Recent Advances in Derivative-Free Optimization

Luís Nunes Vicente[1]

[1]Department of Mathematics
University of Coimbra

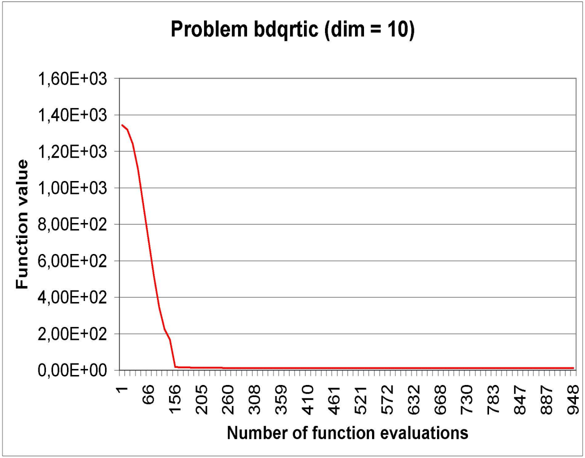
Iberian Conference in Optimization — CIM 2006

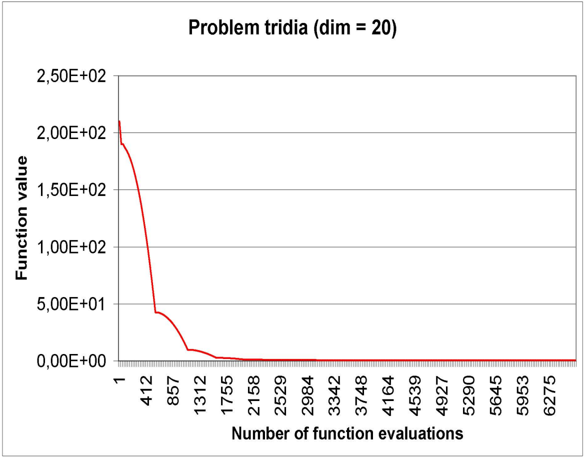
Contents

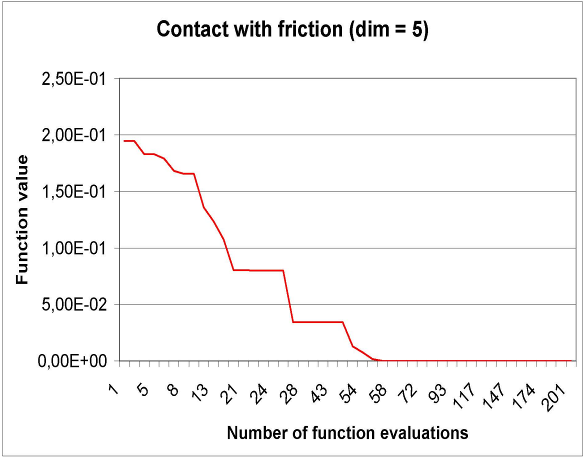
Contents

Contents

Contents

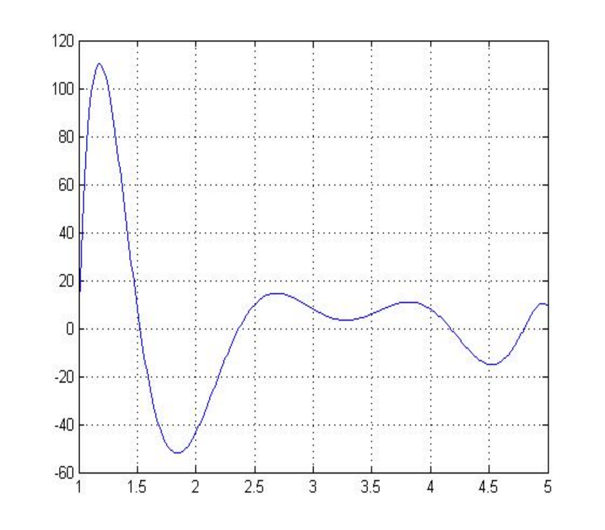

Problem bdqrtic (dim = 10)
Function value
1,60E+03
1,40E+03
1,20E+03
1,00E+03
8,00E+02
6,00E+02
4,00E+02
2,00E+02
0,00E+00
1
66
156
205
260
308
359
410
461
521
572
632
668
730
783
847
887
948
Number of function evaluations

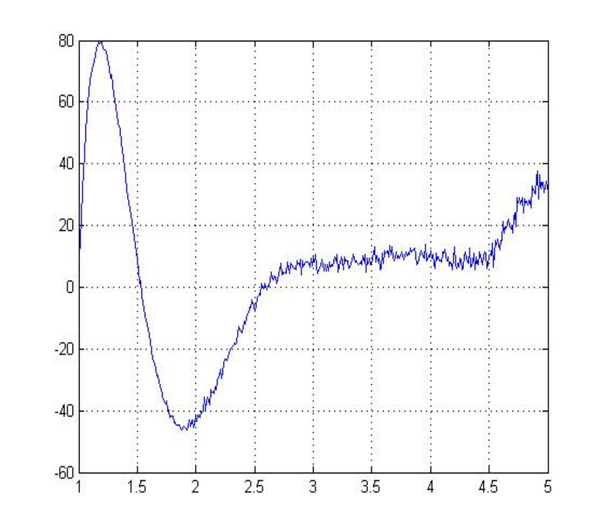

Problem tridia (dim = 20)
2,50E+02
2,00E+02
1,50E+02
1,00E+02
5,00E+01
0,00E+00
Function value
1
412
857
1312
1755
2158
2529
2984
3342
3748
4164
4539
4927
5290
5645
5953
6275
Number of function evaluations

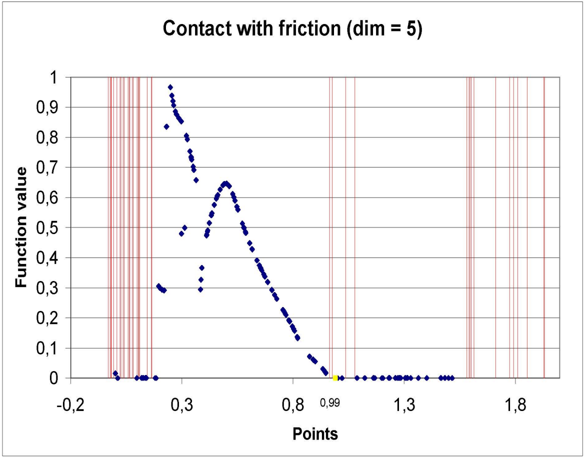

Contact with friction (dim = 5)
Function value
2,50E-01
2,00E-01
1,50E-01
1,00E-01
5,00E-02
0,00E+00
1
5
8
13
21
24
28
43
54
58
72
93
117
147
174
201
Number of function evaluations

Besides,

1. the function might have several relative/local minimizers,
2. the value of the function might be noisy,
3. the evaluation can be expensive,
4. the domain of the function is unknown.


120
100
80
60
40
20
0
-20
-40
-60
1
1.5
2
2.5
3
3.5
4
4.5
5


80
60
40
20
0
-20
-40
-60
1
1.5
2
2.5
3
3.5
4
4.5
5


Contact with friction (dim = 5)
Function value
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
-0,2
0,3
0,8
0,99
1,3
1,8
Points

There is a need for algorithms that:

1. minimize f without the knowledge of its derivatives (gradient),
2. are rigorous (guaranteeing at least stationary points),

... select where to evaluate the function intelligently,

... might be expensive in the underlying linear algebra.

Nelder-Mead

- J. A. Nelder and R. Mead, *A simplex-method for function minimization*, J. Comput., 7 (1965) 308–313.

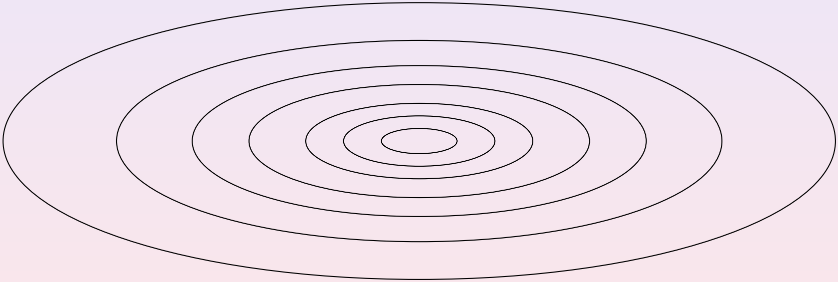
$\longrightarrow$ 6605 (!) citations (SCI).

$\longrightarrow$ Globally convergent for $n = 1$ (1998).

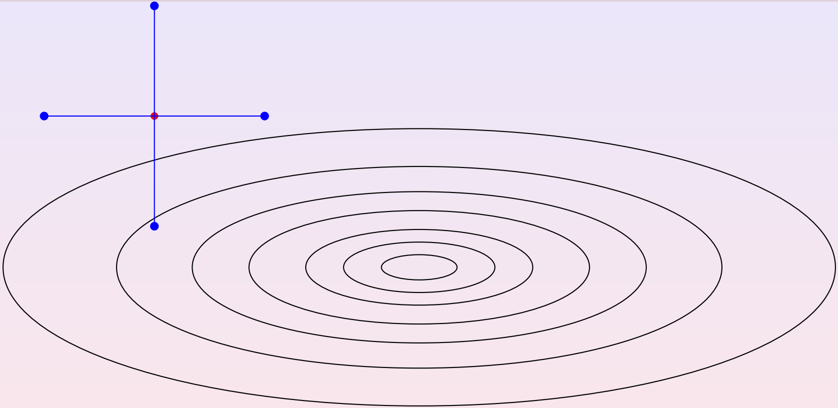
$\longrightarrow$ Counter-examples for $n \geq 2$ (1998).

$\longrightarrow$ Convergence for modifying versions (1999-2002).

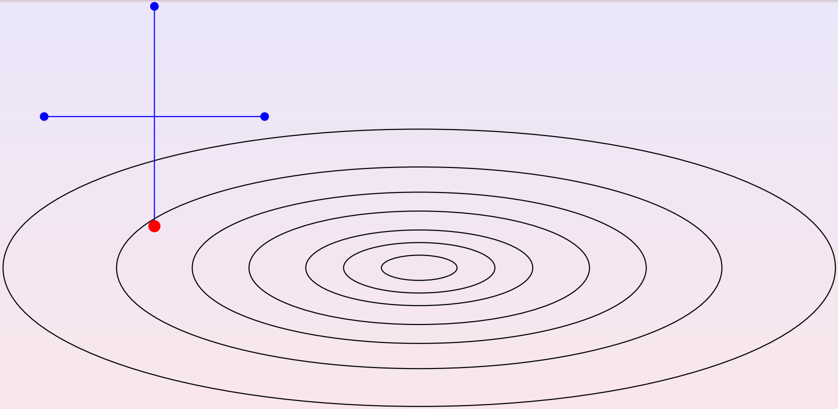
Coordinate Search (poll step)



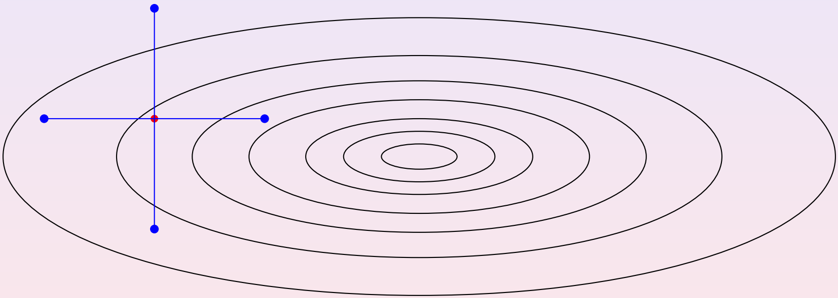
Coordinate Search (poll step)



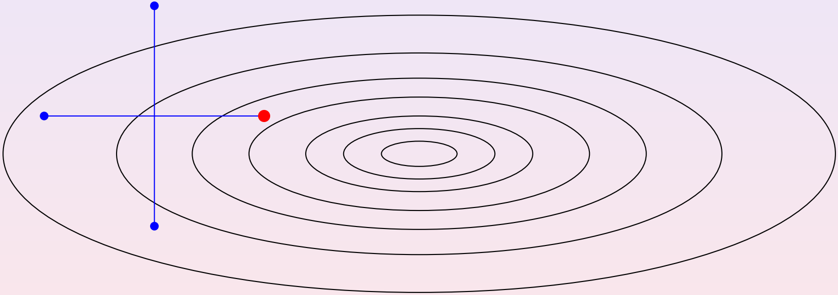
Coordinate Search (poll step)



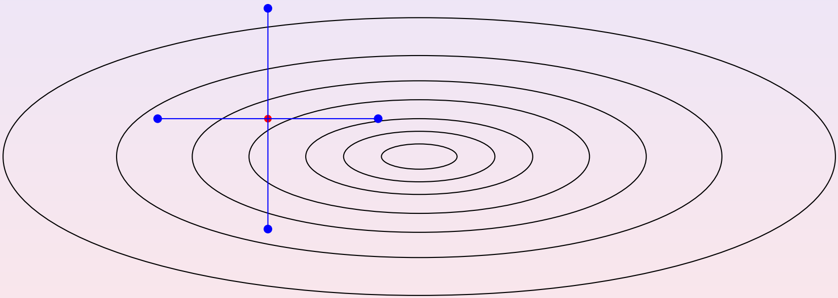
Coordinate Search (poll step)



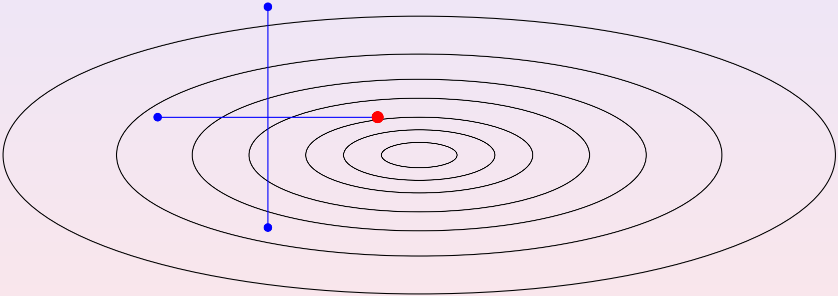
Coordinate Search (poll step)



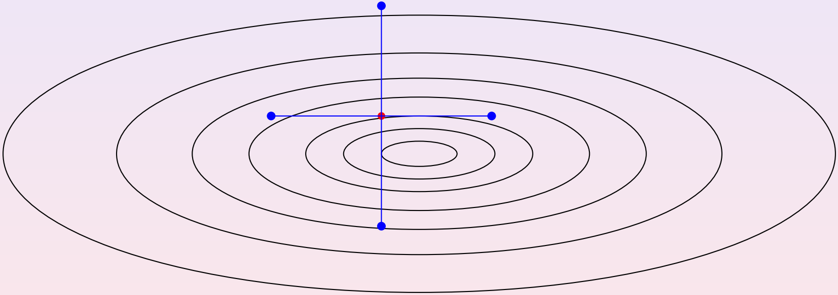
Coordinate Search (poll step)



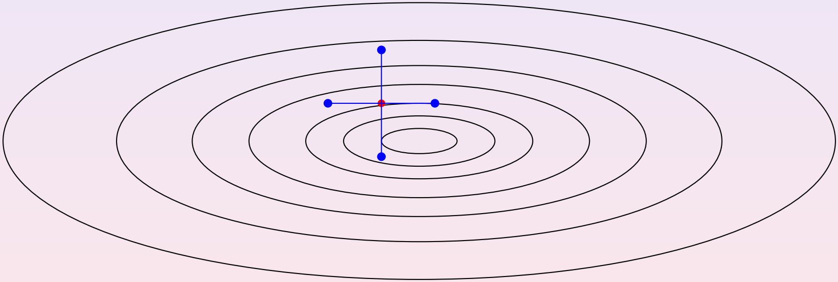
Coordinate Search (poll step)



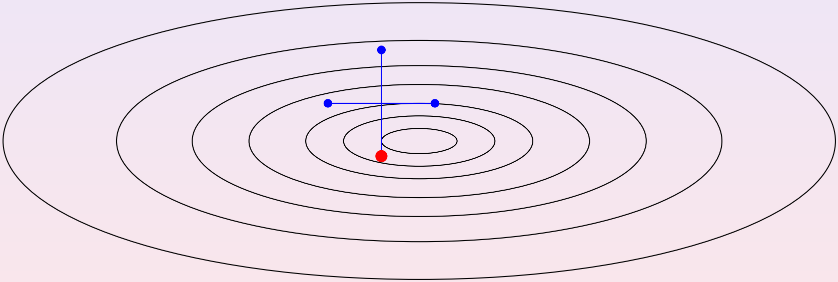
Coordinate Search (poll step)



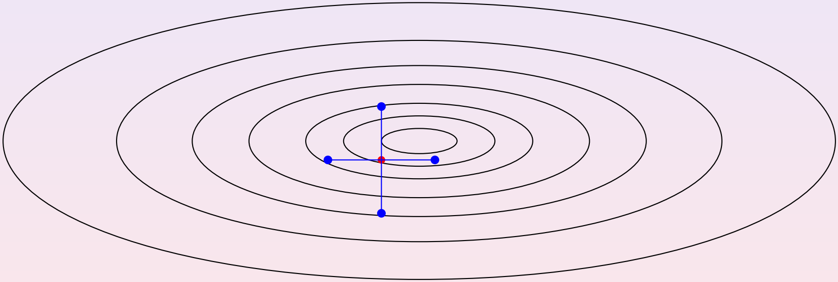
Coordinate Search (poll step)



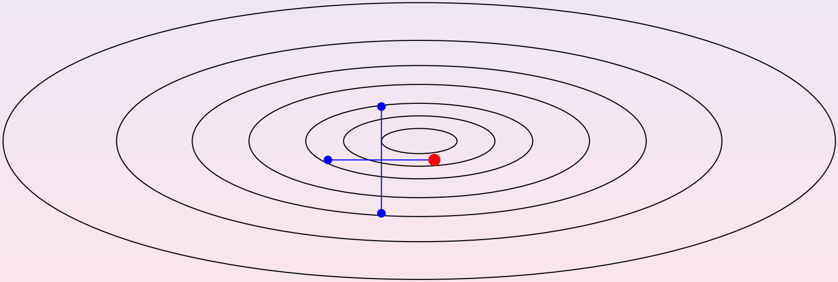
Coordinate Search (poll step)



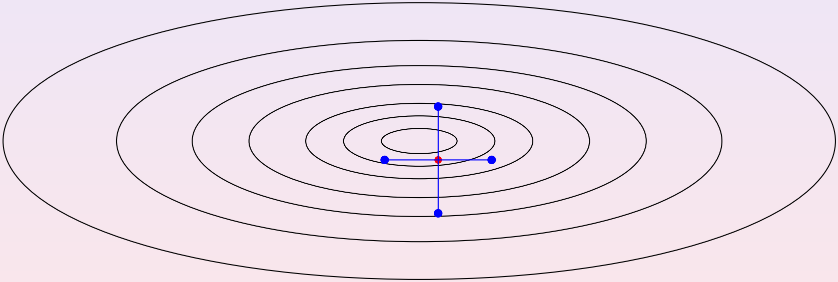
Coordinate Search (poll step)



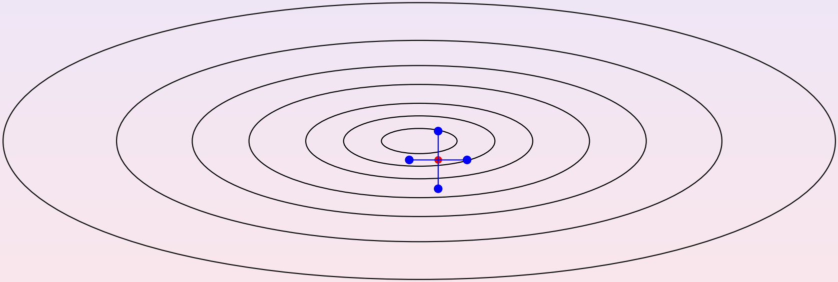
Coordinate Search (poll step)



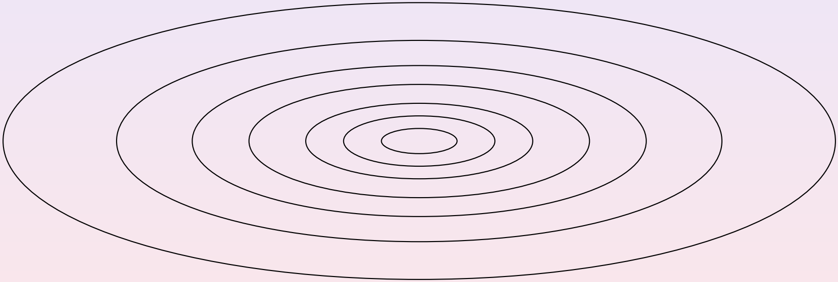
Coordinate Search (poll step)



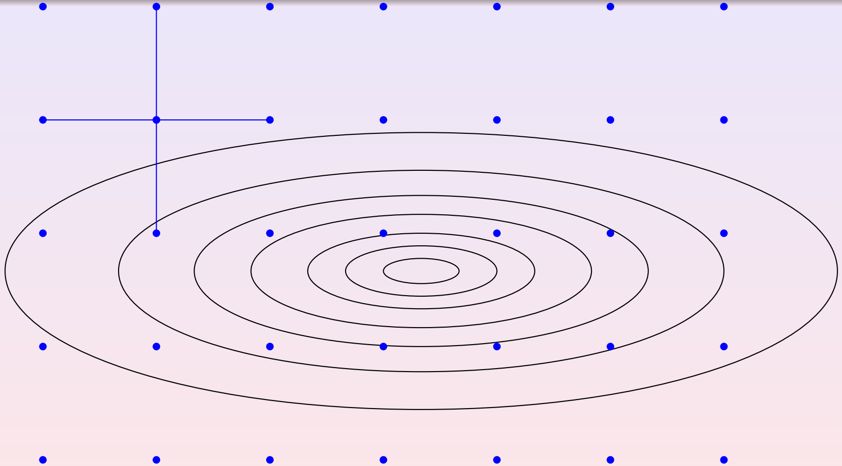
Coordinate Search (poll step)



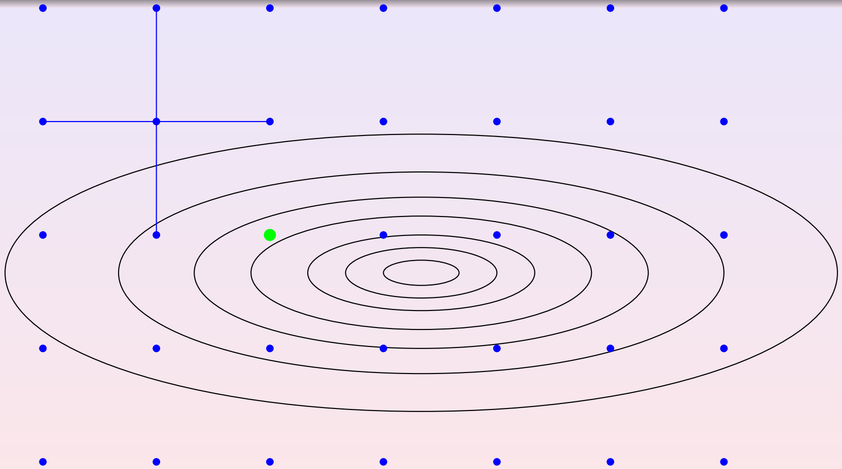
Coordinate Search (with search step)



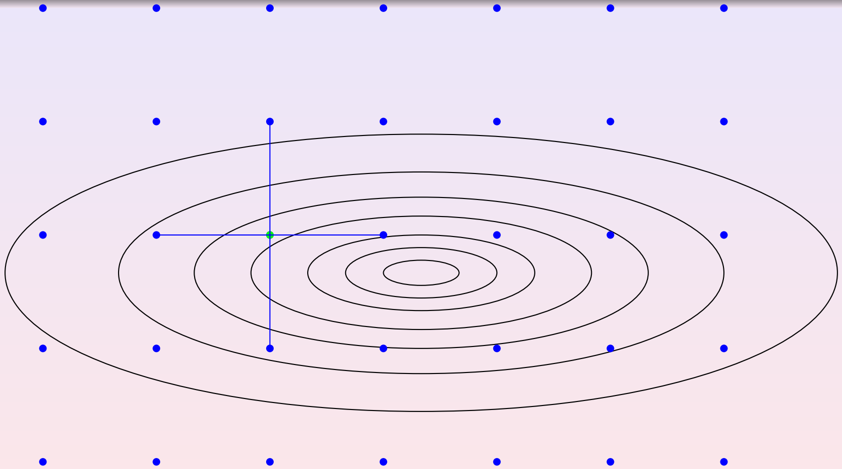
Coordinate Search (with search step)



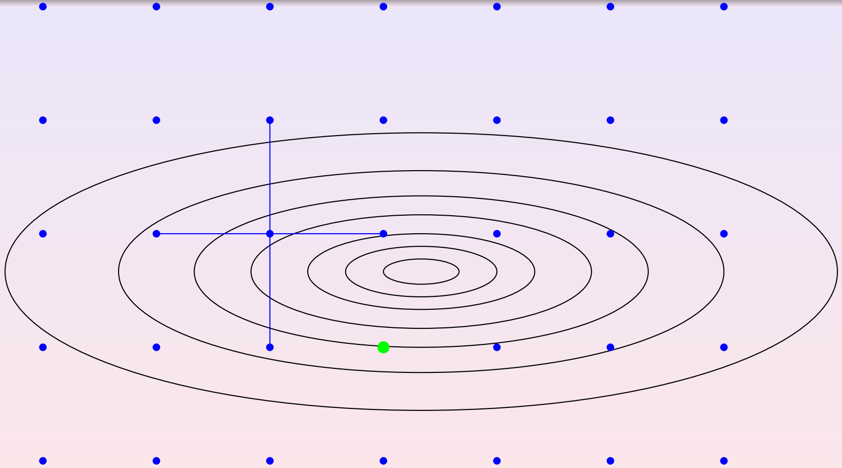
Coordinate Search (with search step)



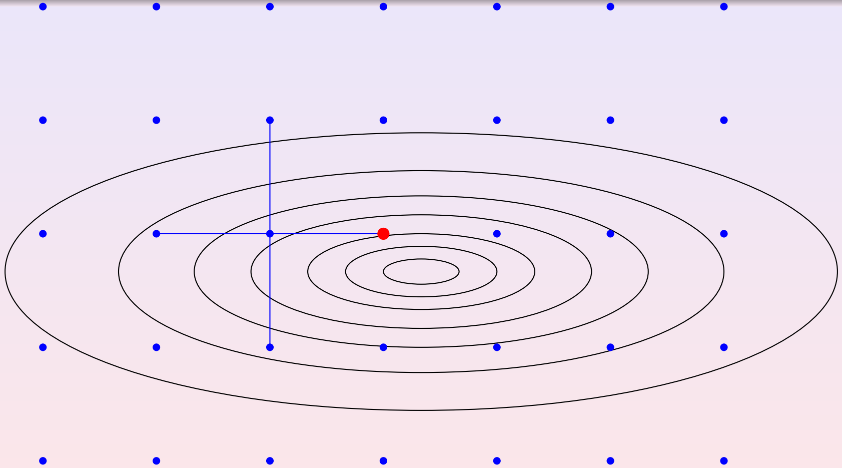
Coordinate Search (with search step)



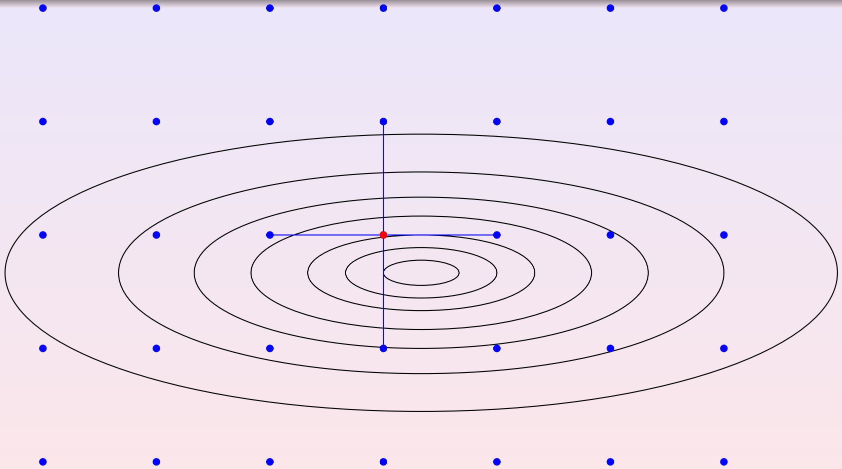
Coordinate Search (with search step)



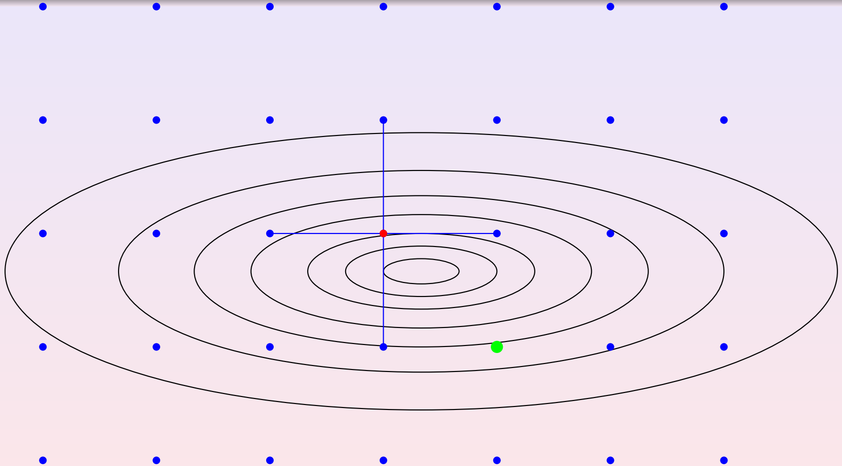
Coordinate Search (with search step)



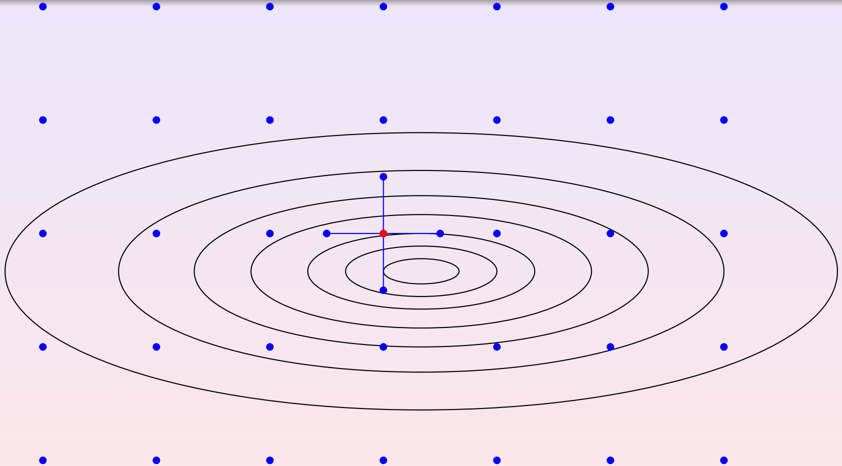
Coordinate Search (with search step)



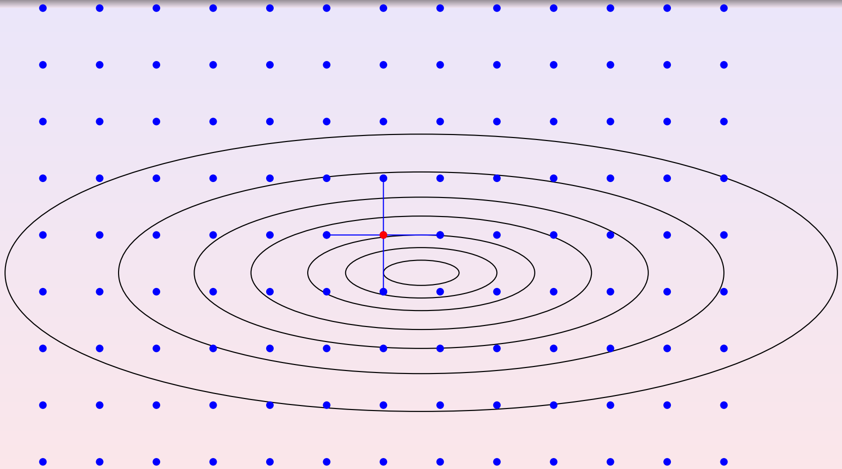
Coordinate Search (with search step)



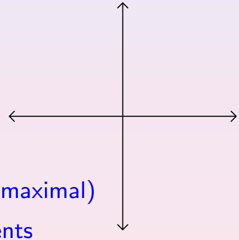
Coordinate Search (with search step)



Coordinate Search (with search step)

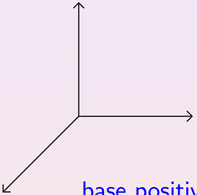


Positive Spanning Sets



positive basis (maximal)

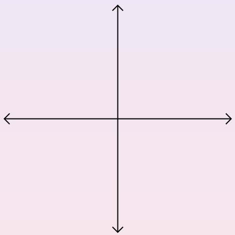
$2n$ elements

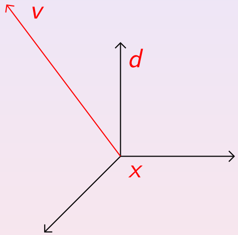


base positive basis (minimal)

$n+1$ elements

Positive Spanning Sets

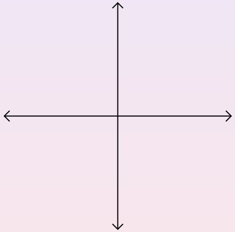


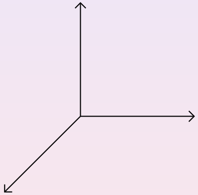


If $v = -\nabla f(x)$ then d is a descent direction.

Positive Spanning Sets

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

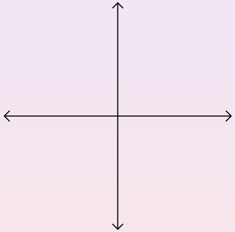


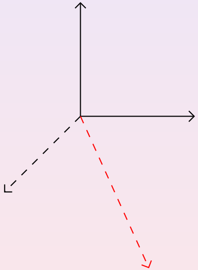


$$\kappa(D) \;=\; \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|} \;>\; 0.$$

Positive Spanning Sets

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



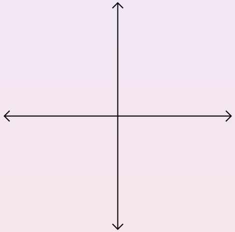


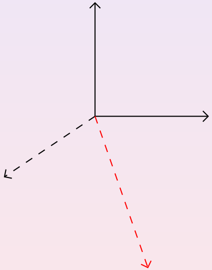
$$\kappa(D) \; = \; \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|} \; > \; 0.$$

Positive Spanning Sets

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

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



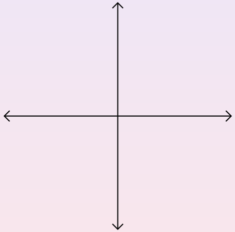


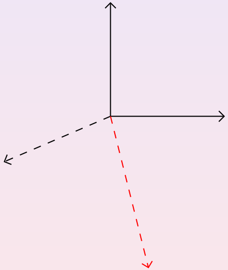
$$\kappa(D) \;=\; \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|} \;>\; 0.$$

Positive Spanning Sets

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

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

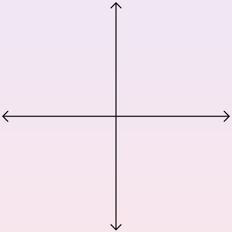


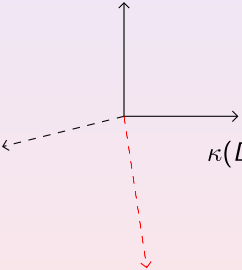


$$\kappa(D) \;=\; \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|} \;>\; 0.$$

Positive Spanning Sets

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





$\kappa(D) \longrightarrow 0$

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

(Directional) Direct Search Methods

1. Search step (finite):

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

2. Poll step ($D_k \subset D$ positive basis):

$$P_k = \{x_k + \alpha_k d, \;\; d \in D_k\} \;\subset\; M_k.$$

3. Update the parameter α_k.

Definition of the positive spanning set D and the update of α_k must satisfy integral/rational requirements.

(Directional) Direct Search Methods

1. Search step (finite):

$$M_k = \{x_k + \alpha_k D z, \ \ z \text{ integer vector}\}.$$

2. Poll step ($D_k \subset D$ positive basis):

$$P_k = \{x_k + \alpha_k d, \ \ d \in D_k\} \ \subset \ M_k.$$

3. Update the parameter α_k.

Definition of the positive spanning set D and the update of α_k must satisfy integral/rational requirements.

(Directional) Direct Search Methods

1. Search step (finite):

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

2. Poll step ($D_k \subset D$ positive basis):

$$P_k = \{x_k + \alpha_k d, \;\; d \in D_k\} \; \subset \; M_k.$$

3. Update the parameter α_k.

Definition of the positive spanning set D and the update of α_k must satisfy integral/rational requirements.

(Directional) Direct Search Methods

1. Search step (finite):

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

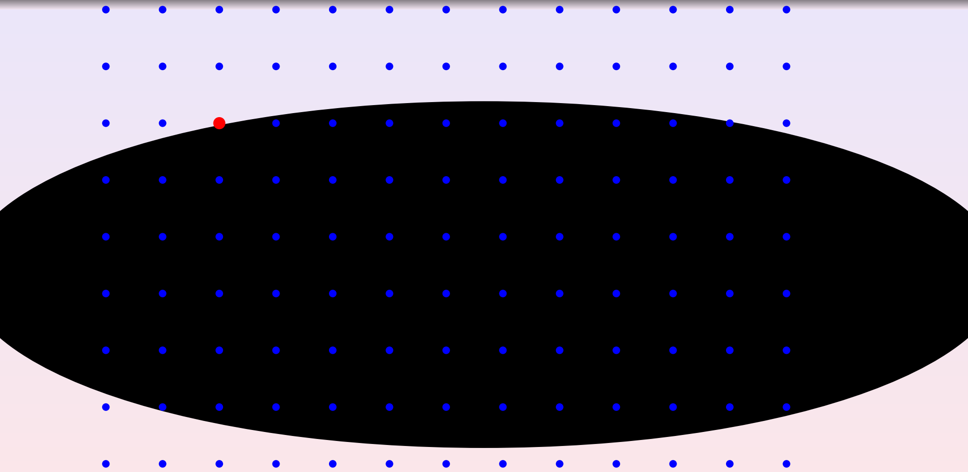
2. Poll step ($D_k \subset D$ positive basis):

$$P_k = \{x_k + \alpha_k d, \;\; d \in D_k\} \;\subset\; M_k.$$

3. Update the parameter α_k.

Definition of the positive spanning set D and the update of α_k must satisfy integral/rational requirements.

Integer/Rational Requirements



$$L(x_0) = \{x \in \mathbb{R}^n : \; f(x) \leq f(x_0)\}$$

Lemma (Infinity of unsuccessful iterations)

If $L(x_0)$ is compact then there exists a subsequence K:

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

For unsuccessful k (∇f Lipschitz continuous):

$$\|\nabla f(x_k)\| \leq \left(C\, \kappa(D_k)^{-1}\right) \alpha_k.$$

Theorem (Global convergence — direct search)

If $L(x_0)$ is compact and $f \in C^1(L(x_0))$:

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

Torczon 97

Lemma (Infinity of unsuccessful iterations)

If $L(x_0)$ is compact then there exists a subsequence K:

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

For unsuccessful k (∇f Lipschitz continuous):

$$\|\nabla f(x_k)\| \;\leq\; \left(C\, \kappa(D_k)^{-1}\right) \alpha_k .$$

Theorem (Global convergence — direct search)

If $L(x_0)$ is compact and $f \in C^1(L(x_0))$:

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

Torczon 97

Lemma (Infinity of unsuccessful iterations)

If $L(x_0)$ is compact then there exists a subsequence K:

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

For unsuccessful k (∇f Lipschitz continuous):

$$\|\nabla f(x_k)\| \leq \left(C\, \kappa(D_k)^{-1}\right) \alpha_k.$$

Theorem (Global convergence — direct search)

If $L(x_0)$ is compact and $f \in C^1(L(x_0))$:

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

Torczon 97

Definition (Clarke Generalized Derivative)

For f Lipschitz continuous near x:

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

If f is increasing from x, along d, then

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

x is a (Clarke) stationary point if

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

Theorem (Global convergence — direct search)

There exists $B \subset D$ such that

If f is increasing from x, along d, then

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

x is a (Clarke) stationary point if

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

Theorem (Global convergence — direct search)

There exists $B \subset D$ such that

$$f^\circ(x_*;d) \geq 0, \quad \forall \textit{poll directions } d \in B.$$

Audet and Dennis 03 (GPS)

If f is increasing from x, along d, then

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

x is a (Clarke) stationary point if

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

Theorem (Global convergence — direct search)

There exists $B \subset D$ such that

$$f^\circ(x_*;d) \geq 0, \quad \forall \textit{poll directions } d \in B.$$

Audet and Dennis 03 (GPS)

If f is increasing from x, along d, then

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

x is a (Clarke) stationary point if

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

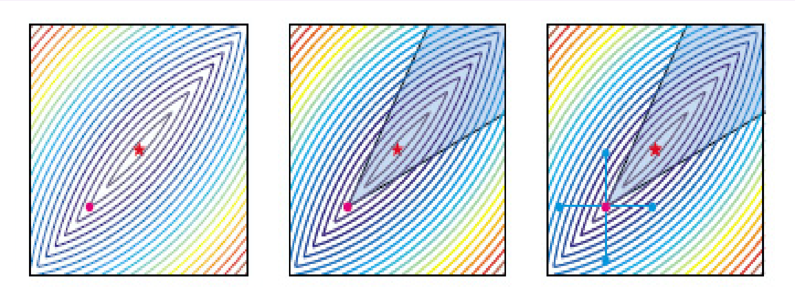
Theorem (Global convergence — direct search)

There exists $B \subset D$ such that

$$f^\circ(x_*; d) \geq 0, \quad \forall \textit{poll directions } d \in B.$$

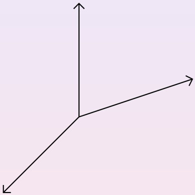
Audet and Dennis 03 (GPS)

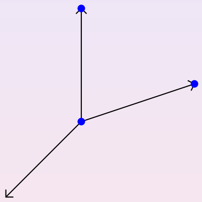
If f is not strictly differentiable near x_*, then the point x_* might not be stationary (Kolda, Lewis, and Torczon 03):

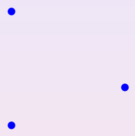


One approach that guarantees proper asymptotic results in the nonsmooth context is MADS:

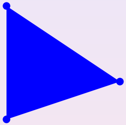
C. Audet and J. E. Dennis Jr., *Mesh adaptive direct search algorithms for constrained*, SIAM Journal on Optimization, 17 (2006) 188–217.



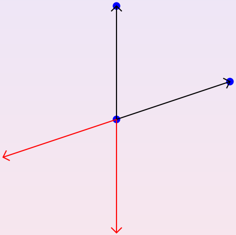




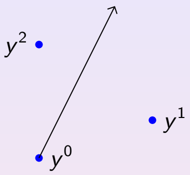
The $3 = n + 1$ points form an affine independent set.



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

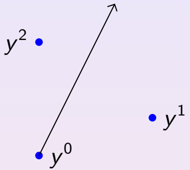


Its reflection produces a (maximal) positive basis.



It is possible to build a simplex gradient:

$$\nabla_s f(y^0) \; = \; \left[\begin{array}{cc} y^1 - y^0 & y^2 - y^0 \end{array} \right]^{-\top} \left[\begin{array}{c} f(y^1) - f(y^0) \\ f(y^2) - f(y^0) \end{array} \right].$$



It is possible to build a simplex gradient:

$$\nabla_s f(y^0) \;=\; S^{-\top} \left[\begin{array}{c} f(y^1) - f(y^0) \\ f(y^2) - f(y^0) \end{array} \right].$$

Theorem

$$\|V^\top[\nabla f(y^0) - \nabla_s f(y^0)]\| \leq \left(q^{\frac{1}{2}}\|\Sigma^{-1}\|\frac{\gamma}{2}\right)\Delta$$

where $V = I$ *if* $q \geq n$.

- $y^0, y^1, \ldots, y^q \in B(y^0;\Delta)$
- ∇f Lipschitz continuous in $B(y^0;\Delta)$ with constant $\gamma > 0$
- $U\Sigma V^\top$ is the reduced SVD of $S^\top/\Delta$

The set is Λ-poised if

$$\|\Sigma^{-1}\| \leq \Lambda.$$

Theorem

$$\|V^\top[\nabla f(y^0) - \nabla_s f(y^0)]\| \leq \left(q^{\frac{1}{2}}\|\Sigma^{-1}\|\frac{\gamma}{2}\right)\Delta$$

where $V = I$ if $q \geq n$.

- $y^0, y^1, \ldots, y^q \in B(y^0;\Delta)$
- ∇f Lipschitz continuous in $B(y^0;\Delta)$ with constant $\gamma > 0$
- $U\Sigma V^\top$ is the reduced SVD of $S^\top/\Delta$

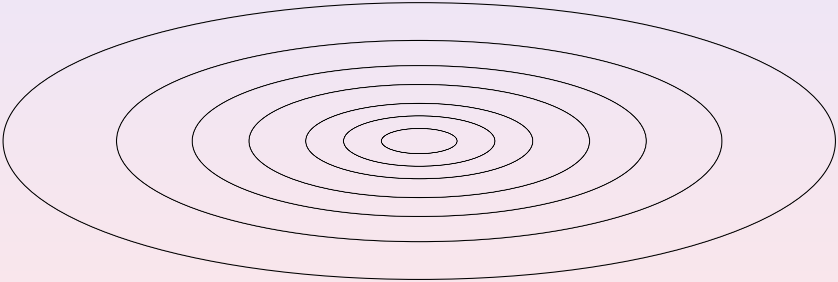
The set is Λ-poised if

$$\|\Sigma^{-1}\| \leq \Lambda.$$

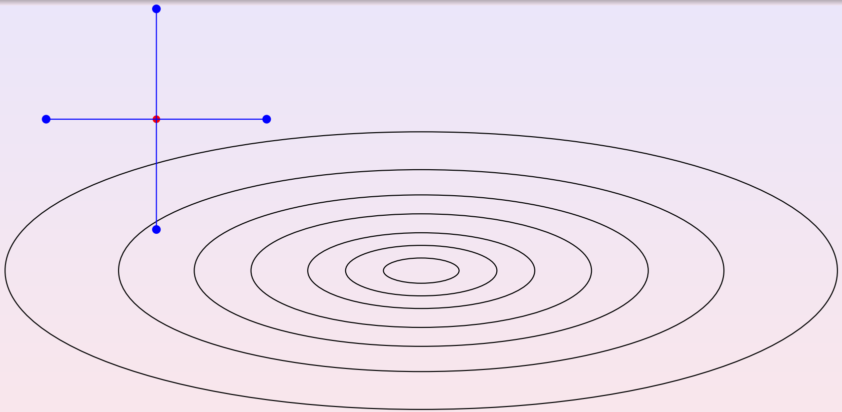
References

1. A. R. Conn, K. Scheinberg, and L. N. Vicente, *Geometry of interpolation sets in derivative free optimization*, to appear in Mathematical Programming.
2. A. R. Conn, K. Scheinberg, and L. N. Vicente, *Geometry of sample sets in derivative free optimization: Polynomial regression and underdetermined interpolation*, submitted, 2005.

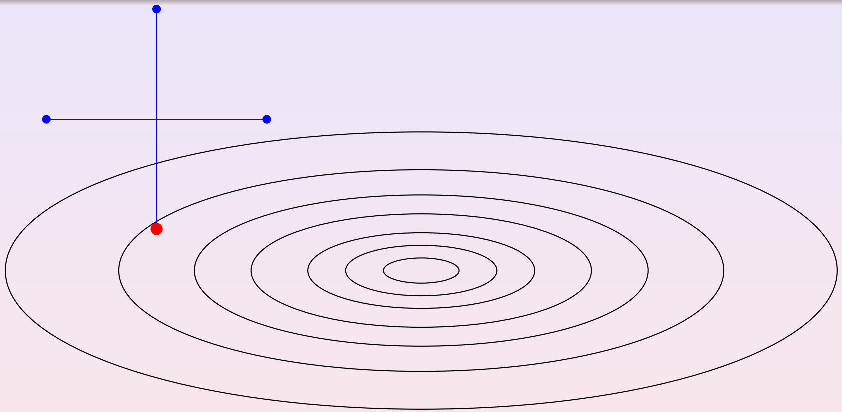
Coordinate Search (poll ordering)



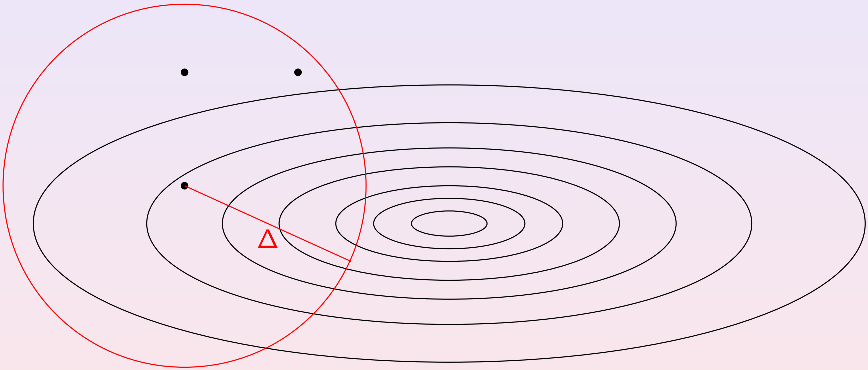
Coordinate Search (poll ordering)



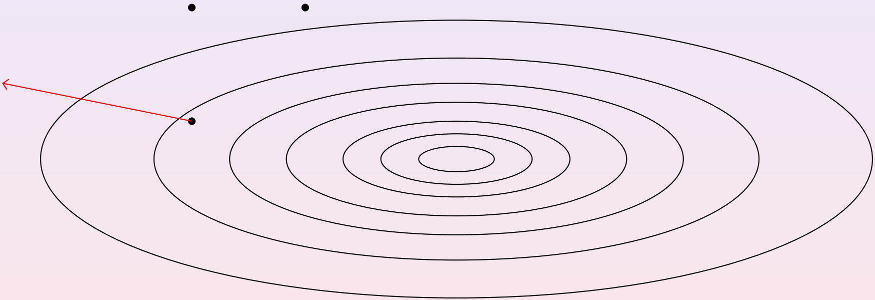
Coordinate Search (poll ordering)



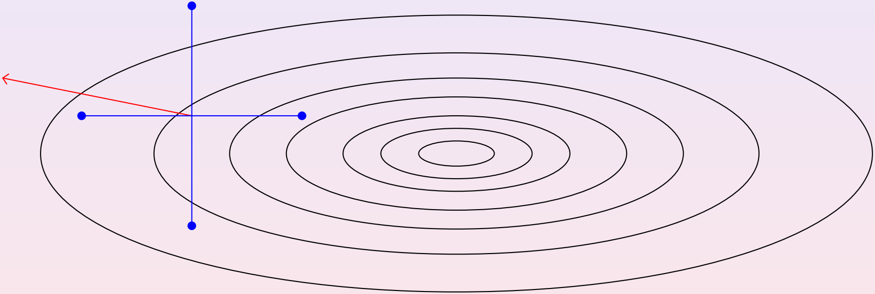
Coordinate Search (poll ordering)



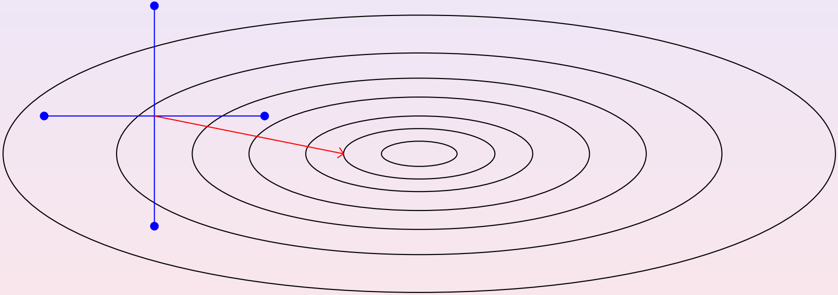
Coordinate Search (poll ordering)



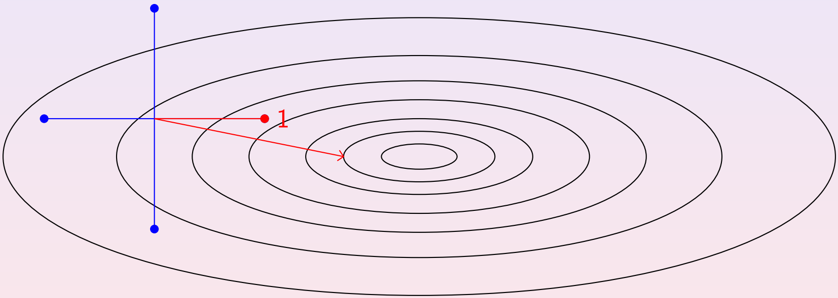
Coordinate Search (poll ordering)



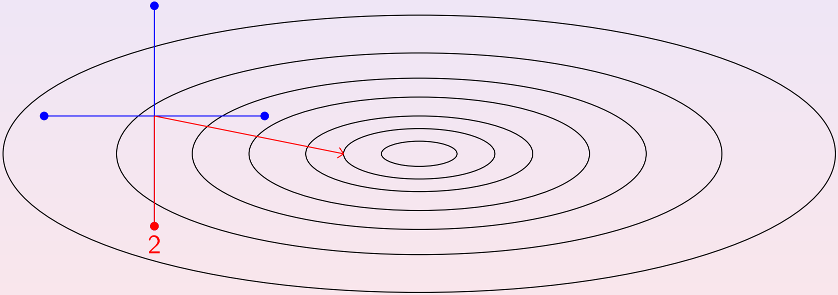
Coordinate Search (poll ordering)



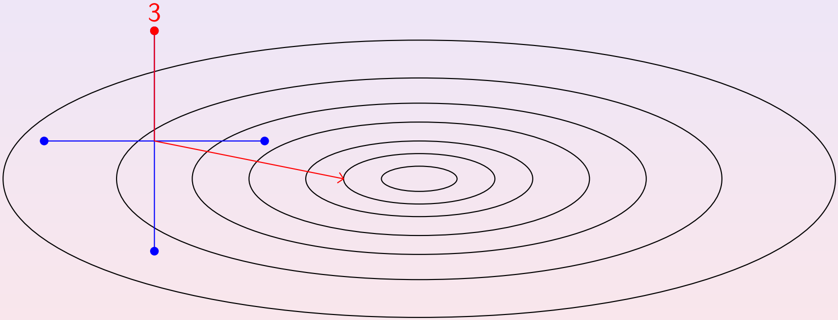
Coordinate Search (poll ordering)



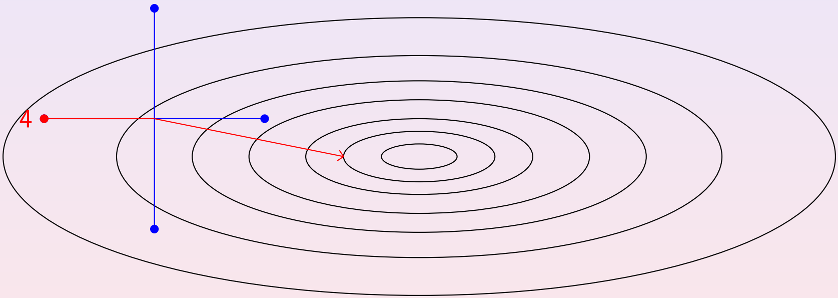
Coordinate Search (poll ordering)



Coordinate Search (poll ordering)



Coordinate Search (poll ordering)



In the context of direct search methods, one can use $-\nabla_s f(x_k)$ to:

1. Order the poll directions.

   Reduction of 50%.
2. Prune the poll directions (Abramson, Audet, and Dennis 04)

   Reduction of 10–40%.
3. Make a search step.

   Reduction of 15–30%.
4. Update the step size parameter.

   Reduction of 0–20% — less recommended.

Custódio and Vicente 04

In the context of direct search methods, one can use $-\nabla_s f(x_k)$ to:

1. Order the poll directions.

   Reduction of 50%.

2. Prune the poll directions (Abramson, Audet, and Dennis 04)

   Reduction of 10–40%.

3. Make a search step.

   Reduction of 15–30%.

4. Update the step size parameter.

   Reduction of 0–20% — less recommended.

Custódio and Vicente 04

In the context of direct search methods, one can use $-\nabla_s f(x_k)$ to:

1. Order the poll directions.
   Reduction of 50%.
2. Prune the poll directions (Abramson, Audet, and Dennis 04)
   Reduction of 10–40%.
3. Make a search step.
   Reduction of 15–30%.
4. Update the step size parameter.
   Reduction of 0–20% — less recommended.

Custódio and Vicente 04

In the context of direct search methods, one can use $-\nabla_s f(x_k)$ to:

1. Order the poll directions.

   Reduction of 50%.
2. Prune the poll directions (Abramson, Audet, and Dennis 04)

   Reduction of 10–40%.
3. Make a search step.

   Reduction of 15–30%.
4. Update the step size parameter.

   Reduction of 0–20% — less recommended.

Custódio and Vicente 04

In the context of direct search methods, one can use $-\nabla_s f(x_k)$ to:

1. Order the poll directions.

   Reduction of 50%.

2. Prune the poll directions (Abramson, Audet, and Dennis 04)

   Reduction of 10–40%.

3. Make a search step.

   Reduction of 15–30%.

4. Update the step size parameter.

   Reduction of 0–20% — less recommended.

Custódio and Vicente 04

Problem	Dimension	DS	DS poll ordering
bdqrtic	20	4120	2138
integreq	20	4244	1573
penalty2	10	496275	93192
srosenbr	20	649621	358656
tridia	20	6635	2828
vardim	10	86316	6550

The use of simplex gradients is validated by:

Theorem (Building simplex gradients)

It is possible to identify subsequences of iterates yielding Λ-poised sets.

Custódio and Vicente 04

In the context of direct search of non-smooth functions, the use of simplex gradients is validated by:

Theorem (Non-smooth context)

There exists a $B \subset D$ such that

$$f^o(x_*; d) \;\geq\; \langle \nabla_s f(x_*), d \rangle, \quad \forall \textit{poll directions } d \in B.$$

$$\neq \quad \nabla_s f(x_*) \in \{ v \in \mathbb{R}^n : \; f^o(x_*; d) \;\geq\; \langle v, d \rangle, \quad \forall d \in \mathbb{R}^n \}.$$

Custódio, Dennis, and Vicente 06

In the context of direct search of non-smooth functions, the use of simplex gradients is validated by:

Theorem (Non-smooth context)

There exists a $B \subset D$ such that

$$f^o(x_*; d) \; \geq \; \langle \nabla_s f(x_*), d \rangle, \quad \forall \textit{poll directions } d \in B.$$

$$\neq \quad \nabla_s f(x_*) \in \{ v \in \mathbb{R}^n : \; f^o(x_*; d) \; \geq \; \langle v, d \rangle, \quad \forall d \in \mathbb{R}^n \}.$$

Custódio, Dennis, and Vicente 06

In the context of direct search of non-smooth functions, the use of simplex gradients is validated by:

Theorem (Non-smooth context)

There exists a $B \subset D$ such that

$$f^o(x_*; d) \;\geq\; \langle \nabla_s f(x_*), d \rangle, \quad \forall \textit{poll directions } d \in B.$$

$$\neq \quad \nabla_s f(x_*) \in \{ v \in \mathbb{R}^n : \; f^o(x_*; d) \;\geq\; \langle v, d \rangle, \quad \forall d \in \mathbb{R}^n \}.$$

Custódio, Dennis, and Vicente 06

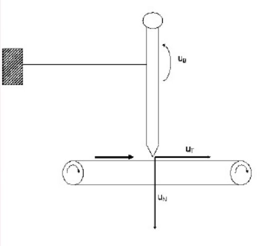
Code and References

Matlab code `sid-psm` available (constraints with derivatives).

1. A. L. Custódio and L. N. Vicente, *Using sampling and simplex derivatives in pattern search methods*, under review in SIAM Journal on Optimization.
2. A. L. Custódio, J. E. Dennis Jr., and L. N. Vicente, *Using simplex gradients of nonsmooth functions in direct search methods*, submitted, 2006.

Mechanical System with Potential Contact I

- Instability may induce oscillations leading to wear of the contacting surfaces.
- Prototype for the non-associative plastic/elastic behavior (soils, rocks, concrete).



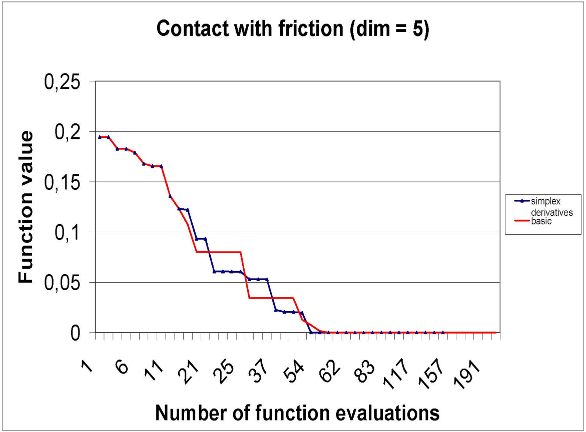
Mechanical System with Potential Contact II

- Simulation of a mechanical system with potential contact by friction.
- Minimization of the normal displacement, computed by numerical simulation (ODEs).
- There is non-differentiability, resulting from constraints treated implicitly (Coulomb friction law, classical conditions for unilateral contact).
- 5 variables (non-dimensional).
- Bounds on the variables and one nonlinear constraint (with known derivatives).
- Each function evaluation takes 1-2 sec., but not always succeed (increases 10-20 times the computational time).

Mechanical System with Potential Contact II

- Simulation of a mechanical system with potential contact by friction.
- Minimization of the normal displacement, computed by numerical simulation (ODEs).
- There is non-differentiability, resulting from constraints treated implicitly (Coulomb friction law, classical conditions for unilateral contact).
- 5 variables (non-dimensional).
- Bounds on the variables and one nonlinear constraint (with known derivatives).
- Each function evaluation takes 1-2 sec., but not always succeed (increases 10-20 times the computational time).

Mechanical System with Potential Contact III



Molecular Geometry Optimization — Physics

P. Alberto, F. Nogueira, H. Rocha, and L. N. Vicente, *Pattern search methods for user-provided points: Application to molecular geometry problems*, SIAM Journal on Optimization, 14 (2004) 1216–1236.

- The goal is to minimize the total energy of clusters (computed by heavy simulations).
- The function is possibly non-convex (low number of relative/local minimizers).
- Direct Search was applied in a parallel environment and took advantage of the problem information.
- $\longrightarrow$ Need for better approach to deal with the presence of non-convexity.

Global Optimization

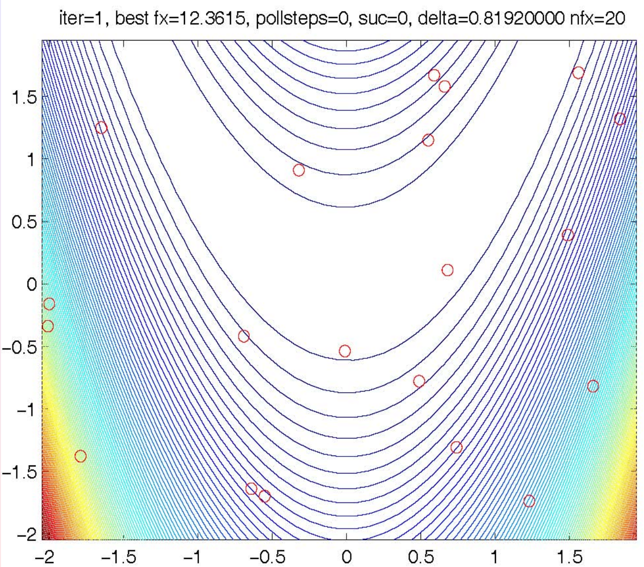
It is possible to adapt direct search to look for the best relative/local minimizer:

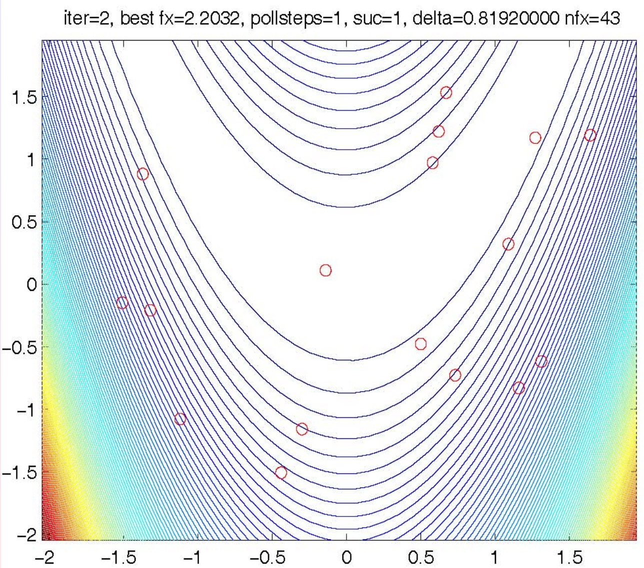
1. incorporating an heuristic in the search step,

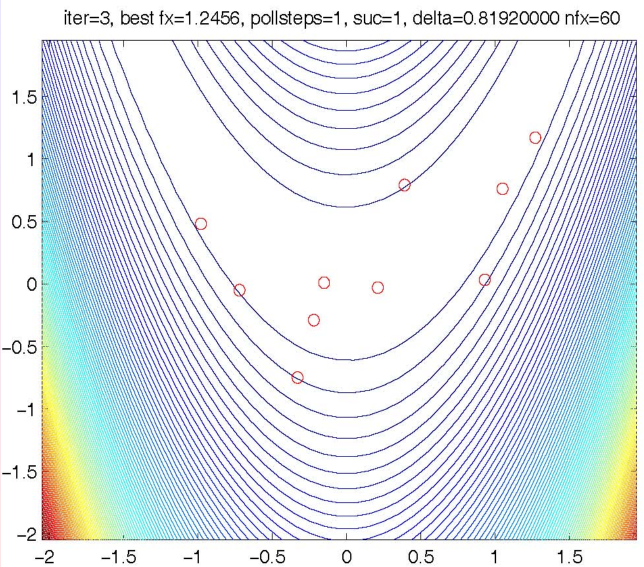
   preferably stochastic and population oriented (e.g. particle swarm)

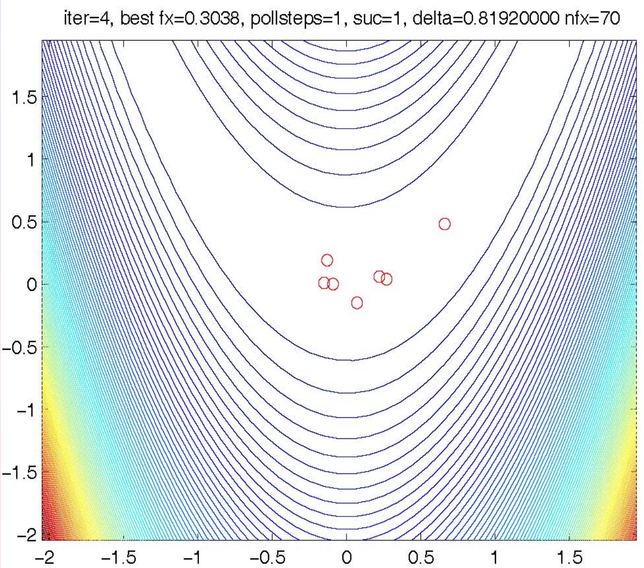
2. and, from a numerical point-of-view, 'postponing' the use of the poll step.

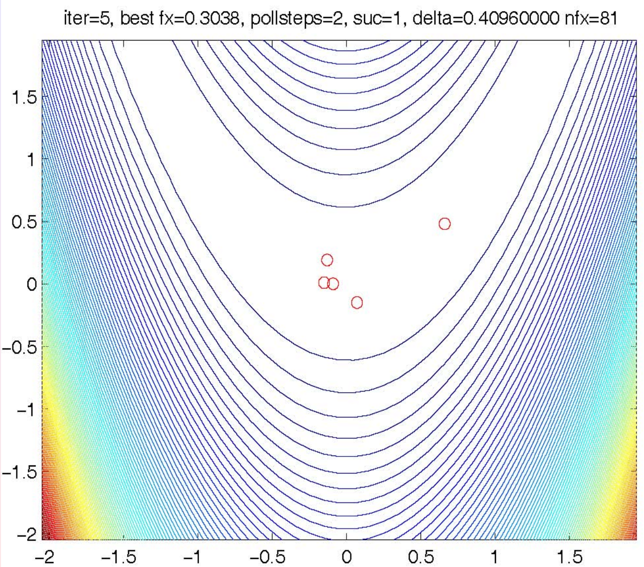
Vaz and Vicente 06

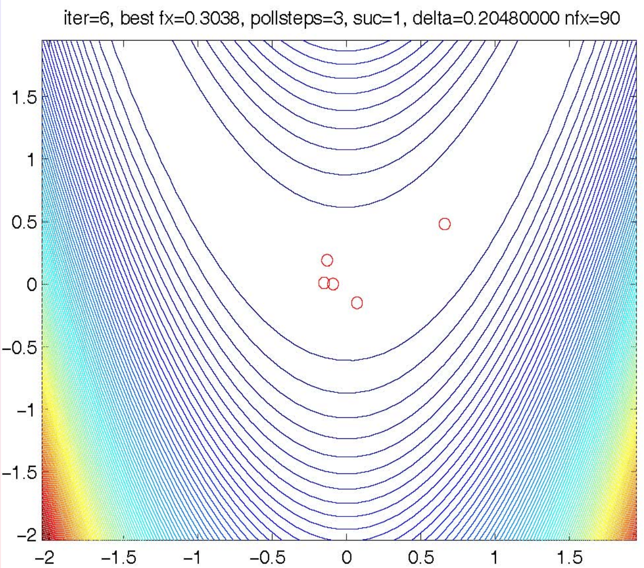

iter=1, best fx=12.3615, pollsteps=0, suc=0, delta=0.81920000 nfx=20
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5

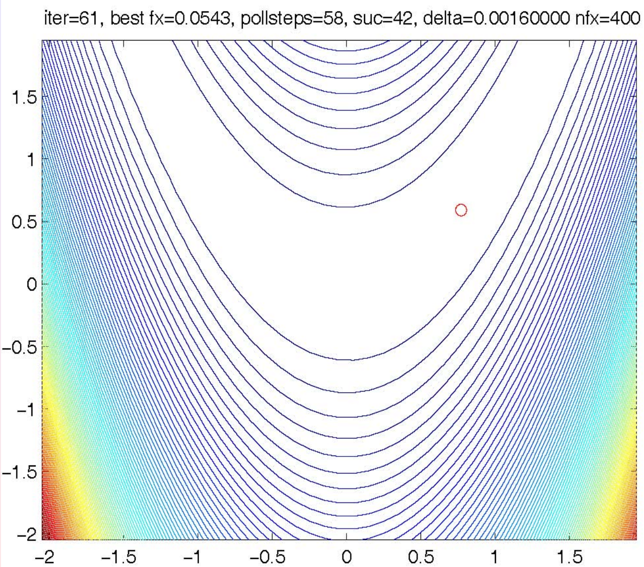

iter=2, best fx=2.2032, pollsteps=1, suc=1, delta=0.81920000 nfx=43
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5

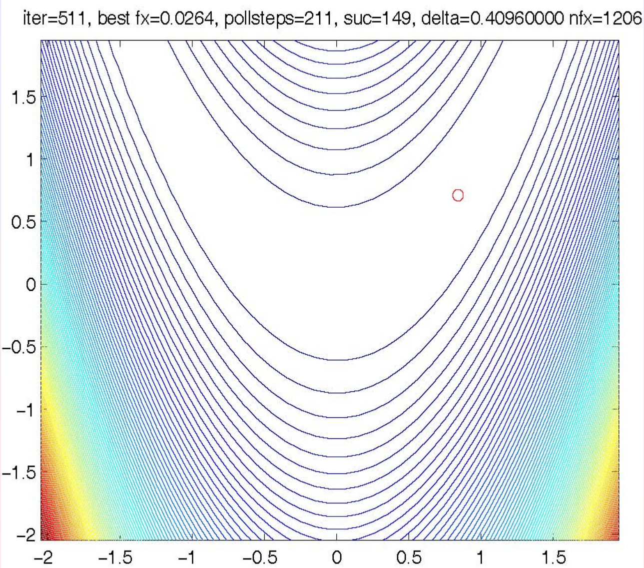

iter=3, best fx=1.2456, pollsteps=1, suc=1, delta=0.81920000 nfx=60
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5

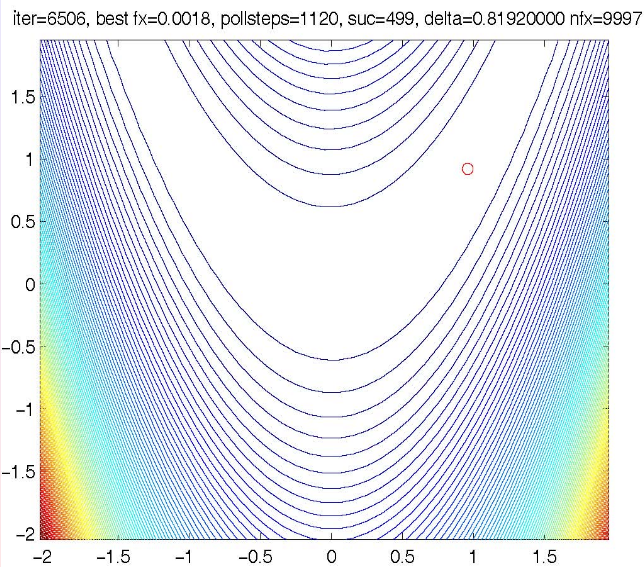

iter=4, best fx=0.3038, pollsteps=1, suc=1, delta=0.81920000 nfx=70
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5

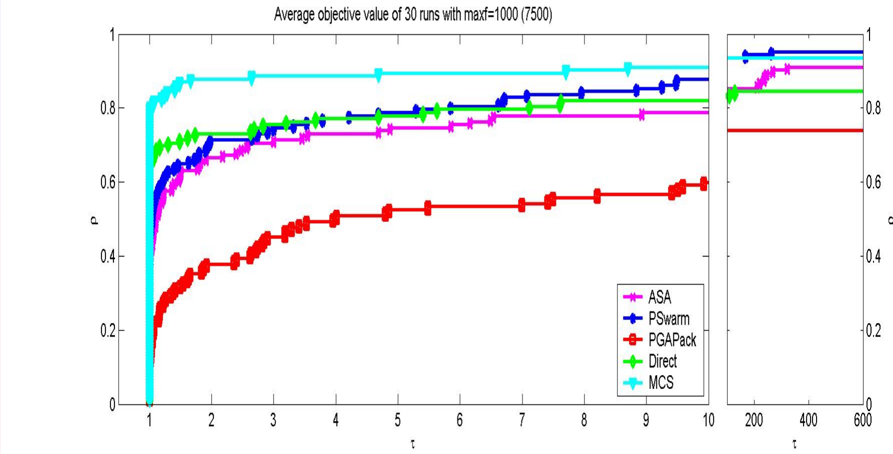

iter=5, best fx=0.3038, pollsteps=2, suc=1, delta=0.40960000 nfx=81
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5

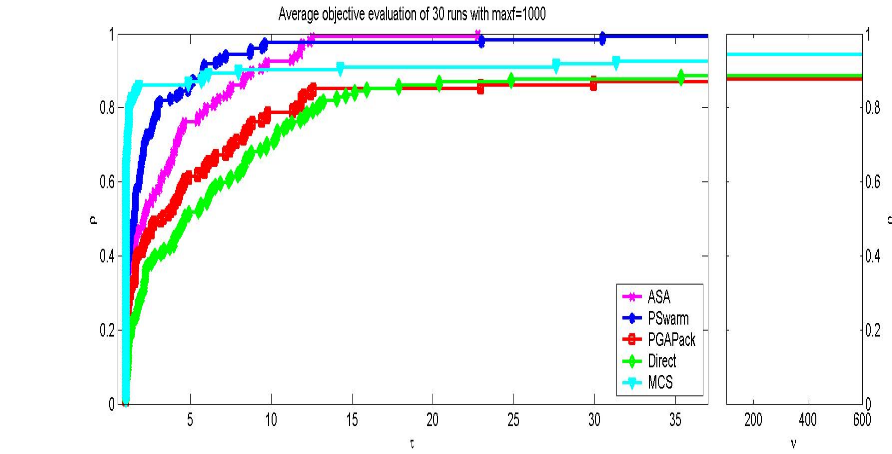

iter=6, best fx=0.3038, pollsteps=3, suc=1, delta=0.20480000 nfx=90
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5


iter=61, best fx=0.0543, pollsteps=58, suc=42, delta=0.00160000 nfx=400
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5


iter=511, best fx=0.0264, pollsteps=211, suc=149, delta=0.40960000 nfx=1206
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5


iter=6506, best fx=0.0018, pollsteps=1120, suc=499, delta=0.81920000 nfx=9997
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5


Average objective value of 30 runs with maxf=1000 (7500)
ρ
τ
0
0.2
0.4
0.6
0.8
1
1
2
3
4
5
6
7
8
9
10
200
400
600
ASA
PSwarm
PGAPack
Direct
MCS


Average objective evaluation of 30 runs with maxf=1000
ASA
PSwarm
PGAPack
Direct
MCS
ρ
τ
ν
0
0.2
0.4
0.6
0.8
1
5
10
15
20
25
30
35
200
400
600

Code and Reference

C code `PSwarm` available (bound constraints).

A. Ismael F. Vaz and L. N. Vicente, *A particle swarm pattern search method for bound constrained nonlinear optimization*, under review in Journal of Global Optimization.

Parameter Estimation in Astrophysics

The knowledge of the internal structure and evolution of a star is dependent on six variables: stellar mass (M), initial individual abundance of helium (Y), hydrogen (X), age (t), and two other parameters (a, ov).

These variables are not available by observation (except for Sun).

What is observable: surface temperature, called effective temperature (teff), and total stellar luminosity (lum).

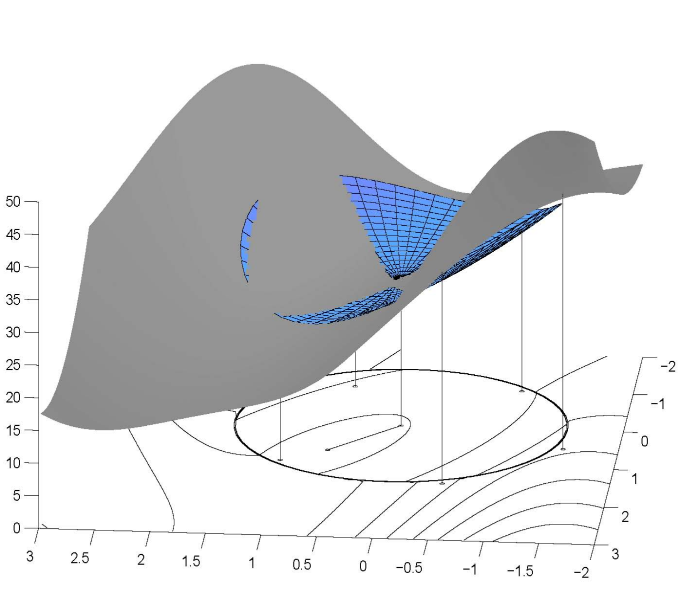
Parameter Estimation in Astrophysics

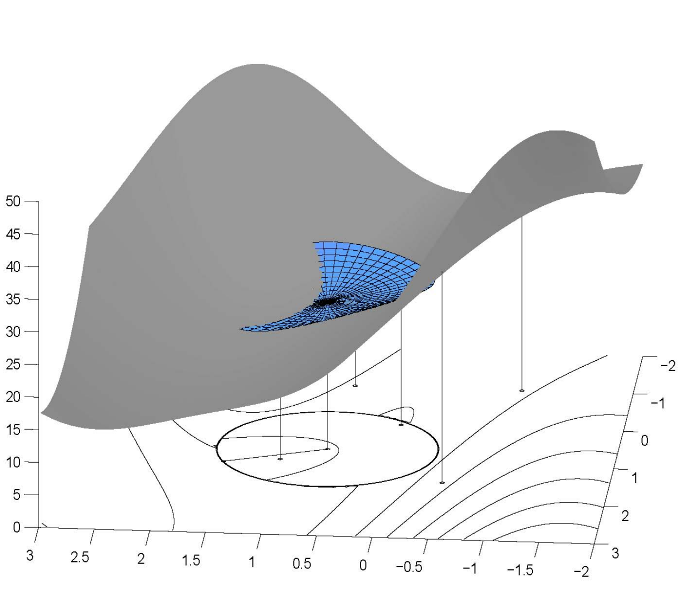
To model a star correctly we must estimate the unknown six parameters from the observations (teff and lum).

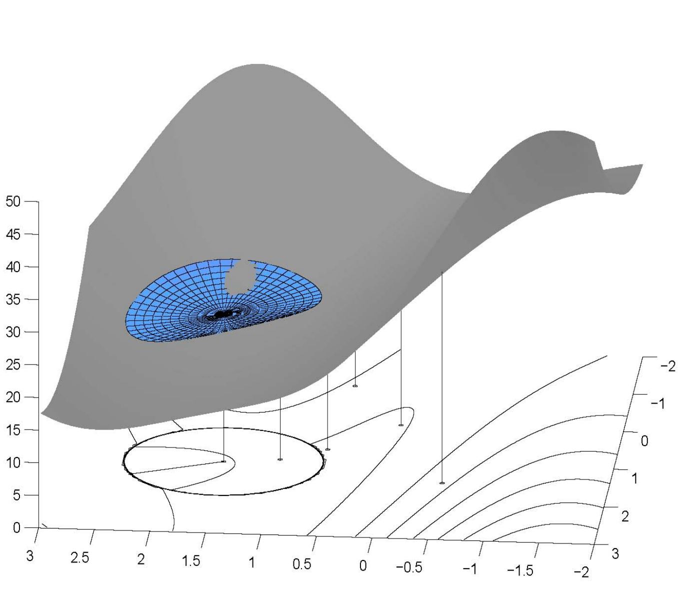
The equations of internal structure are five: conservation of mass and energy, hydrostatic equilibrium, energy transport, production and destruction of chemical elements by thermonuclear reactions.

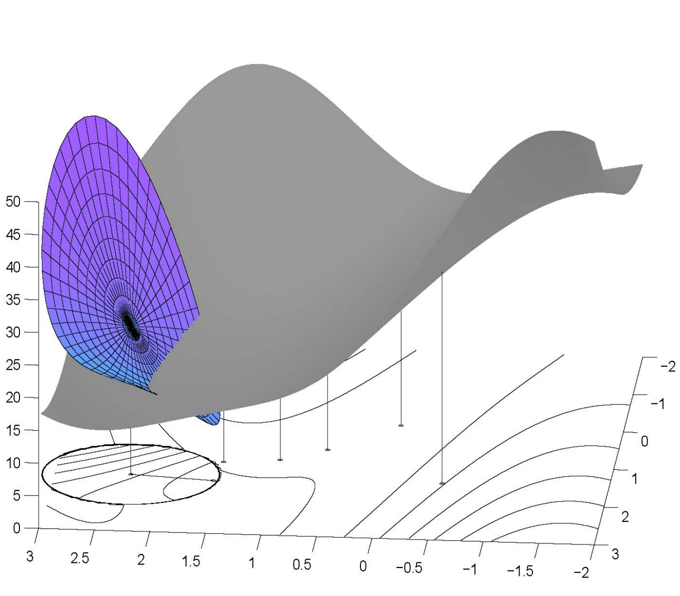
Each function evaluation takes 10-100 sec., when well succeeded.

`PSwarm` on the Sun: $f = 4833.737593 \longrightarrow f = 1.465536$ in $\#fevals = 938$, $\#poll = 29$, $\#sucpoll = 18$.


50
45
40
35
30
25
20
15
10
5
0
3
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1
0
1
2
3


50
45
40
35
30
25
20
15
10
5
0
3
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1
0
1
2
3


50
45
40
35
30
25
20
15
10
5
0
3
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1
0
1
2
3


50
45
40
35
30
25
20
15
10
5
0
3
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1
0
1
2
3

Trust-Region Methods in DFO

- Form a model (interpolation or regression)

$$m(\Delta x) \;=\; f(x_k) + \langle \nabla_s f(x_k), \Delta x \rangle + 1/2 \langle \Delta x, \nabla_s^2 f(x_k) \Delta x \rangle$$

  based on a Λ-poised sampling set.

Trust-Region Methods in DFO

- Calculate a step Δx_k, by approximately solving

$$\min_{\Delta x \in B(x_k;\Delta_k)} \quad m(\Delta x).$$

- Set x_{k+1} to $x_k + \Delta x_k$ (success) or to x_k (unsuccess) and update Δ_k depending on the value of

$$\rho_k \;=\; \frac{f(x_k) - f(x_k + \Delta x_k)}{m(0) - m(\Delta x_k)}.$$

Trust-Region Methods in DFO

- Calculate a step Δx_k, by approximately solving

$$\min_{\Delta x \in B(x_k;\Delta_k)} \quad m(\Delta x).$$

- Set x_{k+1} to $x_k + \Delta x_k$ (success) or to x_k (unsuccess) and update Δ_k depending on the value of

$$\rho_k \;=\; \frac{f(x_k) - f(x_k + \Delta x_k)}{m(0) - m(\Delta x_k)}.$$

Trust-Region Methods in DFO

- Incorporate a stationary step (1st or 2nd order) when the 'stationarity' of the model is small.

  $\longrightarrow$ Internal cycle of reductions of Δ_k (NEW: same radius).

- Do not reduce Δ_k if the sampling set is not sufficiently well poised (fully linear/quadratic models).

- NEW: Only increase Δ_k when it is small relative to the measure of stationarity:

$$\Delta_k \leq \textit{constant} \times \|\nabla_s f(x_k)\|.$$

Trust-Region Methods in DFO

- Incorporate a stationary step (1st or 2nd order) when the 'stationarity' of the model is small.

  $\longrightarrow$ Internal cycle of reductions of Δ_k (NEW: same radius).

- Do not reduce Δ_k if the sampling set is not sufficiently well poised (fully linear/quadratic models).

- NEW: Only increase Δ_k when it is small relative to the measure of stationarity:

$$\Delta_k \leq \textit{constant} \times \|\nabla_s f(x_k)\|.$$

Trust-Region Methods in DFO

- Incorporate a stationary step (1st or 2nd order) when the 'stationarity' of the model is small.

  $\longrightarrow$ Internal cycle of reductions of Δ_k (NEW: same radius).

- Do not reduce Δ_k if the sampling set is not sufficiently well poised (fully linear/quadratic models).

- NEW: Only increase Δ_k when it is small relative to the measure of stationarity:

$$\Delta_k \ \leq \ constant \times \|\nabla_s f(x_k)\|.$$

Trust-Region Methods in DFO

Iterations can be:

1. **successful:** $\rho_k \geq \eta_1$.

   $\longrightarrow x_k + \Delta x_k$ is accepted and Δ_k is retained or increased.

2. **acceptable:** $\eta_1 > \rho_k \geq \eta_0$ and m_k is fully linear/quadratic.
   $\longrightarrow x_k + \Delta x_k$ is accepted and Δ_k is decreased.

3. **model-improving:** $\rho_k < \eta_1$ and m_k is not fully linear/quadratic.
   $\longrightarrow$ The model is improved. The new point might be included in the sample set but is rejected as a new iterate.

4. **unsuccessful:**
   $\rho_k < \eta_0$ and m_k is fully linear/quadratic.
   $\longrightarrow x_k + \Delta x_k$ is rejected and Δ_k is decreased.

Trust-Region Methods in DFO

Iterations can be:

1. **successful:** $\rho_k \geq \eta_1$.

   $\longrightarrow$ $x_k + \Delta x_k$ is accepted and Δ_k is retained or increased.

2. **acceptable:** $\eta_1 > \rho_k \geq \eta_0$ and m_k is fully linear/quadratic.
   $\longrightarrow$ $x_k + \Delta x_k$ is accepted and Δ_k is decreased.

3. **model-improving:** $\rho_k < \eta_1$ and m_k is not fully linear/quadratic.
   $\longrightarrow$ The model is improved. The new point might be included in the sample set but is rejected as a new iterate.

4. **unsuccessful:**
   $\rho_k < \eta_0$ and m_k is fully linear/quadratic.
   $\longrightarrow$ $x_k + \Delta x_k$ is rejected and Δ_k is decreased.

Trust-Region Methods in DFO

Iterations can be:

1. **successful:** $\rho_k \geq \eta_1$.

   $\longrightarrow$ $x_k + \Delta x_k$ is accepted and Δ_k is retained or increased.

2. **acceptable:** $\eta_1 > \rho_k \geq \eta_0$ and m_k is fully linear/quadratic.
   $\longrightarrow$ $x_k + \Delta x_k$ is accepted and Δ_k is decreased.

3. **model-improving:** $\rho_k < \eta_1$ and m_k is not fully linear/quadratic.
   $\longrightarrow$ The model is improved. The new point might be included in the sample set but is rejected as a new iterate.

4. **unsuccessful:**
   $\rho_k < \eta_0$ and m_k is fully linear/quadratic.
   $\longrightarrow$ $x_k + \Delta x_k$ is rejected and Δ_k is decreased.

Trust-Region Methods in DFO

Iterations can be:

1. **successful:** $\rho_k \geq \eta_1$.

   $\longrightarrow x_k + \Delta x_k$ is accepted and Δ_k is retained or increased.

2. **acceptable:** $\eta_1 > \rho_k \geq \eta_0$ and m_k is fully linear/quadratic.
   $\longrightarrow x_k + \Delta x_k$ is accepted and Δ_k is decreased.

3. **model-improving:** $\rho_k < \eta_1$ and m_k is not fully linear/quadratic.
   $\longrightarrow$ The model is improved. The new point might be included in the sample set but is rejected as a new iterate.

4. **unsuccessful:**
   $\rho_k < \eta_0$ and m_k is fully linear/quadratic.
   $\longrightarrow x_k + \Delta x_k$ is rejected and Δ_k is decreased.

Theorem (Global convergence (1st order) — TRM)

$$\|\nabla f(x_k)\| \longrightarrow 0.$$

- $f \in C^1(L(x_0)) \cdots$
- Steps must be as good as the Cauchy step.
- Compactness of $L(x_0)$ is not necessary.

Conn, Scheinberg, and Toint 97

Conn, Scheinberg, and Vicente 06 (simple decrease, more practical algorithm)

Theorem (Global convergence (2nd order) — TRM)

$$\max\left\{\|\nabla f(x_k)\|, -\lambda_{min}[\nabla^2 f(x_k)]\right\} \longrightarrow 0.$$

- $f \in C^2(L(x_0)) \cdots$
- Steps must be as good as the optimal step.
- Compactness of $L(x_0)$ is not necessary.

Conn, Scheinberg, and Vicente 06 (including simple decrease, practical algorithm)

Problem	Dim.	DS poll ordering	TRM	NM
bdqrtic	20	2138	7426	32409
integreq	20	1573	1389	977 (worse f)
penalty2	10	93192	1418	6020
srosenbr	20	358656	25011	12504 (worse f)
tridia	20	2828	2071	12111 (better f)
vardim	10	6550	2626	1223

References

1. 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*, submitted, 2006.

2. A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.

References

1. A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, in preparation for SIAM/MPS Book Series on Optimization. ...ready in 2017...

Some Open Questions in DFO

1. Prove global convergence for the *wedge* trust-region method (Marazzi and Nocedal 02).
2. Understand better direct search for nonlinear constraints (with/without derivatives).
3. Develop a TR method for nonlinear constraints, globally convergent.

July 23-25, 2007 — University of Porto, Portugal

http://www.fep.up.pt/opti2007