.

This symmetric system has a unique solution for the examples of ϕ given above provided P is full rank and $d \geq 2$.

RBFs (properties)

Such a property is a consequence of the fact that, for the examples above, ϕ is conditionally positive definite of order d with $d = 2$.

One says that ϕ is conditionally positive definite of order d when $\sum_{i,j=0}^{p} \phi(\|y^i - y^j\|)\lambda_i\lambda_j$ is positive for all distinct points $y^0, \ldots, y^p$ and $\lambda \neq 0$ satisfying $\sum_{i=0}^{p} \lambda_i p_j(y^i) = 0$, $j = 0, \ldots, q$, where the p's represent a basis for $\mathcal{P}_n^{d-1}$.

If ϕ is conditionally positive definite of order d, then it is so for any larger order.

For instance, in the case of Gaussian and inverse multiquadric radial basis functions, the matrix Φ is already a positive definite one (and so trivially conditionally positive definite of order 2).

RBFs in DFO

An approach used in DFO is cubic radial basis functions and linear polynomial tails:

$$sm(x) = \sum_{i=0}^{p} \lambda_i \|x - y^i\|^3 + c + g^\top x.$$

In this case poisedness is equivalent to the existence of $n+1$ affinely independent (and thus distinct) points in the interpolation set.

If the number of interpolation points is $p+1$, then the model has $(p+1)+(n+1)$ parameters.

When the number of points is $n+1$ (or less), the solution of the interpolation system gives rise to a linear polynomial, since all the parameters λ_i, $i = 0, \ldots, p$, are zero.

Kriging models

Functional models may incorporate a stochastic component, which may make them better suited for global optimization purposes. A popular example is Kriging.

Kriging models are decomposed into two components. One is typically a simple model intended to capture the trend in the data. The other measures the deviation between the simple model and the true function.

We consider the simple model to be a constant for ease of illustration. Assume then that the true function is of the form

$$f(x) \;=\; \beta + z(x),$$

where $z(x)$ follows a stationary Gaussian process of mean 0 and variance σ^2.

Kriging models

Let the covariance between two sample points y and w be

$$R(y, w) \;=\; \mathbf{Cov}(z(y), z(w)),$$

where $R(y, w)$ is such that $R(Y, Y)$ is positive definite for any set Y of distinct sample points. A popular choice is to model the covariance by RBFs: $R(y, w) = \phi(\|y - w\|)$.

Given a set Y of distinct sample points, the Kriging model $ksm(x)$ is defined as the expected value of the true function given the observed values $f(Y)$:

$$ksm(x) \;=\; E(f(x) \,|\, Y).$$

One can prove that $ksm(x)$ is of the form:

$$ksm(x) \;=\; \hat{\beta} + R(Y, x)^\top R(Y, Y)^{-1}(f(Y) - \hat{\beta}e),$$

where e is a vector of ones and

$$\hat{\beta} \;=\; \frac{e^\top R(Y, Y)^{-1} f(Y)}{e^\top R(Y, Y)^{-1} e}.$$

Design of experiments

In the statistical literature the process of determining the location of the points in the sampling space is called design of experiments, where one of the main goals focuses on reducing the noise of the experiments.

The sample sets are derived by techniques like Latin or factorial designs, supposed to spread well the points in the sampling space by typically placing them at hypercube type vertices.

$\longrightarrow$ Such sampling sets are likely to be well poised for linear interpolation or regression.

There are more elaborated criteria to choose the sampling locations, such as D-optimality, where the sample set Y is chosen so that quantities related to $|\det((Y^\top Y)^{-1})|$ are minimized.

Another technique with interesting spread out type sampling properties is called orthogonal arrays.

Response surface methodologies

Response surface methodology (RSM) is a framework to minimize a function (resulting from the response of a system) by sequentially building and minimizing functional surrogate models.

An RSM typically starts by selecting the most important and relevant variables by applying some analysis of variance.

Then, some heuristic optimization iterative procedure is started where at each iteration a functional surrogate model $sm_k(x)$ is built (for instance by Kriging) and used to produce a direction of potential descent for the true function f.

The current iterate can then be improved by minimizing the true function along this direction. Such an iterative procedure is terminated when the iterates approach a point $\bar{x}$.

Response surface methodologies

An RSM can then be concluded by minimizing a potentially more accurate functional surrogate model (like a second-order polynomial model) locally around $\bar{x}$, without concern about noise.

Rigorously, one can regard other response surface methodologies as one-shot optimization approaches, where a functional surrogate model is first built and then minimized to determine the optimum.

Such approaches, however, are likely to require dense sampling to produce meaningful results.

However, the statistical nature of many applications allows modeling techniques like Kriging to assign confidence intervals to the models generated and to identify outliers and lack of fit.

Space mapping

It is often the case that the function $f(x)$ considered for optimization is of the form $H(F(x))$, where $F : \mathbb{R}^n \to \mathbb{R}^m$ describes a response of a system and $H : \mathbb{R}^m \to \mathbb{R}$ is some merit function, for instance, a norm.

Similarly, suppose we have a surrogate model $S(x)$ for $F(x)$, and thus a surrogate model $sm(x) = H(S(x))$ for $f(x)$.

More generally, one can have a family of surrogate models $S(x; p)$ for $F(x)$, parametrized by some $p \in \mathbb{R}^p$, and a corresponding family of surrogate models $sm(x; p) = H(S(x; p))$ for $f(x)$.

One can then think of the following iterative optimization process for the approximated minimization of $f(x)$ (the parameters extracted are called space-mapping parameters).

Space mapping

Choose $x^{(0)}$.

For every k:

1. **Parameter extraction:** Compute $p^{(k)}$ as a solution for

$$\min_p \ \|S(x^{(k)}; p) - F(x^{(k)})\|.$$

2. **Minimizing the surrogate:** Compute $x^{(k+1)}$ as a solution for

$$\min_x \ sm(x; p^{(k)}) = H(S(x; p^{(k)})).$$

We assumed for simplicity that the domains of x (in both S and F) and of p are unrestricted.

Another natural generalization is to extend the fitting to all previous iterates.

Analysis of space mapping

If the Lipschitz constants of the optimal mappings are sufficiently small (with respect to the variables x and the parameters p, respectively), one can show that this process generates a convergent sequence $\{(x^{(k)}, p^{(k)})\}$.

If, in addition, one imposes other conditions, including the fact that the space-mapping parameter extraction is exact at the limit point (x^*, p^*), meaning that $S(x^*; p^*) = F(x^*)$, then x^* is an optimal solution for the original problem of minimizing the true function $f(x) = H(F(x))$.

Space mapping

The name space mapping is associated with the mapping $P : \Omega_F \to \Omega_S$, defined by

$$P(x_f) \in \operatorname{argmin}_{x \in \Omega_S} \|S(x) - F(x_f)\|.$$

The original space-mapping approach consisted of the minimization of the space-mapping surrogate model $smsm(x) = sm(P(x)) = H(S(P(x)))$.

In practice, one needs to regularize the definition of the space mapping to guarantee existence and uniqueness of solution above.

Secant updates have been derived to approximate the Jacobian of $P(x)$ based only on responses of the system (i.e., evaluations of F) and integrated in trust-region methods.

Constrained DFO problem formulation

Now we briefly consider the case of constrained nonlinear optimization problems written in the form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega, \\ & h_i(x) \leq 0, \quad i = 1, \ldots, m_h, \end{aligned}$$

where $f, h_i : \Omega \subseteq \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$.

$h_i(x) \leq 0,\ i = 1, \ldots, m_h$ are DFO constraints or constraints without available derivatives.

Constraints with available derivatives

We call all the constraints that define Ω constraints with available derivatives.

The constraints $x \in \Omega$ are typically simple bounds of the form $l \leq x \leq u$, or linear constraints of the form $Ax \leq b$.

Relaxable and unrelaxable constraints

The objective function f is often not defined outside Ω, hence (a possible subset of) the constraints defining Ω have to be satisfied at all iterations in an algorithmic framework for which the objective function (and/or some h_i's) is (are) evaluated.

Such constraints are not relaxable. In contrast, relaxable constraints need only to be satisfied approximately or asymptotically.

Other authors refer to relaxable and unrelaxable constraints as soft and hard constraints, or as open and closed constraints, respectively.

Hidden constraints

An extreme case of DFO unrelaxable constraints that does occur in practice are the so-called hidden constraints.

Hidden constraints are not part of the problem specification/formulation and their manifestation comes in the form of some indication that the objective function could not be evaluated.

For example, the objective function $f(x)$ may be computed by a simulation package which may not converge for certain (unknown a priori) values of input parameters, failing to produce the objective function value.

So far these constraints are treated in practical implementations by a heuristic approach or by using the extreme barrier function approach.

Feasible methods

A significant number of methods for constrained DFO problems are feasible methods, in the sense that the iterates produced are always kept feasible.

Feasible approaches might be preferred for several reasons.

The constraints might not be relaxable and the objective function value cannot be evaluated outside the feasible region.

Generating a sequence of feasible points allows the iterative process to be terminated prematurely, a procedure commonly applied when the objective function is very expensive to evaluate.

Extreme barrier function

Since it may be desirable that a derivative-free algorithm generates feasible iterates, the barrier approaches are particularly appealing.

The feasibility may be enforced with respect to only Ω or with respect to the whole feasible region of problem:

$$X = \{x \in \Omega : h_i(x) \leq 0, \ i = 1, \ldots, m_h\}.$$

Directional direct-search methods can be applied not to f directly but to the extreme barrier function f_Ω or f_X.

For any S, one defines f_S as

$$f_S(x) = \begin{cases} f(x) & \text{if } x \in S, \\ +\infty & \text{otherwise.} \end{cases}$$

Extreme barrier function

It is not necessary (in many of the existing approaches) to evaluate f at infeasible points.

Rather, the value of the extreme barrier function is set to $+\infty$ at such points, — and here we should recall that direct-search methods compare function values rather than building models.

Clearly, such an approach could be inappropriate for methods based on interpolation or regression.

Directional direct search (no DFO constraints)

Suppose $X = \Omega$ (no DFO constraints):

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega. \end{aligned}$$

Directional direct-search methods for unconstrained optimization are directly applicable to the minimization of f_Ω.

However, the extreme barrier technique should NOT be applied using an arbitrary positive spanning set.

A descent direction for the objective function (e.g., a direction that makes an acute angle with the negative gradient of a continuously differentiable function) may not be feasible.

To guarantee global convergence, the directions chosen must reflect properly the geometry of the feasible region near the current iterate.

Directional direct search (only simple bounds)

When the constraints amount to simple bounds,

$$\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\},$$

where $l \in (\{-\infty\} \cup \mathbb{R})^n$ and $u \in (\mathbb{R} \cup \{+\infty\})^n$, then (a subset of) the positive spanning set $D_\oplus$ reflects adequately the feasible region near any feasible point.

As a consequence, a directional direct-search method that includes $D_\oplus$ among the vectors used for polling is globally convergent for first-order stationary points when f is continuously differentiable.

In the simple-bounded case it is very easy to check if a point is outside Ω. In those cases one sets f_Ω to $+\infty$ right away, saving one evaluation of f.

Equality constraints

Equality constraints, when included in the original problem formulation, could be converted into two inequalities.

However, such procedure can introduce degeneracy and complicate the calculation of a feasible point.

One alternative is to get rid of the equalities by eliminating some of the problem variables.

The equality linearly constraints case can be treated differently.

Directional direct search (no DFO constraints)

Under the presence of more general constraints, for which the derivatives are known, it becomes necessary to identify the set of active (or nearly active) constraints in order to construct an appropriate set of poll positive generators.

Suppose, for the purpose of the current discussion, that

$$\Omega = \{x \in \mathbb{R}^n : c_i(x) \leq 0, \ i = 1, \ldots, m_c\}.$$

Definition

Given a point $x \in \Omega$ and a parameter $\epsilon > 0$, the index set of the ϵ–active constraints is

$$I(x; \epsilon) = \{i \in \{1, \ldots, m_c\} : c_i(x) \geq -\epsilon\}.$$

Tangent and normal cones

Given $x \in \Omega$, we call $N(x;\epsilon)$ the cone positively generated by the vectors $\nabla c_i(x)$ for $i \in I(x;\epsilon)$:

$$N(x;\epsilon) \;=\; \left\{ \sum_{i \in I(x;\epsilon)} \lambda_i \nabla c_i(x) : \; \lambda_i \geq 0, i \in I(x;\epsilon) \right\}.$$

The polar cone $T(x;\epsilon) = N(x;\epsilon)^\circ$ is then defined by

$$T(x;\epsilon) \;=\; \left\{ v \in \mathbb{R}^n : \; w^\top v \leq 0, \; \forall w \in N(x;\epsilon) \right\}.$$

For proper choices of ϵ and under a constraint qualification, $x + T(x;\epsilon)$ approximates well the local geometry of the feasible region near x.
$\longrightarrow$ This property allows an algorithm to make feasible displacements from x along any direction chosen in $T(x;\epsilon)$.

$T(x;0)$ and $N(x;0)$ are, respectively, the tangent and normal cones for Ω at x.

Directional direct search (no DFO constraints)

If at a given iteration k, $N(x_k; \epsilon) = \{0\}$, then $T(x_k; \epsilon) = \mathbb{R}^n$, which allows the problem to be locally seen as unconstrained.

In the context of a directional direct-search method, such an occurrence suggests the use in the poll step of a positive spanning set for $\mathbb{R}^n$.

In the case $N(x_k; \epsilon) \neq \{0\}$, the set of poll directions must reflect well the local feasible region near the iterate x_k and therefore must contain the positive generators of $T(x_k; \epsilon)$.

Directional direct search (no DFO constraints)

Let us consider for simplicity only the nondegenerate case where $N(x_k;\epsilon)$ has a set of linear independent generators (columns of the matrix N_k).

Let us also consider a full QR decomposition of N_k:

$$N_k = \begin{bmatrix} Y_k & Z_k \end{bmatrix} \begin{bmatrix} R_k \\ 0 \end{bmatrix}.$$

Then, the following includes a set of positive generators for $T(x_k;\epsilon)$:

$$\begin{bmatrix} Z_k & -Z_k & Y_k R_k^{-\top} & -Y_k R_k^{-\top} \end{bmatrix}.$$

Directional direct search (only linear constraints)

In the linearly constrained case, this construction provides the positive generators for all the cones of the form $T(x_k; \varepsilon)$ for all $\varepsilon \in [0, \epsilon]$.

If the linear algebra is performed via Gaussian elimination and N_k has rational entries then the positive generators have rational entries too meeting the integrality requirements.

In the linearly constrained case, a number of directional direct-search approaches have been investigated involving either the extreme barrier function or a projection onto Ω.

The degenerate case poses additional computational difficulties, especially when the number of nonredundant constraints is high.

Directional direct search (no DFO constraints)

The initial point must lie in Ω in all approaches, which is relatively easy to enforce in the linearly constrained case.

There are several approaches provably globally convergent to a stationary point.

The nonlinear case has also been studied (but involves sufficient decrease and projections onto the feasible set).

Directional direct search (DFO constraints)

Now we turn our attention to the situation where there are derivative-free constraints of the type $h_i(x) \leq 0,\ i = 1, \ldots, m_h$.

The augmented Lagrangian method considers then the solution of a sequence of subproblems where the augmented Lagrangian function is minimized subject to the remaining constraints (bounds on the variables or more general linear constraints).

Original nonlinear inequality constraints must be converted into equalities by means of slack variables. Each problem can then be approximately solved using an appropriate directional direct-search method.

This application of augmented Lagrangian methods yields global convergence results to first-order stationary points of the same type of those obtained under the presence of derivatives

Directional direct search (DFO constraints)

Another approach suggested is based on a nonsmooth exact penalty function and on the imposition of sufficient decrease. Linear constraints are handled separately by the use of positive generators.

It is not clear how the augmented Lagrangian or exact penalty approaches can handle general unrelaxable constraints, other than the linear ones.

However, these methods allow one to start infeasible with respect to the relaxable constraints.

Dense generation in the unit sphere

In the context of (directional) direct search, one can derive a general approach for DFO constraints using sets of poll directions whose union is asymptotically dense in $\mathbb{R}^n$.

When globalizing with integer lattices and simple decrease this is just MADS. Again, globalizing with sufficient decrease is easier.

This strategy is applicable to problems with general DFO constraints, including the situation where constraints are hidden.

All constraints are considered as unrelaxable and a feasible starting point is required.

Infeasibility is ruled out by means of the extreme barrier for the whole feasible set.

Clarke stationarity — constrained case

To avoid further notation, let Ω be the feasible set.

Assuming that f is Lipschitz continuous near x_*.

Definition

x_* is a Clarke critical point if

$$\forall d \in T_\Omega(x_*), f^\circ(x_*; d) \geq 0.$$

$T_\Omega(x_*)$ is the tangent cone to Ω at x_* (redefined in the nonsmooth, Clarke way).

Moreover, the Clarke derivative must be appropriately redefined.

Clarke stationarity — constrained case

Let f Lipschitz continuous near x_*. Clarke-Jahn Generalized directional derivative:

For $v \in \text{int}(T_\Omega(x_*))$

$$f^\circ(x_*; v) = \limsup_{\substack{x' \to x_*, x' \in \Omega \\ t \downarrow 0, x' + tv \in \Omega}} \frac{f(x' + tv) - f(x')}{t}.$$

For $d \in T_\Omega(x_*)$ (Audet and Dennis [2006])

$$f^\circ(x_*; d) = \lim_{v \in \text{int}(T_\Omega(x_*)), v \to d} f^\circ(x_*; v).$$

Refining directions (constraints)

Stationarity results for these type of DS consist of nonnegativity of generalized directional derivatives along certain limit directions.

Definition (refining directions)

Let K be a refining subsequence converging to x_*.

Refining directions for x_* are limit points of $\{d_k/\|d_k\|\}_{k\in K}$, where $d_k \in D_k$ and $x_k \in \Omega$ and $x_k + \alpha_k d_k \in \Omega$.

Audet and Dennis [2006]

Consider a refining subsequence converging to x_* (and assume that f is Lipschitz continuous near x_*).

Let $d \in \text{int}(T_\Omega(x_*))$ be a refining direction.

Proof sketch (assuming normalized directions)

Then $d \in \operatorname{int}(T_\Omega(x_*))$ is a limit point of $\{d_k\}$ (assume $\|d_k\| = 1$) for which $x_k, x_k + \alpha_k d_k \in \Omega$. W.l.o.g. assume $d_k \to d$ in K.

$$
\begin{aligned}
f^\circ(x_*; d) &= \limsup_{\substack{x' \to x_*,\, x' \in \Omega \\ t \downarrow 0,\, x' + td \in \Omega}} \frac{f(x' + td) - f(x')}{t} \\
&\geq \limsup_{k \in K} \left\{ \frac{f(x_k + \alpha_k d_k) - f(x_k)}{\alpha_k} + o(\alpha_k) \right\} \\
&= \limsup_{k \in K} \left\{ \frac{f(x_k + \alpha_k d_k) - f(x_k) + \rho(\alpha_k)}{\alpha_k} - \frac{\rho(\alpha_k)}{\alpha_k} \right\}.
\end{aligned}
$$

Since $\{x_k\}_{k \in K}$ is a refining subsequence, for each $k \in K$,

$$f(x_k + \alpha_k d_k) - f(x_k) + \rho(\alpha_k) \geq 0.$$

Global convergence of DS in the non-smooth case (constraints)

By passing from the interior to the closure of the tangent cone:

Theorem

If $L(x_0)$ is bounded, then (for either integer lattices and simple decrease or for sufficient decrease) there exists a refining subsequence converging to a point x_*.

Let f be Lipschitz continuous near x_*. Assume $\text{int}(T_\Omega(x_*)) \neq \emptyset$.

If all refining directions for that refining subsequence are asymptotically dense in the unit sphere intersected with $T_\Omega(x_*)$, then:

$$\forall d \in T_\Omega(x_*), f^\circ(x_*; d) \geq 0.$$

(We have considered directions in $T_\Omega(x_*)$ normalized (w.l.o.g.).)

Progressive barrier / merit function

A more recent approach called MADS with a progressive barrier allows the handling of both types of constraints, by combining extreme barrier for unrelaxable constraints (again globalizing by simple decrease and integer lattices) with non-dominance filter type concepts for the relaxable constraints.

An interesting feature is that a constraint can be considered relaxable until it becomes feasible and transferrable to the set of unrelaxable ones.

A comparable approach has been developed for globalization by sufficient decrease (using a merit function and a feasibility restoration procedure).

- C. Audet and J. E. Dennis, A progressive barrier for derivative-free nonlinear programming, SIAM J. Optim., 20 (2009) 445–472.
- S. Gratton and L. N. Vicente, A merit function approach for direct search, SIAM J. Optim., 24 (2014) 1980–1998.

Constrained TR interpolation-based methods

A number of pratical trust-region SQP type methods have been proposed (see available software).

The main ideas are the following:

- Use of quadratic models for the Lagrangian function.
- Models for the constraints can be linear, or quadratic (especially when function evaluations are expensive but leading to quadratically constrained TR subproblems).
- Globalization requires a merit function (typically f) or a filter.
- Relaxable constraints have no influence on the poisedness for f.

Currently, there is no (non-obvious) convergence theory developed for TR interpolation-based methods (in the constrained case).

For example, for (i) linear or box constraints (unrelaxable) and (ii) relaxable constraints with derivatives, the unconstrained theory should be reasonably easy to adapt.

Towards global optimization

(A) One possibility is to apply a multistart strategy, to whatever (local) method one has in hands.

(B) Another is to adapt direct search:

- Using the search step of the search-poll framework.

  $\longrightarrow$ Global convergence (of the overall algorithm) to a stationary point is still guaranteed.

Towards global optimization

There are two main possibilities in (B):

- Build a model using previously evaluated points and RBFs or quadratics. Minimize the possibly non-convex model in a certain region dependent of the DS step size.

- Incorporate a dissemination method or heuristic for global optimization purposes.

  $\longrightarrow$ Such schemes provide a wider exploration of the variable domain or feasible region.

  $\longrightarrow$ Examples are particle swarm (see PSwarm) and variable neighborhood search.

  $\longrightarrow$ Robustness and efficiency of the heuristic (used in the search step) are generally improved.

Presentation outline

Software (directional direct search)

APPSPACK: Asynchronous parallel pattern search (constraints)
http://software.sandia.gov/appspack

NOMAD: Generalized pattern search and mesh adaptive direct search (constraints)
https://www.gerad.ca/nomad

SID-PSM: Generalized pattern search guided by simplex derivatives (constraints with derivatives)
http://www.mat.uc.pt/sid-psm

Software (direct search — others in Matlab)

For unconstrained optimization:

Iterative Methods for Optimization: `Matlab` Codes
Hooke-Jeeves, multidirectional search, and Nelder-Mead methods
`http://www4.ncsu.edu/~ctk/matlab_darts.html`

The Matrix Computation Toolbox
Multidirectional search, alternating directions, and Nelder-Mead methods
`http://www.maths.manchester.ac.uk/~higham/mctoolbox`

`fminsearch`: `Matlab`'s implementation of the Nelder-Mead method
`http://www.mathworks.fr/fr/help/matlab/ref/fminsearch.html`

Software (trust-region interpolation based methods)

`DFO`: Trust-region interpolation-based method (constraints)
`https://projects.coin-or.org/Dfo`

For unconstrained problems/simple bounds:

`ORBIT`: Trust-region interpolation-based method (based on radial basis functions)
`http://www.mcs.anl.gov/~wild/orbit`

`NEWUOA`, `BOBYQA`, `LINCOA`: Trust-region interpolation-based methods by M. J. D. Powell
`http://mat.uc.pt/~zhang/software.html#powell_software`

`WEDGE`: Trust-region interpolation-based method
`http://www.ece.northwestern.edu/~nocedal/wedge.html`

Software (others)

CONDOR: Trust-region interpolation-based method (version of UOBYQA in parallel)
http://www.applied-mathematics.net/optimization/CONDORdownload.html

Implicit Filtering: implicit filtering method (simple bounds)
http://www4.ncsu.edu/~ctk/iffco.html

DFL – Derivative-Free Library, A software library for derivative-free optimization (minimax, global, mixed-integer, smooth/non-smooth)
http://www.dis.uniroma1.it/~lucidi/DFL

VXQR1 (smooth functions of many continuous variables)
http://solon.cma.univie.ac.at/software/vxqr1

Software (global DFO)

`DIRECT`: DIRECT – A Global Optimization Algorithm (simple bounds)
`http://www4.ncsu.edu/~ctk/Finkel_Direct`

MATLAB®Global Optimization Toolbox, The MathWorks^{TM}: It includes global search, multistart, pattern search, genetic algorithm, and simulated annealing solvers.
`http://www.mathworks.com/products/global-optimization/index.html?requestedDomain=www.mathworks.com`

`MCS`: Global optimization by Multilevel Coordinate Search (simple bounds)
`http://www.mat.univie.ac.at/~neum/software/mcs`

`PSwarm`: Coordinate search and particle swarm for global optimization (simple bounds and linear constraints)
`http://www.norg.uminho.pt/aivaz/pswarm`

Benchmarking

- K. R. Fowler, J. P. Reese, C. E. Kees, J. E. Dennis Jr., C. T. Kelley, C. T. Miller, C. Audet, A. J. Booker, G. Couture, R. W. Darwin, M. W. Farthing, D. E. Finkel, J. M. Gablonsky, G. Gray, and T. G. Kolda, *A comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems*, Advances in Water Resources, 31(2): 743–757, 2008.
- L. M. Rios and N. Sahinidis, *Derivative-free optimization: A review of algorithms and comparison of software implementations*, J. Global Optim., 56 (2013) 1247–1293.

- (data profiles) J. J. Moré and S. M. Wild, *Benchmarking derivative-free optimization algorithms*, SIAM J. Optim., 20 (2009) 172–191.
- (performance profiles) E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002) 201–213.

Data profiles and test set

Data profiles (Moré and Wild) indicate how likely is a solver to reach a specific reduction in function value,

$$f(x_0) - f(x) \geq (1 - \tau)[f(x_0) - f_L],$$

given some computational budget (f_L is the best objective value found by all solvers).

Test problems have been divided into four classes (Moré and Wild):

- Smooth (53 nonlinear least squares problems obtained from CUTEr functions, with $n \in [2, 12]$).
- Non-stochastic noisy (adding oscillatory noise to the smooth ones).
- Non-differentiable (as in the smooth case but by taking ℓ_1 norms).
- Stochastic noisy (adding random noise to the smooth ones).

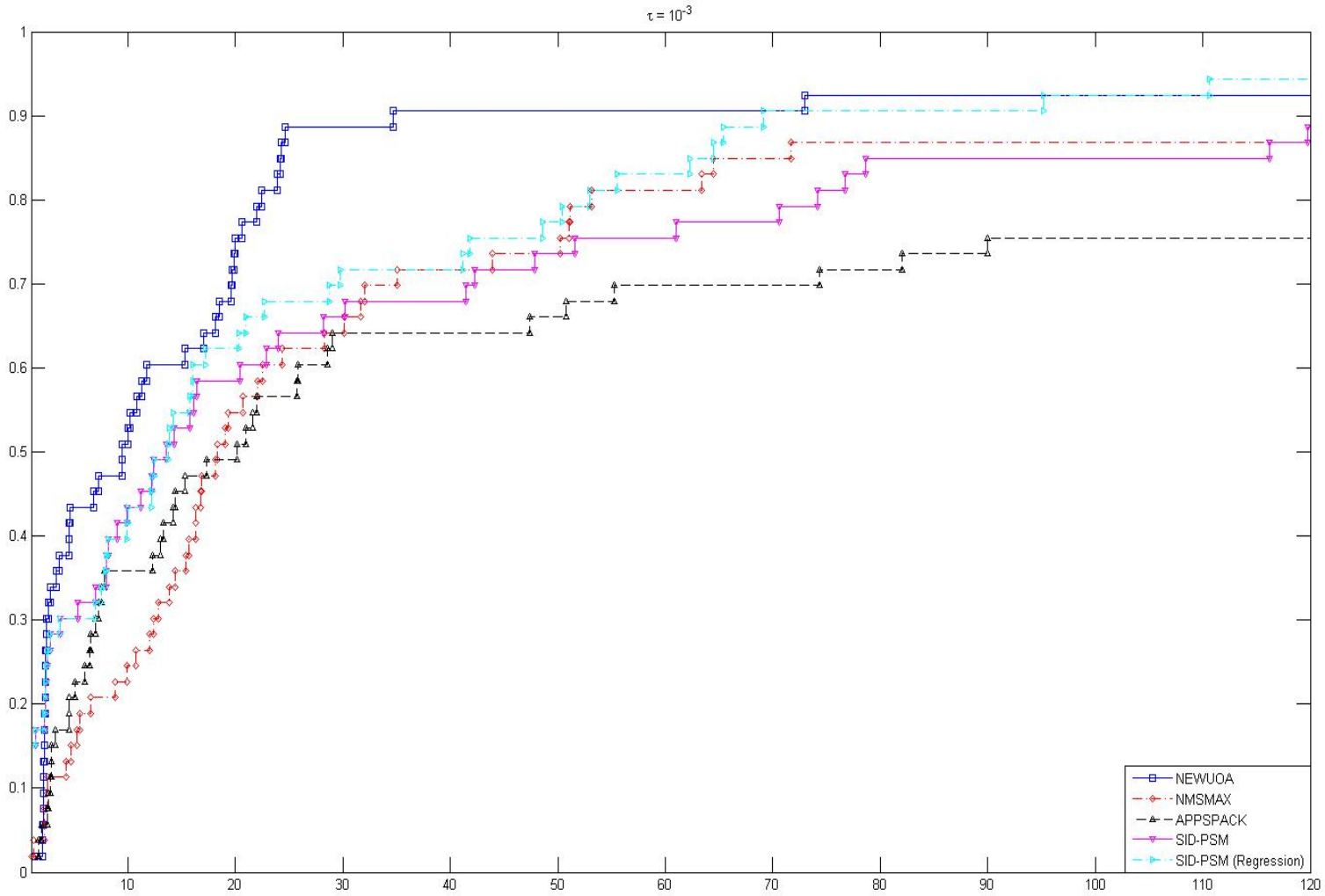
A DFO comparison

We present a comparison among the codes:

- APPSPACK — generalized pattern search (poll by random order), by T. G. Kolda's group.
- NEWUOA — interpolation-based trust-region method (least updating MFN models), by M. J. D. Powell.
- NMSMAX — Nelder-Mead method, by N. J. Higham.
- SID-PSM — generalized pattern search guided by simplex derivatives (uses MFN models in the search step), by A. L. Custódio and L. N. Vicente.

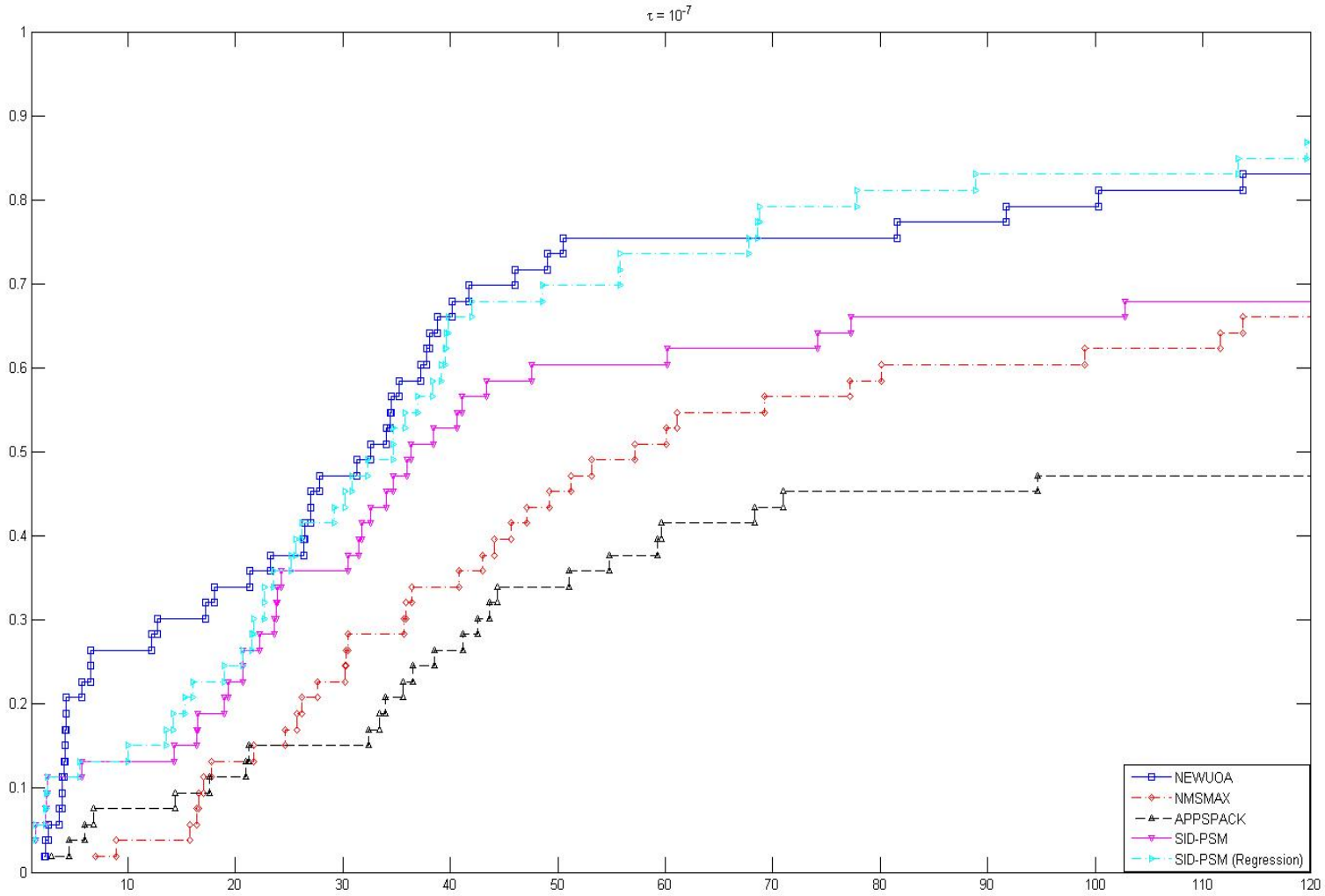
Data profiles — smooth

$\tau = 10^{-3}$



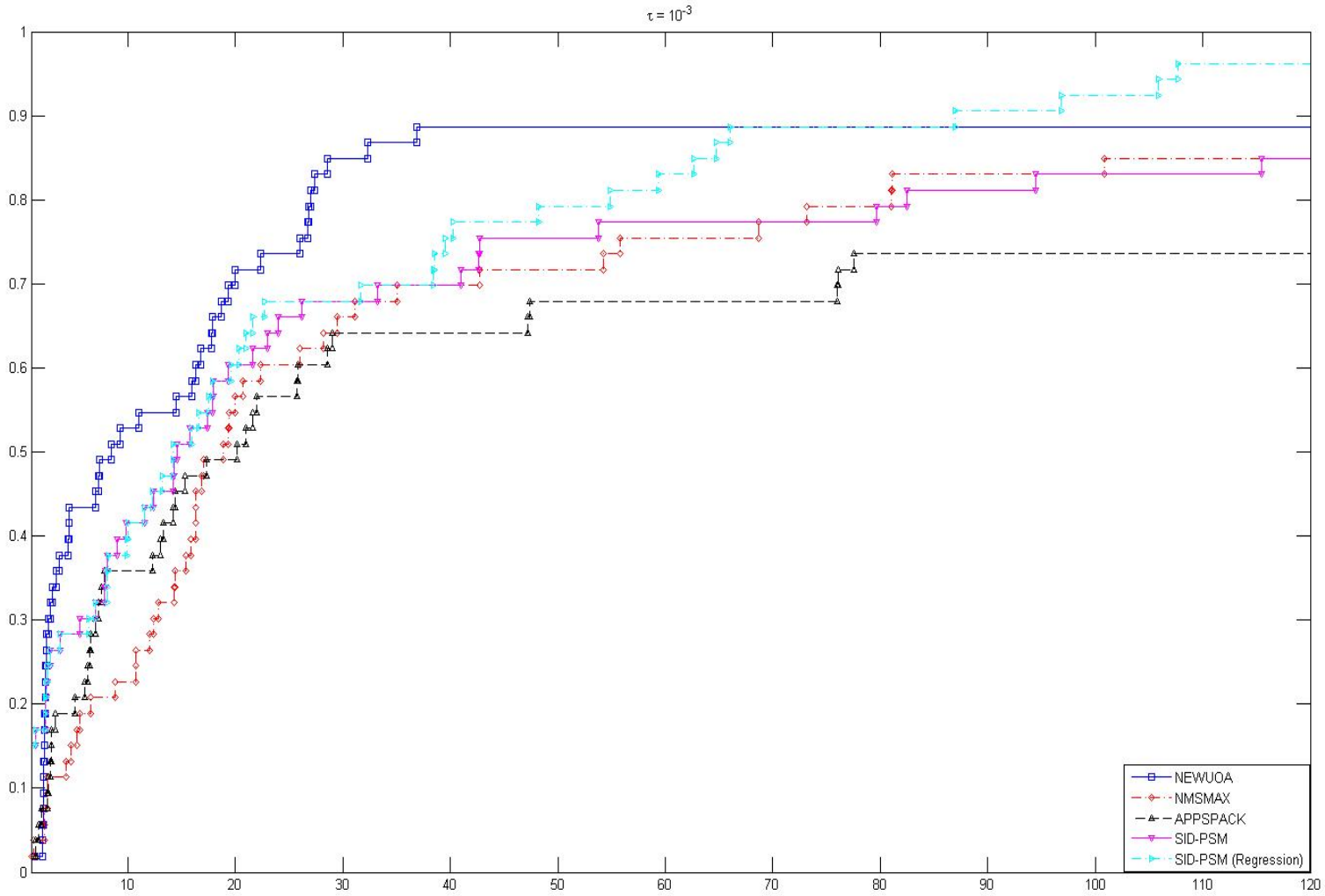
Data profiles — smooth

$\tau = 10^{-7}$



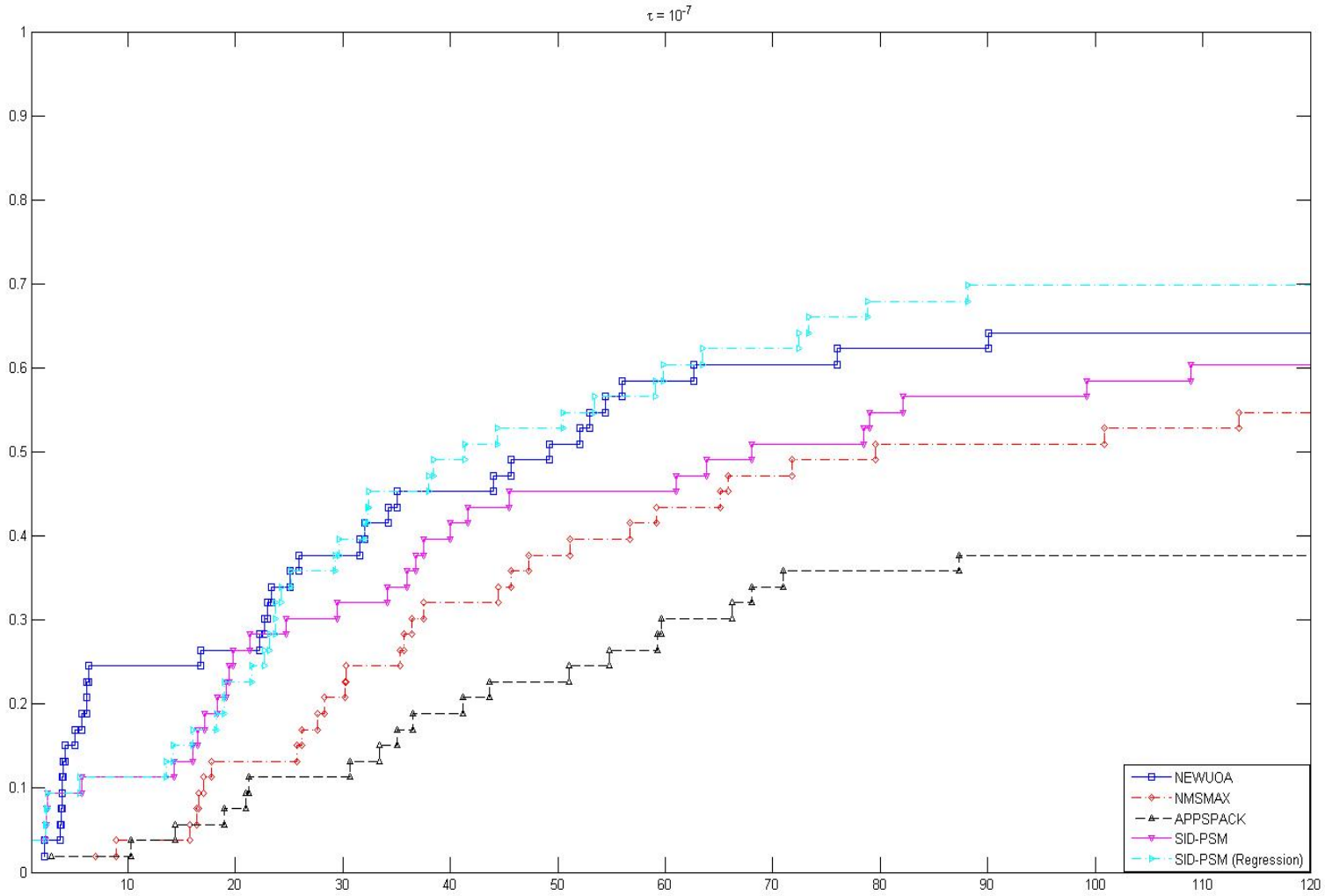
Data profiles — non-stochastic noisy

$\tau = 10^{-3}$



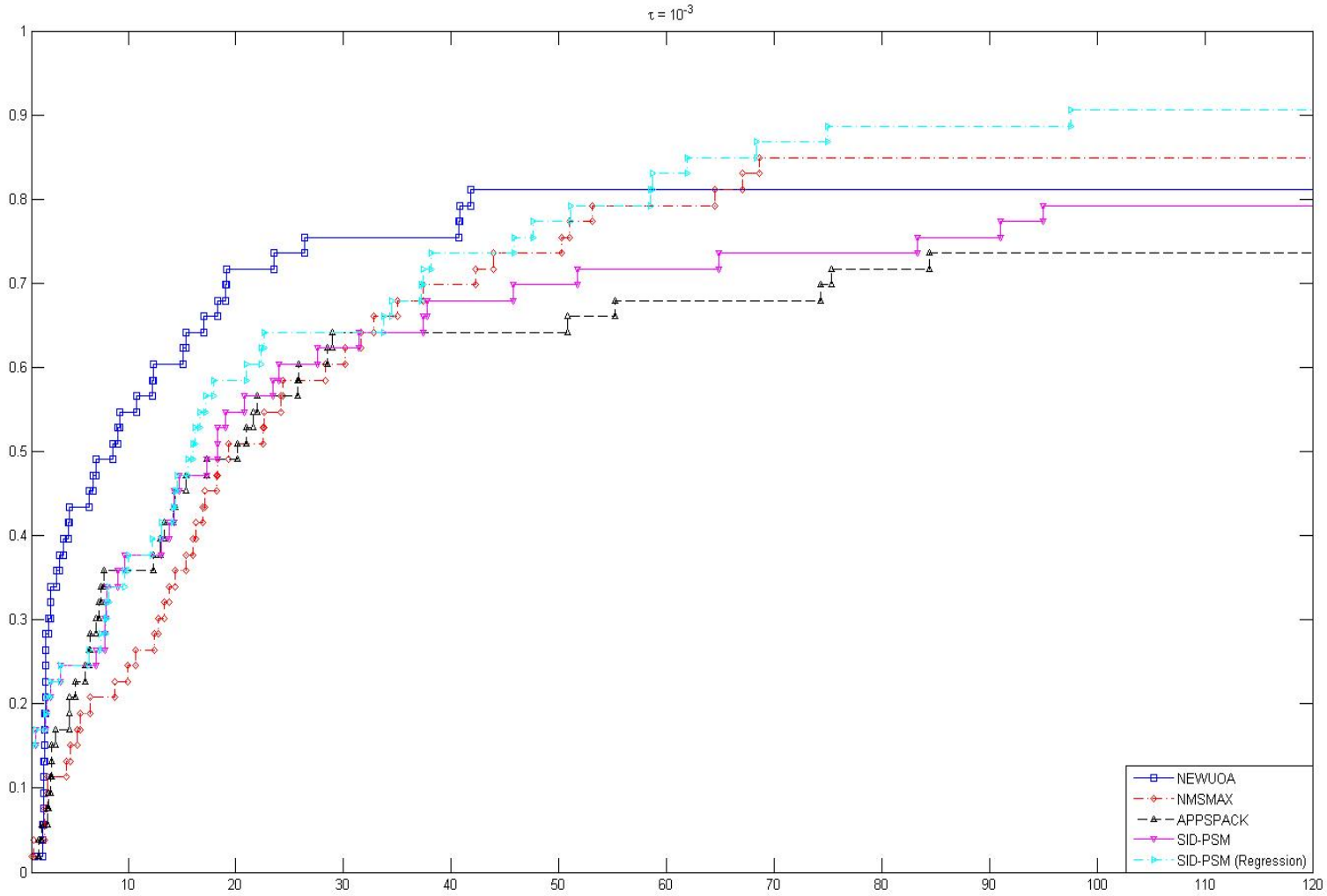
Data profiles — non-stochastic noisy

$\tau = 10^{-7}$



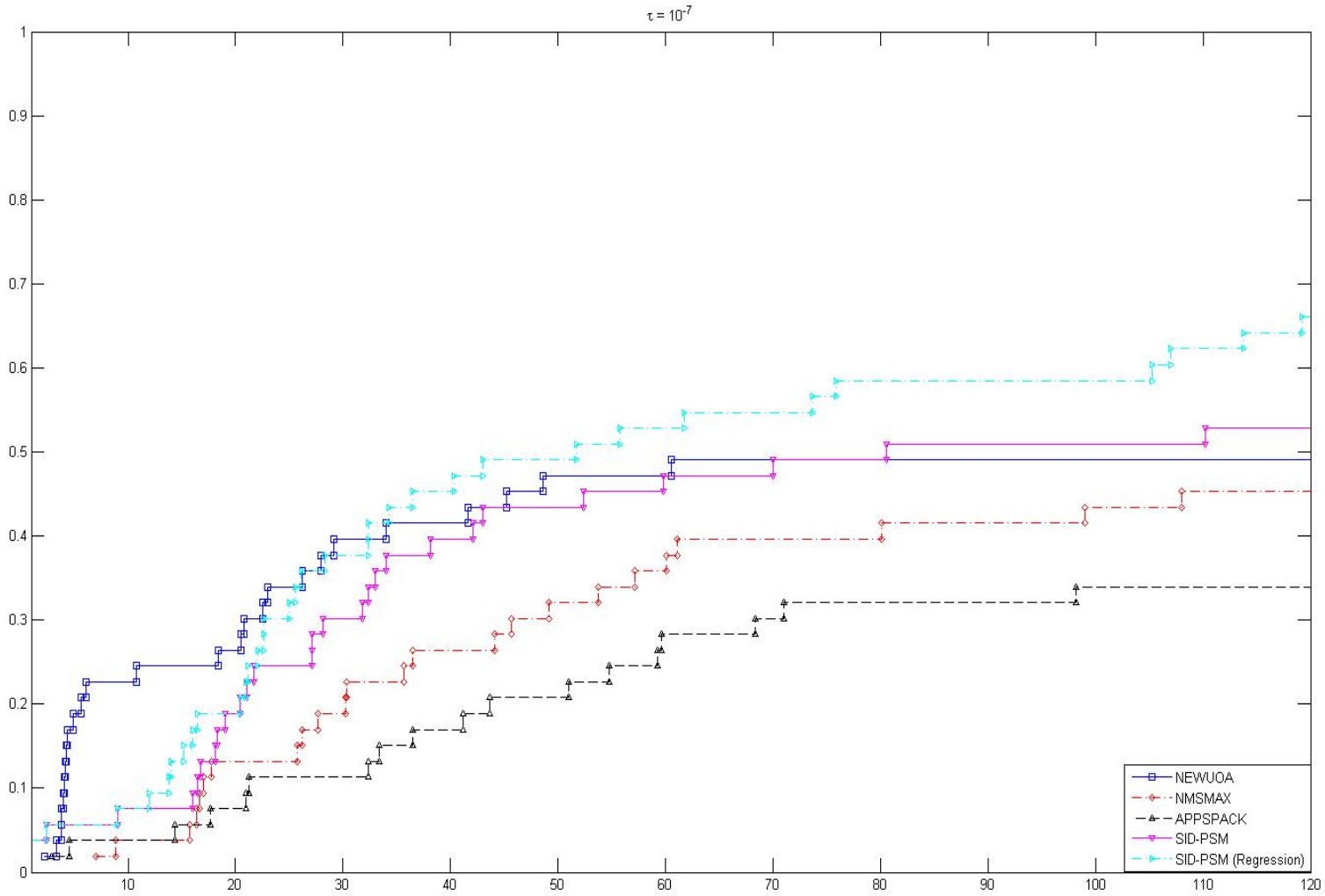
Data profiles — stochastic noisy

$\tau = 10^{-3}$



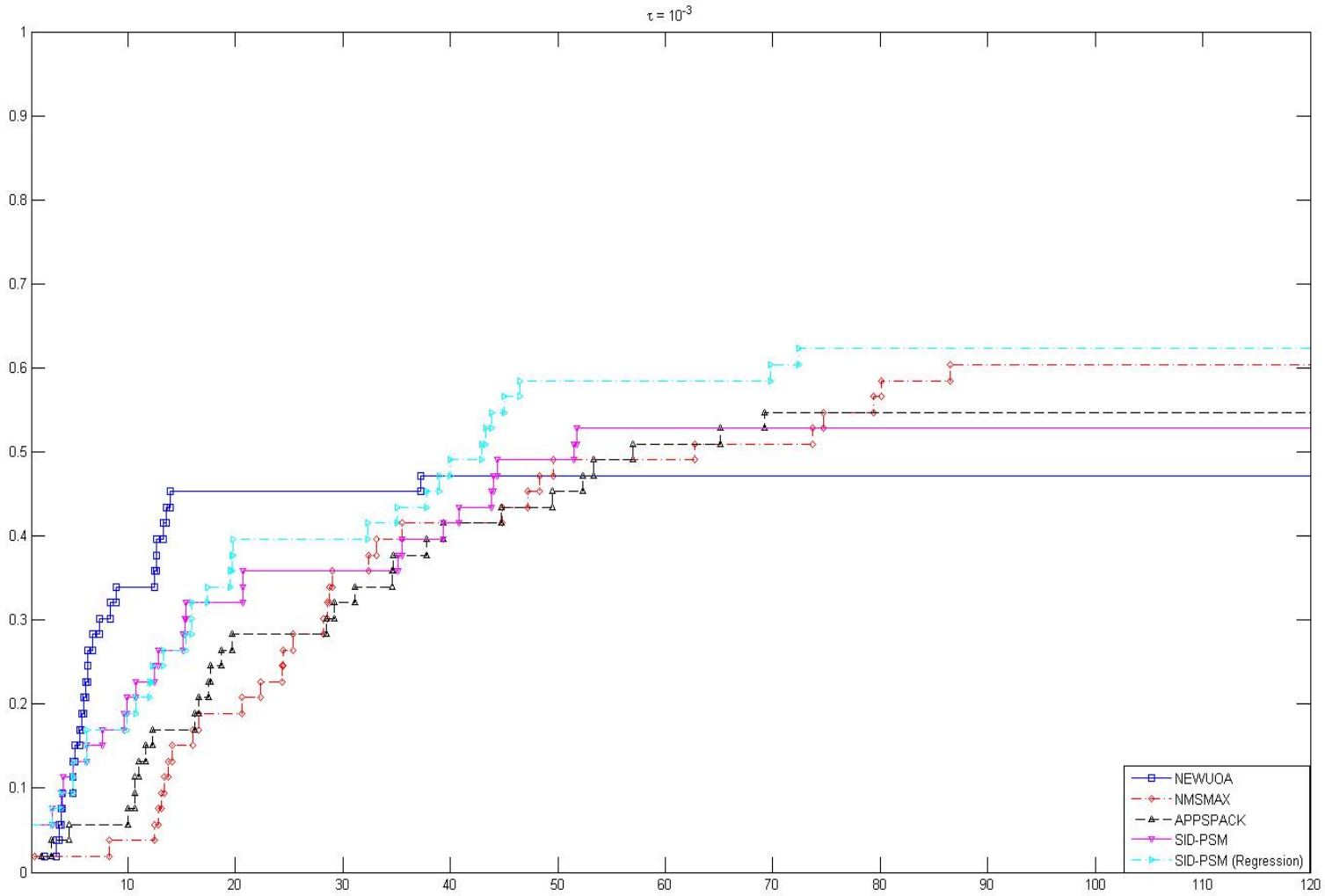
Data profiles — stochastic noisy

$\tau = 10^{-7}$



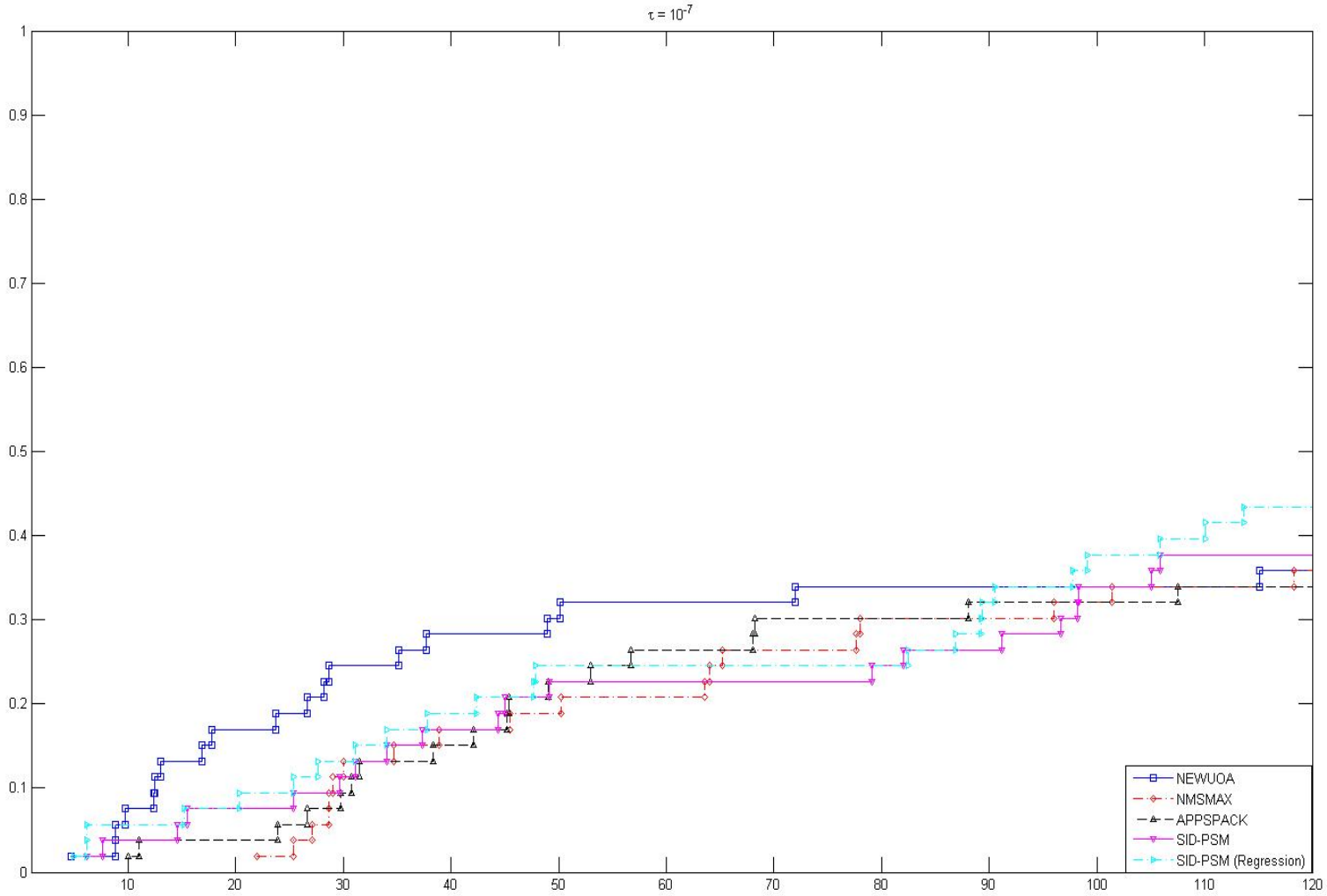
Data profiles — non-smooth

$\tau = 10^{-3}$



Data profiles — non-smooth

$\tau = 10^{-7}$



Main references

- A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2009.

- T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Review, 45 (2003) 385–482.

- D. C. Montgomery, *Design and Analysis of Experiments*, 5th Edition, Wiley, 2000.

Thanks...

Thank you...

Thank a number of people who helped me in the preparation of my slides:

A. L. Custódio, M. Dodengeh, R. Garmanjani, K. Scheinberg, A. I. F. Vaz, Z. Zhang

C. Audet, J. M. Martínez, L. P. Pedroso, Ph. L. Toint, M. W. Wright