A multistart multisplit direct search methodology for global optimization

Ismael Vaz (Univ. Minho)
Luis Nunes Vicente (Univ. Coimbra)

IPAM, Optimization and Optimal Control for Complex Energy and Property Landscapes

October, 2017

Outline

1. LOCAL: Probabilistic direct search.

   Deterministic case first. Motivation, performance, complexity.

Outline

1. LOCAL: Probabilistic direct search.

   Deterministic case first. Motivation, performance, complexity.

2. GLOBAL: By multistarting & multispliting probabilistic DS.

   How to split and merge runs. Use of non-convex modelling.

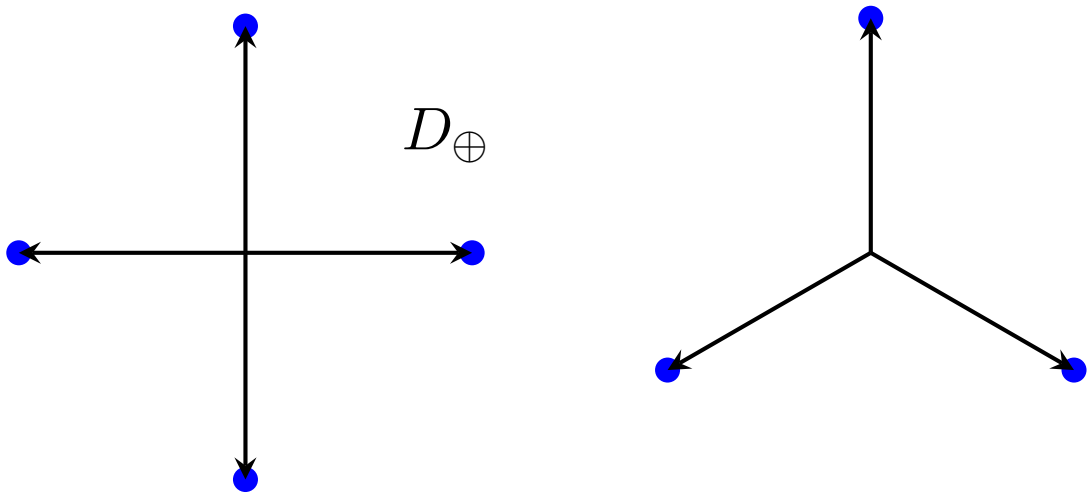
Direct-search methods

Definition

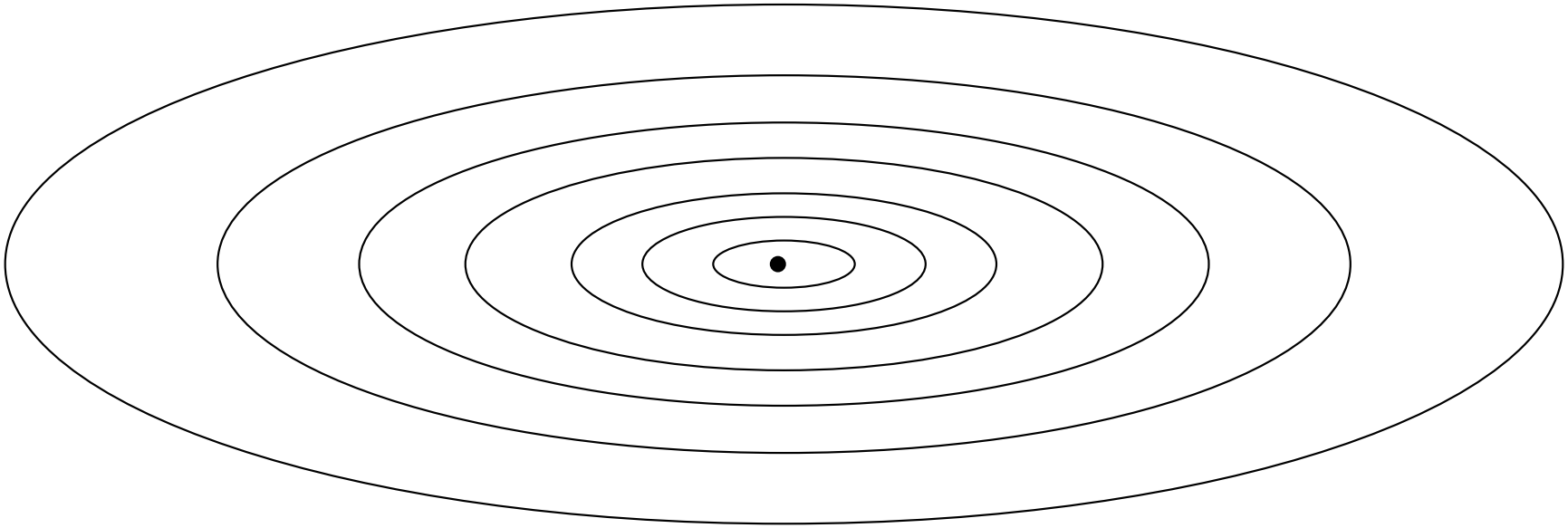
- *Sample the objective function at a finite number of points at each iteration.*
- *Achieve descent by moving in the direction of potentially better points.*
- *In the smooth and deterministic case, these points are defined by directions in positive spanning sets (PSS):*

Definition

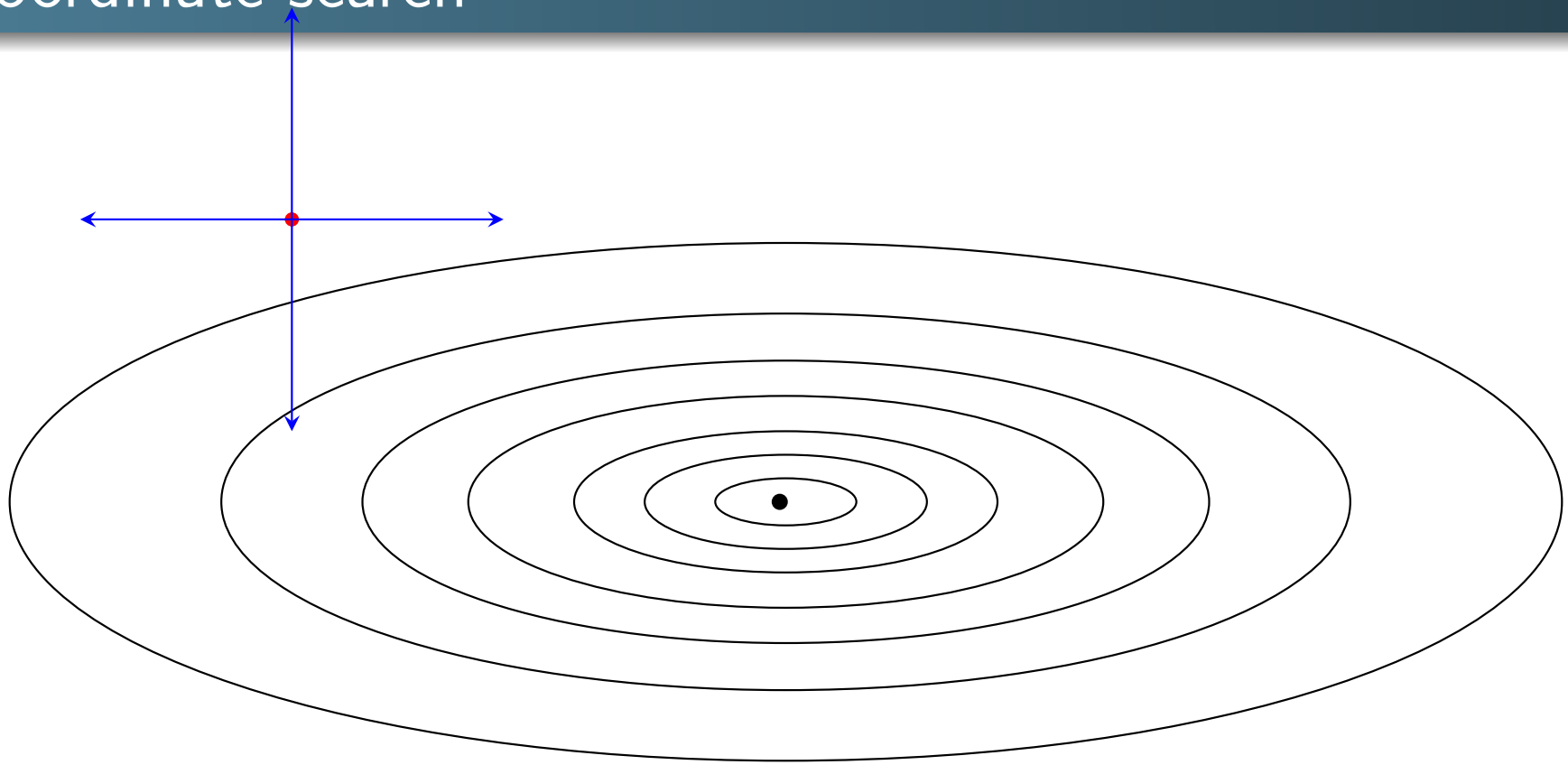
- *Sample the objective function at a finite number of points at each iteration.*
- *Achieve descent by moving in the direction of potentially better points.*
- *In the smooth and deterministic case, these points are defined by directions in positive spanning sets (PSS):*



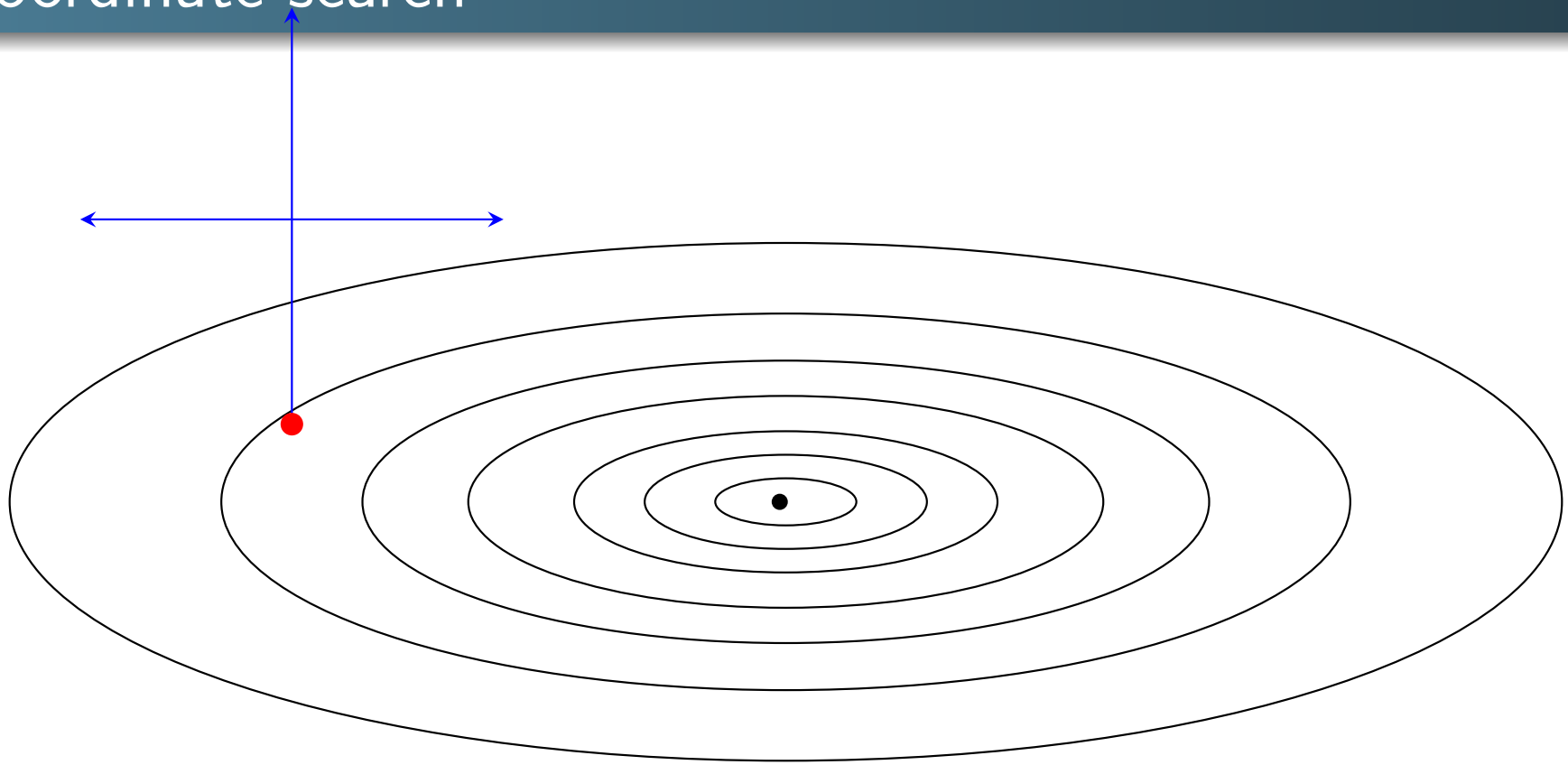
Coordinate search



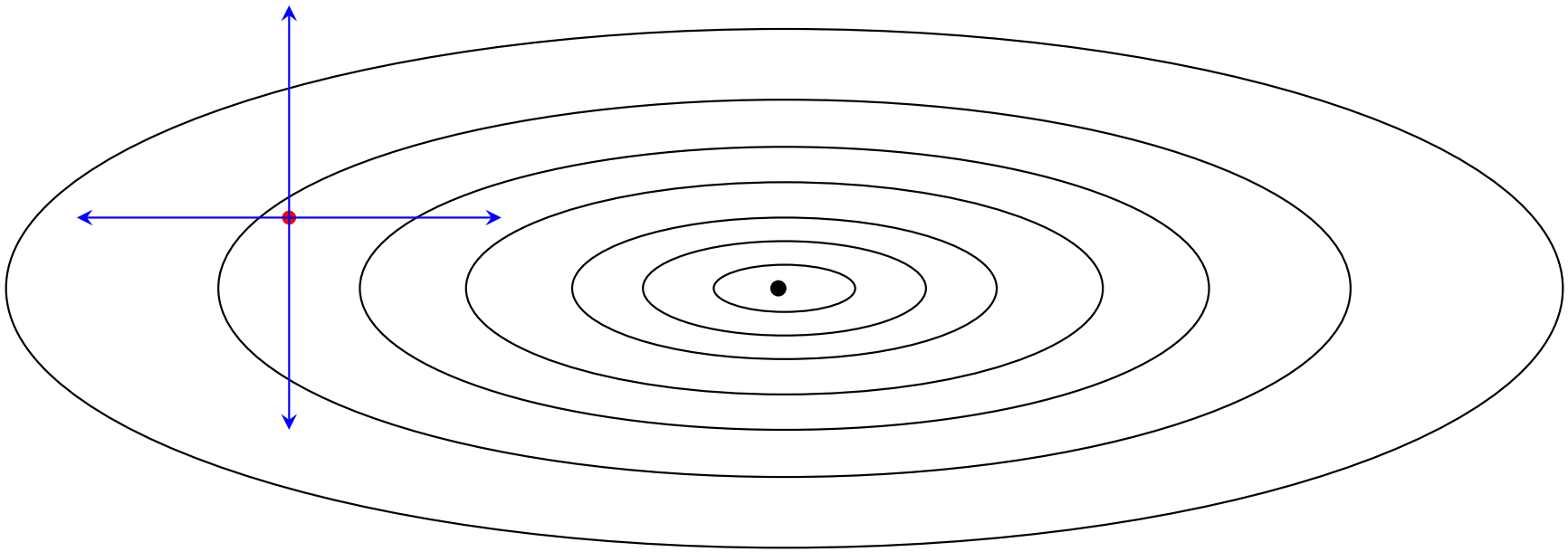
Coordinate search



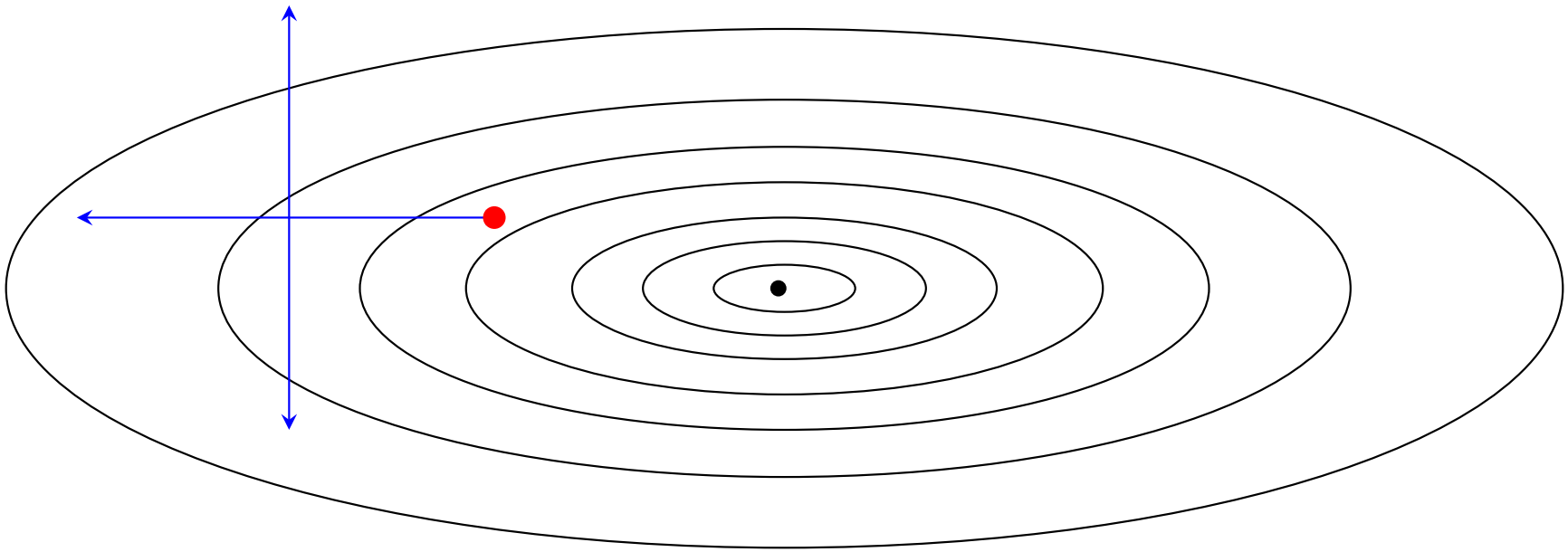
Coordinate search



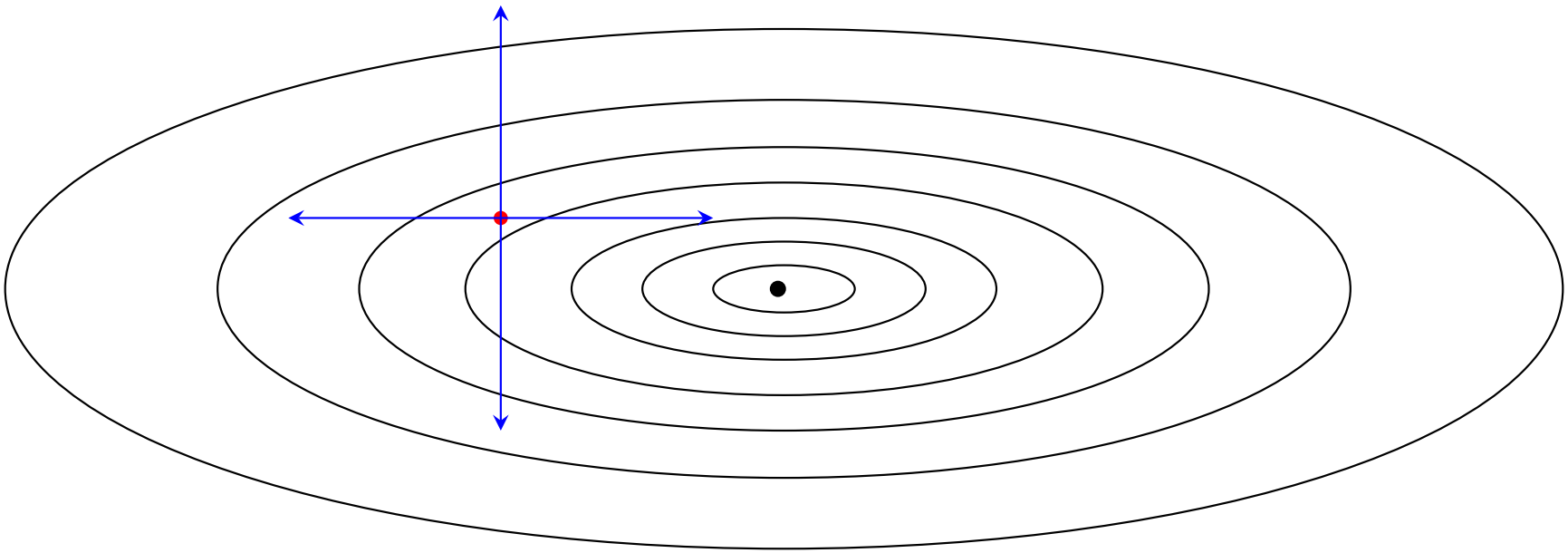
Coordinate search



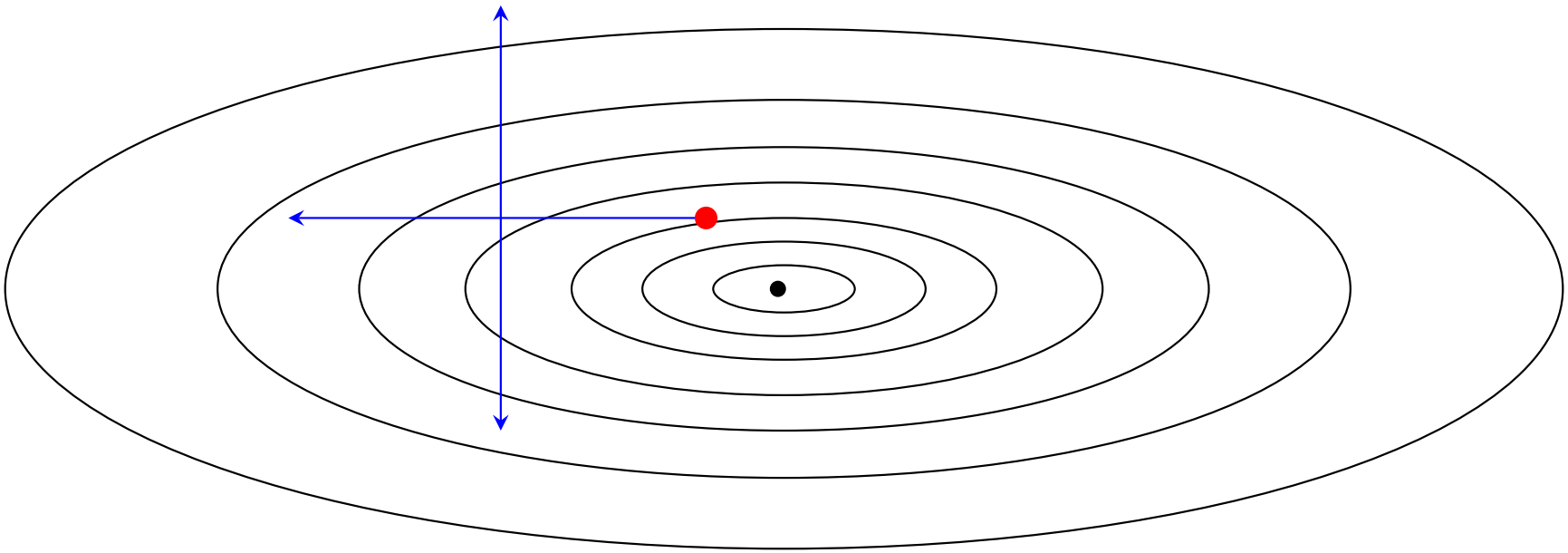
Coordinate search



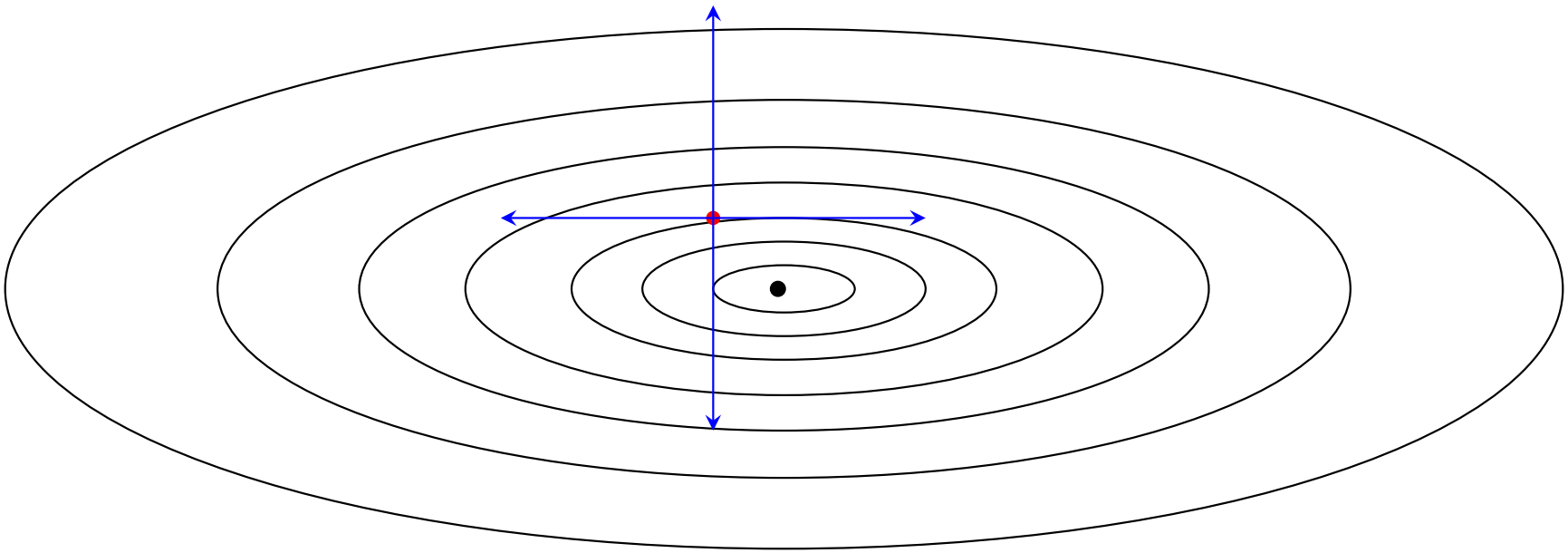
Coordinate search



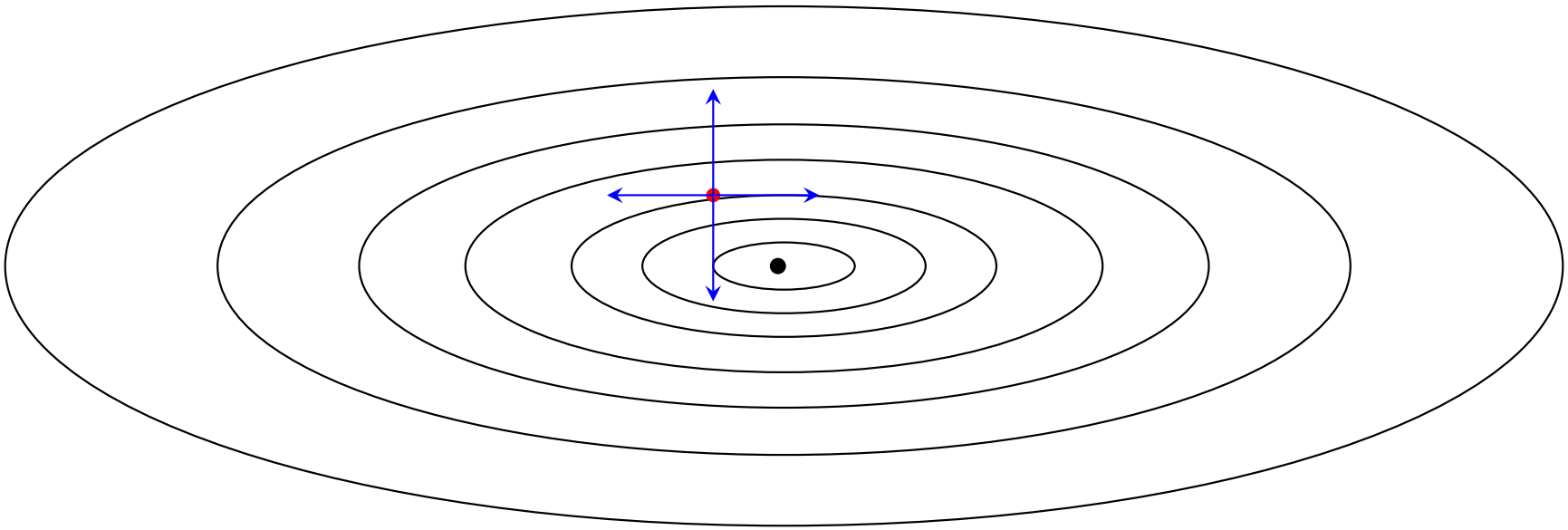
Coordinate search



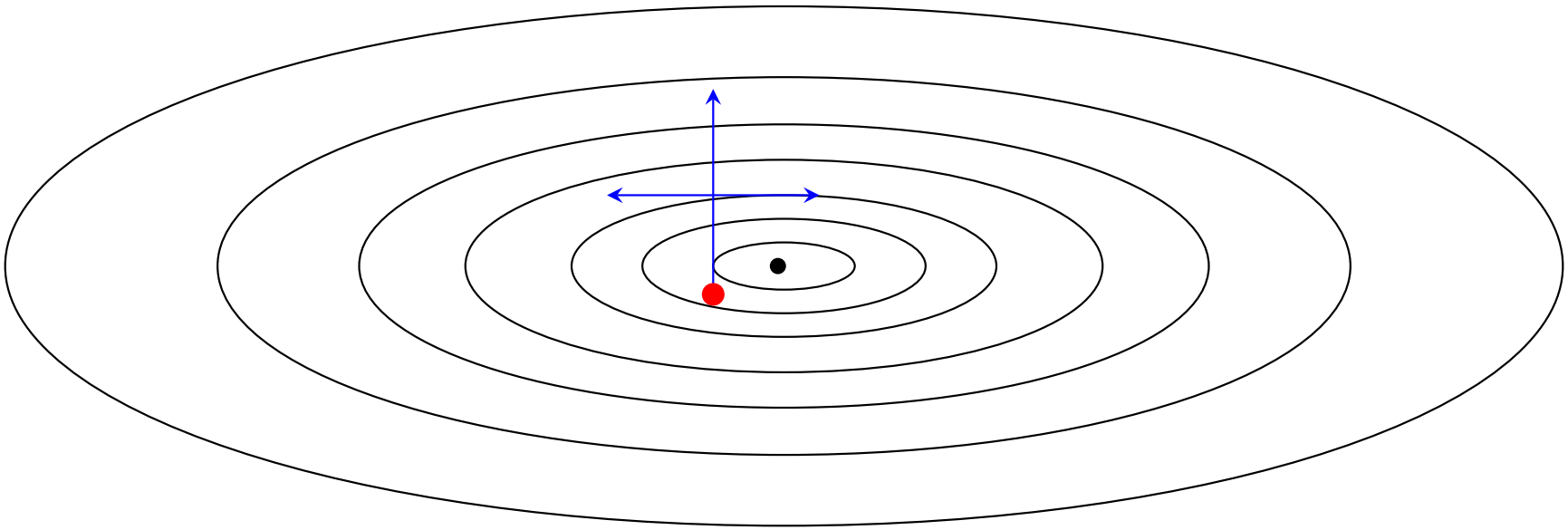
Coordinate search



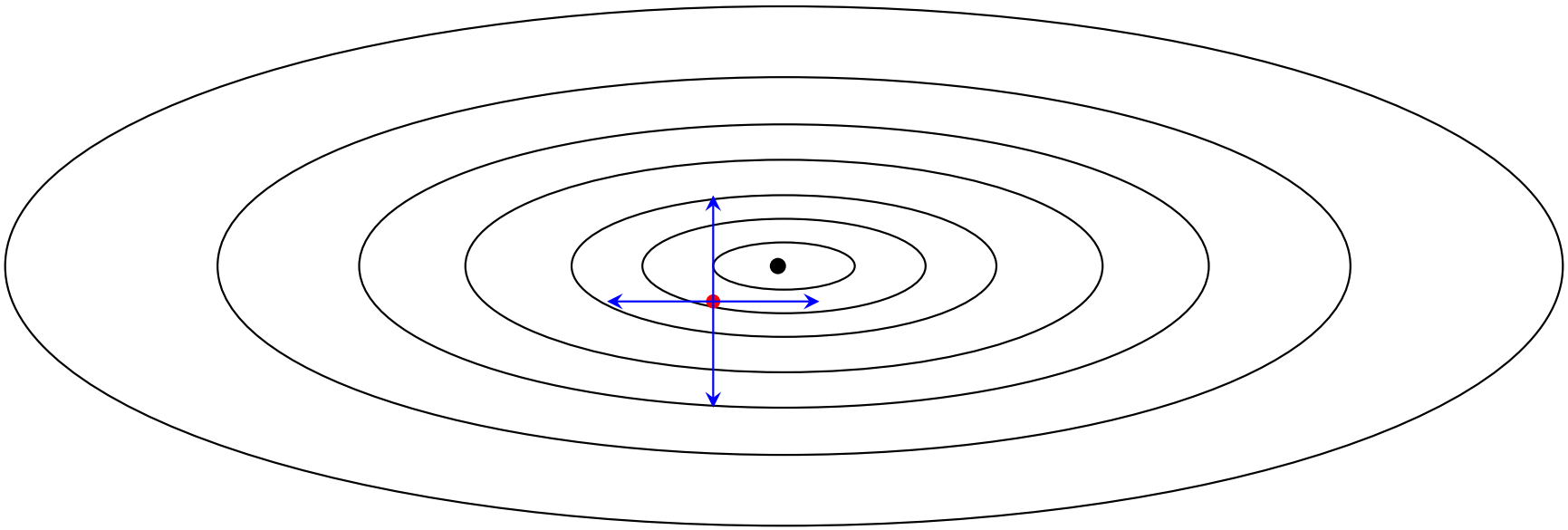
Coordinate search



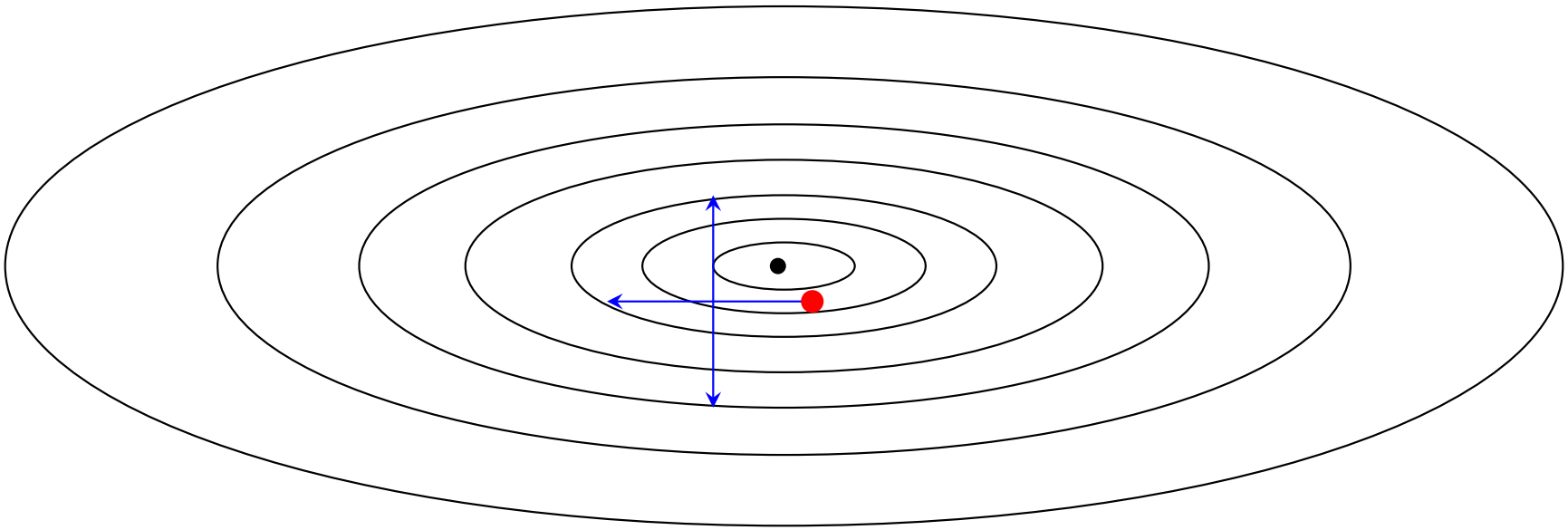
Coordinate search



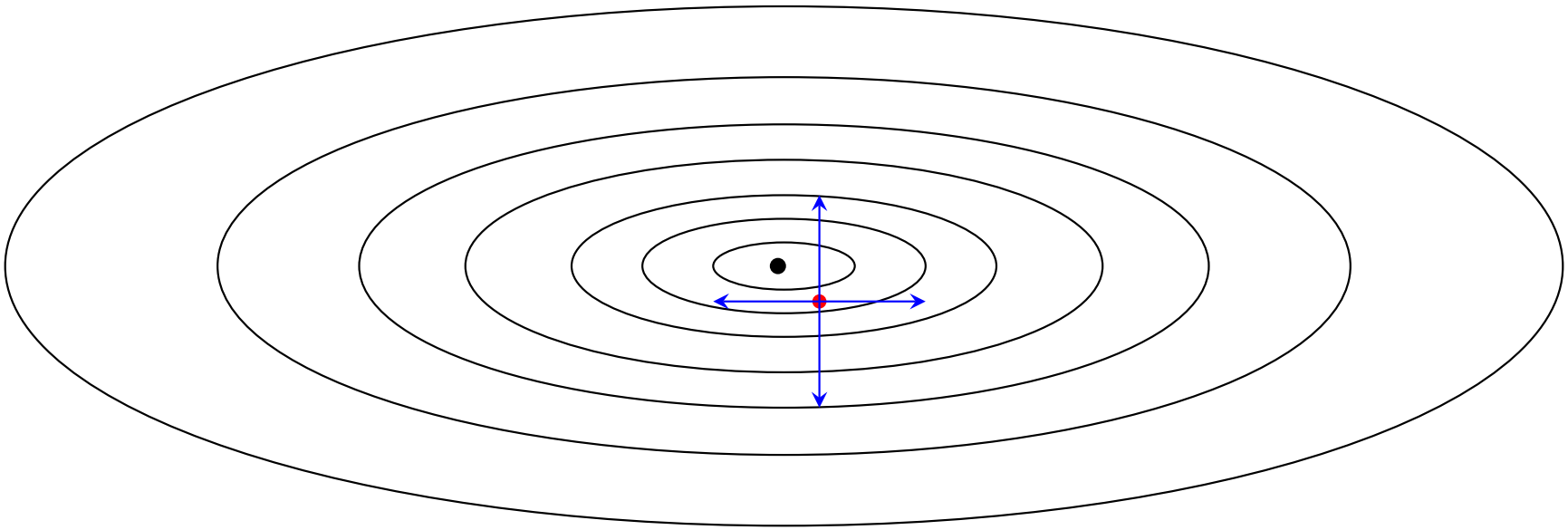
Coordinate search



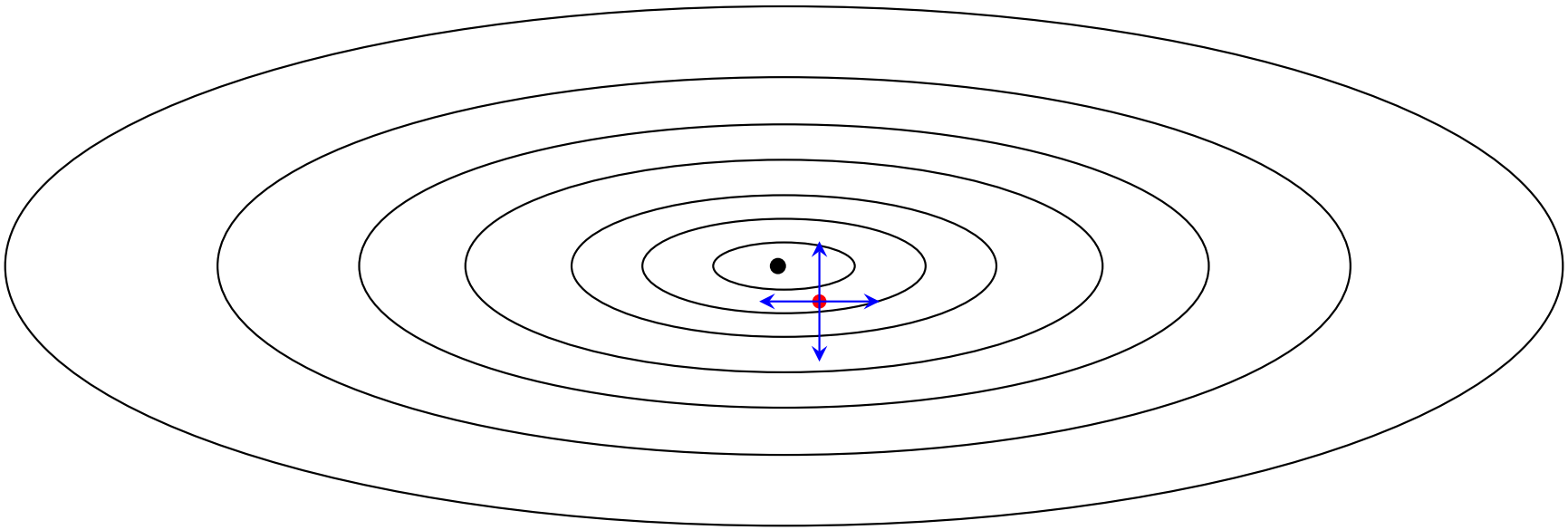
Coordinate search



Coordinate search



Coordinate search



A class of DS methods

Choose: x_0 and α_0.

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

- **Search step (optional)**

A class of DS 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 $x_k + \alpha_k d_k$ $(d_k \in D_k)$:

$$f(x_k + \alpha_k d_k) < f(x_k) - \rho(\alpha_k) \qquad \text{like } \rho(\alpha) = \alpha^2/2.$$

A class of DS methods

Choose: x_0 and α_0.

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

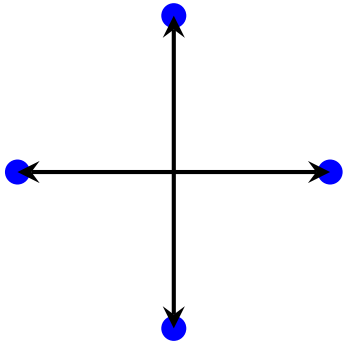
- **Search step (optional)**
- **Poll step:** Select D_k PSS and find $x_k + \alpha_k d_k$ $(d_k \in D_k)$:

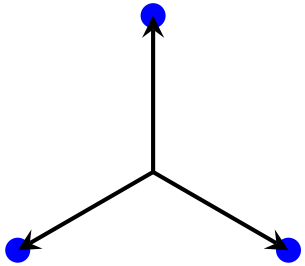
$$f(x_k + \alpha_k d_k) < f(x_k) - \rho(\alpha_k) \qquad \text{like } \rho(\alpha) = \alpha^2/2.$$

- **SUCCESS:** Move $x_{k+1} = x_k + \alpha_k d_k$ and possibly increase $\alpha_{k+1} = \gamma\, \alpha_k$ $(\gamma = 1 \text{ or } 2)$.
- **UNSUCCESS:** Stay $x_{k+1} = x_k$ and decrease $\alpha_{k+1} = \theta\, \alpha_k$ $(\theta = 1/2)$.

Deterministic approach

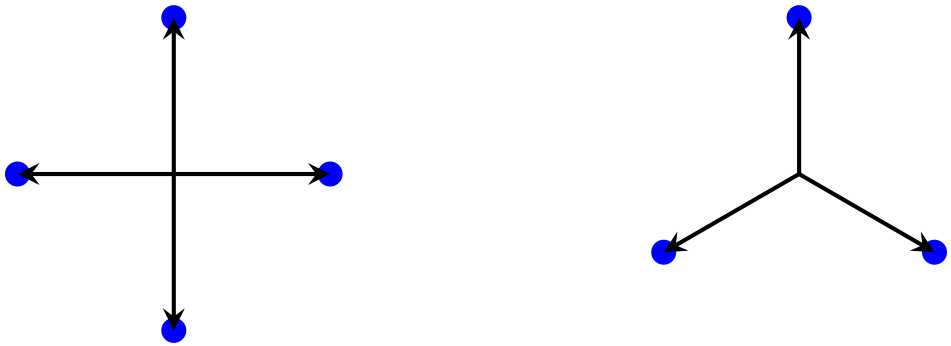
- Positive spanning set (PSS)





Deterministic approach

- Positive spanning set (PSS)

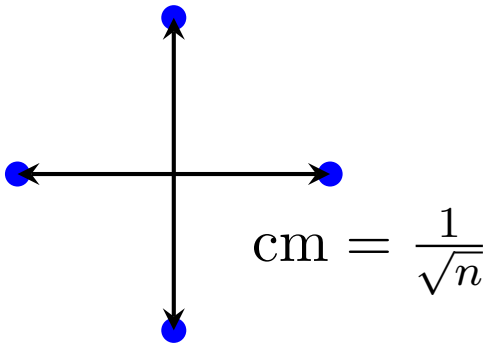


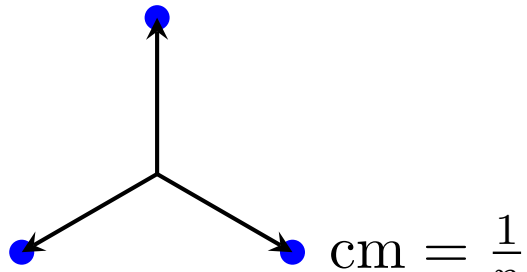
- Cosine measure of a PSS D

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

Deterministic approach

- Positive spanning set (PSS)



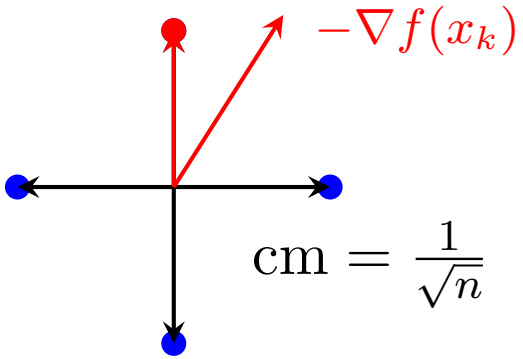


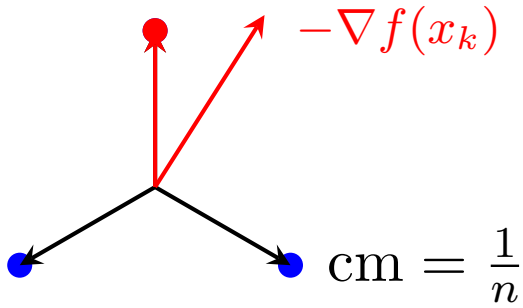
- Cosine measure of a PSS D

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

Deterministic approach

- Positive spanning set (PSS)





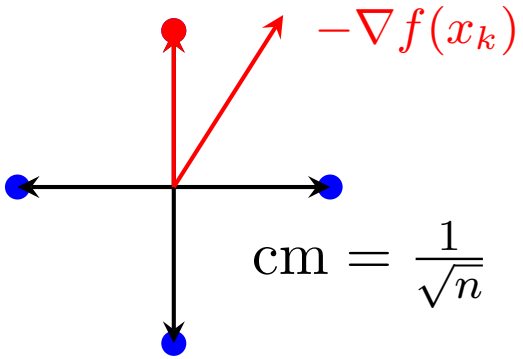
- Cosine measure of a PSS D

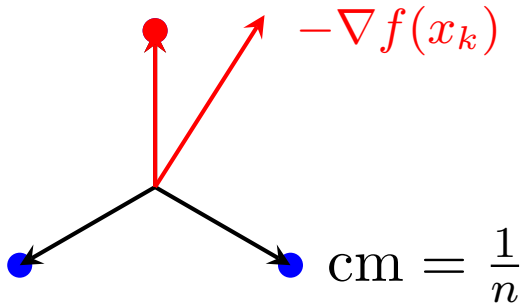
$$\mathrm{cm}(D) = \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{d^\top v}{\|d\| \|v\|} > 0.$$

- Thus $\exists\, d \in D$ descent when $\nabla f(x_k) \neq 0$.

Deterministic approach

- Positive spanning set (PSS)





- Cosine measure of a PSS D

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

- Thus $\exists\, d \in D$ descent when $\nabla f(x_k) \neq 0$.

  $\Longrightarrow$ α_k small leads to success!

WCC of DS (smooth case)

Insight: $\underbrace{(\text{decrease in } f) \geq \mathcal{O}(\alpha_k^2)}_{\text{success}} \geq \cdots \geq \underbrace{\mathcal{O}(\alpha_{k_u}^2) \geq \mathcal{O}(\|\nabla f(x_{k_u})\|^2)}_{\text{unsuccess}}$

Kolda, Lewis, Torczon, 2003 SIREV

WCC of DS (smooth case)

Insight: $\underbrace{(\text{decrease in } f) \;\geq\; \mathcal{O}(\alpha_k^2)}_{\text{success}} \;\geq\; \cdots \;\geq\; \underbrace{\mathcal{O}(\alpha_{k_u}^2) \;\geq\; \mathcal{O}(\|\nabla f(x_{k_u})\|^2)}_{\text{unsuccess}}$

Kolda, Lewis, Torczon, 2003 SIREV

Theorem (LNV, 2013 EURO J. Comp. Optim.)

Any such DS method 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)

Insight: $\underbrace{(\text{decrease in } f) \;\geq\; \mathcal{O}(\alpha_k^2)}_{\text{success}} \;\geq\; \cdots \;\geq\; \underbrace{\mathcal{O}(\alpha_{k_u}^2) \;\geq\; \mathcal{O}(\|\nabla f(x_{k_u})\|^2)}_{\text{unsuccess}}$

Kolda, Lewis, Torczon, 2003 SIREV

Theorem (LNV, 2013 EURO J. Comp. Optim.)

Any such DS method 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

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

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

WCC of DS (smooth case)

Insight: $\underbrace{(\text{decrease in } f) \;\geq\; \mathcal{O}(\alpha_k^2)}_{\text{success}} \;\geq\; \cdots \;\geq\; \underbrace{\mathcal{O}(\alpha_{k_u}^2) \;\geq\; \mathcal{O}(\|\nabla f(x_{k_u})\|^2)}_{\text{unsuccess}}$

Kolda, Lewis, Torczon, 2003 SIREV

Theorem (LNV, 2013 EURO J. Comp. Optim.)

Any such DS method 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

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

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)

Insight: $\underbrace{(\text{decrease in } f) \;\geq\; \mathcal{O}(\alpha_k^2)}_{\text{success}} \;\geq\; \cdots \;\geq\; \underbrace{\mathcal{O}(\alpha_{k_u}^2) \;\geq\; \mathcal{O}(\|\nabla f(x_{k_u})\|^2)}_{\text{unsuccess}}$

Kolda, Lewis, Torczon, 2003 SIREV

Theorem (LNV, 2013 EURO J. Comp. Optim.)

Any such DS method 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

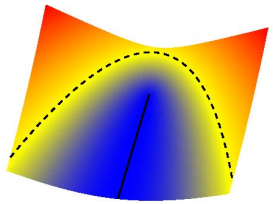
$$k_\epsilon \;\leq\; \mathcal{O}\left(n\,\epsilon^{-2}\right)$$

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 $L_{\nabla f}^2$ (instead of $L_{\nabla f}$ as in gradient method).

WCC of DS (smooth, convex case)

Ruling out cases where the supreme distance from the initial level set $L_f(x_0)$ to the solution set X_*^f is infinite...



Theorem (M. Dodangeh and LNV, 2016 Math. Program.)

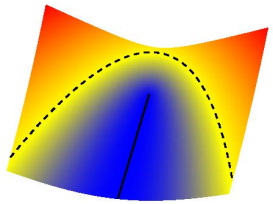
Any such DS method generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$f(x_k) - f_* \;=\; \mathcal{O}(1/k) \qquad k_\epsilon \;\leq\; \mathcal{O}\left(n\,\epsilon^{-1}\right).$$

Again, the # of fevals must be multiplied by n*:* $\mathcal{O}\left(n^2\,\epsilon^{-1}\right)$.

WCC of DS (smooth, convex case)

Ruling out cases where the supreme distance from the initial level set $L_f(x_0)$ to the solution set X_*^f is infinite...



Theorem (M. Dodangeh and LNV, 2016 Math. Program.)

Any such DS method generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

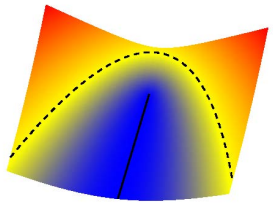
$$f(x_k) - f_* \;=\; \mathcal{O}(1/k) \qquad k_\epsilon \;\leq\; \mathcal{O}\left(n\,\epsilon^{-1}\right).$$

Again, the # of fevals must be multiplied by n*:* $\mathcal{O}\left(n^2\,\epsilon^{-1}\right)$.

The n^2 factor comes from $\frac{|D|}{\mathrm{cm}(D)^2}$.

WCC of DS (smooth, convex case)

Ruling out cases where the supreme distance from the initial level set $L_f(x_0)$ to the solution set X_*^f is infinite...



Theorem (M. Dodangeh and LNV, 2016 Math. Program.)

Any such DS method generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$f(x_k) - f_* \;=\; \mathcal{O}(1/k) \qquad k_\epsilon \;\leq\; \mathcal{O}\left(n\,\epsilon^{-1}\right).$$

Again, the # of fevals must be multiplied by n*:* $\mathcal{O}\left(n^2\,\epsilon^{-1}\right)$.

The n^2 factor comes from $\frac{|D|}{\mathrm{cm}(D)^2}$. For $D = D_\oplus$ one obtains

$$\frac{2n}{(1/\sqrt{n})^2} \;=\; 2n^2. \qquad \text{Is this optimal?}$$

Optimality of the n^2 factor

Theorem (M. Dodangeh, LNV, and Z. Zhang, 2016 Optim. Lett.)

The factor n^2 is optimal since any PSS D in $\mathbb{R}^n$ satisfies

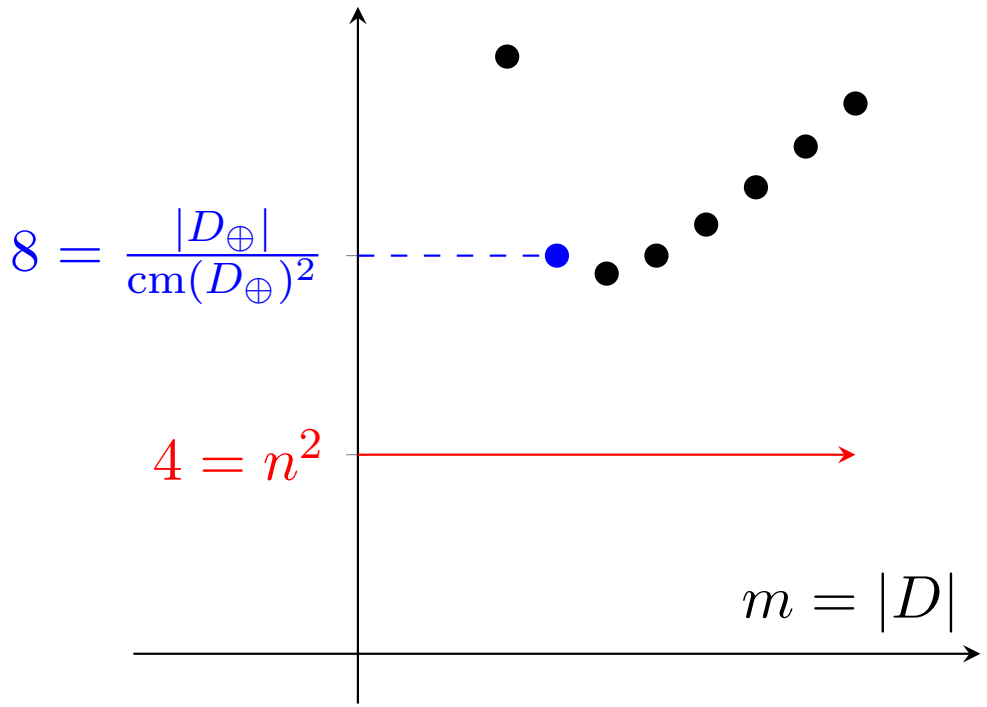
$$\frac{|D|}{\mathrm{cm}(D)^2} \geq \frac{1}{\zeta^2} n^2$$

Optimality of the n^2 factor

Theorem (M. Dodangeh, LNV, and Z. Zhang, 2016 Optim. Lett.)

The factor n^2 is optimal since any PSS D in $\mathbb{R}^n$ satisfies

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



Plot of

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

for the case $n = 2$ and D's with uniform angles,

Global rate of DS (smooth, strongly convex case)

Theorem (M. Dodangeh and LNV, 2014 Math. Program.)

Any such DS method 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 Math. Program.)

Any such DS method 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 Math. Program.)

Any such DS method generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

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

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

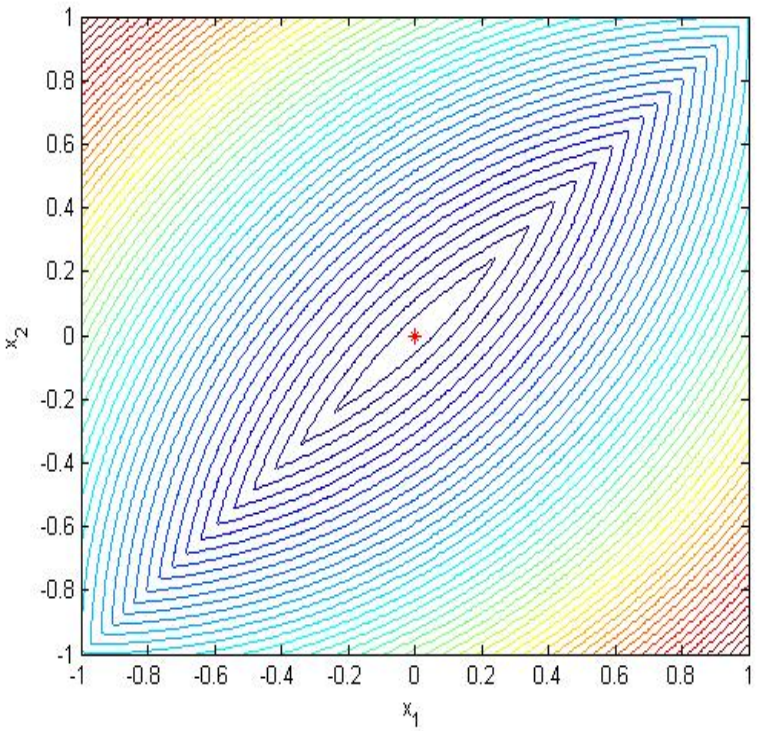
- When f is SC (constant $\mu > 0$), one has (k_0 first unsucc.)

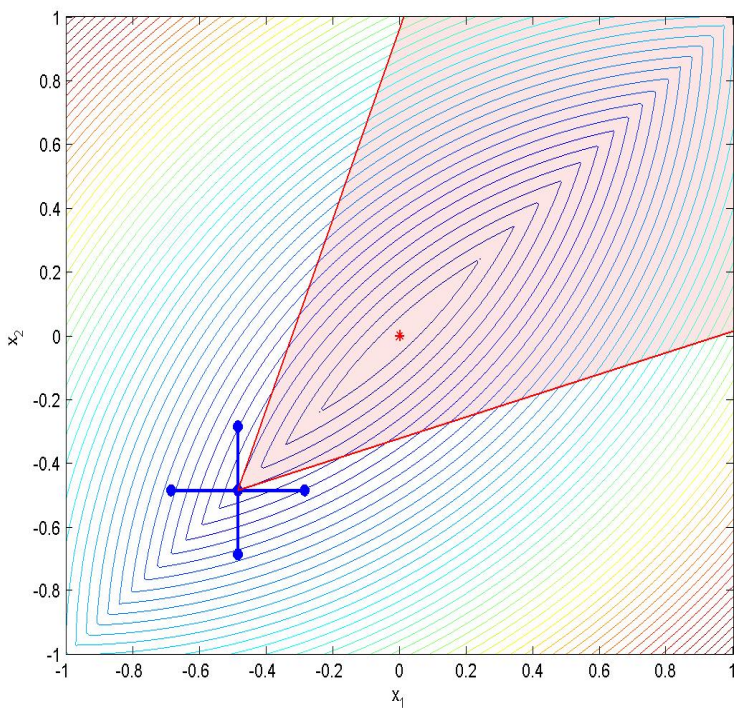
$$\|x_k - x_*\| \ \leq \ \sqrt{L_{\nabla f}/\mu}\, \|x_{k_0} - x_*\|.$$

- A linear rate for the iterates can be derived from

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

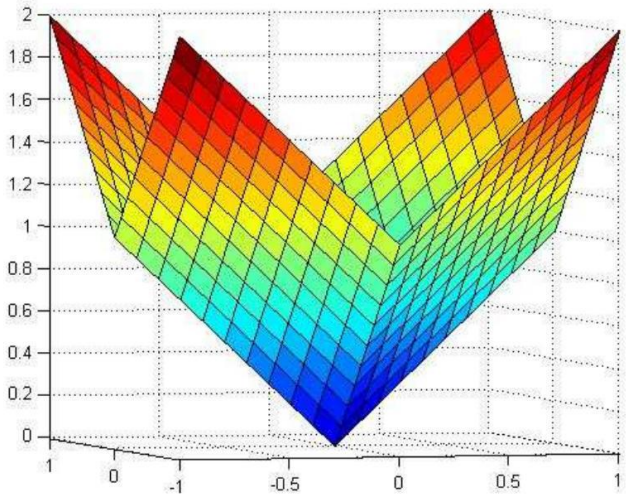
Difficulties in the nonsmooth case



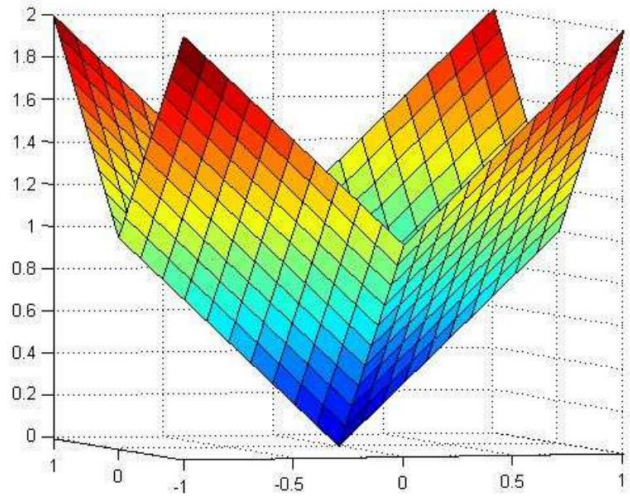


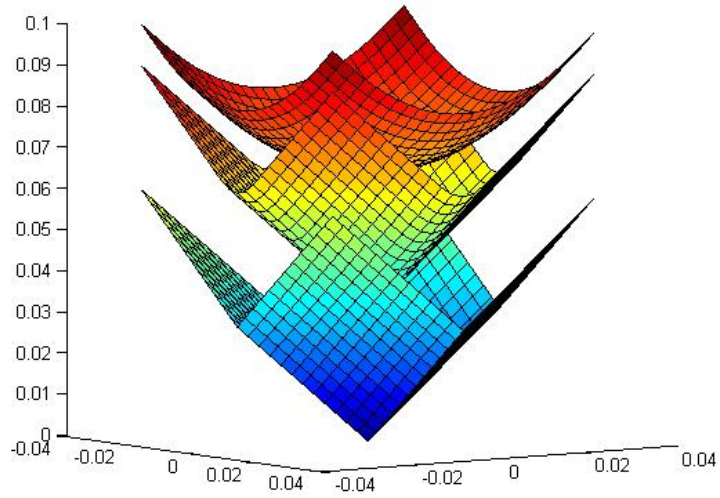
The cone of descent directions at the poll center is shaded.

One possible fix: Combine DS with Smoothing

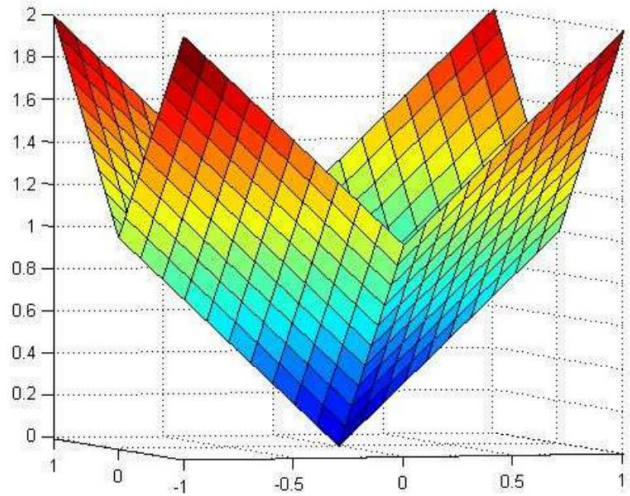


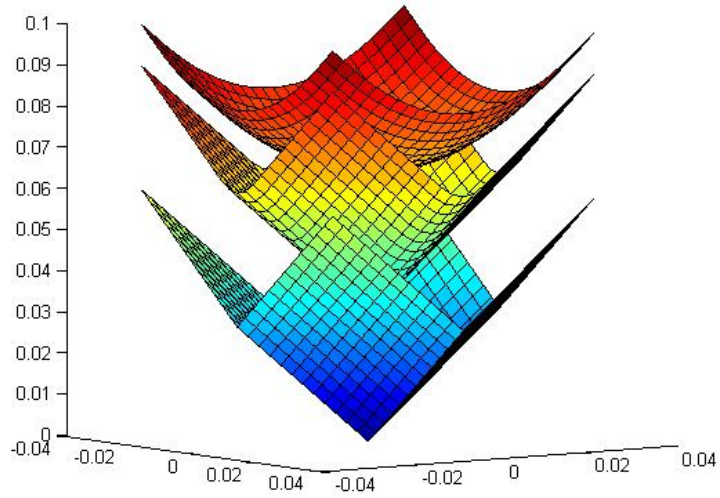
One possible fix: Combine DS with Smoothing





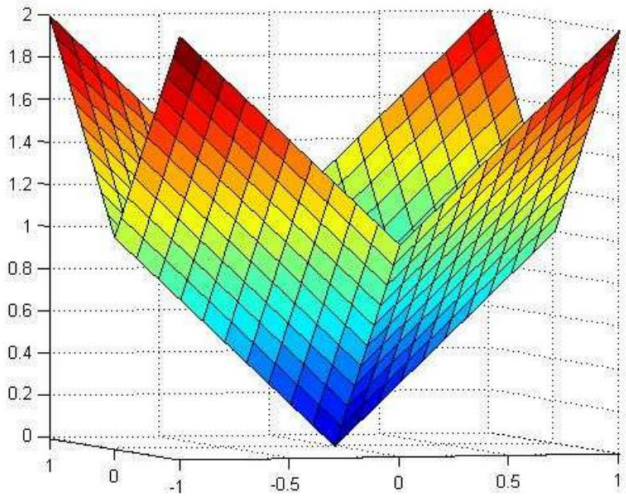
One possible fix: Combine DS with Smoothing

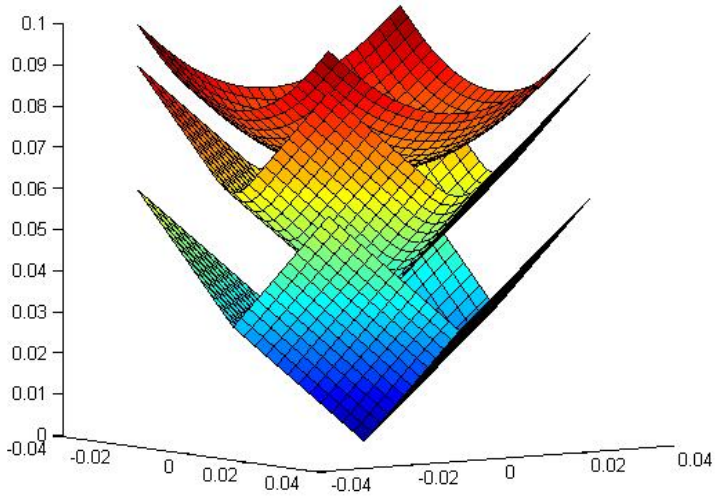




- Essentially the WCC cost increases from $\mathcal{O}\left(\epsilon^{-2}\right)$ to $\mathcal{O}\left(\epsilon^{-3}\right)$ [R. Garmanjani and LNV, 2013 IMA J. Numer. Anal.].

One possible fix: Combine DS with Smoothing





- Essentially the WCC cost increases from $\mathcal{O}\left(\epsilon^{-2}\right)$ to $\mathcal{O}\left(\epsilon^{-3}\right)$ [R. Garmanjani and LNV, 2013 IMA J. Numer. Anal.].
- The # of function evaluations increases from $\mathcal{O}\left(n^2\epsilon^{-2}\right)$ to $\mathcal{O}\left(n^{\frac{5}{2}}\epsilon^{-3}\right)$.

Summary of DS global rates

Imposing sufficient decrease to accept new iterates:

Summary of DS global rates

Imposing sufficient decrease to accept new iterates:

- $\mathcal{O}(-\log(\epsilon))$ strongly convex — linear global rate for f, ∇f, and absolute error in iterates.

Summary of DS global rates

Imposing sufficient decrease to accept new iterates:

- $\mathcal{O}(-\log(\epsilon))$ strongly convex — linear global rate for f, ∇f, and absolute error in iterates.

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

Summary of DS global rates

Imposing sufficient decrease to accept new iterates:

- $\mathcal{O}(-\log(\epsilon))$ strongly convex — linear global rate for f, ∇f, and absolute error in iterates.

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

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

Summary of DS global rates

Imposing sufficient decrease to accept new iterates:

- $\mathcal{O}(-\log(\epsilon))$ strongly convex — linear global rate for f, ∇f, and absolute error in iterates.
- $\mathcal{O}(\epsilon^{-1})$ convex — global rate $1/k$ for f and ∇f.
- $\mathcal{O}(\epsilon^{-2})$ 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:

- $\mathcal{O}(-\log(\epsilon))$ strongly convex — linear global rate for f, ∇f, and absolute error in iterates.
- $\mathcal{O}(\epsilon^{-1})$ convex — global rate $1/k$ for f and ∇f.
- $\mathcal{O}(\epsilon^{-2})$ 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).

Descent condition and successful iterations

Assume the polling directions are normalized.

Lemma

If

$$\operatorname{cm}\left(D_k, -\nabla f(x_k)\right) \geq \kappa \quad \textit{and} \quad \alpha_k < \frac{2\kappa\|\nabla f(x_k)\|}{L_{\nabla f}+1},$$

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.

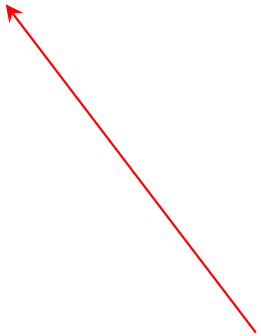
where $\mathrm{cm}(D,v)$ is the cosine measure of D given v, defined by

$$\mathrm{cm}(D,v) = \max_{d\in D} \frac{d^\top v}{\|d\|\|v\|}$$

and $L_{\nabla f}$ is a Lipschitz constant of ∇f.

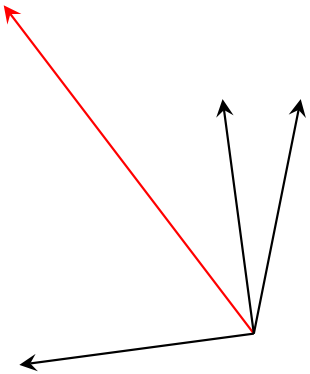
Randomly generating ‘positive spanning sets’ ...

$-\nabla f(x_k)$



Randomly generating ‘positive spanning sets’ ...

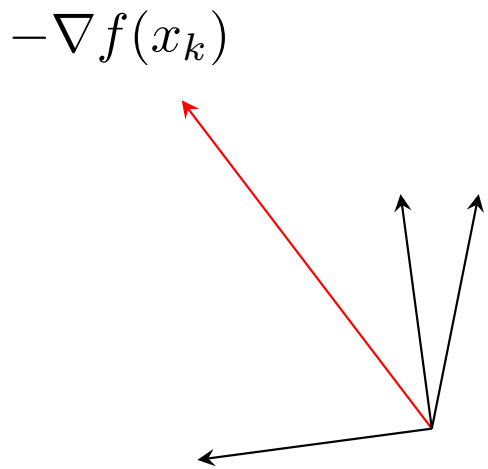
$-\nabla f(x_k)$



$n+1$ random polling directions

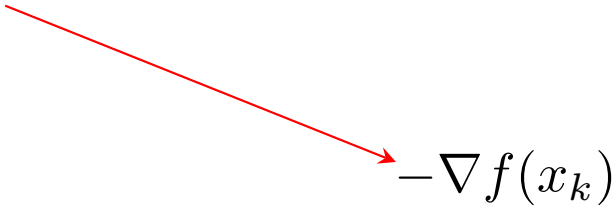
in this case not a PSS

Randomly generating 'positive spanning sets' ...

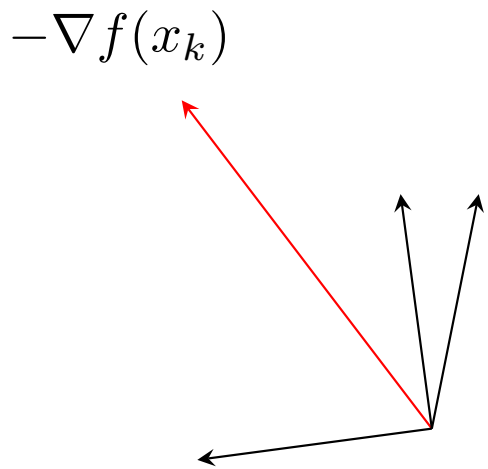


$n+1$ random polling directions

in this case not a PSS

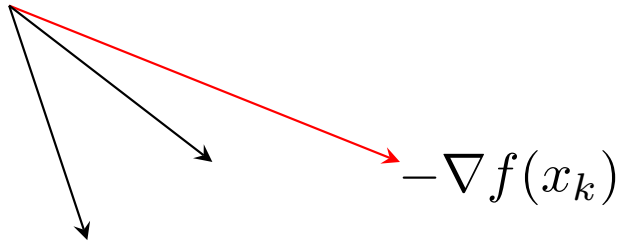


Randomly generating 'positive spanning sets' ...



$n+1$ random polling directions

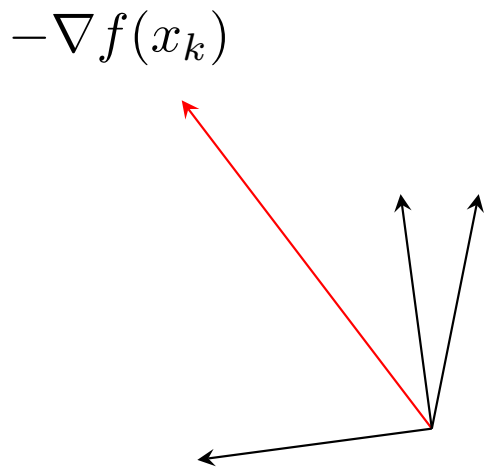
in this case not a PSS



$\leq n$ random polling directions

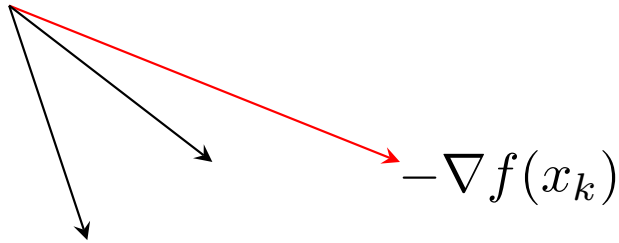
certainly not a PSS ...

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 the definition of probabilistic models (Bandeira, LNV, Scheinberg, 2014 SIOPT):

Definition

The sequence $\{\mathfrak{D}_k\}$ *is* (p, κ)*-probabilistically descent if, for each* $k \geq 0$,

$$\Pr\left(\mathrm{cm}(\mathfrak{D}_k, -\nabla f(X_k)) \geq \kappa \mid \mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}\right) \geq p,$$

Probabilistic descent

From the definition of probabilistic models (Bandeira, LNV, Scheinberg, 2014 SIOPT):

Definition

The sequence $\{\mathfrak{D}_k\}$ *is* (p, κ)*-probabilistically descent if, for each* $k \geq 0$,

$$\Pr\left(\mathrm{cm}(\mathfrak{D}_k, -\nabla f(X_k)) \;\geq\; \kappa \mid \mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}\right) \geq \; p,$$

Let Z_k be the indicator function of $\left\{\mathrm{cm}\left(\mathfrak{D}_k, -\nabla f(X_k)\right) \geq \kappa\right\}$, and

$$p_0 \;=\; \frac{\ln \theta}{\ln(\gamma^{-1}\theta)} \;=\; \frac{1}{2} \qquad \theta = 1/2,\ \gamma = 2.$$

Global rate: Counting descent

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: Counting descent

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: Counting descent

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: Counting descent

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.

Let z_ℓ denote the realization of $Z_\ell = \left\{\mathrm{cm}\big(\mathfrak{D}_\ell, -\nabla f(X_\ell)\big) \geq \kappa\right\}$ $(\ell \geq 0)$.

Intuition: If $\|\tilde{g}_k\|$ is 'big', then $\sum_{\ell=0}^{k-1} z_\ell$ is probably 'small'.

Global rate: Counting descent

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

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

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,

$$\left\{\|\tilde{G}_k\| > \epsilon\right\} \;\subset\; \left\{\sum_{\ell=0}^{k-1} Z_\ell \leq \underbrace{\left[\mathcal{O}\left(\frac{1}{k\epsilon^2}\right) + p_0\right]}_{\lambda} k\right\}.$$

Global rate: Counting descent

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,

$$\left\{\|\tilde{G}_k\| > \epsilon\right\} \;\subset\; \left\{\sum_{\ell=0}^{k-1} Z_\ell \leq \underbrace{\left[\mathcal{O}\left(\frac{1}{k\epsilon^2}\right) + p_0\right]}_{\lambda} k\right\}.$$

Hence

$$\Pr\left(\|\tilde{G}_k\| \leq \epsilon\right) \;=\; 1 - \Pr\left(\|\tilde{G}_k\| > \epsilon\right) \;\geq\; 1 - \underbrace{\Pr\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right)}_{\text{apply Chernoff \& Submartingale Theory}}.$$

Global rate and WCC bound

Theorem (Gratton, Royer, LNV, and Zhang, 2015 SIOPT)

Suppose that $\{\mathfrak{D}_k\}$ *is* (p, κ)*-probabilistically descent with* $p > p_0$*. Then*

$$\Pr\left(\|\tilde{G}_k\| \leq \mathcal{O}\left(\frac{1}{\kappa\sqrt{k}}\right)\right) \geq 1 - \exp\left[-\mathcal{O}(k)\right].$$

Global rate and WCC bound

Theorem (Gratton, Royer, LNV, and Zhang, 2015 SIOPT)

Suppose that $\{\mathfrak{D}_k\}$ *is* (p, κ)*-probabilistically descent with* $p > p_0$*. Then*

$$\Pr\left(\|\tilde{G}_k\| \leq \mathcal{O}\left(\frac{1}{\kappa\sqrt{k}}\right)\right) \geq 1 - \exp\left[-\mathcal{O}(k)\right].$$

$\longrightarrow$ $\mathcal{O}(1/\sqrt{k})$ sublinear rate with overwhelmingly high probability.

Global rate and WCC bound

Theorem (Gratton, Royer, LNV, and Zhang, 2015 SIOPT)

Suppose that $\{\mathfrak{D}_k\}$ *is* (p, κ)*-probabilistically descent with* $p > p_0$*. Then*

$$\Pr\left(\|\tilde{G}_k\| \leq \mathcal{O}\left(\frac{1}{\kappa\sqrt{k}}\right)\right) \geq 1 - \exp\left[-\mathcal{O}(k)\right].$$

$\longrightarrow$ $\mathcal{O}(1/\sqrt{k})$ sublinear rate with overwhelmingly high probability.

Since $\Pr(K_\epsilon \leq k) = \Pr(\|\tilde{G}_k\| \leq \epsilon)$, we also get:

Theorem (Gratton, Royer, LNV, and Zhang, 2015 SIOPT)

Suppose that $\{\mathfrak{D}_k\}$ *is* (p, κ)*-probabilistically descent with* $p > p_0$*. Then*

$$\Pr\left(K_\epsilon \leq \left\lceil \mathcal{O}\left(\frac{\epsilon^{-2}}{\kappa^2}\right)\right\rceil\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right].$$

Global rate and WCC bound

Theorem (Gratton, Royer, LNV, and Zhang, 2015 SIOPT)

Suppose that $\{\mathfrak{D}_k\}$ *is* (p, κ)*-probabilistically descent with* $p > p_0$*. Then*

$$\Pr\left(\|\tilde{G}_k\| \leq \mathcal{O}\left(\frac{1}{\kappa\sqrt{k}}\right)\right) \geq 1 - \exp\left[-\mathcal{O}(k)\right].$$

$\longrightarrow \mathcal{O}(1/\sqrt{k})$ sublinear rate with overwhelmingly high probability.

Since $\Pr(K_\epsilon \leq k) = \Pr(\|\tilde{G}_k\| \leq \epsilon)$, we also get:

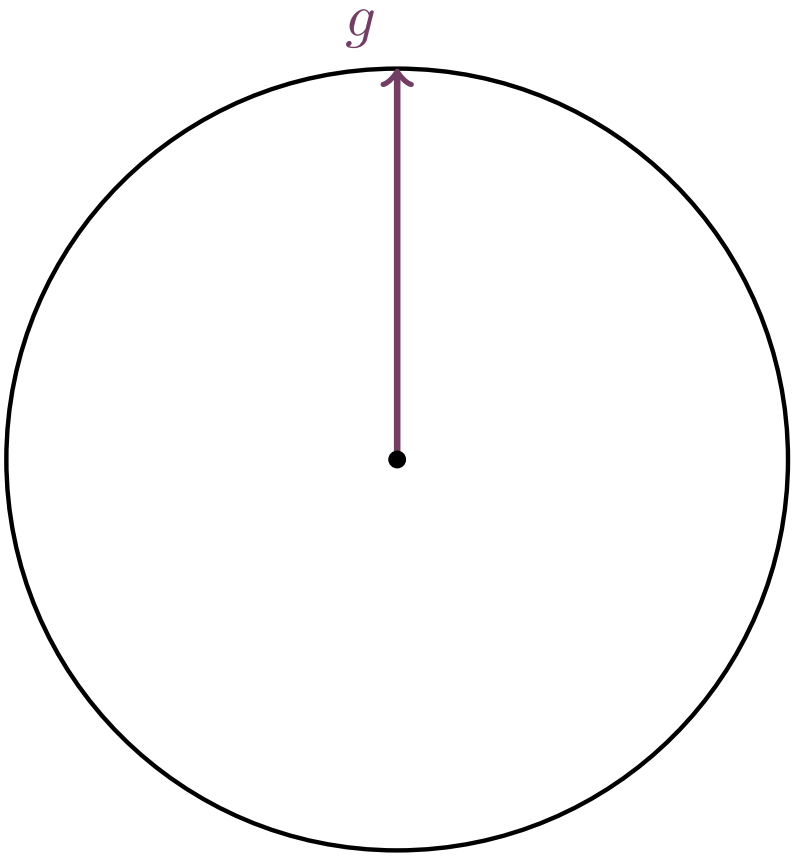
Theorem (Gratton, Royer, LNV, and Zhang, 2015 SIOPT)

Suppose that $\{\mathfrak{D}_k\}$ *is* (p, κ)*-probabilistically descent with* $p > p_0$*. Then*

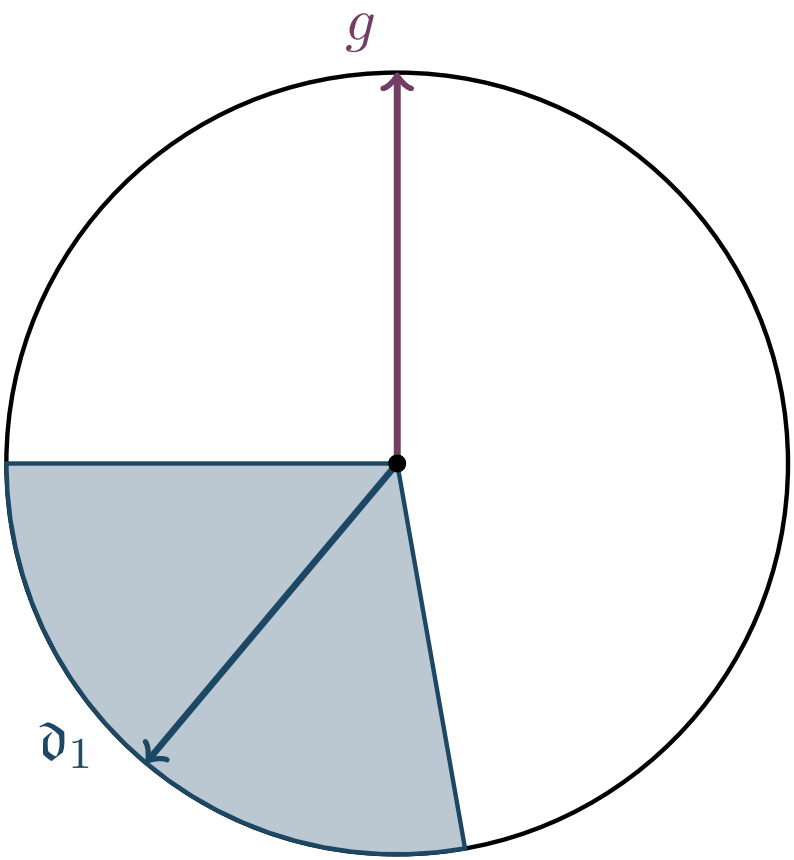
$$\Pr\left(K_\epsilon \leq \left\lceil \mathcal{O}\left(\frac{\epsilon^{-2}}{\kappa^2}\right)\right\rceil\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right].$$

$\longrightarrow \mathcal{O}(\epsilon^{-2})$ bound for # of iter. with overwhelmingly high probability.

Two uniform directions are enough, one is not

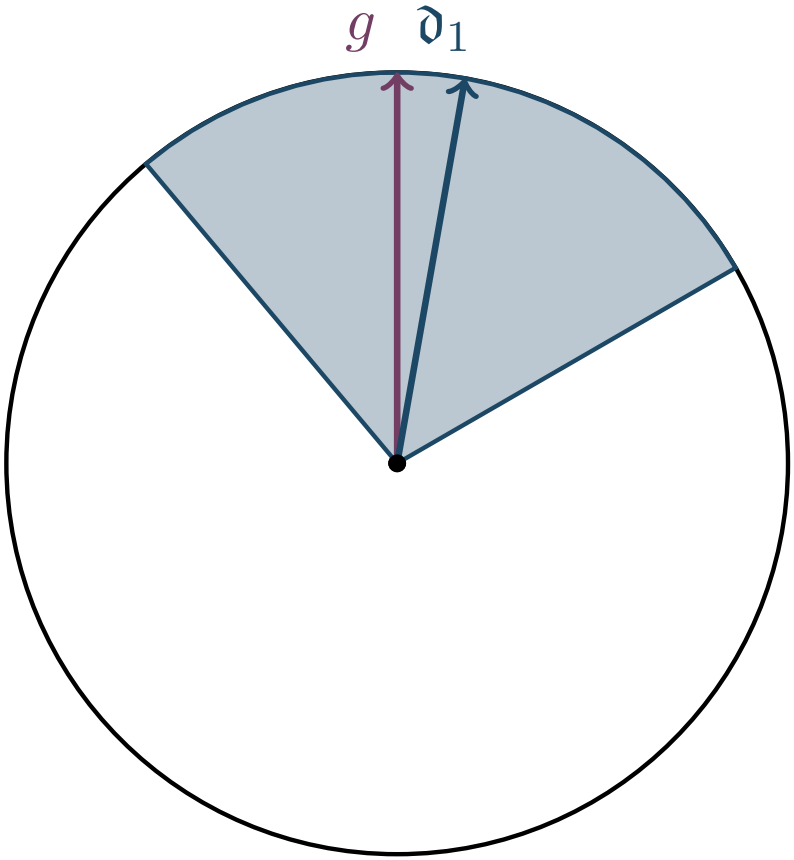


Two uniform directions are enough, one is not



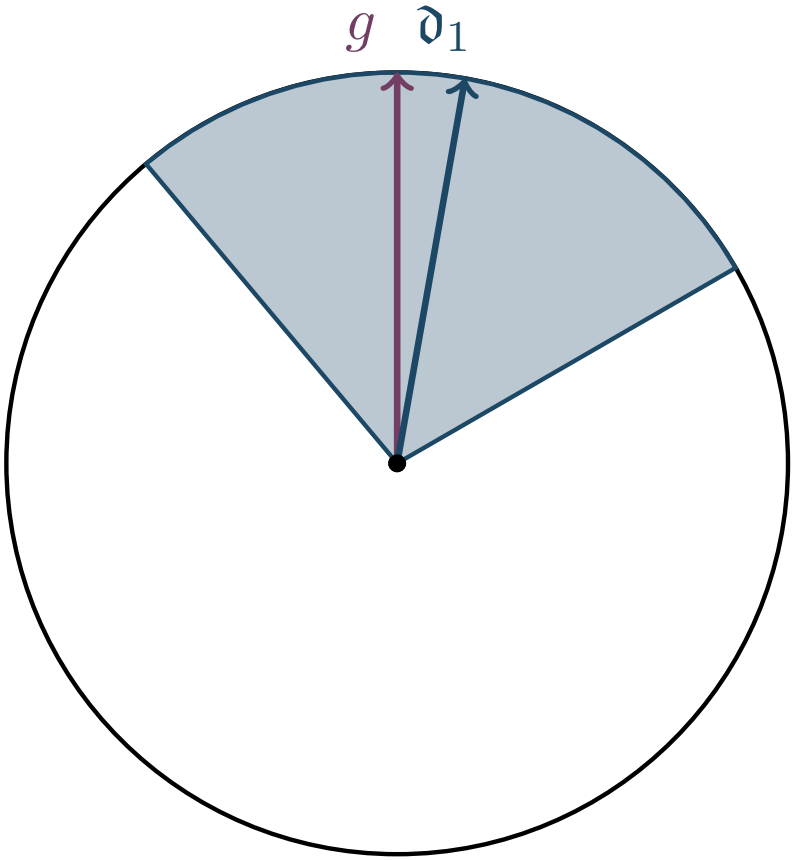
$$\mathfrak{d}_1 \sim \mathcal{U}(\mathbb{S}^1) \Rightarrow \forall \kappa \in (0,1), \quad \Pr\left(\mathrm{cm}\left(\mathfrak{d}_1, g\right) = \mathfrak{d}_1^\top g \geq \kappa\right) < 1/2.$$

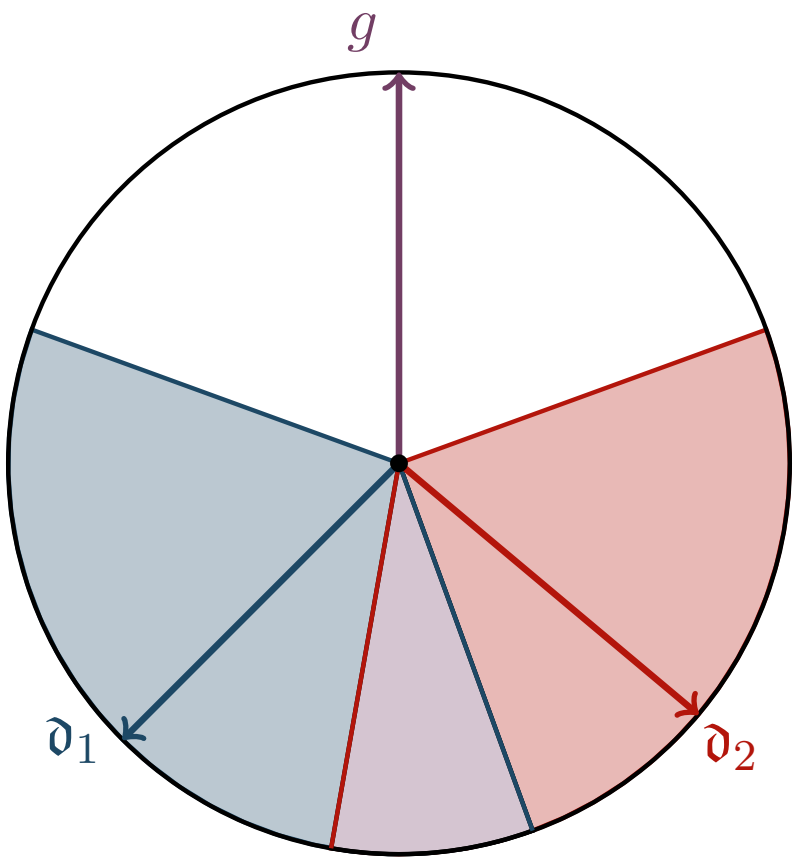
Two uniform directions are enough, one is not



$$\mathfrak{d}_1 \sim \mathcal{U}(\mathbb{S}^1) \Rightarrow \forall \kappa \in (0,1), \quad \Pr\left(\mathrm{cm}\left(\mathfrak{d}_1, g\right) = \mathfrak{d}_1^\top g \geq \kappa\right) < 1/2.$$

Two uniform directions are enough, one is not





$$\mathfrak{d}_1 \sim \mathcal{U}(\mathbb{S}^1) \Rightarrow \forall \kappa \in (0,1), \quad \Pr\left(\mathrm{cm}\left(\mathfrak{d}_1, g\right) = \mathfrak{d}_1^\top g \geq \kappa\right) < 1/2.$$

$$\mathfrak{d}_1, \mathfrak{d}_2 \sim \mathcal{U}(\mathbb{S}^1) \Rightarrow \exists \kappa^* \in (0,1), \quad \Pr\left(\mathrm{cm}\left(\{\mathfrak{d}_1, \mathfrak{d}_2\}, g\right) \geq \kappa^*\right) > 1/2.$$

Worst case complexity: Dependence on the dimension

Then, when $r = |D| > 1$, $\{\mathfrak{D}_k\}$ is p-probabilistically $(1/\sqrt{n})$-descent for some $p > p_0 = 1/2$ independent of n.

Worst case complexity: Dependence on the dimension

Then, when $r = |D| > 1$, $\{\mathfrak{D}_k\}$ is p-probabilistically $(1/\sqrt{n})$-descent for some $p > p_0 = 1/2$ independent of n.

Plugging $\kappa = 1/\sqrt{n}$ into the WCC bound, one obtains

Worst case complexity: Dependence on the dimension

Then, when $r = |D| > 1$, $\{\mathfrak{D}_k\}$ is p-probabilistically $(1/\sqrt{n})$-descent for some $p > p_0 = 1/2$ independent of n.

Plugging $\kappa = 1/\sqrt{n}$ into the WCC bound, one obtains

WCC (number of function evaluations)

$$\Pr\left(K_\epsilon^f \leq \left\lceil \mathcal{O}\left(n\epsilon^{-2}\right)\right\rceil r\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right].$$

Worst case complexity: Dependence on the dimension

Then, when $r = |D| > 1$, $\{\mathfrak{D}_k\}$ is p-probabilistically $(1/\sqrt{n})$-descent for some $p > p_0 = 1/2$ independent of n.

Plugging $\kappa = 1/\sqrt{n}$ into the WCC bound, one obtains

WCC (number of function evaluations)

$$\Pr\left(K_\epsilon^f \leq \left\lceil \mathcal{O}\left(n\epsilon^{-2}\right)\right\rceil r\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right].$$

The WCC bound is $\mathcal{O}(rn\epsilon^{-2})$, better than when $\mathcal{O}(n^2\epsilon^{-2})$ when $r \ll n$.

Worst case complexity: Dependence on the dimension

Then, when $r = |D| > 1$, $\{\mathfrak{D}_k\}$ is p-probabilistically $(1/\sqrt{n})$-descent for some $p > p_0 = 1/2$ independent of n.

Plugging $\kappa = 1/\sqrt{n}$ into the WCC bound, one obtains

WCC (number of function evaluations)

$$\Pr\left(K_\epsilon^f \leq \left\lceil \mathcal{O}\left(n\epsilon^{-2}\right)\right\rceil r\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right].$$

The WCC bound is $\mathcal{O}(rn\epsilon^{-2})$, better than when $\mathcal{O}(n^2\epsilon^{-2})$ when $r \ll n$.

Theory admits $r = 2$, 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 $[Q \ -Q]$.

Even better than 2 directions

Two random vectors $\mathfrak{D}_k = \{\mathfrak{d}, -\mathfrak{d}\}$ u.d. on the unit sphere of $\mathbb{R}^n$ works even better, and is an optimal choice:

Even better than 2 directions

Two random vectors $\mathfrak{D}_k = \{\mathfrak{d}, -\mathfrak{d}\}$ u.d. on the unit sphere of $\mathbb{R}^n$ works even better, and is an optimal choice:

Given $v \in \mathbb{R}^n$ with $\|v\| = 1$ and $\kappa \in [0, 1]$,

$$\Pr\left(\operatorname{cm}(\mathfrak{D}_k, v) \geq \kappa\right) = \Pr\left(\mathfrak{d}^\top v \geq \kappa\right) + \Pr\left(-\mathfrak{d}^\top v \geq \kappa\right) = 2\varrho.$$

with $\varrho = \Pr\left(\{\mathfrak{d}^\top v \geq \kappa\}\right)$.

Even better than 2 directions

Two random vectors $\mathfrak{D}_k = \{\mathfrak{d}, -\mathfrak{d}\}$ u.d. on the unit sphere of $\mathbb{R}^n$ works even better, and is an optimal choice:

Given $v \in \mathbb{R}^n$ with $\|v\| = 1$ and $\kappa \in [0, 1]$,

$$\Pr\big(\operatorname{cm}(\mathfrak{D}_k, v) \geq \kappa\big) \;=\; \Pr\big(\mathfrak{d}^\top v \geq \kappa\big) + \Pr\big(-\mathfrak{d}^\top v \geq \kappa\big) = \; 2\varrho.$$

with $\varrho = \Pr\big(\{\mathfrak{d}^\top v \geq \kappa\}\big)$.

Given any γ, θ satisfying $0 < \theta < 1 < \gamma$, we can pick $\kappa > 0$ sufficiently small so that $2\varrho > p_0 = (\ln \theta)/[\ln(\gamma^{-1}\theta)]$.

Even better than 2 directions

Two random vectors $\mathfrak{D}_k = \{\mathfrak{d}, -\mathfrak{d}\}$ u.d. on the unit sphere of $\mathbb{R}^n$ works even better, and is an optimal choice:

Given $v \in \mathbb{R}^n$ with $\|v\| = 1$ and $\kappa \in [0, 1]$,

$$\Pr\big(\operatorname{cm}(\mathfrak{D}_k, v) \geq \kappa\big) \;=\; \Pr\big(\mathfrak{d}^\top v \geq \kappa\big) + \Pr\big(-\mathfrak{d}^\top v \geq \kappa\big) = \; 2\varrho.$$

with $\varrho = \Pr\big(\{\mathfrak{d}^\top v \geq \kappa\}\big)$.

Given any γ, θ satisfying $0 < \theta < 1 < \gamma$, we can pick $\kappa > 0$ sufficiently small so that $2\varrho > p_0 = (\ln \theta)/[\ln(\gamma^{-1}\theta)]$.

In addition, for any $\mathfrak{D} = \{\mathfrak{d}_1, \mathfrak{d}_2\}$ u.d. in the unit sphere,

$$\Pr\big(\operatorname{cm}(\mathfrak{D}, v) \geq \kappa\big) \;=\; 2\varrho - \Pr\big(\{\mathfrak{d}_1^\top v \geq \kappa\} \cap \{\mathfrak{d}_2^\top v \geq \kappa\}\big) \leq \; 2\varrho.$$

So, a new approach and proof technique

A new proof technique for establishing global rates and WCC bounds for randomized algorithms for which

So, a new approach and 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 approach and 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 approach and 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.

What we obtain is a global rate of $\mathcal{O}(1/\sqrt{k})$, with overwhelmingly high probability.

Examples of application

This technique has also been applied to:

- Trust-region methods based on probabilistic models [Gratton, Royer, LNV, and Zhang, 2017 IMAJNA].

  When models are built iteratively in some random fashion and exhibit good accuracy with sufficiently high probability.

Examples of application

This technique has also been applied to:

- Trust-region methods based on probabilistic models [Gratton, Royer, LNV, and Zhang, 2017 IMAJNA].

  When models are built iteratively in some random fashion and exhibit good accuracy with sufficiently high probability.

- Problems with linear constraints [Gratton, Royer, LNV, and Zhang, 2017].

  Random generation can be efficiently done in subspaces identified within approximate tangent cones.

Global Optimization part of the talk

Our goal is now to develop a solver for a global optimization problem of the type

$$\min_{x \in \Omega} f(x),$$

where $\Omega \in \mathbb{R}^n$ is a box. We want to do this:

Global Optimization part of the talk

Our goal is now to develop a solver for a global optimization problem of the type

$$\min_{x \in \Omega} f(x),$$

where $\Omega \in \mathbb{R}^n$ is a box. We want to do this:

- Relying on probabilistic direct search — which is an efficient, simple, and rigorous local solver.

Global Optimization part of the talk

Our goal is now to develop a solver for a global optimization problem of the type

$$\min_{x \in \Omega} f(x),$$

where $\Omega \in \mathbb{R}^n$ is a box. We want to do this:

- Relying on probabilistic direct search — which is an efficient, simple, and rigorous local solver.
- Taking advantage of parallel computing.

Global Optimization part of the talk

Our goal is now to develop a solver for a global optimization problem of the type

$$\min_{x \in \Omega} f(x),$$

where $\Omega \in \mathbb{R}^n$ is a box. We want to do this:

- Relying on probabilistic direct search — which is an efficient, simple, and rigorous local solver.
- Taking advantage of parallel computing.
- Having an approach capable of identifying several global or local minimizers of interest.

REMEMBER: Probabilistic DS (polling with 2 directions)

Choose: x_0 and α_0.

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

- **Search step (optional)**
- **Poll step:** Select $D_k = [d, -d]$ RANDOMLY and find $x_k + \alpha_k d_k$ ($d_k \in D_k$):

$$f(x_k + \alpha_k d_k) < f(x_k) - \rho(\alpha_k) \qquad \text{like } \rho(\alpha) = \alpha^2/2.$$

- **SUCCESS:** Move $x_{k+1} = x_k + \alpha_k d_k$ and increase $\alpha_{k+1} = \gamma\, \alpha_k$ $(\gamma = 2)$.
- **UNSUCCESS:** Stay $x_{k+1} = x_k$ and decrease $\alpha_{k+1} = \theta\, \alpha_k$ $(\theta = 1/2)$.

A multistart multisplit DS framework (main idea)

MAIN IDEA:

A multistart multisplit DS framework (main idea)

MAIN IDEA:

- Consider a certain number R of DS runs guided by the poll centers (in parallel).

A multistart multisplit DS framework (main idea)

MAIN IDEA:

- Consider a certain number R of DS runs guided by the poll centers (in parallel).
- Create R clusters $C_j, j = 1, \ldots, R$, of points, out of all the points where f has been evaluated.

A multistart multisplit DS framework (main idea)

MAIN IDEA:

- Consider a certain number R of DS runs guided by the poll centers (in parallel).
- Create R clusters $C_j, j = 1, \ldots, R$, of points, out of all the points where f has been evaluated.
- Split or merge the DS runs based on how the poll centers fit into the clusters.

A multistart multisplit DS framework (main idea)

MAIN IDEA:

- Consider a certain number R of DS runs guided by the poll centers (in parallel).
- Create R clusters $C_j, j = 1, \ldots, R$, of points, out of all the points where f has been evaluated.
- Split or merge the DS runs based on how the poll centers fit into the clusters.

IN THIS WAY:

A multistart multisplit DS framework (main idea)

MAIN IDEA:

- Consider a certain number R of DS runs guided by the poll centers (in parallel).
- Create R clusters $C_j, j = 1, \ldots, R$, of points, out of all the points where f has been evaluated.
- Split or merge the DS runs based on how the poll centers fit into the clusters.

IN THIS WAY:

- We parallelize DS runs (where poll steps are typically limited to $2n$ evaluations).

A multistart multisplit DS framework (main idea)

MAIN IDEA:

- Consider a certain number R of DS runs guided by the poll centers (in parallel).
- Create R clusters $C_j, j = 1, \ldots, R$, of points, out of all the points where f has been evaluated.
- Split or merge the DS runs based on how the poll centers fit into the clusters.

IN THIS WAY:

- We parallelize DS runs (where poll steps are typically limited to $2n$ evaluations).
- We allow MULTISTART runs and SPLITTING/MERGING of runs.

A multistart multisplit DS framework (scheme)

MULTISTART: Select initial points for R_0 DS runs: $x_j^0, j = 1, \ldots, R_0$ (and initial step sizes $\alpha_j^0, j = 1, \ldots, R_0$).

A multistart multisplit DS framework (scheme)

MULTISTART: Select initial points for R_0 DS runs: $x_j^0, j = 1, \ldots, R_0$ (and initial step sizes $\alpha_j^0, j = 1, \ldots, R_0$).

At each OUTER iteration ℓ:

A multistart multisplit DS framework (scheme)

MULTISTART: Select initial points for R_0 DS runs: $x_j^0, j = 1, \ldots, R_0$ (and initial step sizes $\alpha_j^0, j = 1, \ldots, R_0$).

At each OUTER iteration ℓ:

- Create R_ℓ clusters $C_j^\ell, j = 1, \ldots, R_\ell$ (using all the history of points).

A multistart multisplit DS framework (scheme)

MULTISTART: Select initial points for R_0 DS runs: $x_j^0, j = 1, \ldots, R_0$ (and initial step sizes $\alpha_j^0, j = 1, \ldots, R_0$).

At each OUTER iteration ℓ:

- Create R_ℓ clusters $C_j^\ell, j = 1, \ldots, R_\ell$ (using all the history of points).
- Decide whether any DS run is split or merged. Let $x_j^{\ell+1}, j = 1, \ldots, R_{\ell+1}$ be the new poll centers.

A multistart multisplit DS framework (scheme)

MULTISTART: Select initial points for R_0 DS runs: $x_j^0, j = 1, \ldots, R_0$ (and initial step sizes $\alpha_j^0, j = 1, \ldots, R_0$).

At each OUTER iteration ℓ:

- Create R_ℓ clusters $C_j^\ell, j = 1, \ldots, R_\ell$ (using all the history of points).
- Decide whether any DS run is split or merged. Let $x_j^{\ell+1}, j = 1, \ldots, R_{\ell+1}$ be the new poll centers.
- INNER ITERATIONS:
  - Perform (in parallel) a certain number of iterations for all the $R_{\ell+1}$ DS runs, starting from $x_j^{\ell+1}, j = 1, \ldots, R_{\ell+1}$.

Requirements and approaches

To align with typical global convergence requirements:

Requirements and approaches

To align with typical global convergence requirements:

- The clustering step must be finite.

Requirements and approaches

To align with typical global convergence requirements:

- The clustering step must be finite.
- The DS runs must comply to the local rules.

Requirements and approaches

To align with typical global convergence requirements:

- The clustering step must be finite.
- The DS runs must comply to the local rules.
- When merging two DS runs, the new poll center must be the one with the lowest objective value.

Requirements and approaches

To align with typical global convergence requirements:

- The clustering step must be finite.
- The DS runs must comply to the local rules.
- When merging two DS runs, the new poll center must be the one with the lowest objective value.

We consider two approaches for the OUTER iterations:

Requirements and approaches

To align with typical global convergence requirements:

- The clustering step must be finite.
- The DS runs must comply to the local rules.
- When merging two DS runs, the new poll center must be the one with the lowest objective value.

We consider two approaches for the OUTER iterations:

- Using space clustering.

Requirements and approaches

To align with typical global convergence requirements:

- The clustering step must be finite.
- The DS runs must comply to the local rules.
- When merging two DS runs, the new poll center must be the one with the lowest objective value.

We consider two approaches for the OUTER iterations:

- Using space clustering.
- Using nonconvex models built from previous function values.

The space clustering approach

- Create R_ℓ clusters using kmeans clustering (using all previously evaluated points).

The space clustering approach

- Create R_ℓ clusters using kmeans clustering (using all previously evaluated points).
- If a cluster contains more than two poll centers, remove the DS run corresponding to the poll center with worst objective value.

The space clustering approach

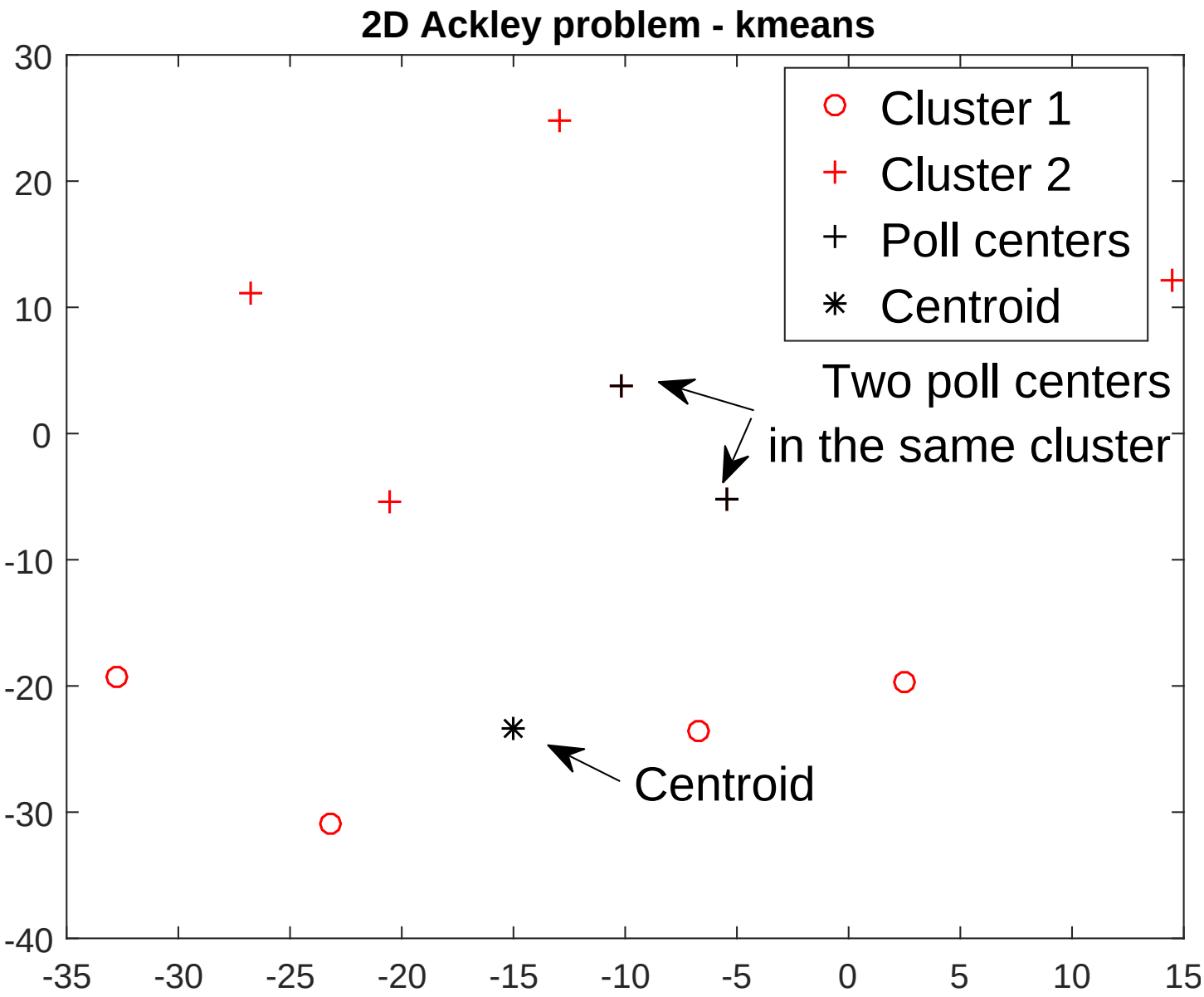
- Create R_ℓ clusters using kmeans clustering (using all previously evaluated points).
- If a cluster contains more than two poll centers, remove the DS run corresponding to the poll center with worst objective value.
- If a cluster contains no poll center, then attempt to start a new DS run from the centroid of the cluster.

The space clustering approach

- Create R_ℓ clusters using kmeans clustering (using all previously evaluated points).
- If a cluster contains more than two poll centers, remove the DS run corresponding to the poll center with worst objective value.
- If a cluster contains no poll center, then attempt to start a new DS run from the centroid of the cluster.

POSSIBLE DRAWBACK: The kmeans approach does not take into account the previous objective function values (clustering is done in the variables space).

The space clustering approach (example)



A nonconvex piecewise quadratic model

Modeling can instead identify valleys of convexity. (Nonconvex) quadratic piecewise models were suggested by Mangasarian et al., JOGO, 2006.

A nonconvex piecewise quadratic model

Modeling can instead identify valleys of convexity. (Nonconvex) quadratic piecewise models were suggested by Mangasarian et al., JOGO, 2006.

Given a sample set $Y = \{y^1, \ldots, y^{n_p}\}$, a nonconvex piecewise quadratic underestimate of f (with n_q quadratics) can be obtained by solving:

A nonconvex piecewise quadratic model

Modeling can instead identify valleys of convexity. (Nonconvex) quadratic piecewise models were suggested by Mangasarian et al., JOGO, 2006.

Given a sample set $Y = \{y^1, \ldots, y^{n_p}\}$, a nonconvex piecewise quadratic underestimate of f (with n_q quadratics) can be obtained by solving:

$$\max_p \sum_{j=1}^{n_p} q(p; y^j) \quad \text{s.t.} \quad q(p; y^j) \leq f(y^j), \quad j = 1, \ldots, n_p,$$

with p containing all the quadratics coefficients $p_i = (a_i, c_i, H_i)$ in

A nonconvex piecewise quadratic model

Modeling can instead identify valleys of convexity. (Nonconvex) quadratic piecewise models were suggested by Mangasarian et al., JOGO, 2006.

Given a sample set $Y = \{y^1, \ldots, y^{n_p}\}$, a nonconvex piecewise quadratic underestimate of f (with n_q quadratics) can be obtained by solving:

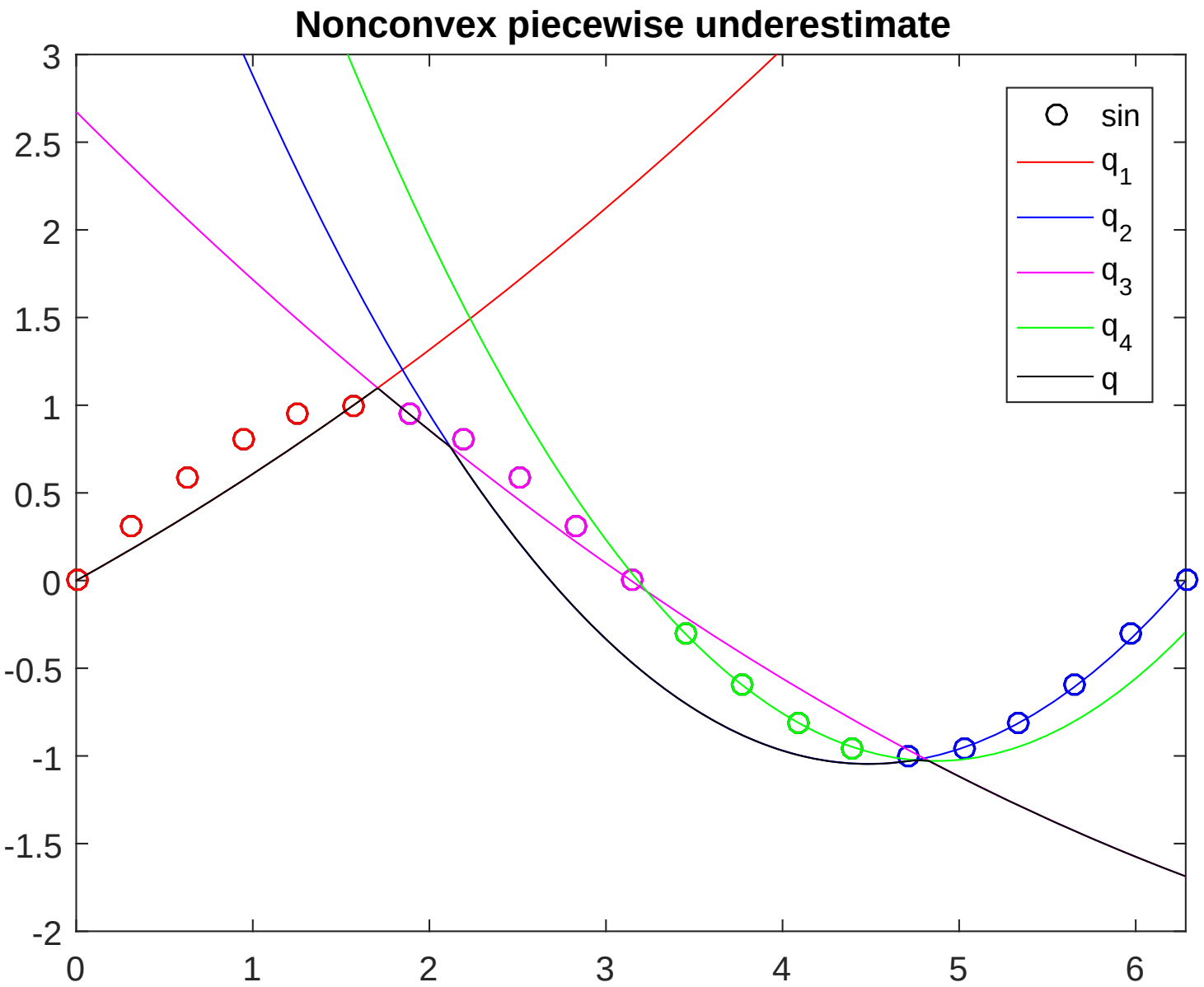
$$\max_p \sum_{j=1}^{n_p} q(p; y^j) \quad \text{s.t.} \quad q(p; y^j) \leq f(y^j), \quad j = 1, \ldots, n_p,$$

with p containing all the quadratics coefficients $p_i = (a_i, c_i, H_i)$ in

$$q(p; y) \;=\; \min_{1 \leq i \leq n_q} \left\{ a_i + c_i^\top y + \frac{1}{2} y^\top H_i y \;=\; q_i(y) \right\},$$

and where the H_i's must be assured symmetric and PD.

A nonconvex piecewise quadratic model (example)



A nonconvex piecewise quadratic model (drawbacks)

One can introduce auxiliary variables (linear objective):

$$\max_{p,\gamma} \sum_{j=1}^{n_p} \gamma_j$$

$$\text{s.t.} \quad q(p; y^j) \le f(y^j), \quad j = 1, \dots, n_p,$$

$$\gamma_j \le q_i(y^j), \quad i = 1, \dots, n_q, \; j = 1, \dots, n_p.$$

A nonconvex piecewise quadratic model (drawbacks)

One can introduce auxiliary variables (linear objective):

$$\begin{aligned} \max_{p,\gamma} \quad & \sum_{j=1}^{n_p} \gamma_j \\ \text{s.t.} \quad & q(p; y^j) \leq f(y^j), \quad j = 1, \ldots, n_p, \\ & \gamma_j \leq q_i(y^j), \quad i = 1, \ldots, n_q, \ j = 1, \ldots, n_p. \end{aligned}$$

Still, this optimization problem has several drawbacks:

- It is nonconvex, and has to be solved itself for global optimality to ensure effective underestimation.

A nonconvex piecewise quadratic model (drawbacks)

One can introduce auxiliary variables (linear objective):

$$\begin{aligned} \max_{p,\gamma} \quad & \sum_{j=1}^{n_p} \gamma_j \\ \text{s.t.} \quad & q(p; y^j) \le f(y^j), \quad j = 1, \dots, n_p, \\ & \gamma_j \le q_i(y^j), \quad i = 1, \dots, n_q, \ j = 1, \dots, n_p. \end{aligned}$$

Still, this optimization problem has several drawbacks:

- It is nonconvex, and has to be solved itself for global optimality to ensure effective underestimation.
- The # of variables is excessive: n_p (# of auxiliary variables) + $(n+1)(n+2)/2 \times n_q$ (quadratic coefficients $\times$ # of quadratics).

A nonconvex piecewise quadratic model (drawbacks)

One can introduce auxiliary variables (linear objective):

$$\begin{aligned} \max_{p,\gamma} \quad & \sum_{j=1}^{n_p} \gamma_j \\ \text{s.t.} \quad & q(p; y^j) \leq f(y^j), \quad j = 1, \ldots, n_p, \\ & \gamma_j \leq q_i(y^j), \quad i = 1, \ldots, n_q, \ j = 1, \ldots, n_p. \end{aligned}$$

Still, this optimization problem has several drawbacks:

- It is nonconvex, and has to be solved itself for global optimality to ensure effective underestimation.
- The # of variables is excessive: n_p (# of auxiliary variables) + $(n+1)(n+2)/2 \times n_q$ (quadratic coefficients $\times$ # of quadratics).
- # of constraints also high when imposing PD on all quadratics.

A nonconvex piecewise quadratic model (simplified)

But still, one could think of computing just a feasible point for the model.

A nonconvex piecewise quadratic model (simplified)

But still, one could think of computing just a feasible point for the model.

Or one can think of taking a subset of $\overline{n_p} \leq n_p$ points and considering $n_q = 1$, i.e., fitting only one quadratic q_i, resulting in the following LP:

A nonconvex piecewise quadratic model (simplified)

But still, one could think of computing just a feasible point for the model.

Or one can think of taking a subset of $\overline{n_p} \leq n_p$ points and considering $n_q = 1$, i.e., fitting only one quadratic q_i, resulting in the following LP:

$$\begin{aligned} \max_{p_i,\gamma} \quad & \sum_{j=1}^{\overline{n_p}} \gamma_j \\ \text{s.t.} \quad & q_i(y^j) \leq f(y^j), \quad j = 1, \ldots, \overline{n_p}, \\ & \gamma_j \leq q_i(y^j), \quad j = 1, \ldots, \overline{n_p}. \end{aligned}$$

A nonconvex piecewise quadratic model (simplified)

But still, one could think of computing just a feasible point for the model.

Or one can think of taking a subset of $\overline{n_p} \leq n_p$ points and considering $n_q = 1$, i.e., fitting only one quadratic q_i, resulting in the following LP:

$$\begin{aligned} \max_{p_i,\gamma} \quad & \sum_{j=1}^{\overline{n_p}} \gamma_j \\ \text{s.t.} \quad & q_i(y^j) \leq f(y^j), \quad j = 1, \ldots, \overline{n_p}, \\ & \gamma_j \leq q_i(y^j), \quad j = 1, \ldots, \overline{n_p}. \end{aligned}$$

The # of variables reduces to $\overline{n_p}$ (# of auxiliary variables) + $(n+1)(n+2)/2$ (one quadratic).

The nonconvex modeling approach

Consider again R runs of DS and all previously evaluated points.

The nonconvex modeling approach

Consider again R runs of DS and all previously evaluated points.

- Form R clusters of points around the R poll centers (using distances to them).

The nonconvex modeling approach

Consider again R runs of DS and all previously evaluated points.

- Form R clusters of points around the R poll centers (using distances to them).
- Compute the quadratic underestimating models for each cluster of points.

The nonconvex modeling approach

Consider again R runs of DS and all previously evaluated points.

- Form R clusters of points around the R poll centers (using distances to them).
- Compute the quadratic underestimating models for each cluster of points.
- Assess how well the quadratic models fit the whole data:

$$\theta_{i,j} = \sum_{l=1}^{(\overline{n_p})_j} \left(\frac{f(y_l^j) - q_i(y_l^j)}{|f(y_l^j)| + 1} \right)^2, \quad i, j = 1, \ldots, R.$$

The nonconvex modeling approach

Consider again R runs of DS and all previously evaluated points.

- Form R clusters of points around the R poll centers (using distances to them).
- Compute the quadratic underestimating models for each cluster of points.
- Assess how well the quadratic models fit the whole data:

$$\theta_{i,j} = \sum_{l=1}^{(\overline{n_p})_j} \left(\frac{f(y_l^j) - q_i(y_l^j)}{|f(y_l^j)| + 1} \right)^2, \quad i, j = 1, \ldots, R.$$

  - If $\theta_{i,i}$ is large, then split the DS run i (the quadratic i does not fit well cluster i).

The nonconvex modeling approach

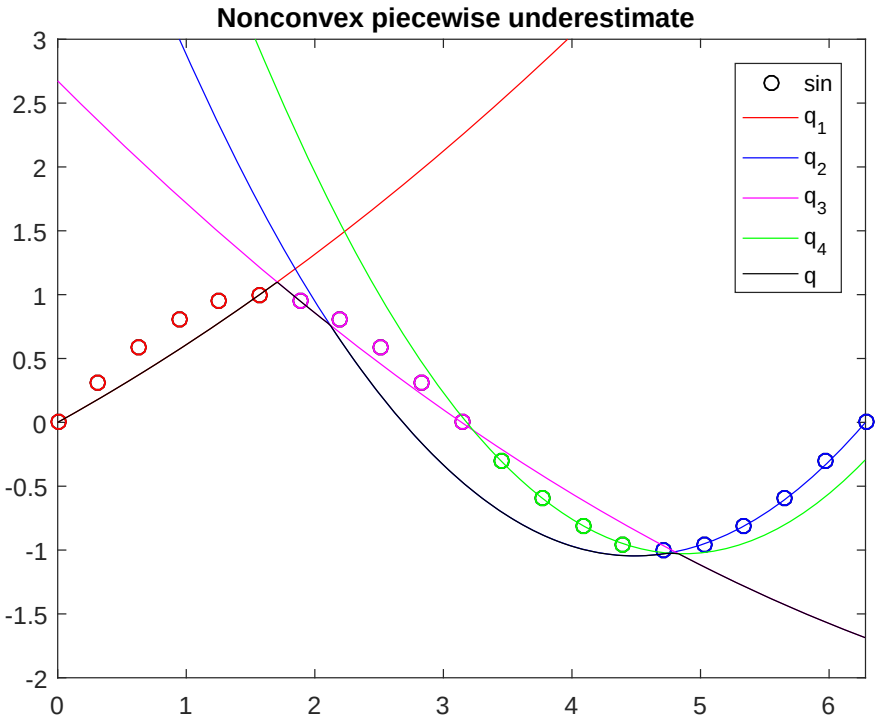
Consider again R runs of DS and all previously evaluated points.

- Form R clusters of points around the R poll centers (using distances to them).
- Compute the quadratic underestimating models for each cluster of points.
- Assess how well the quadratic models fit the whole data:

$$\theta_{i,j} = \sum_{l=1}^{(\overline{n_p})_j} \left(\frac{f(y_l^j) - q_i(y_l^j)}{|f(y_l^j)| + 1} \right)^2, \quad i, j = 1, \ldots, R.$$

  - If $\theta_{i,i}$ is large, then split the DS run i (the quadratic i does not fit well cluster i).
  - In addition, if $\theta_{i,j}$, $i \neq j$, is small, then merge DS runs i and j, using the poll center with lowest objective value.

The nonconvex modeling approach (example)



$\theta_{i,j}$ values:

	clusters or runs			
q_1	0.2068	433.1825	22.4790	75.9993
q_2	168.5406	0.0019	1.1115	0.4049
q_3	36.9686	12.3333	0.0441	0.1387
q_4	394.8617	0.5690	2.1869	0.0000

Numerical results

We provide some numerical results with a set of 102 global optimization problems. Our goal here is:

Numerical results

We provide some numerical results with a set of 102 global optimization problems. Our goal here is:

- To understand if our approach is sound, and to see if it is capable of identifying global minima.

Numerical results

We provide some numerical results with a set of 102 global optimization problems. Our goal here is:

- To understand if our approach is sound, and to see if it is capable of identifying global minima.
- To show that it performs efficiently on a wide and well representative set of problems.

Numerical results

We provide some numerical results with a set of 102 global optimization problems. Our goal here is:

- To understand if our approach is sound, and to see if it is capable of identifying global minima.
- To show that it performs efficiently on a wide and well representative set of problems.
- To develop a solver/code reasonably tested and tuned.

Numerical results

We provide some numerical results with a set of 102 global optimization problems. Our goal here is:

- To understand if our approach is sound, and to see if it is capable of identifying global minima.
- To show that it performs efficiently on a wide and well representative set of problems.
- To develop a solver/code reasonably tested and tuned.

The results are presented using performance profiles.

Numerical results

We provide some numerical results with a set of 102 global optimization problems. Our goal here is:

- To understand if our approach is sound, and to see if it is capable of identifying global minima.
- To show that it performs efficiently on a wide and well representative set of problems.
- To develop a solver/code reasonably tested and tuned.

The results are presented using performance profiles.

The solver starts with 10 poll centers in serial. A maximum of 10 simultaneous runs is imposed.

Numerical results

We provide some numerical results with a set of 102 global optimization problems. Our goal here is:

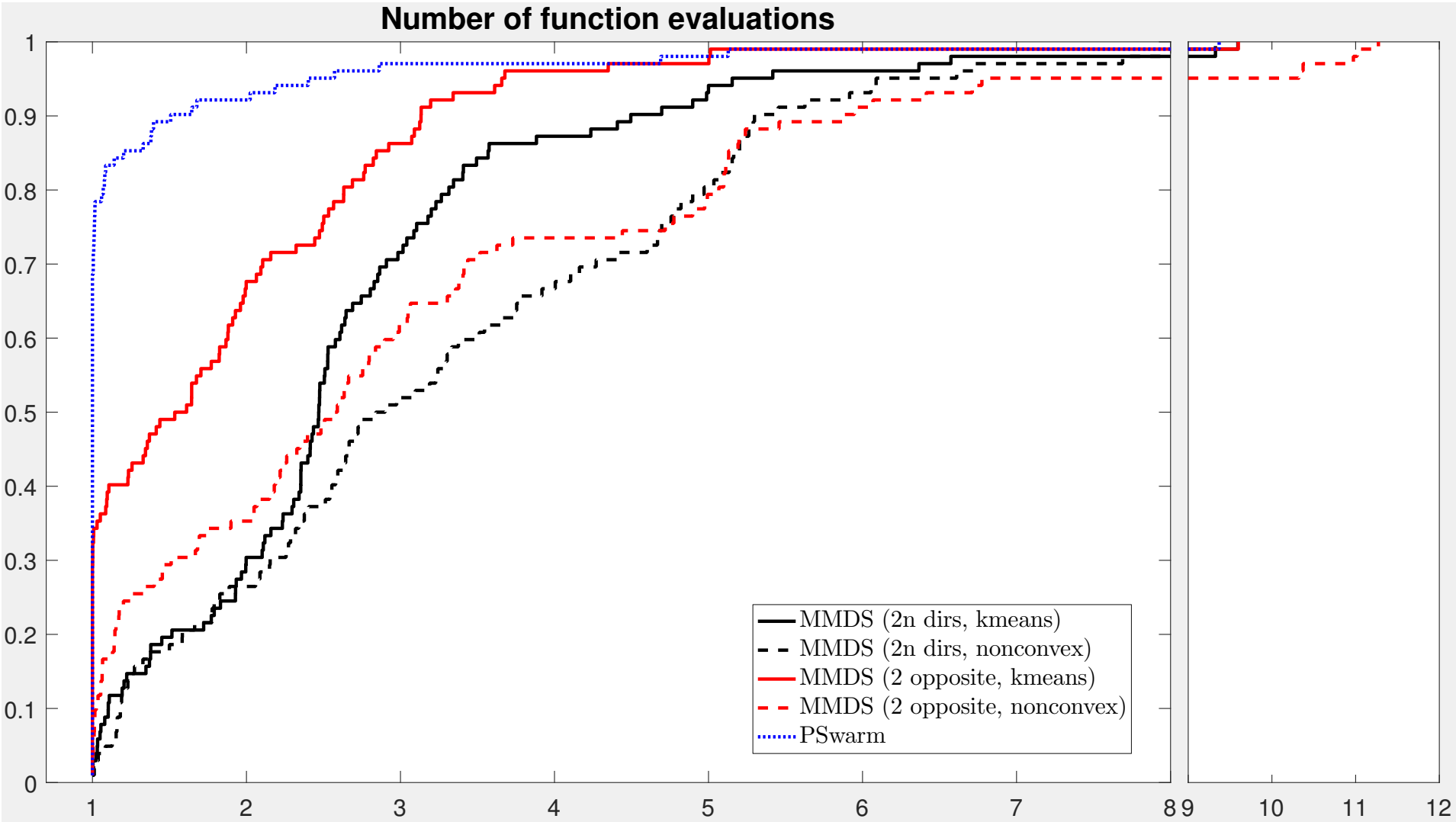
- To understand if our approach is sound, and to see if it is capable of identifying global minima.
- To show that it performs efficiently on a wide and well representative set of problems.
- To develop a solver/code reasonably tested and tuned.

The results are presented using performance profiles.

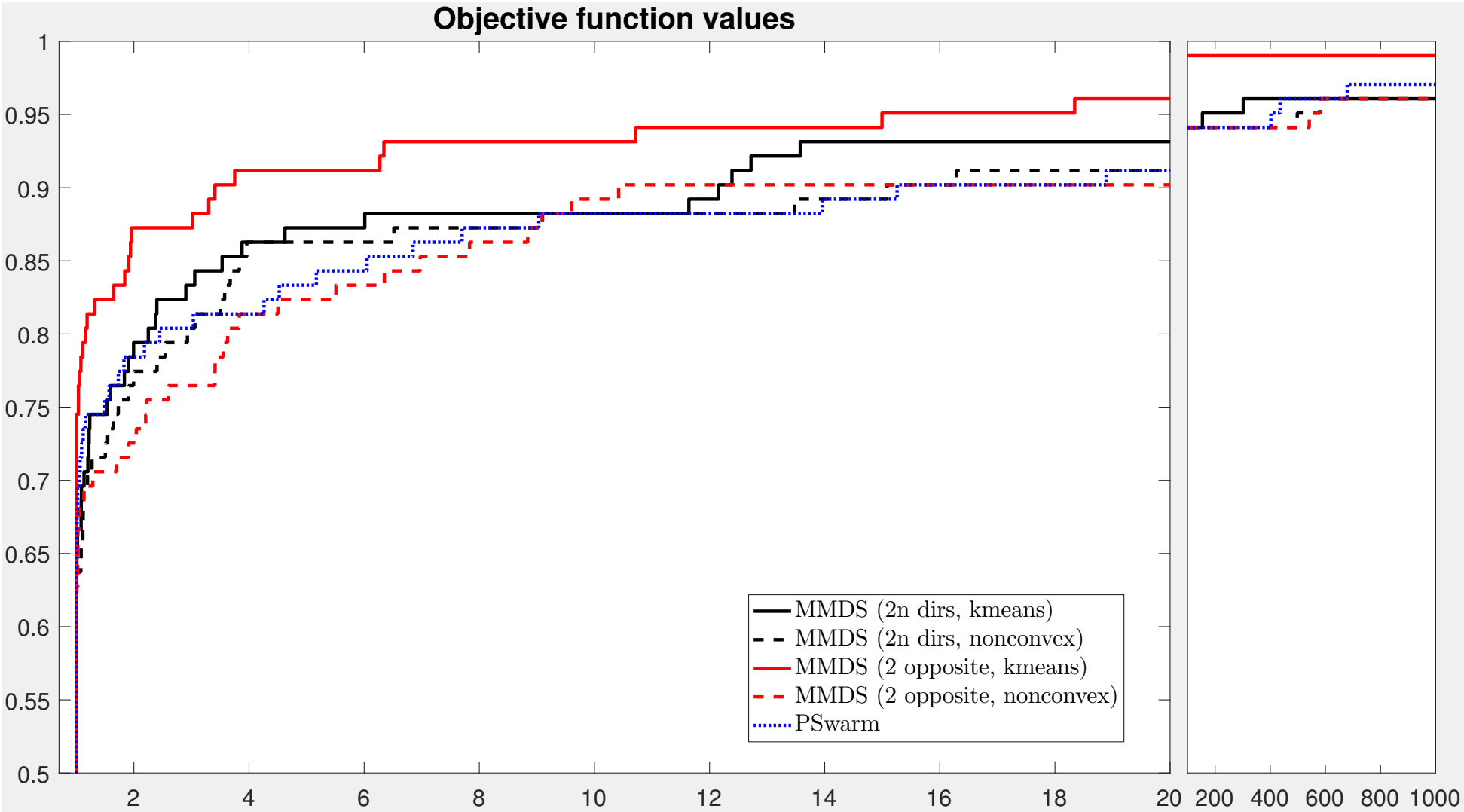
The solver starts with 10 poll centers in serial. A maximum of 10 simultaneous runs is imposed.

A comparison is made with PSwarm (a direct search solver that uses a particle swarm heuristic as a search step).

Performance profiles (# function evaluations)



Performance profiles (quality of final f)



Numerical results (application problem)

Is this approach capable of computing different global minima for an application problem?

Numerical results (application problem)

Is this approach capable of computing different global minima for an application problem?

Consider optimizing the orientation of a part, to be built by additive manufacturing (3D printer), where the staircase effect is to be minimized:

$$\min_{(\theta_x,\theta_y)} \sum_j \begin{cases} \frac{h^2 A_j |d.n_j|}{2} & \text{if } |d.n_j| \neq 1, \\ 0 & \text{otherwise,} \end{cases}$$

where h is the slicing thickness, A_j is the area of the j–triangle, d is the slicing direction, and n_j is the j–triangle normal.

Numerical results (application problem)

Is this approach capable of computing different global minima for an application problem?

Consider optimizing the orientation of a part, to be built by additive manufacturing (3D printer), where the staircase effect is to be minimized:

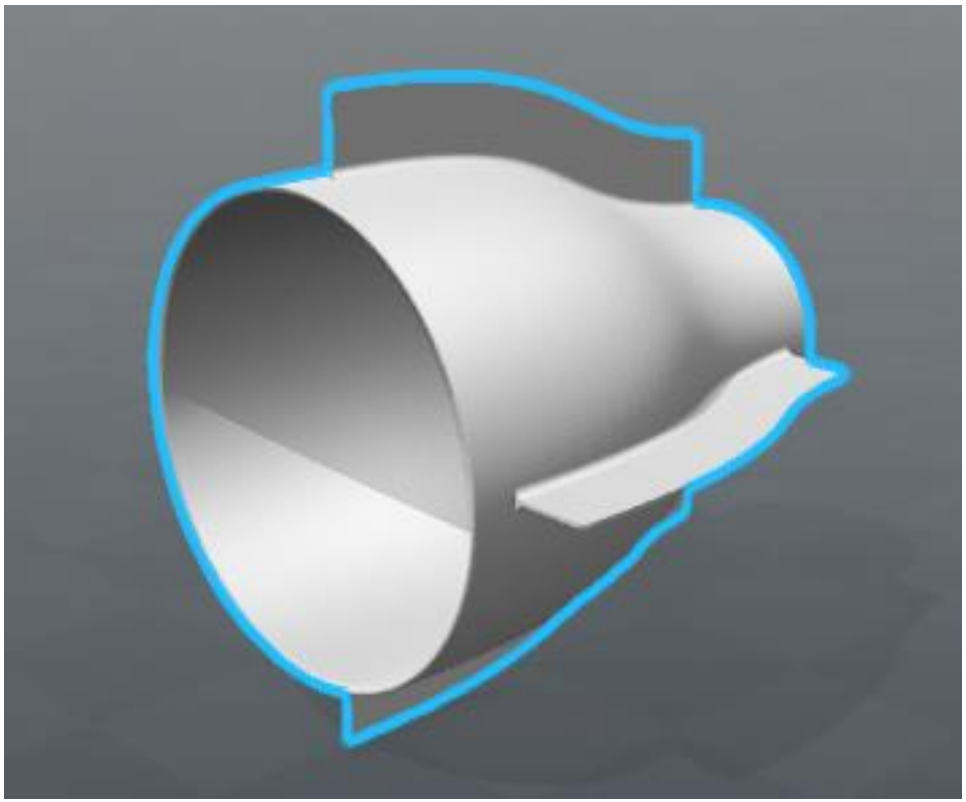
$$\min_{(\theta_x,\theta_y)} \sum_j \begin{cases} \frac{h^2 A_j |d.n_j|}{2} & \text{if } |d.n_j| \neq 1, \\ 0 & \text{otherwise,} \end{cases}$$

where h is the slicing thickness, A_j is the area of the j–triangle, d is the slicing direction, and n_j is the j–triangle normal.

This is a 2D problem where $(\theta_x, \theta_y) \in [0, 180]^2$ are the rotation degrees along the x-axis and y-axis, respectively, in a three dimension space.

A used part

We consider an aerospace complex part for numerical testing.

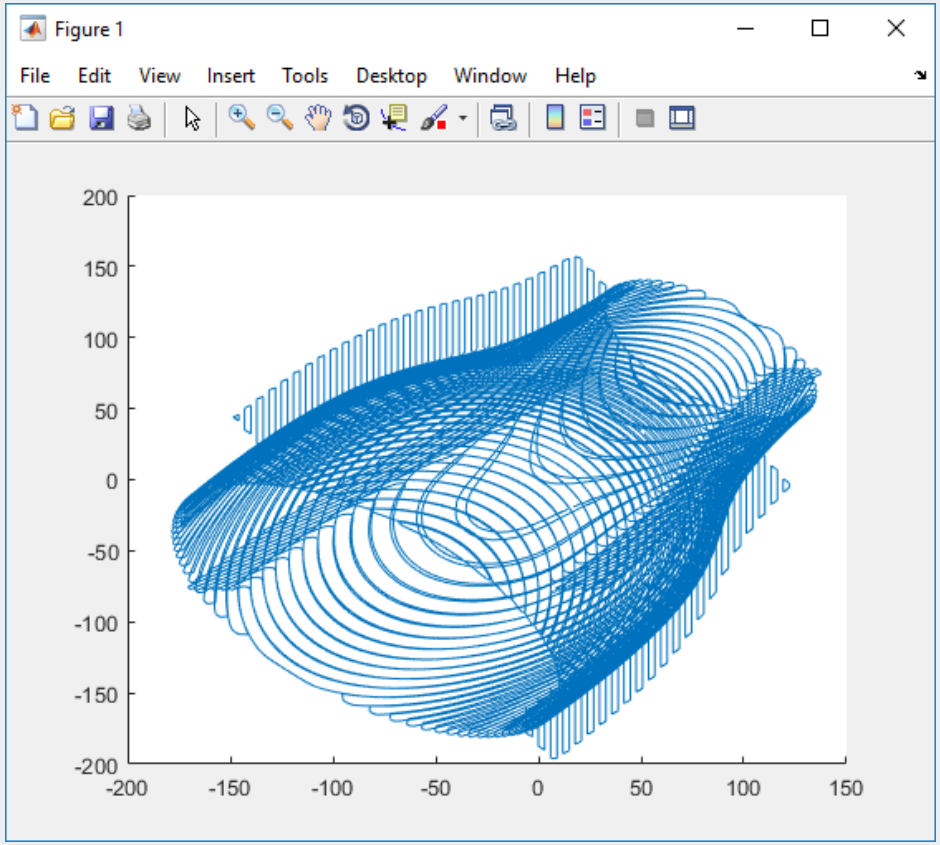


Nearly 80000 triangles are needed to define the part.

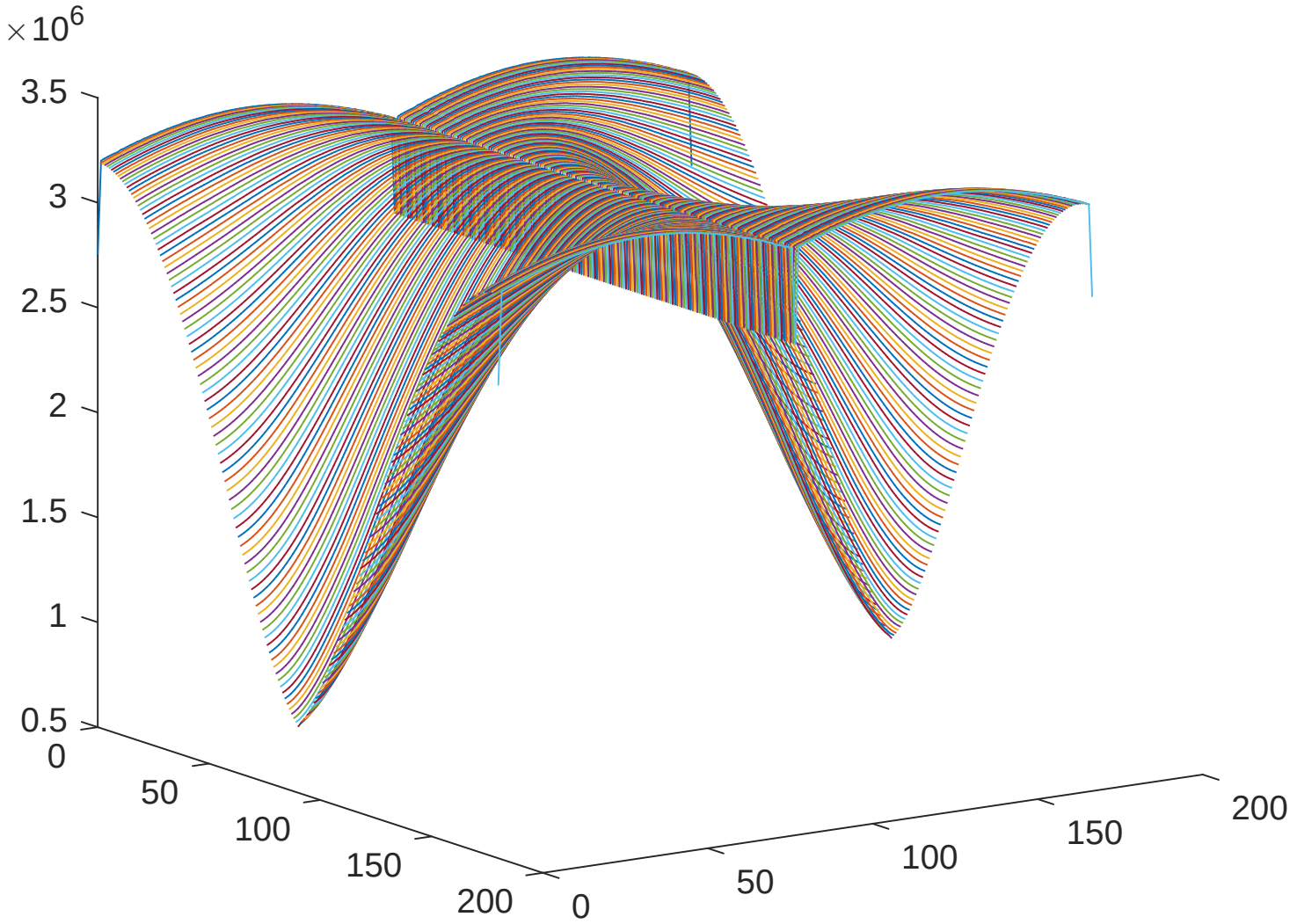
The part/object to be considered

Slices are taken along the z axis with 5mm high, resulting in 64 slices.

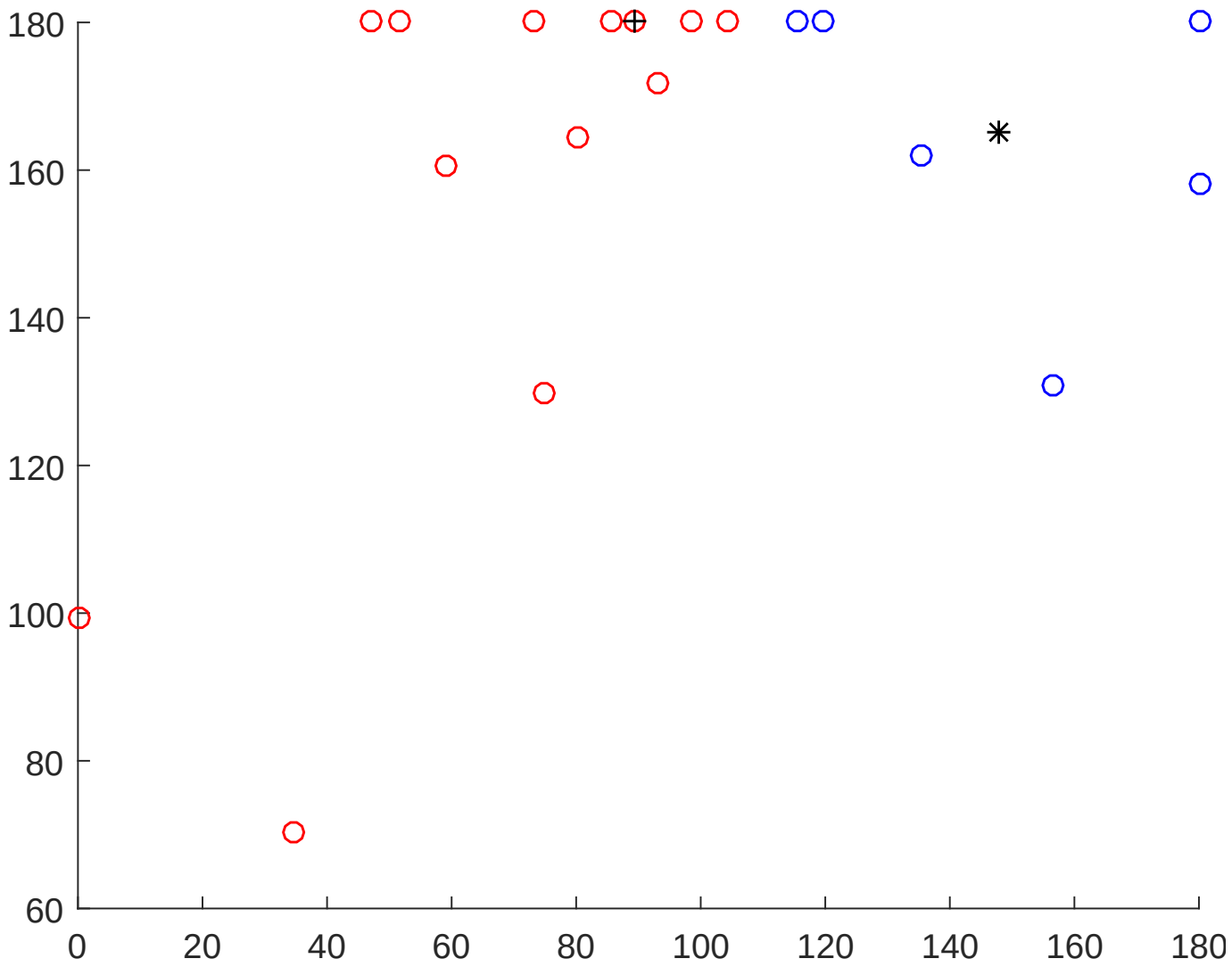
Before optimization



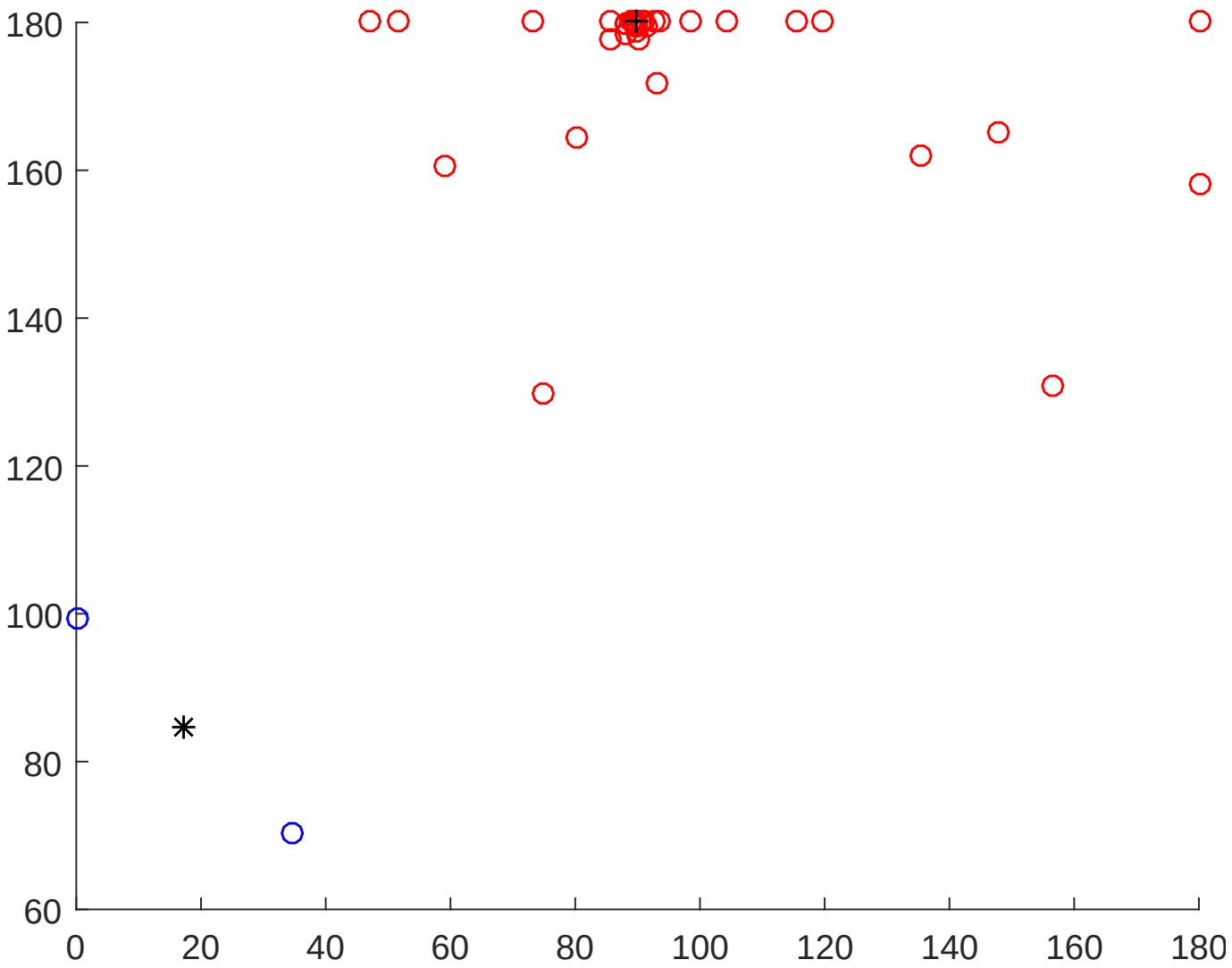
Objective function landscape



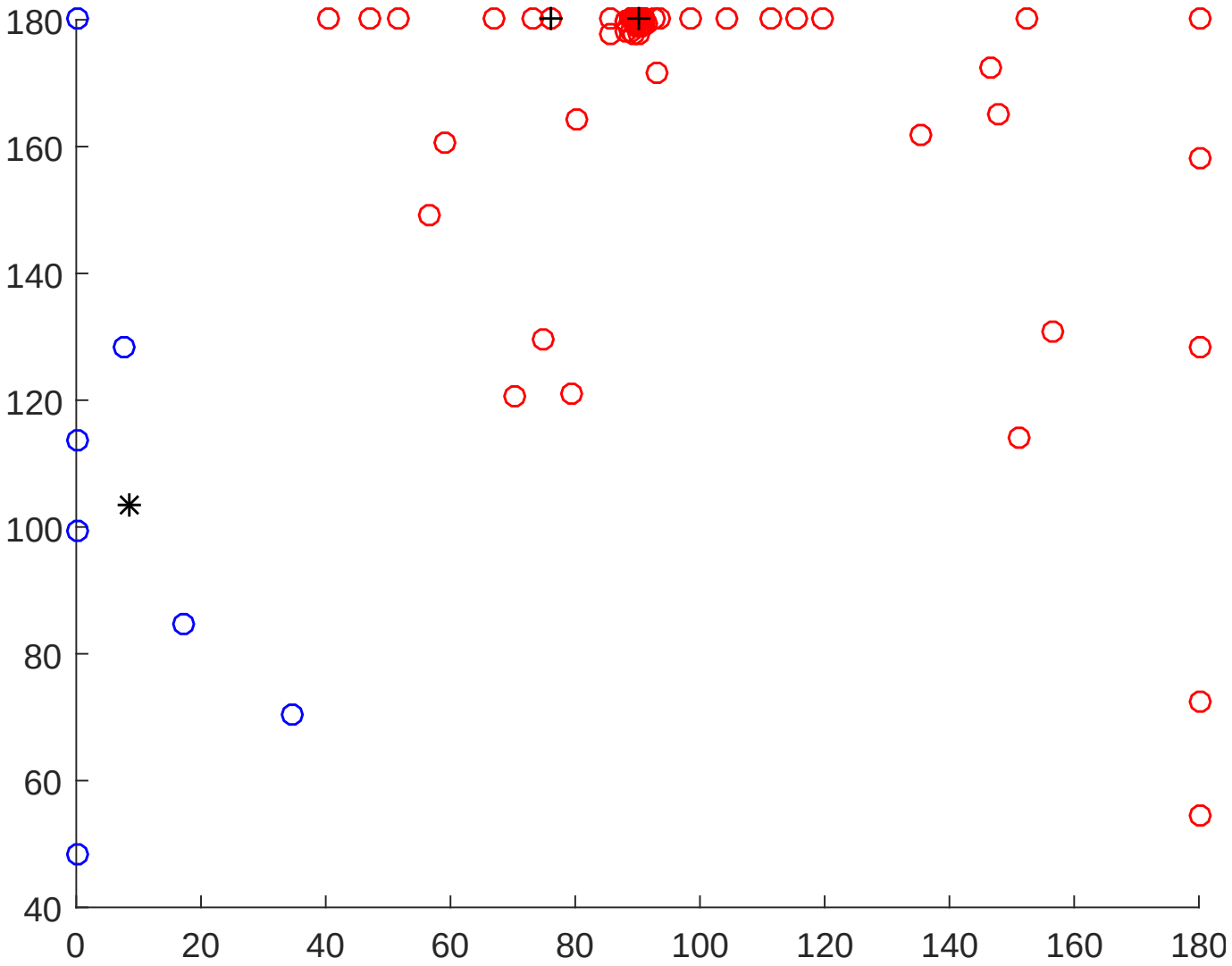
Numerical results (space clustering)



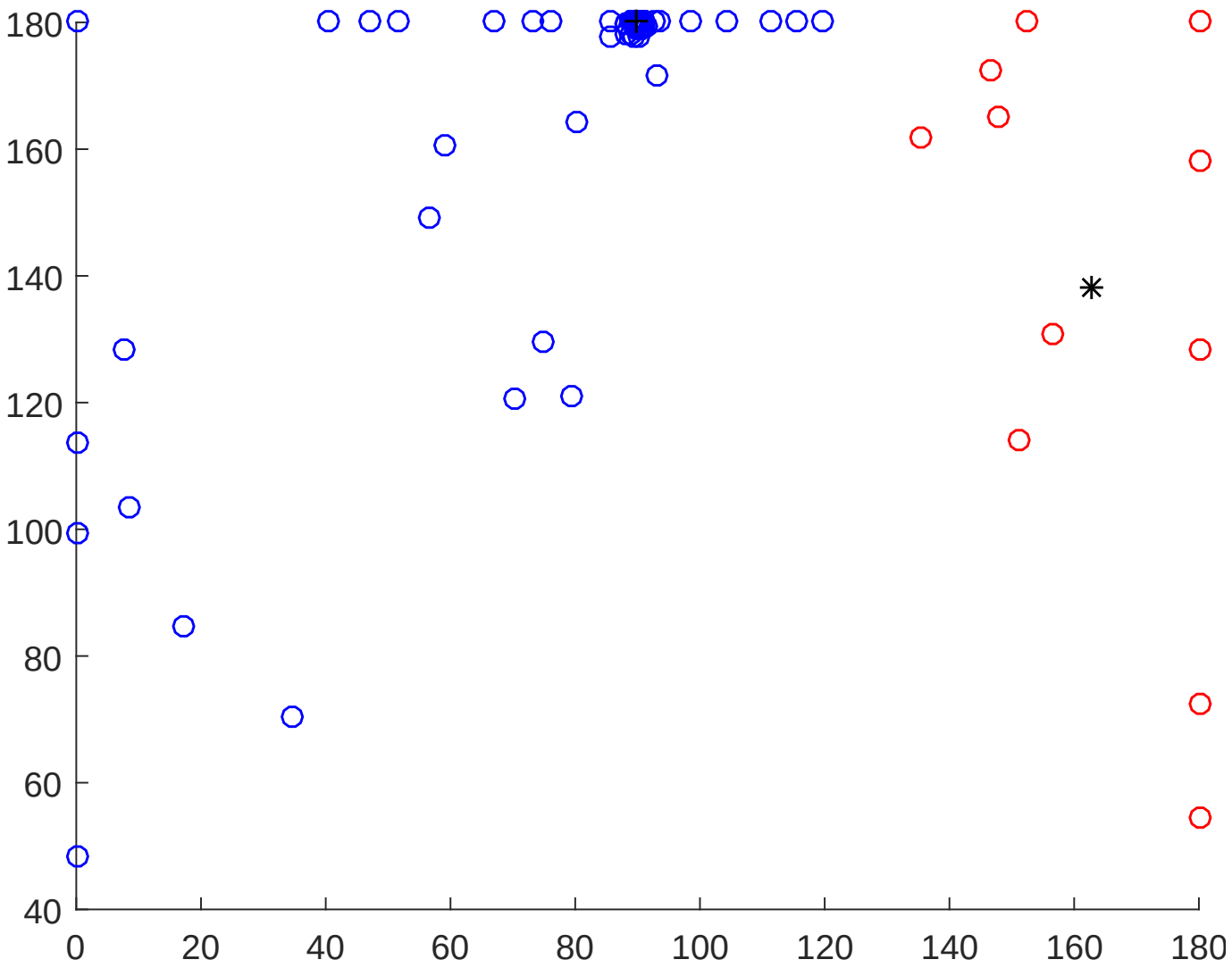
Numerical results (space clustering)



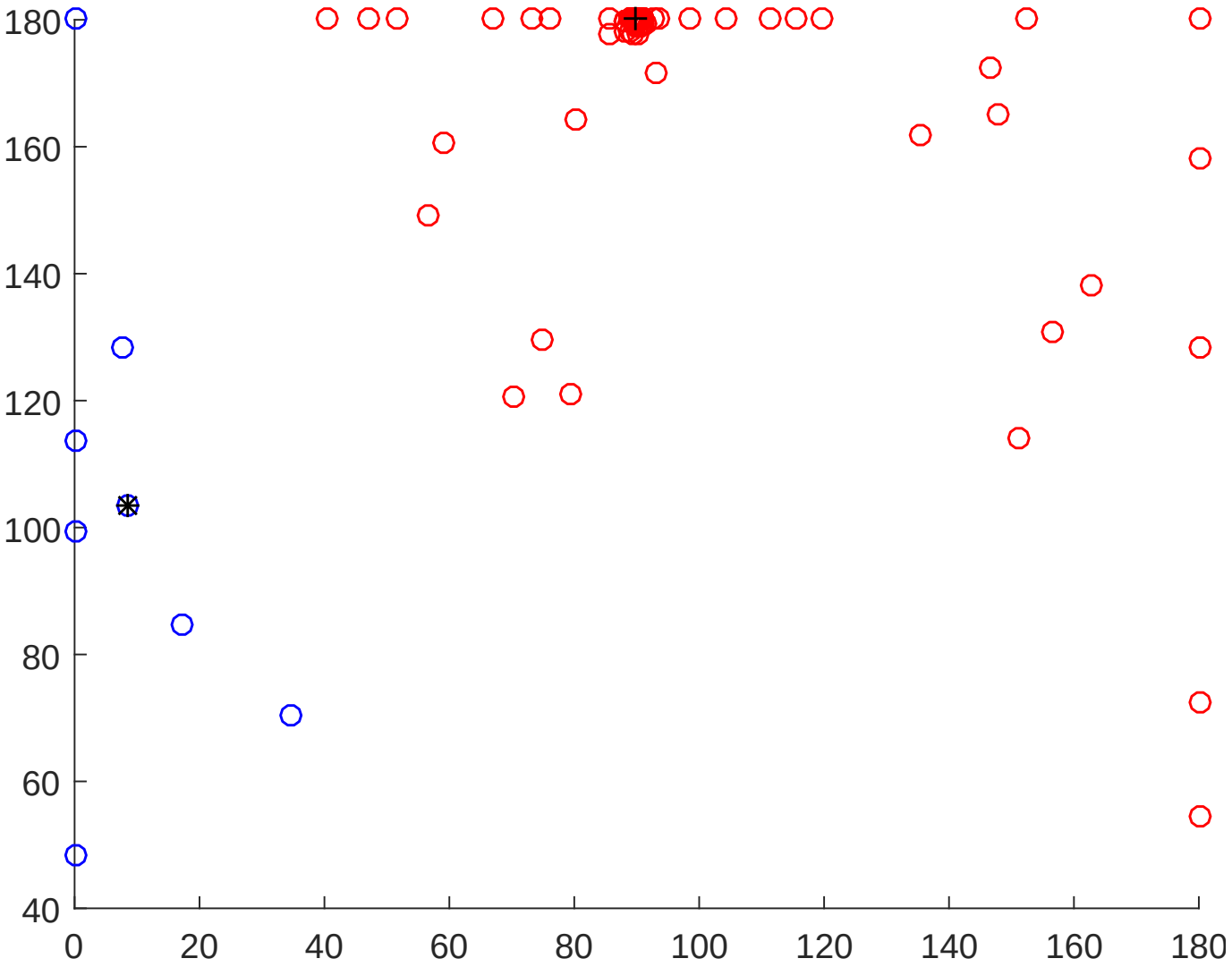
Numerical results (space clustering)



Numerical results (space clustering)



Numerical results (space clustering)



Numerical results (space clustering)

The solver stops at

$$(\theta_x, \theta_y) = (90, 180), \quad \text{with } f^* = 8.1404e + 05,$$

taking 124 function evaluations.

Numerical results (space clustering)

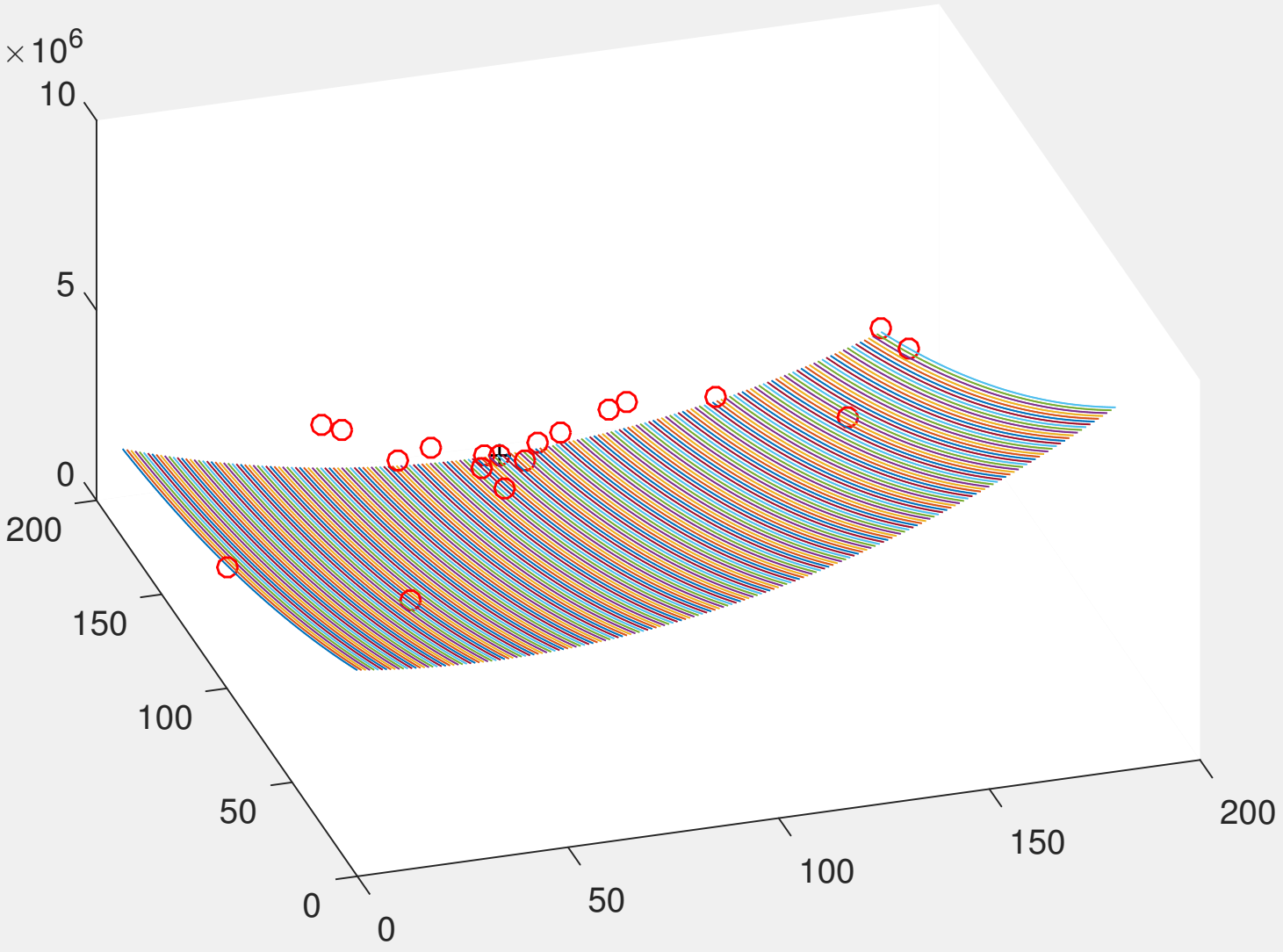
The solver stops at

$$(\theta_x, \theta_y) = (90, 180), \quad \text{with } f^* = 8.1404e + 05,$$

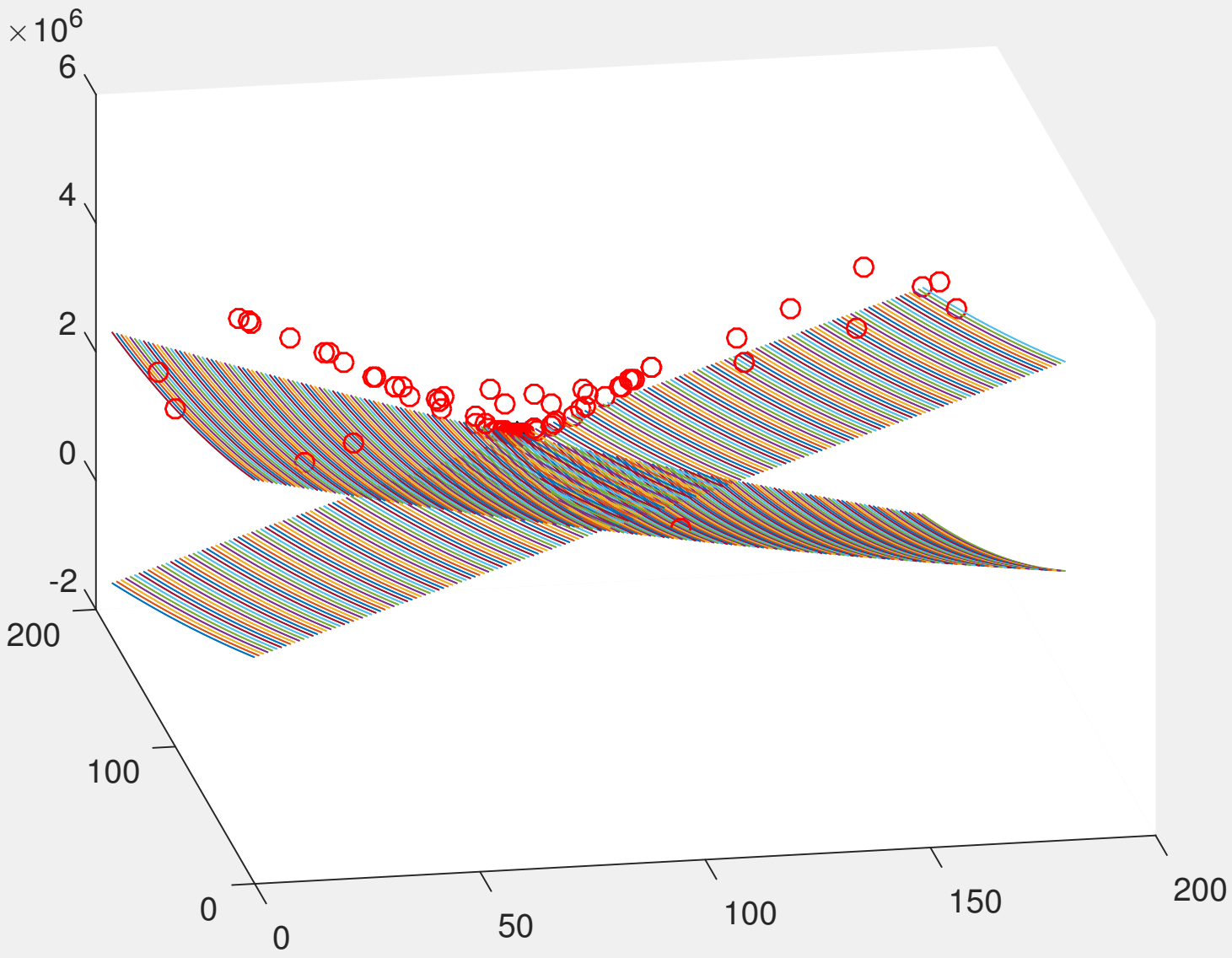
taking 124 function evaluations.

The space clustering approach determined only one of the global minimizers!

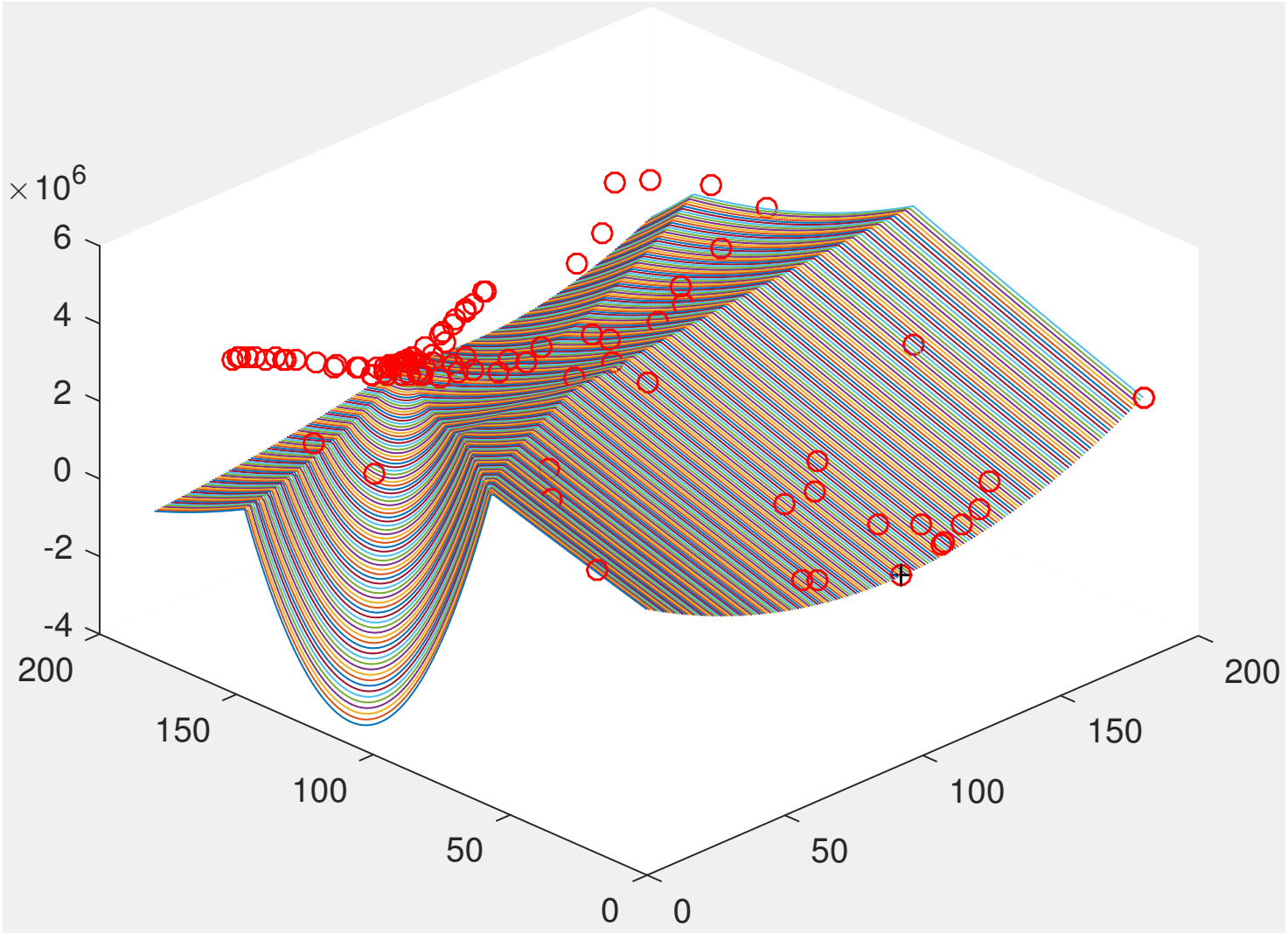
Numerical results (nonconvex modeling) — q_1



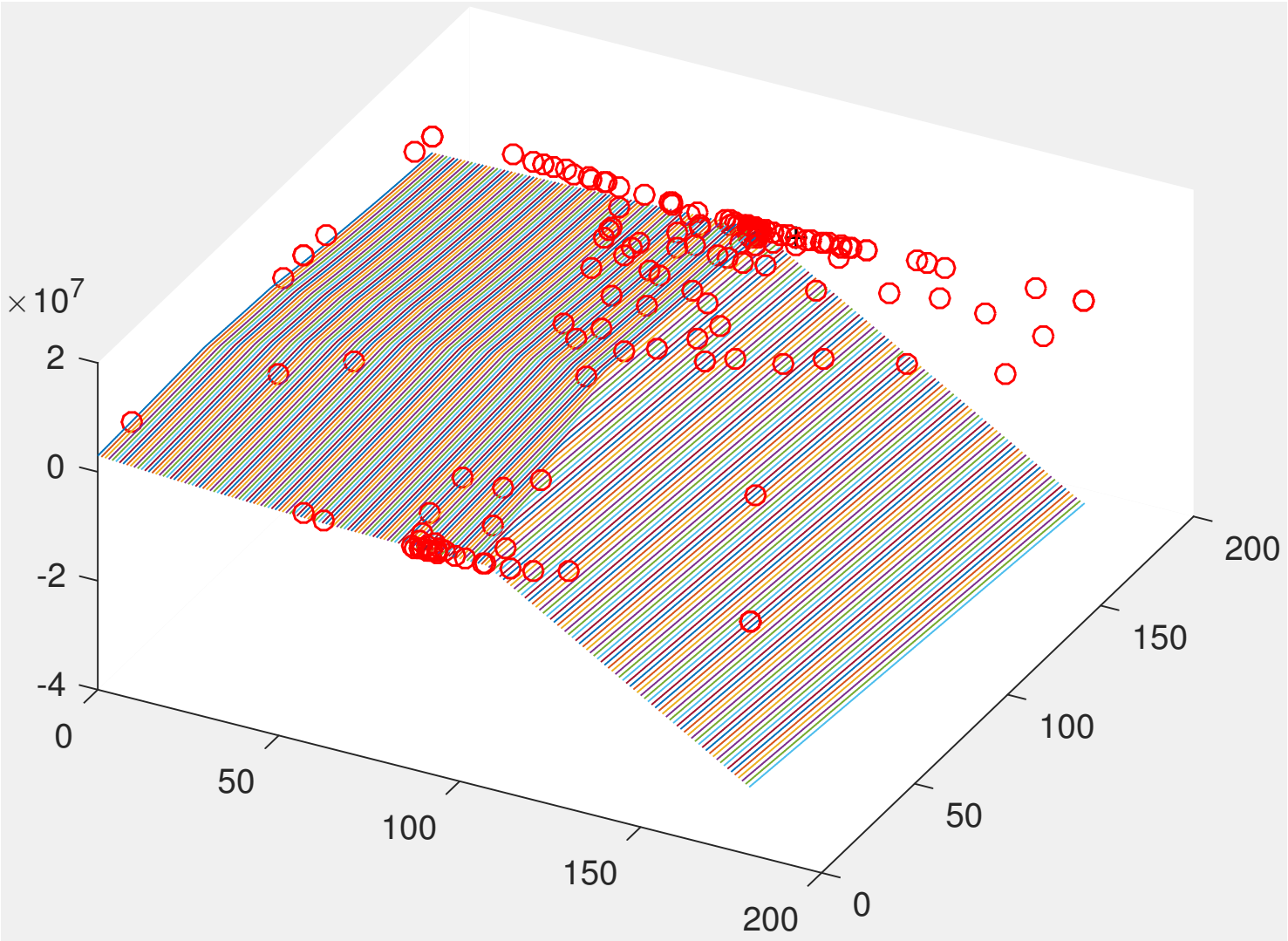
Numerical results (nonconvex modeling) — q_1,q_2



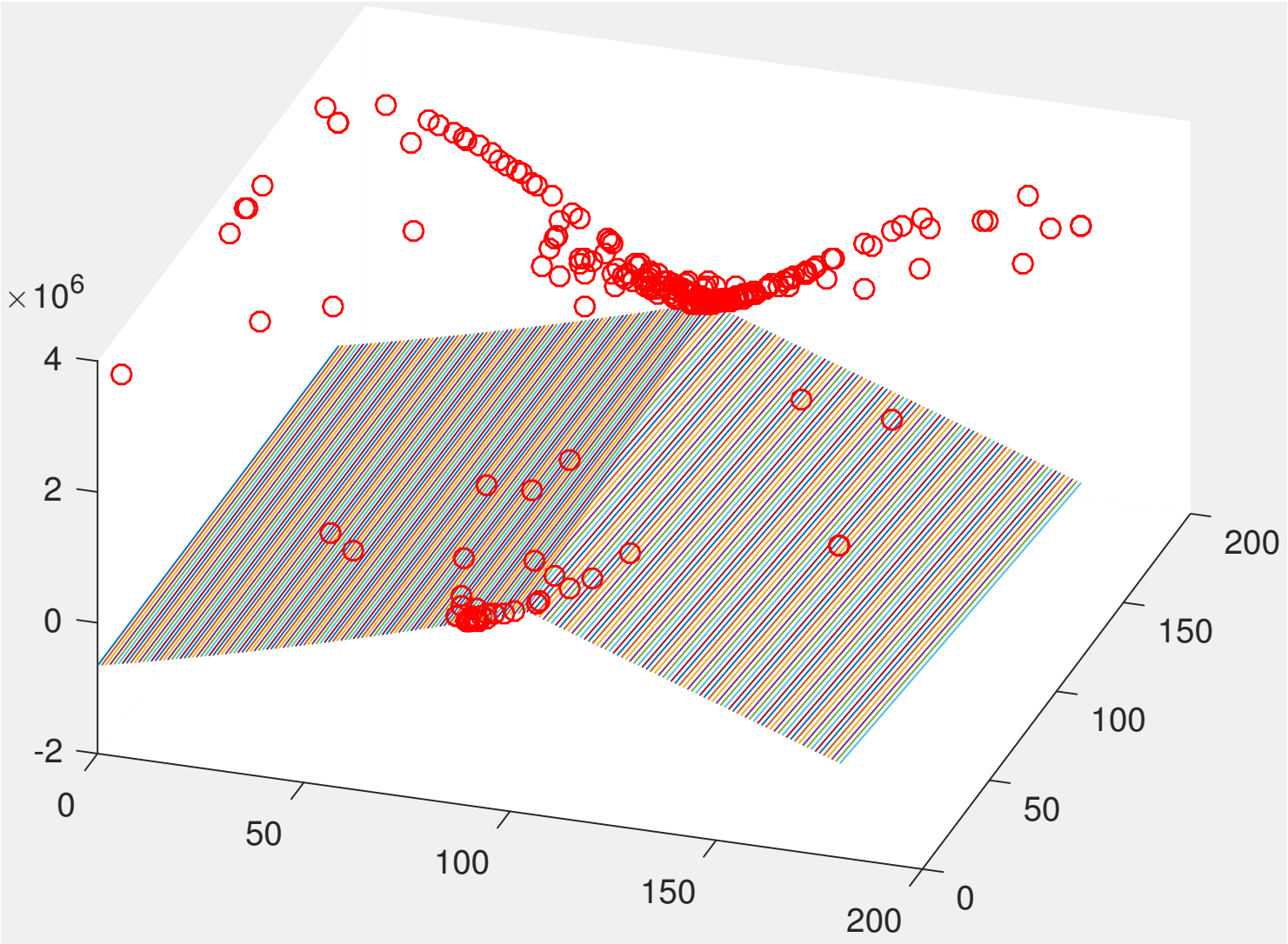
Numerical results (nonconvex modeling) — min of q's



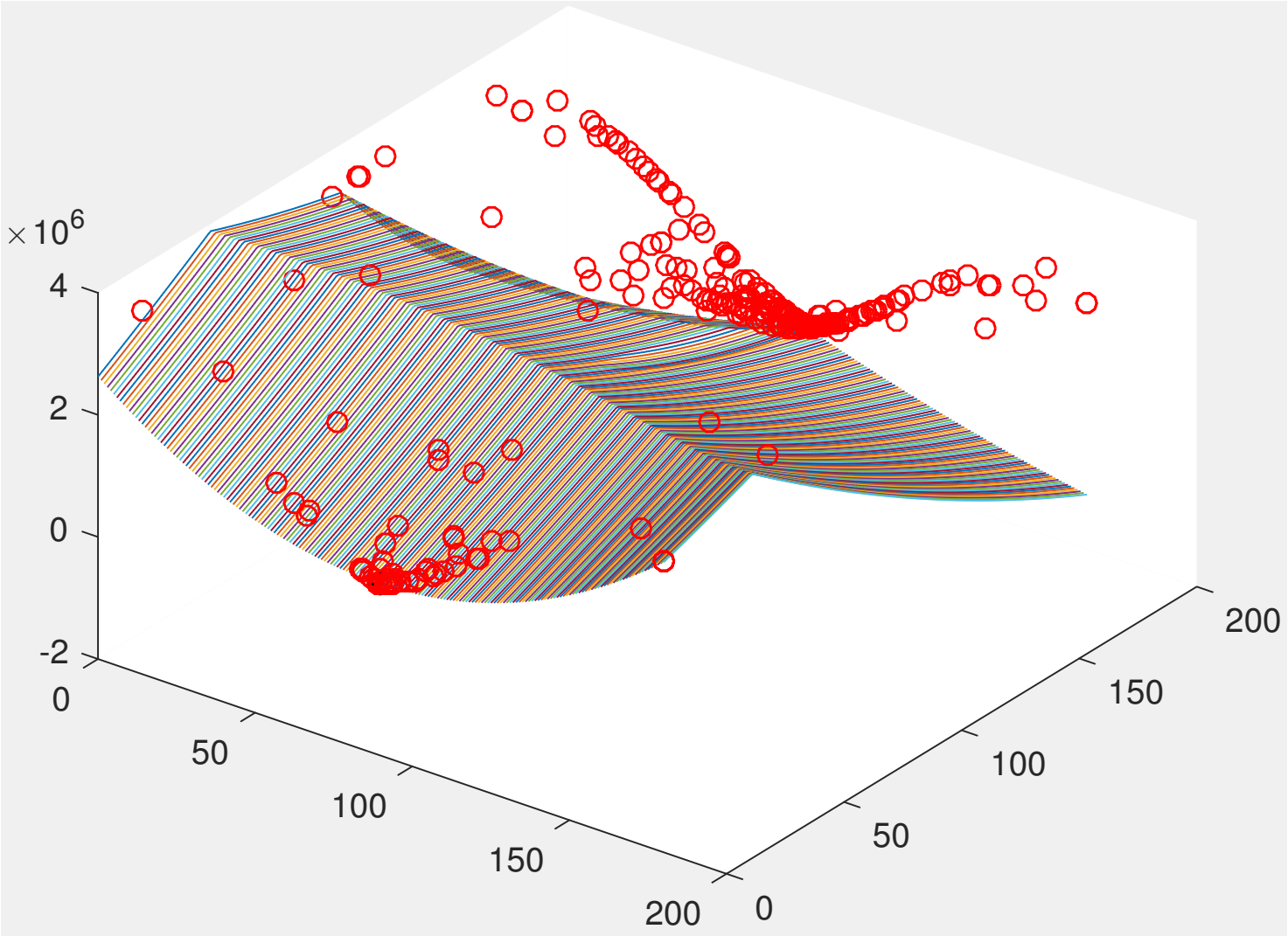
Numerical results (nonconvex modeling) — min of q's



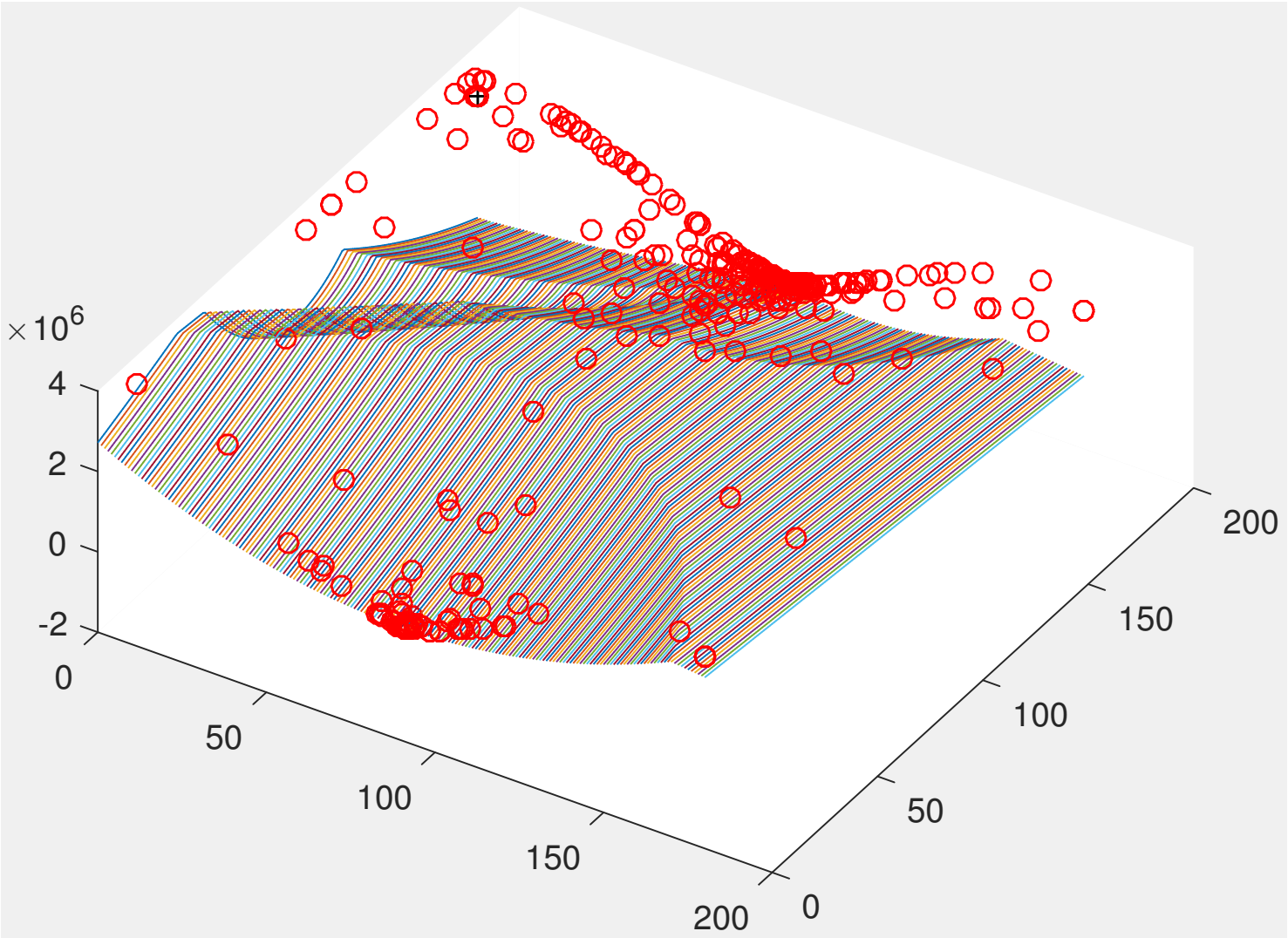
Numerical results (nonconvex modeling) — min of q's



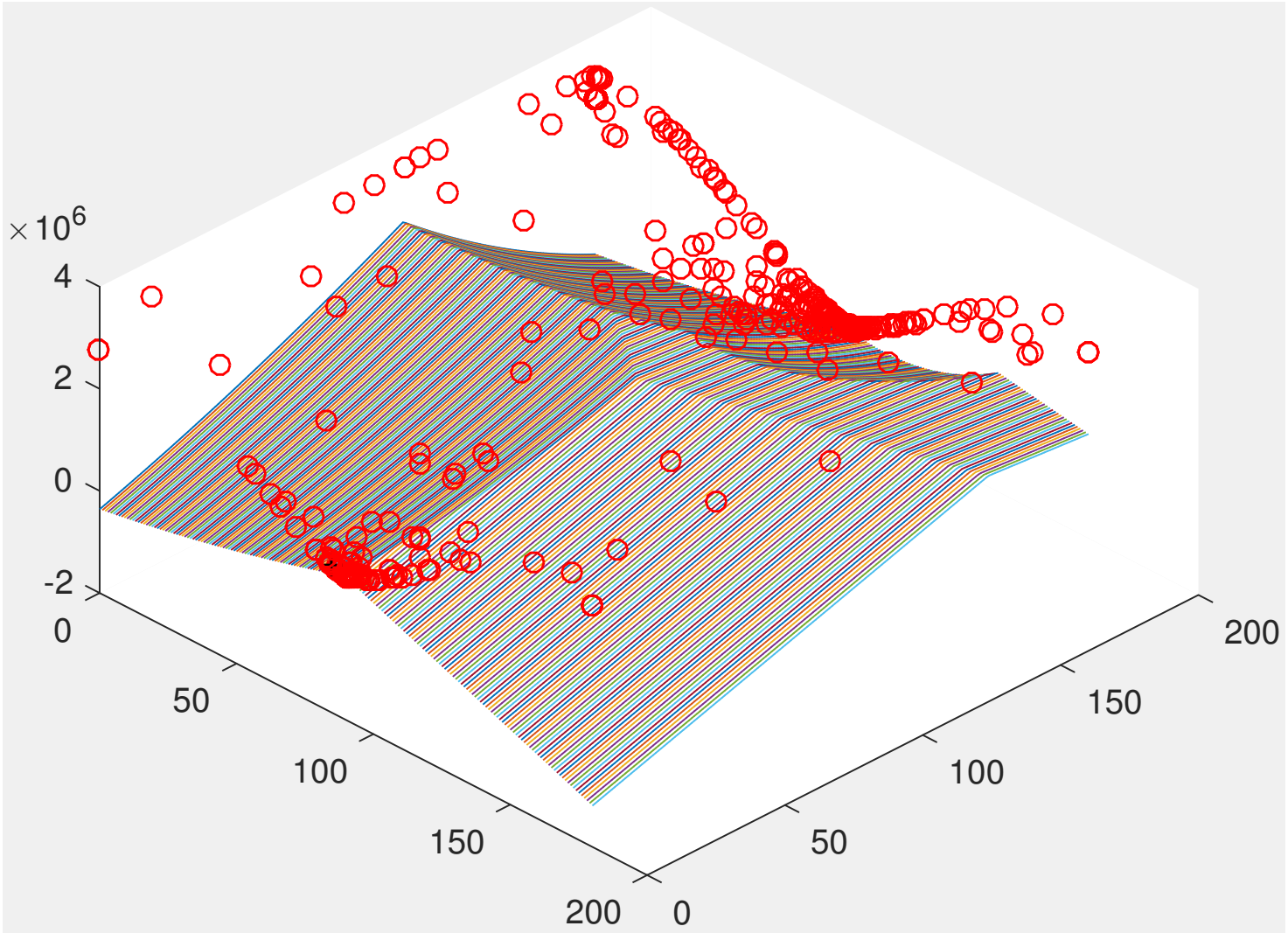
Numerical results (nonconvex modeling) — min of q's



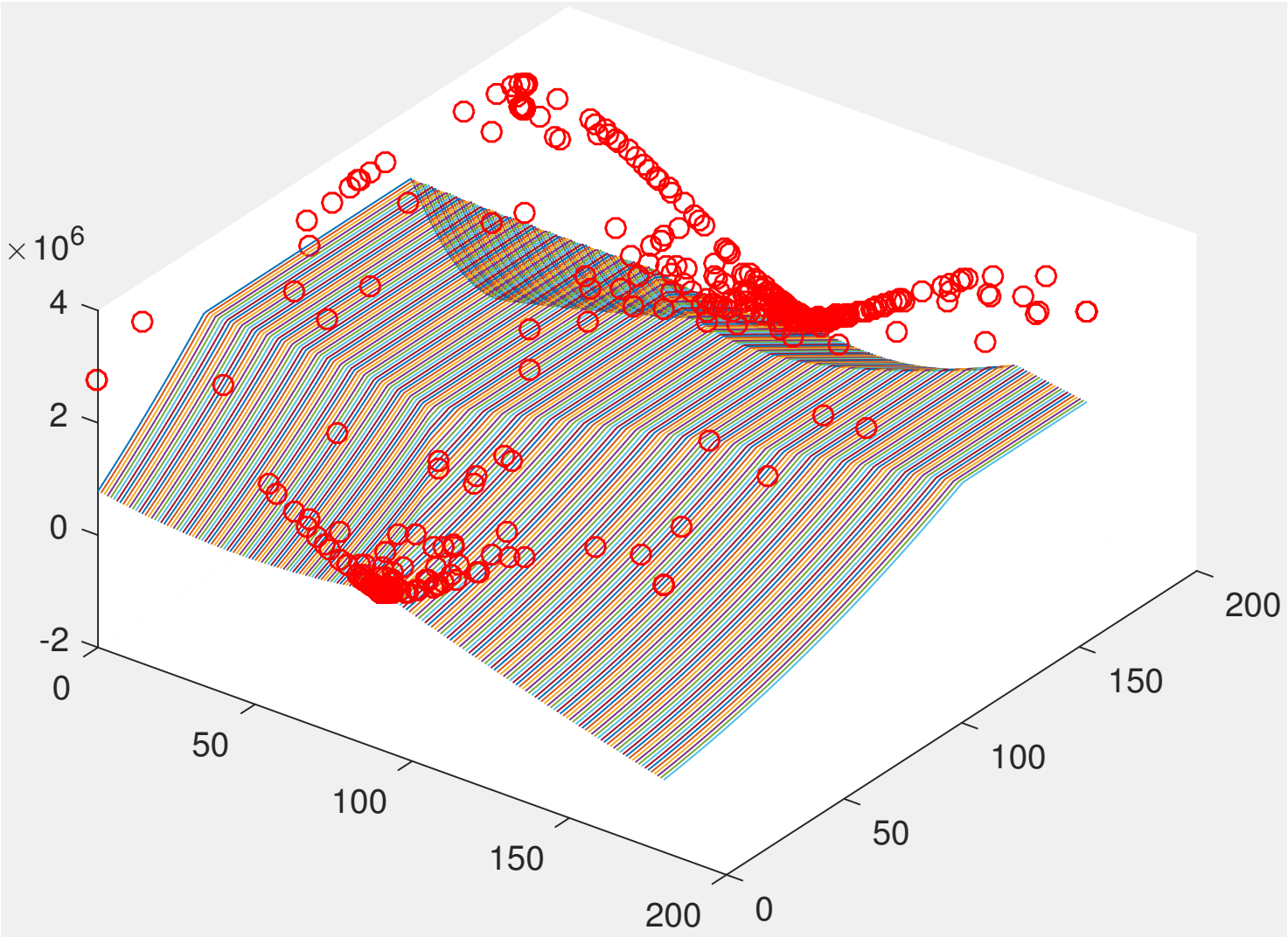
Numerical results (nonconvex modeling) — min of q's



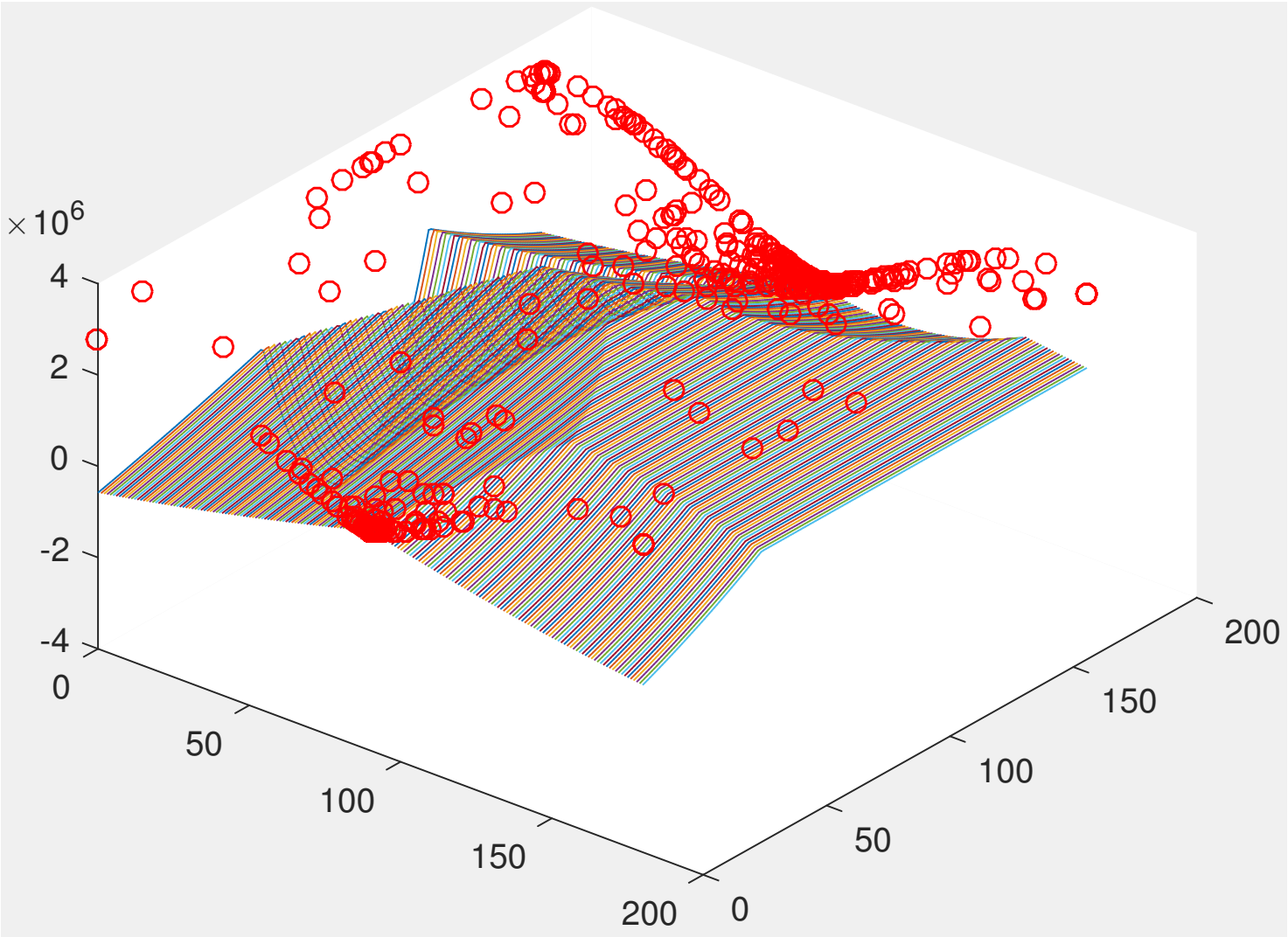
Numerical results (nonconvex modeling) — min of q's



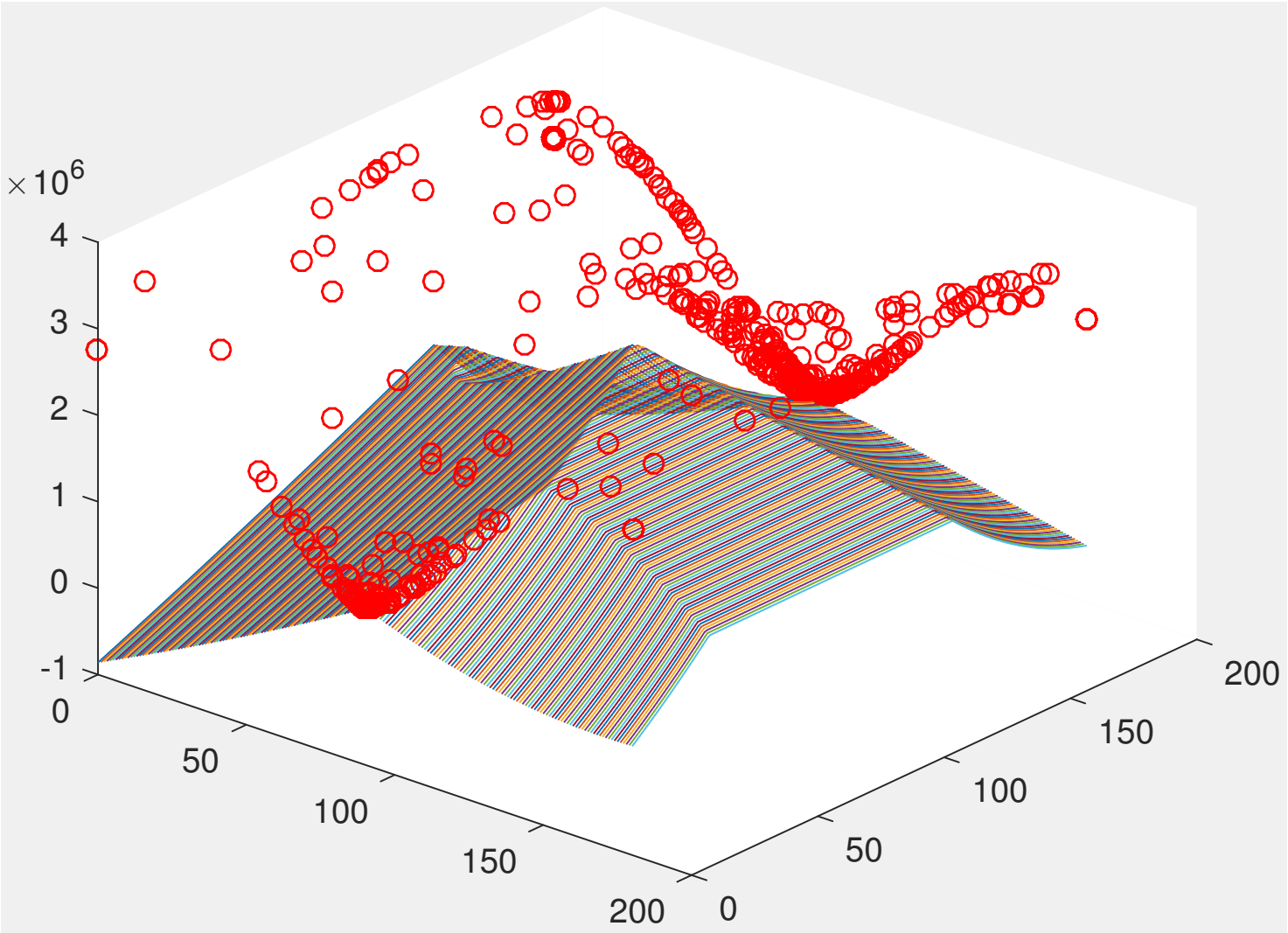
Numerical results (nonconvex modeling) — min of q's



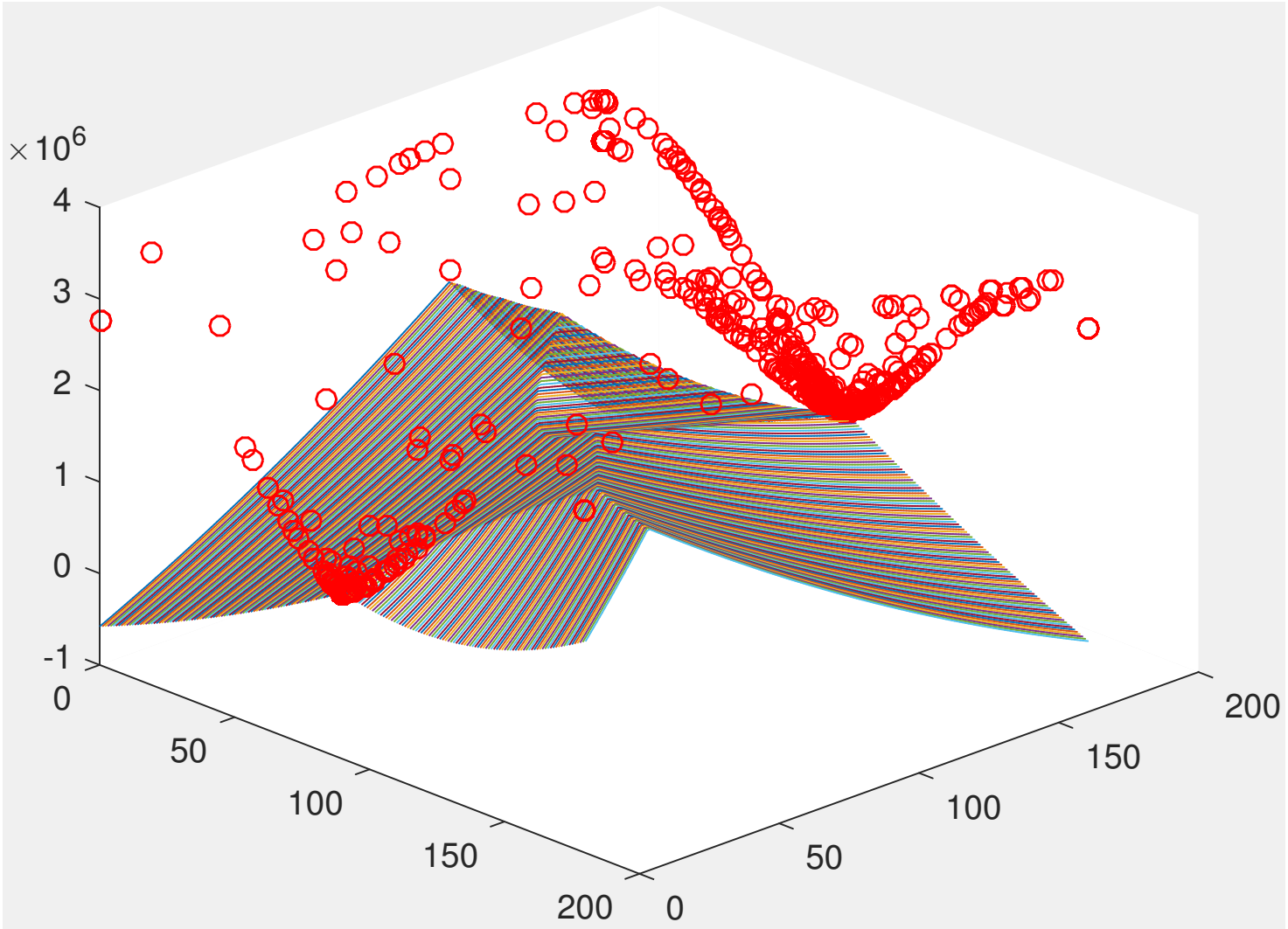
Numerical results (nonconvex modeling) — min of q's



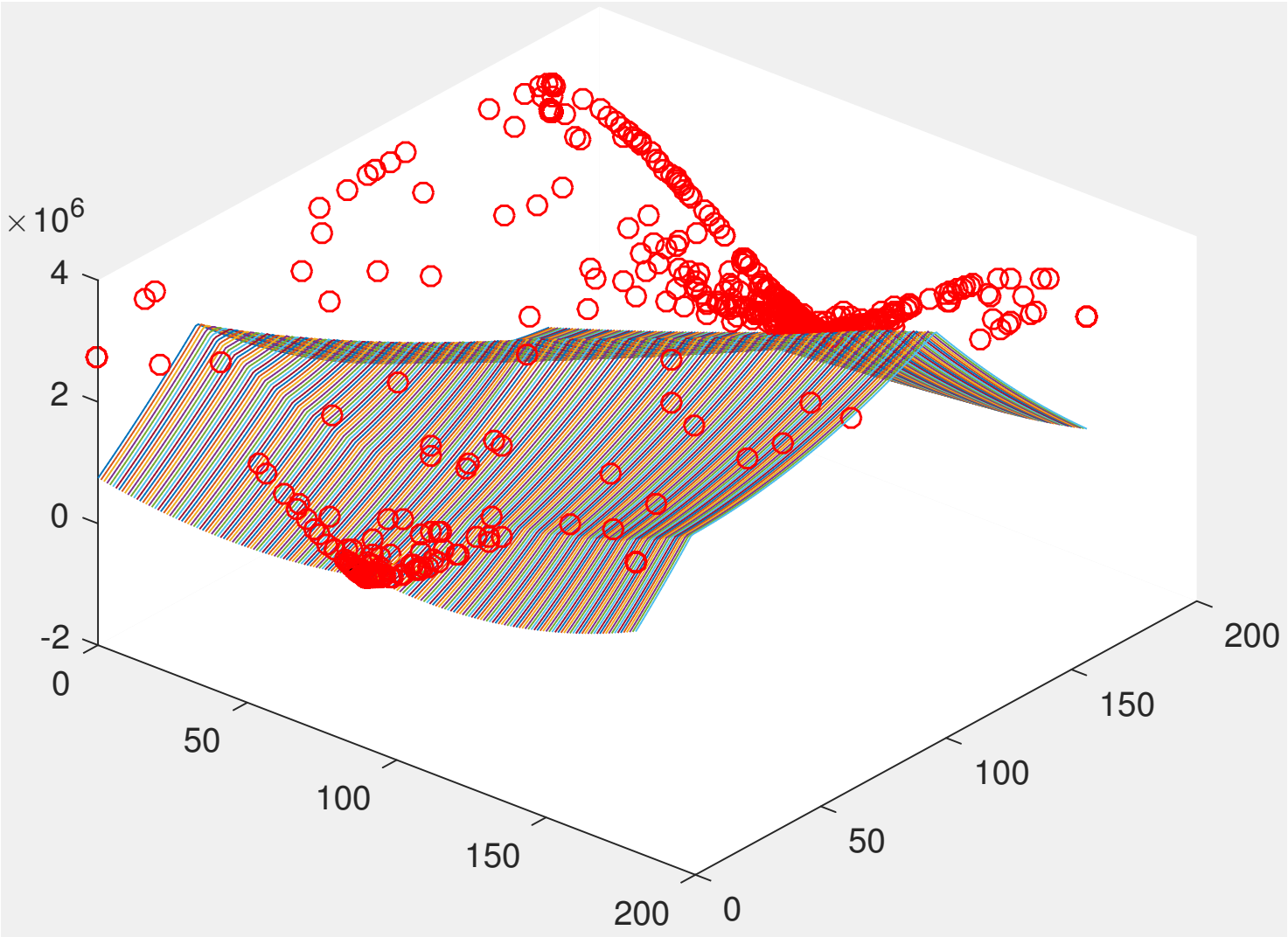
Numerical results (nonconvex modeling) — min of q's



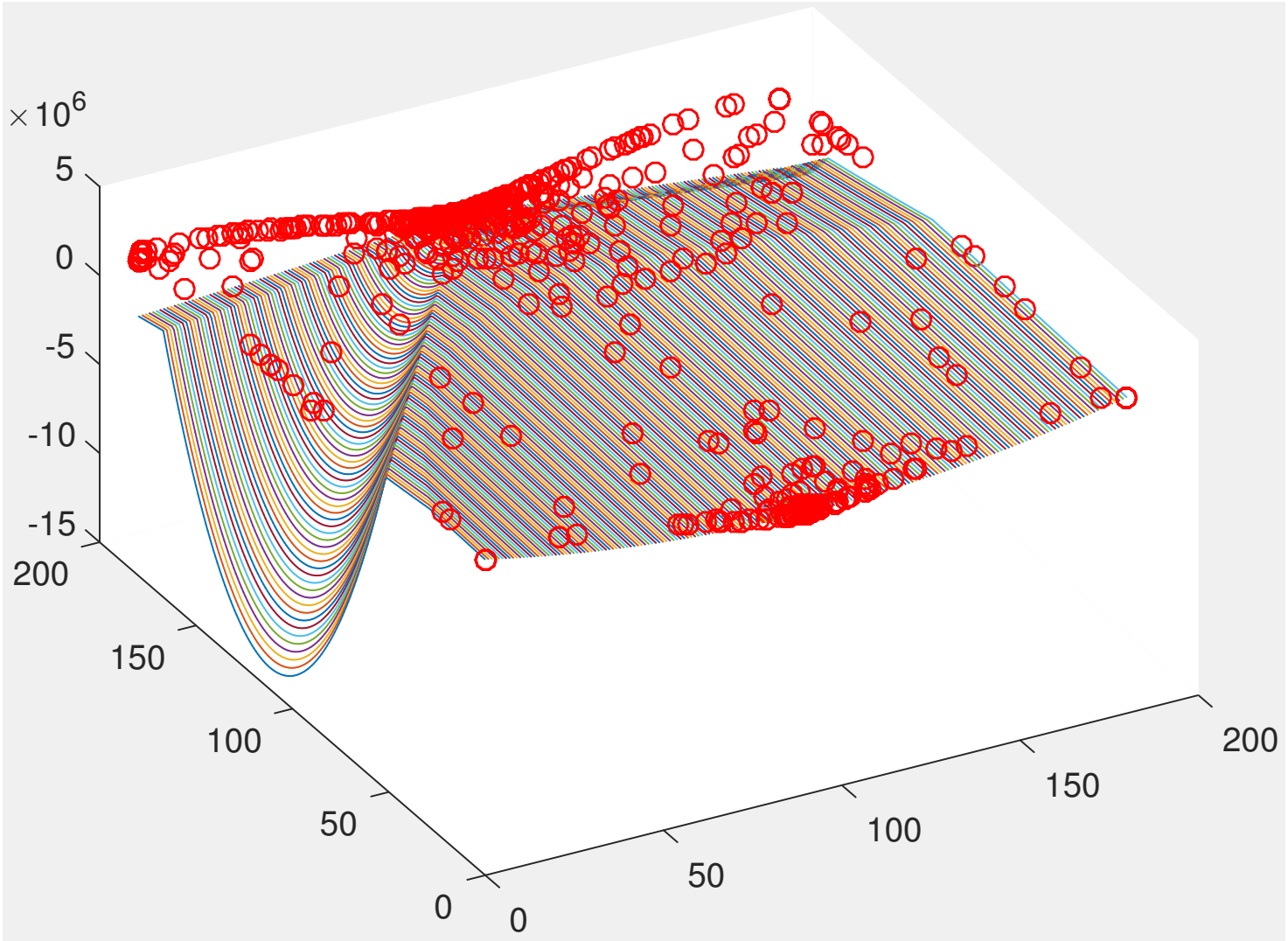
Numerical results (nonconvex modeling) — min of q's



Numerical results (nonconvex modeling) — min of q's



Numerical results (nonconvex modeling) — min of q's



Numerical results (nonconvex modeling)

The solver stops now at

$(\theta_x, \theta_y) = (90, 180)$ and $(\theta_x, \theta_y) = (90, 0)$, both with $f^* = 8.1404e + 05$,

taking 2152 function evaluations.

Numerical results (nonconvex modeling)

The solver stops now at

$(\theta_x, \theta_y) = (90, 180)$ and $(\theta_x, \theta_y) = (90, 0)$, both with $f^* = 8.1404e + 05$,

taking 2152 function evaluations.

The nonconvex modeling approach determined both global minimizers of the problem!

The result of Optimization

Slices are taken along the z axis with 5mm high.

After optimization