Using random models in derivative free optimization

Katya Scheinberg

Lehigh University

(mainly based on work with A. Bandeira and L.N. Vicente and also with A.R. Conn, Ph.Toint and C. Cartis)

Derivative free optimization

- Unconstrained optimization problem

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

- Function f is computed by a black box, no derivative information is available.
- Numerical noise is often present, but we do not account for it in this talk!
- $f \in C^1$ or C^2 and is deterministic.
- May be expensive to compute.

Black box function evaluation

$$x=(x_1, x_2, x_3, \ldots, x_n)$$

$$\downarrow$$

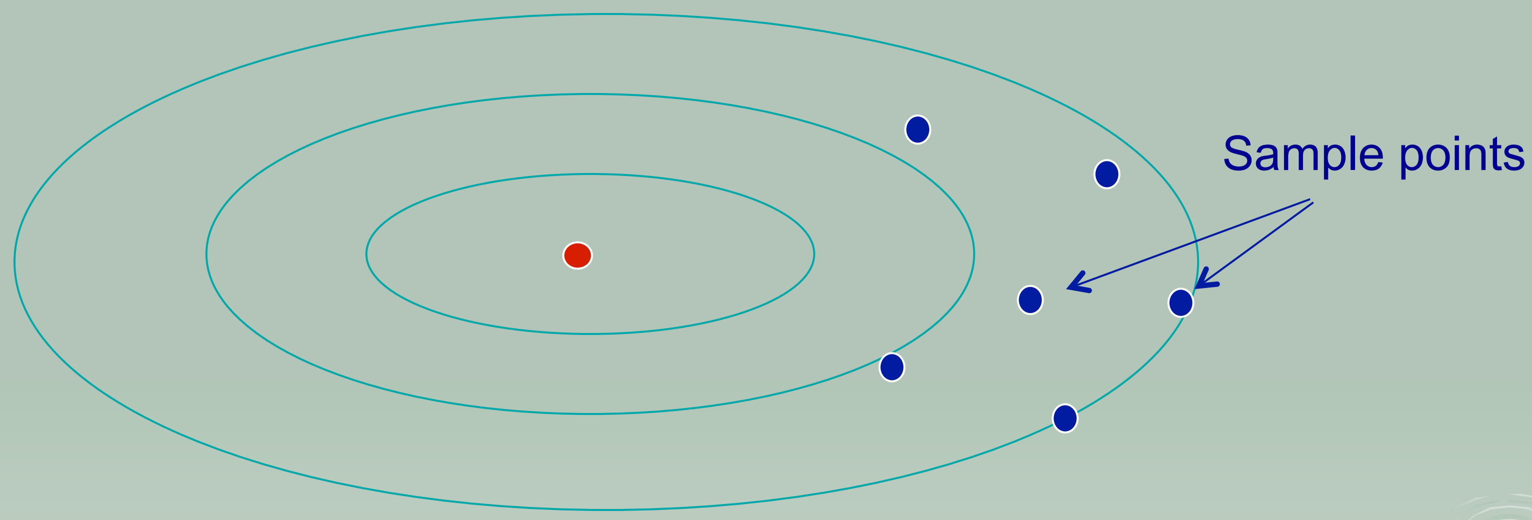
$$v=f(x_1, \ldots, x_n)$$

$$\downarrow$$

$$v$$

All we can do is "sample" the function values at some sample points

Sampling the black box function



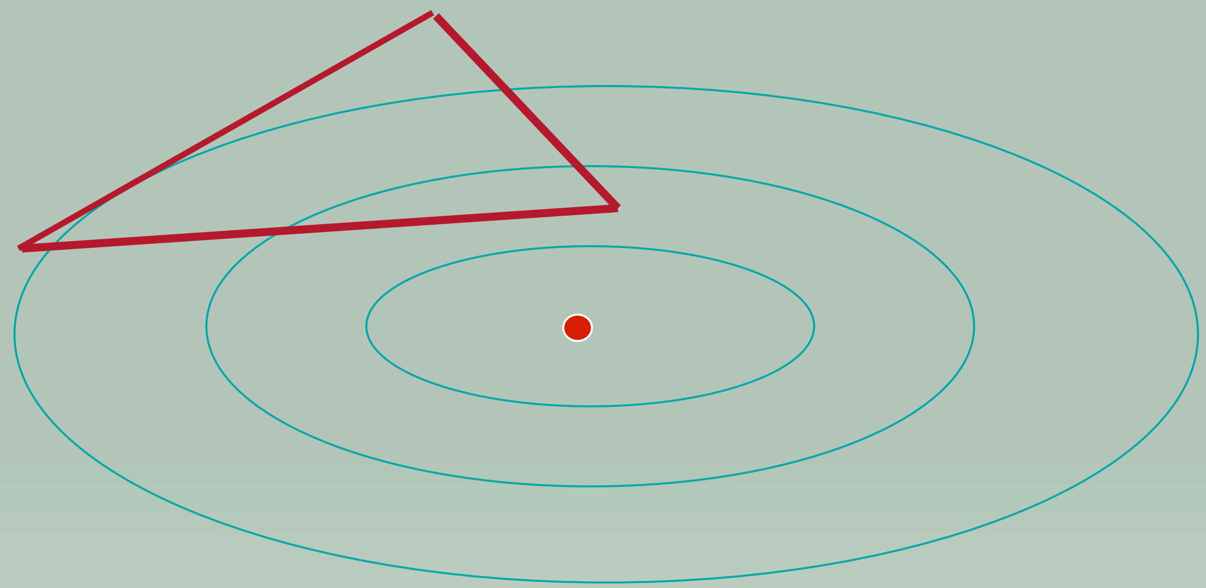
How to choose and to use the sample points and the functions values defines different DFO methods

Outline

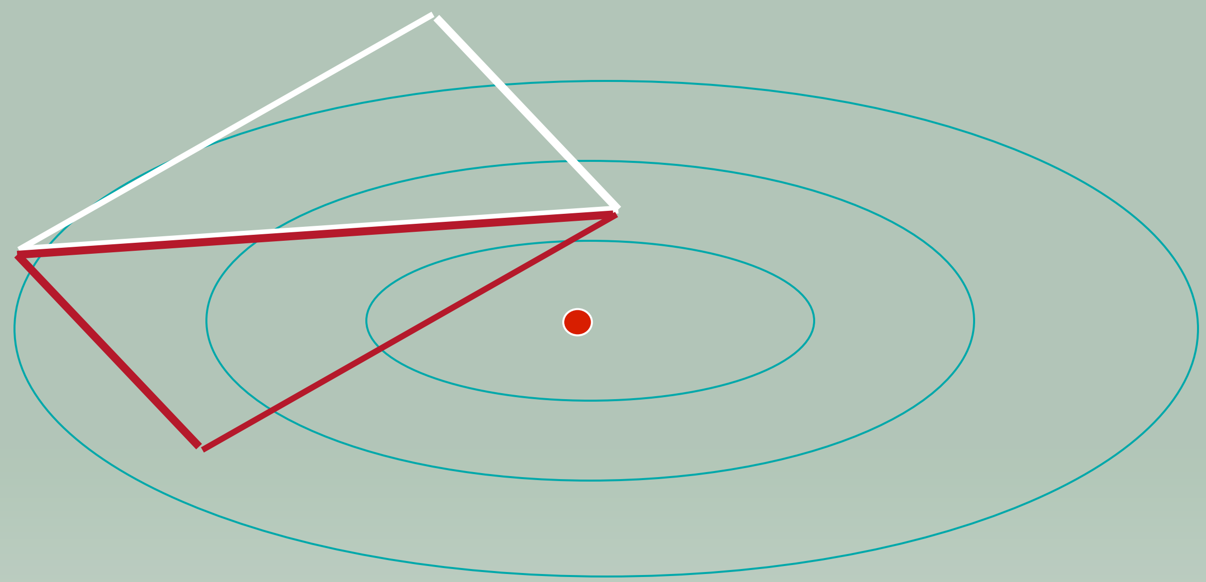
- Review with illustrations of existing methods as motivation for using models.
- Polynomial interpolation models and motivation for models based on random sample sets.
- Structure recovery using random sample sets and compressed sensing in DFO.
- Algorithms using random models and conditions on these models.
- Convergence theory for TR framework based on random models.

Algorithms

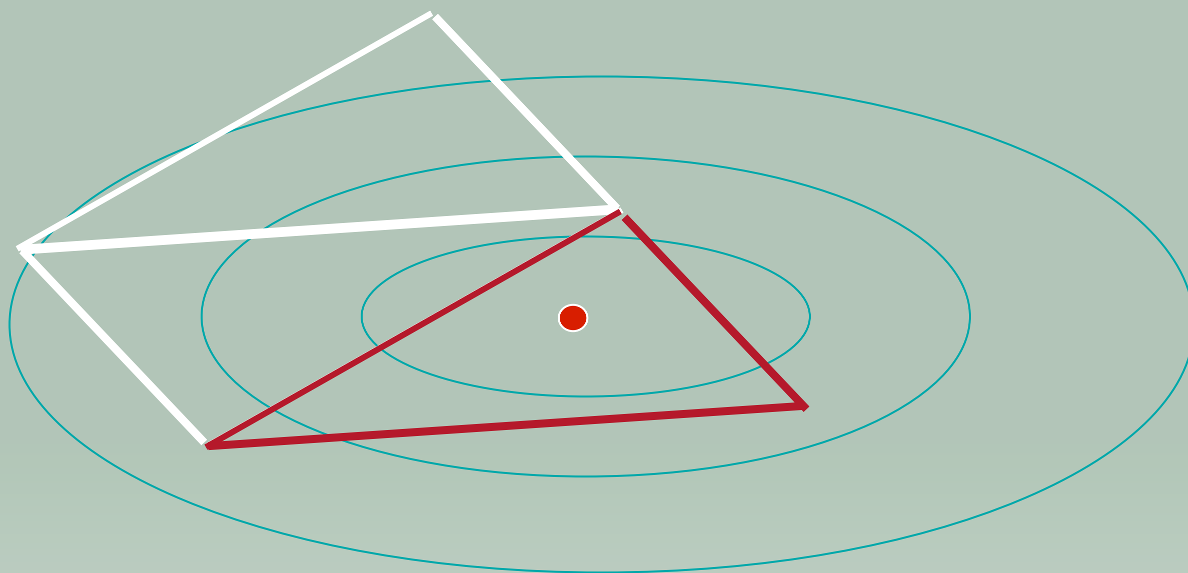
Nelder-Mead method (1965)



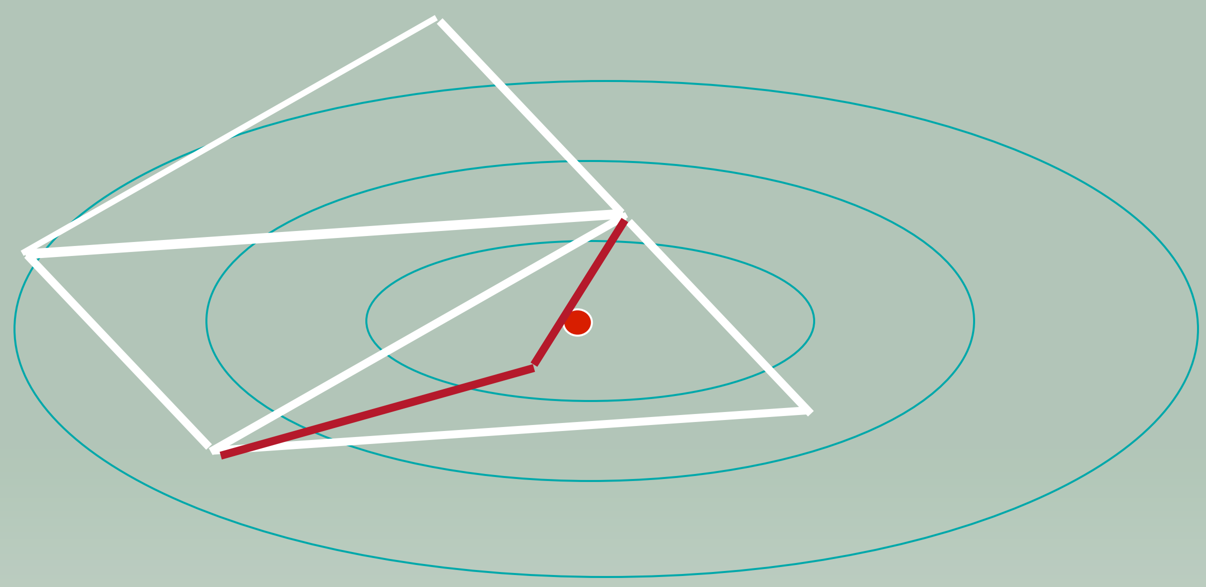
Nelder-Mead method (1965)



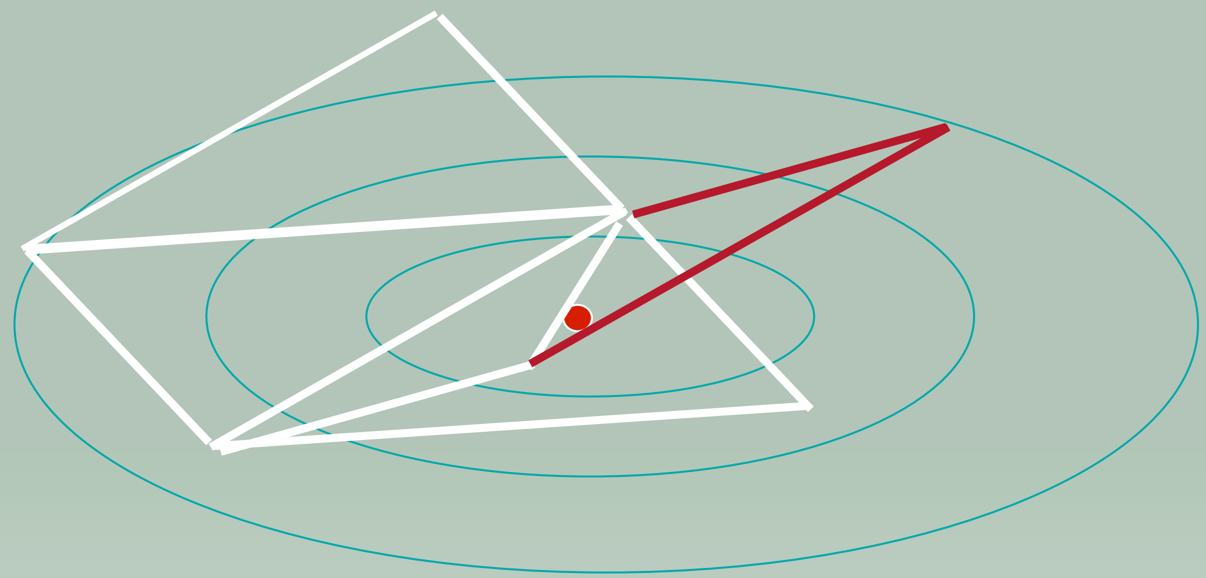
Nelder-Mead method (1965)



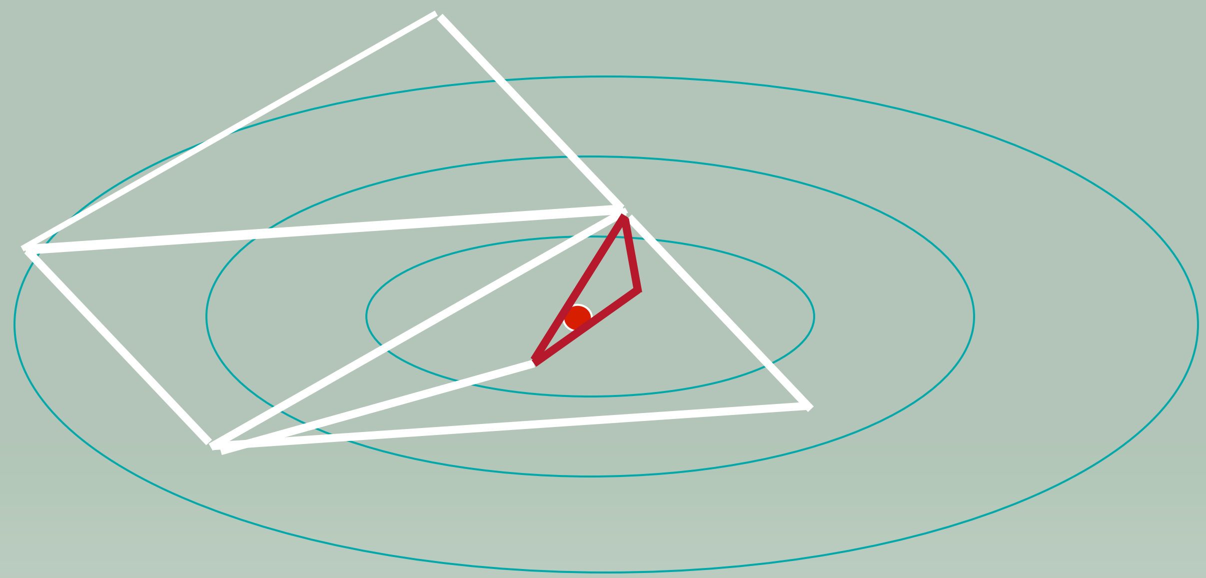
Nelder-Mead method (1965)



Nelder-Mead method (1965)



Nelder-Mead method (1965)

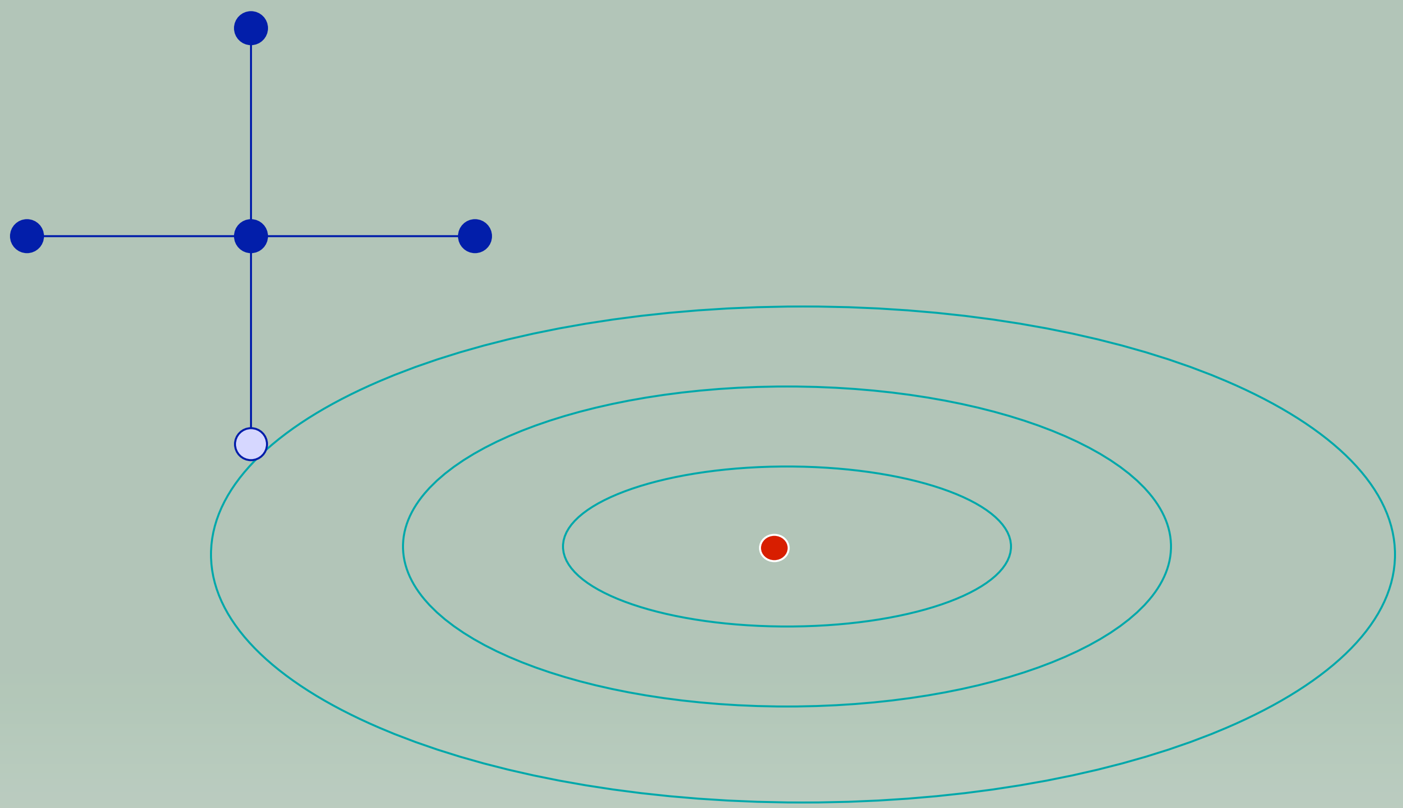


The simplex changes shape during the algorithm to adapt to curvature. But the shape can deteriorate and NM gets stuck

Nelder Mead on Rosenbrock

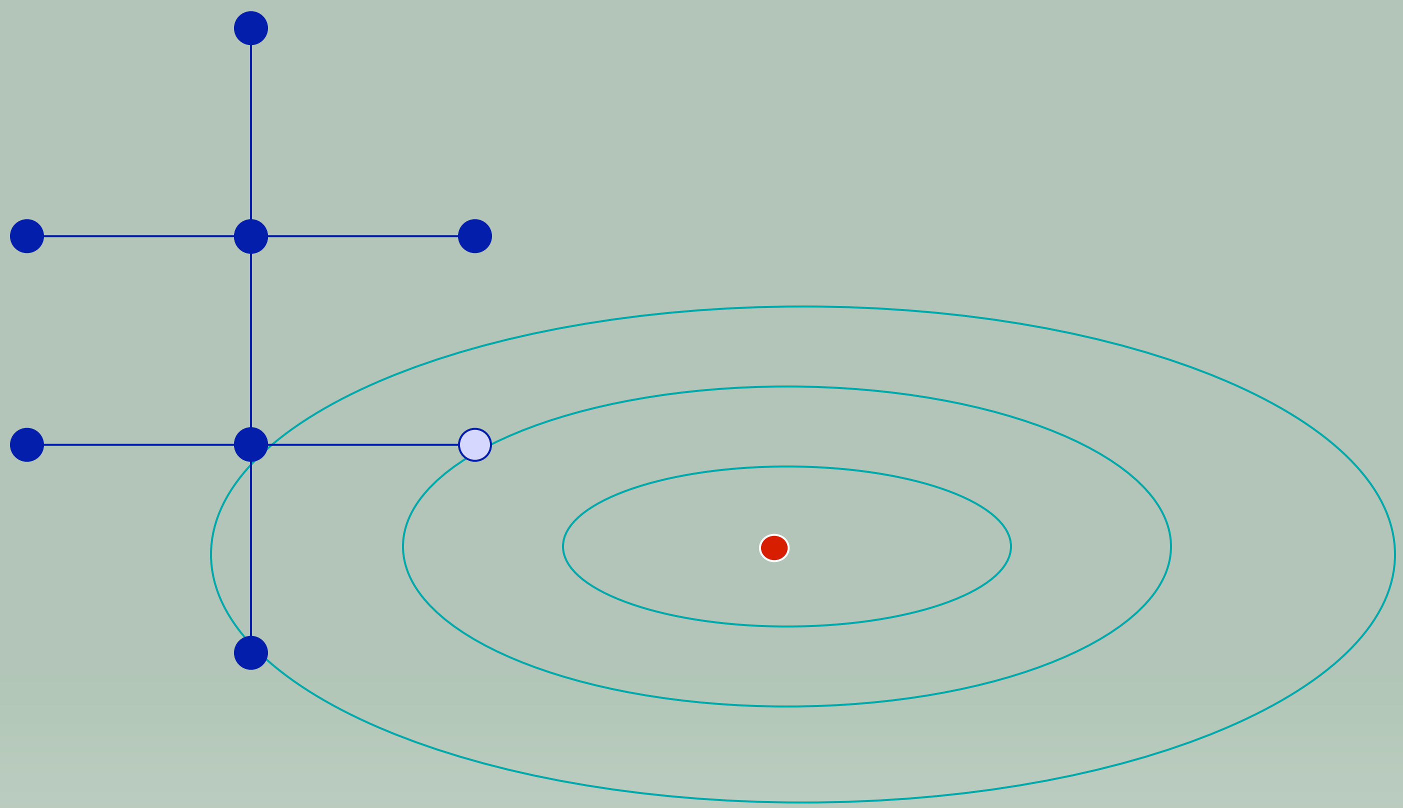
Surprisingly good, but essentially a heuristic

Direct Search methods (early 1990s)



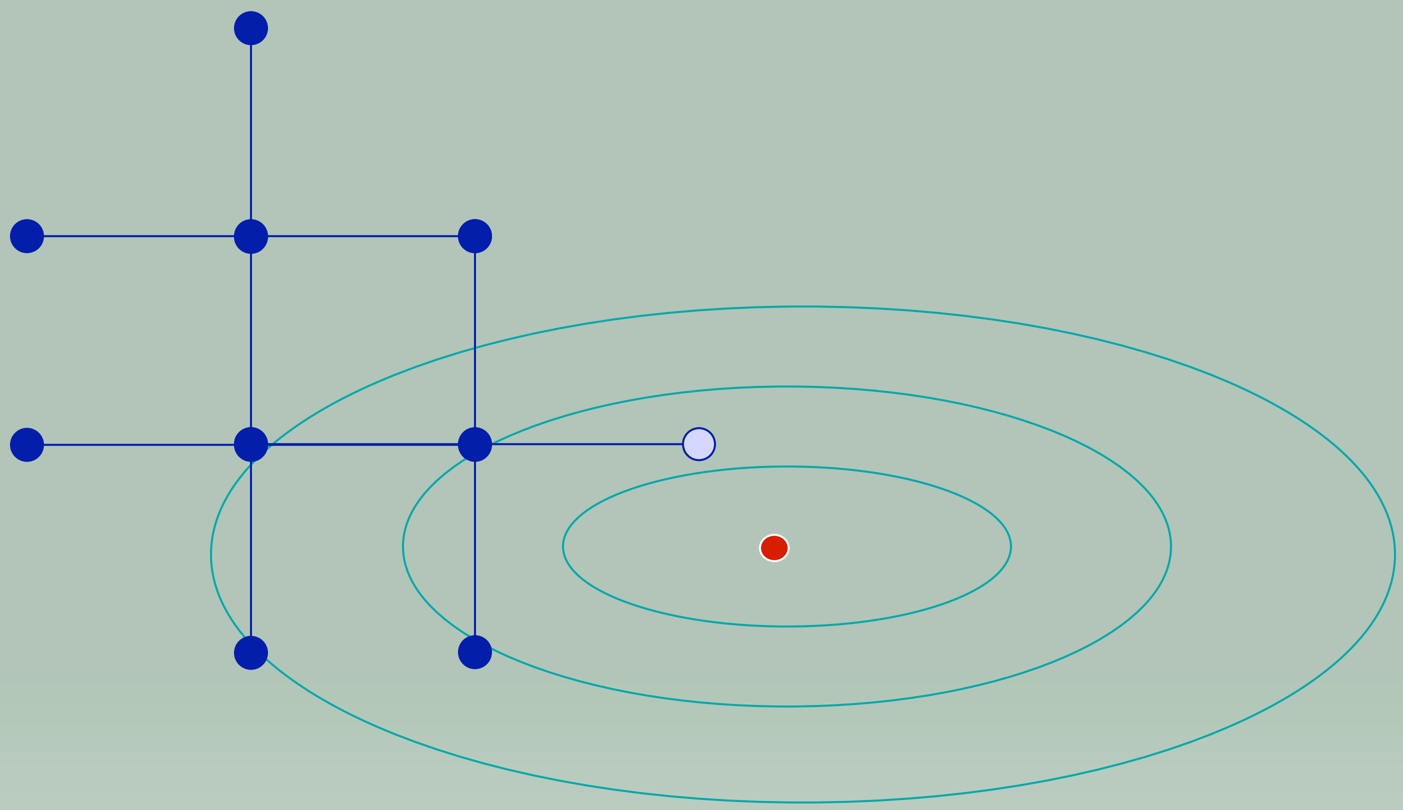
Torczon, Dennis, Audet, Vicente, Luizzi, many others

Direct Search methods



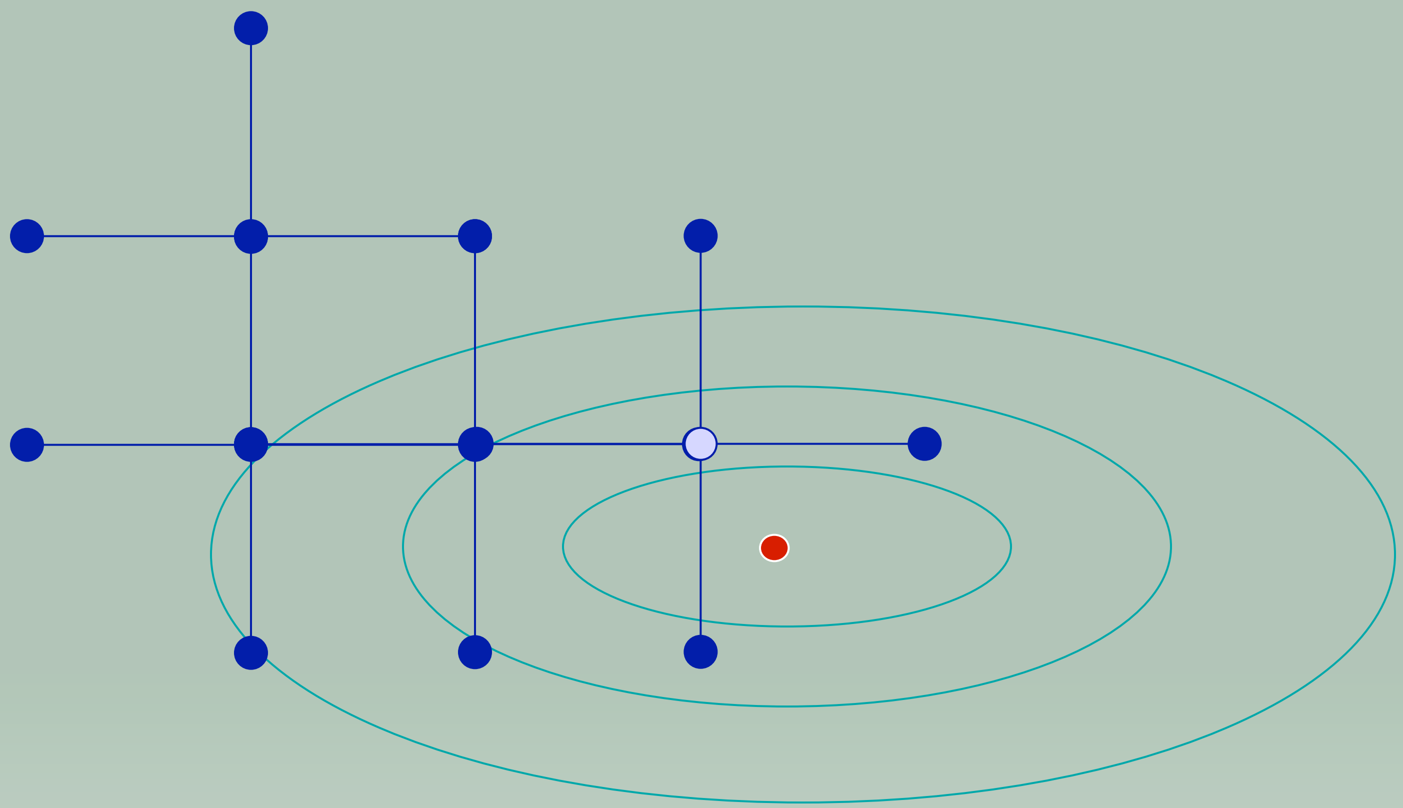
Torczon, Dennis, Audet, Vicente, Luizzi, many others

Direct Search methods



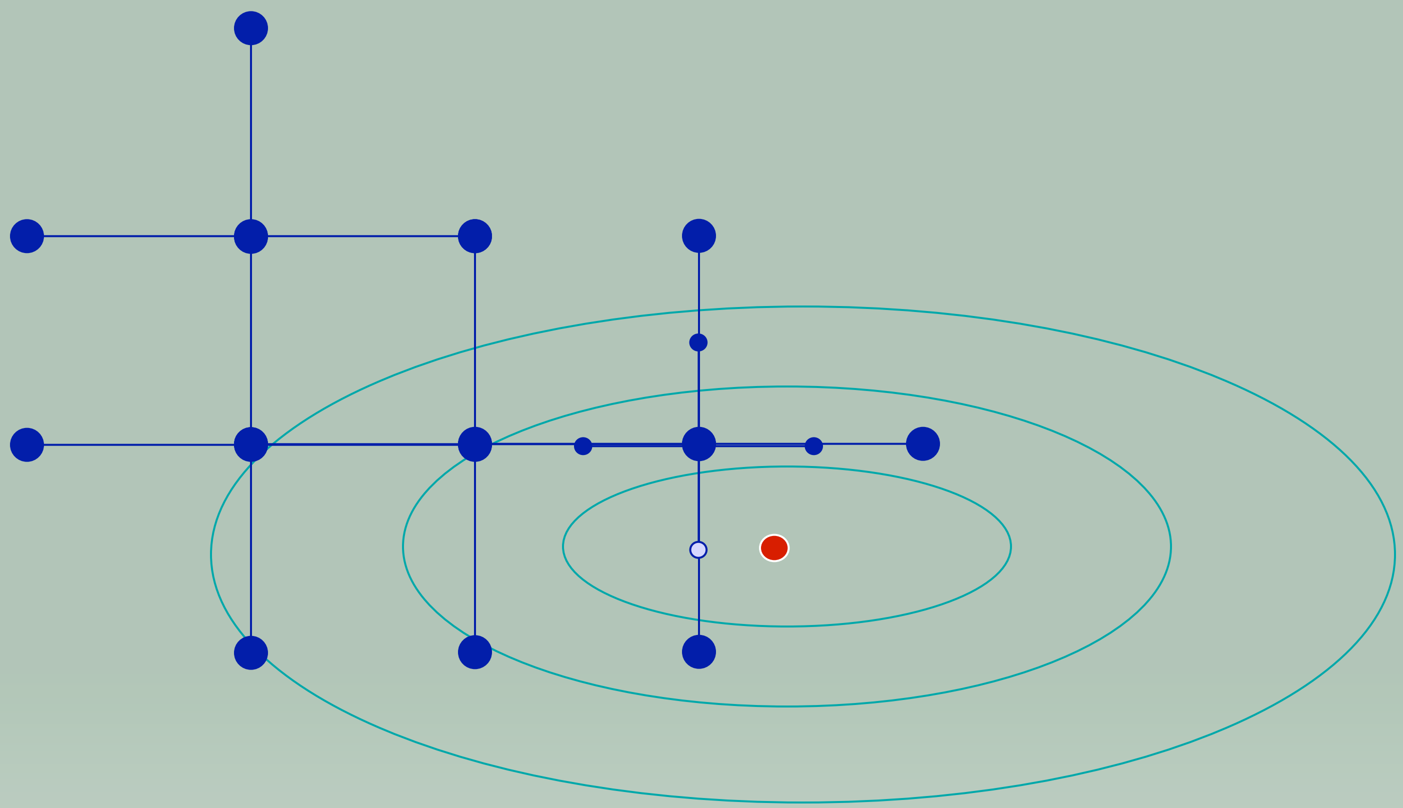
Torczon, Dennis, Audet, Vicente, Luizzi, many others

Direct Search method



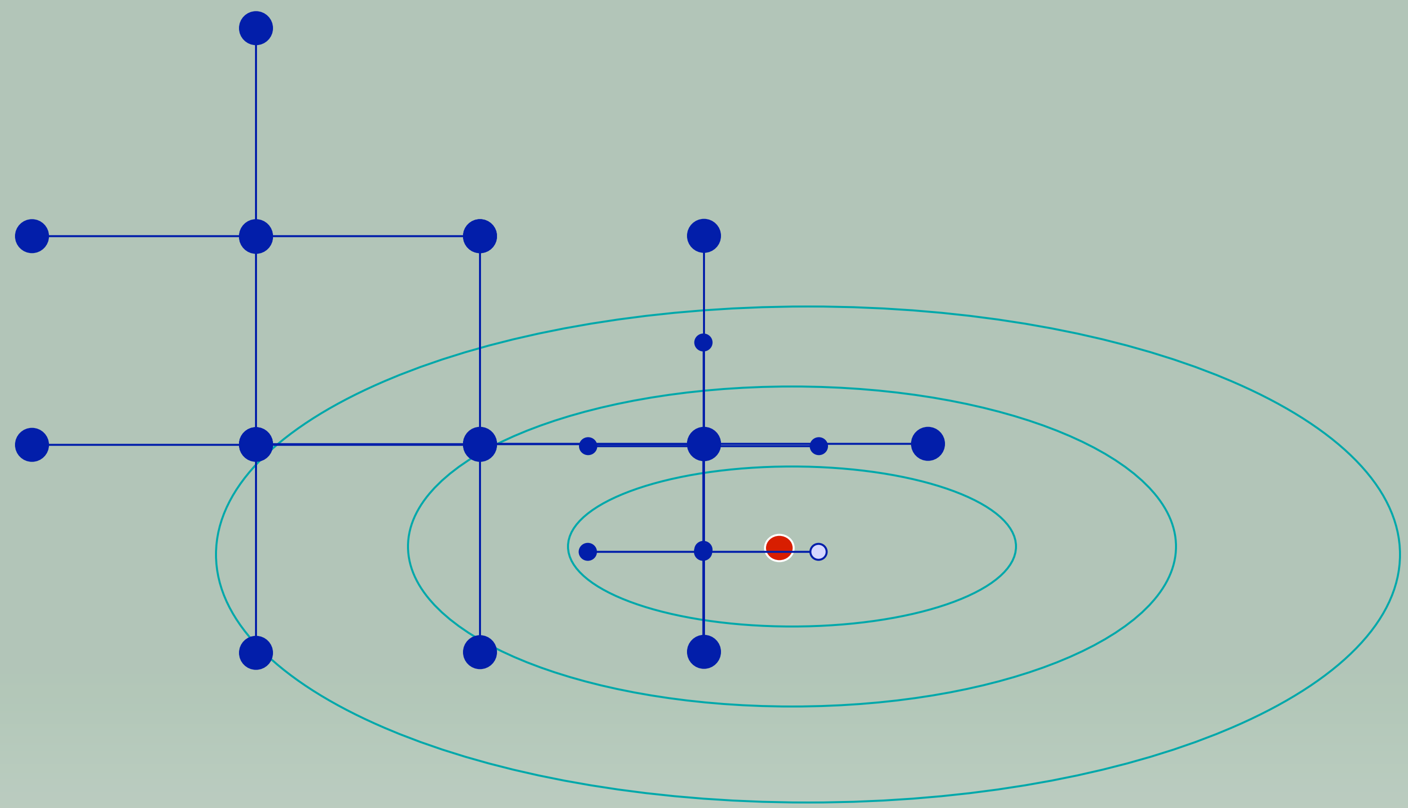
Torczon, Dennis, Audet, Vicente, Luizzi, many others

Direct Search method



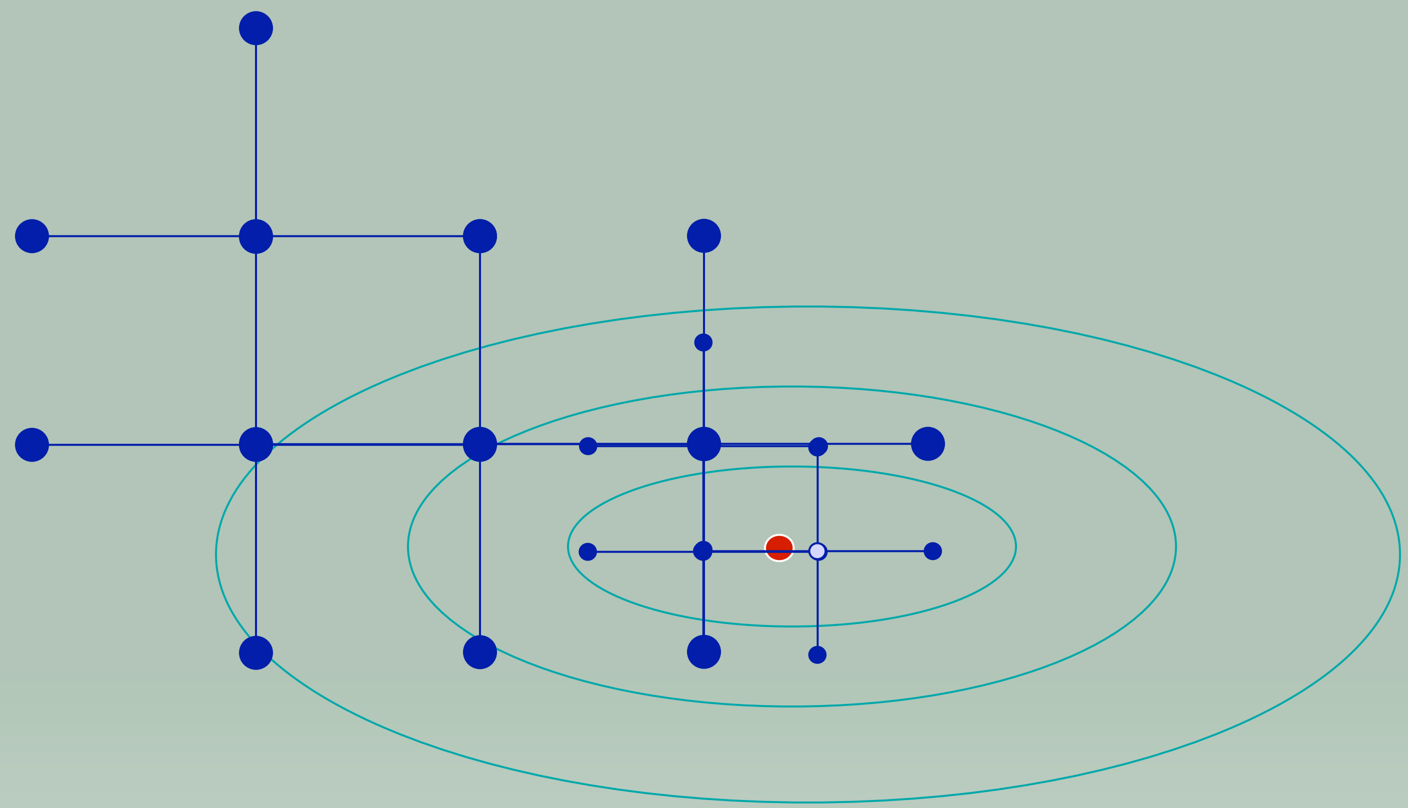
Torczon, Dennis, Audet, Vicente, Luizzi, many others

Direct Search method



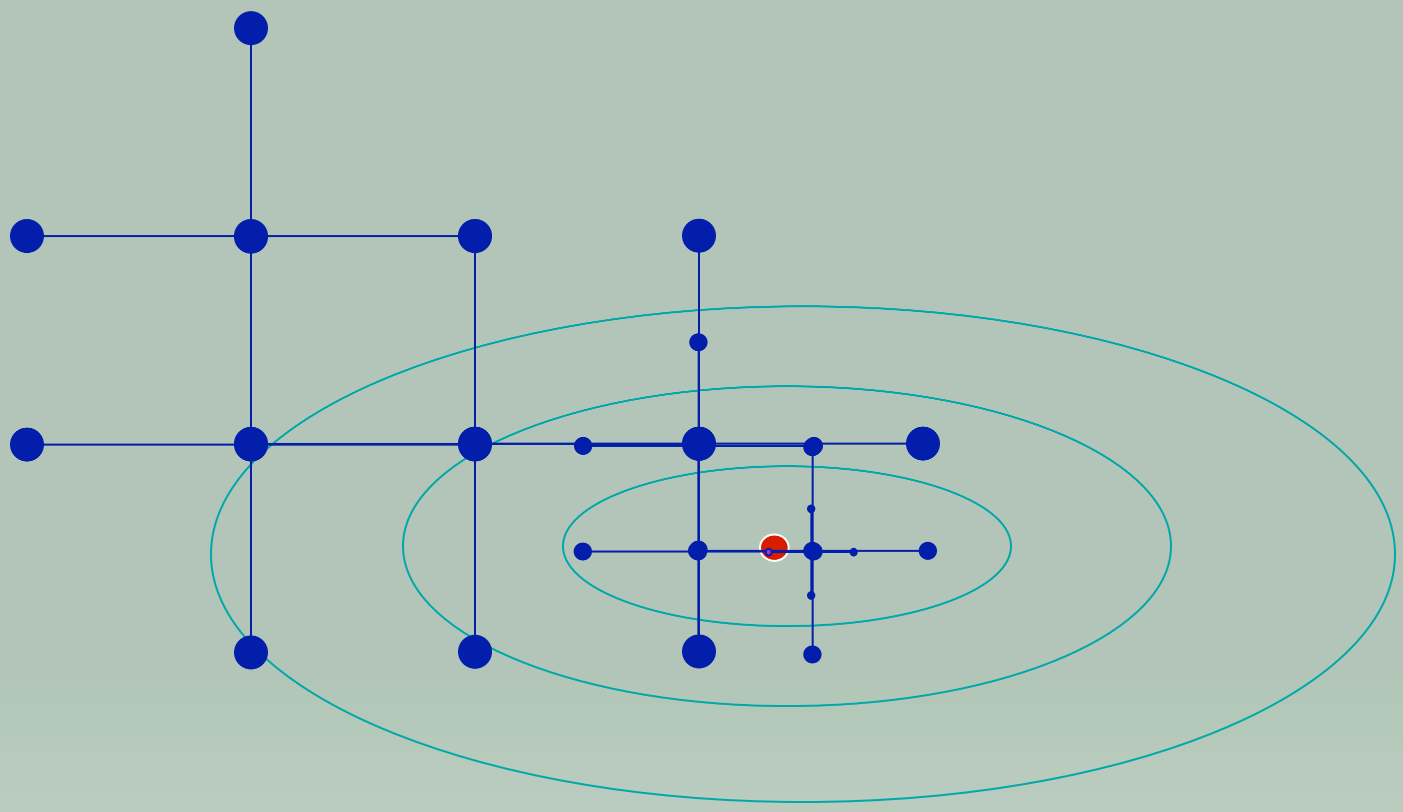
Torczon, Dennis, Audet, Vicente, Luizzi, many others

Direct Search method



Torczon, Dennis, Audet, Vicente, Luizzi, many others

Direct Search method



Fixed pattern, never deteriorates: theoretically convergent, but slow

Compass Search on Rosenbrock

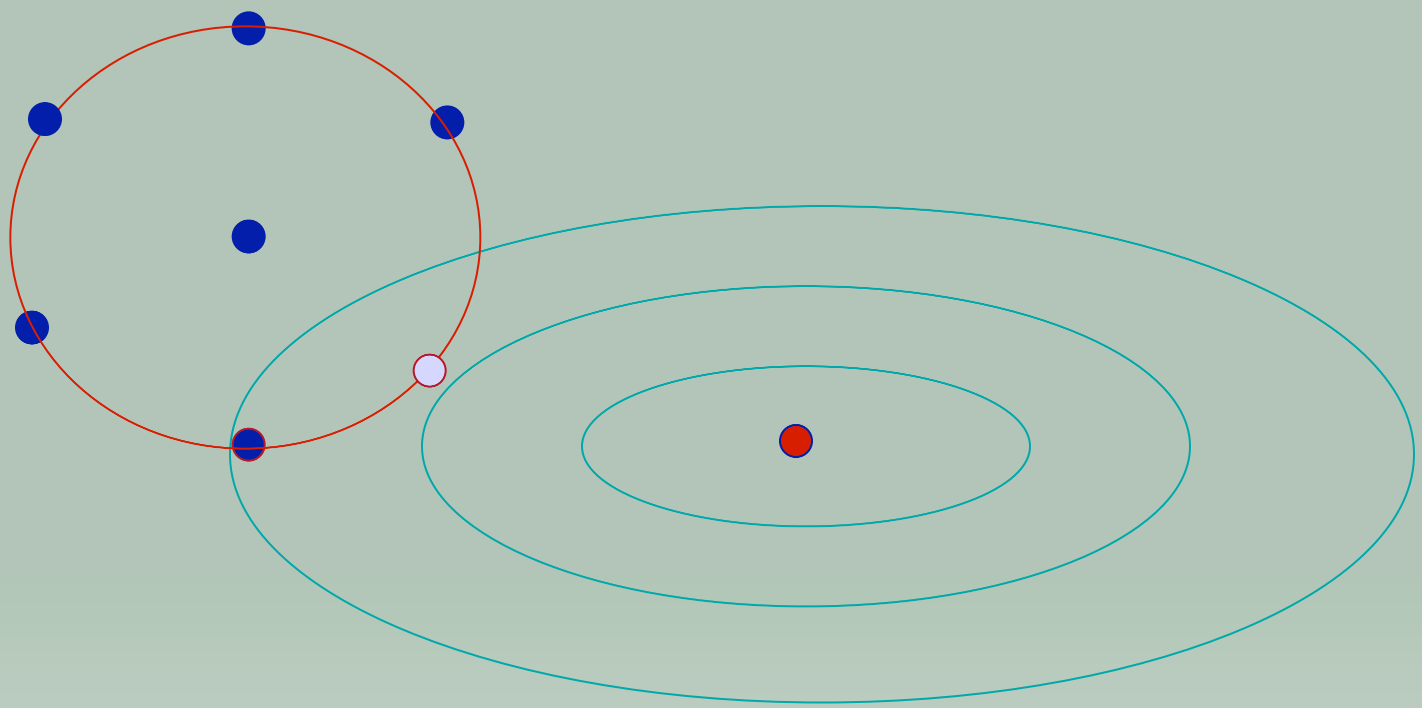
Very slow because of badly aligned axis directions

Random directions on Rosenbrock

Polyak, Yuditski, Nesterov, Lan, Nemirovski, Audet & Dennis, etc

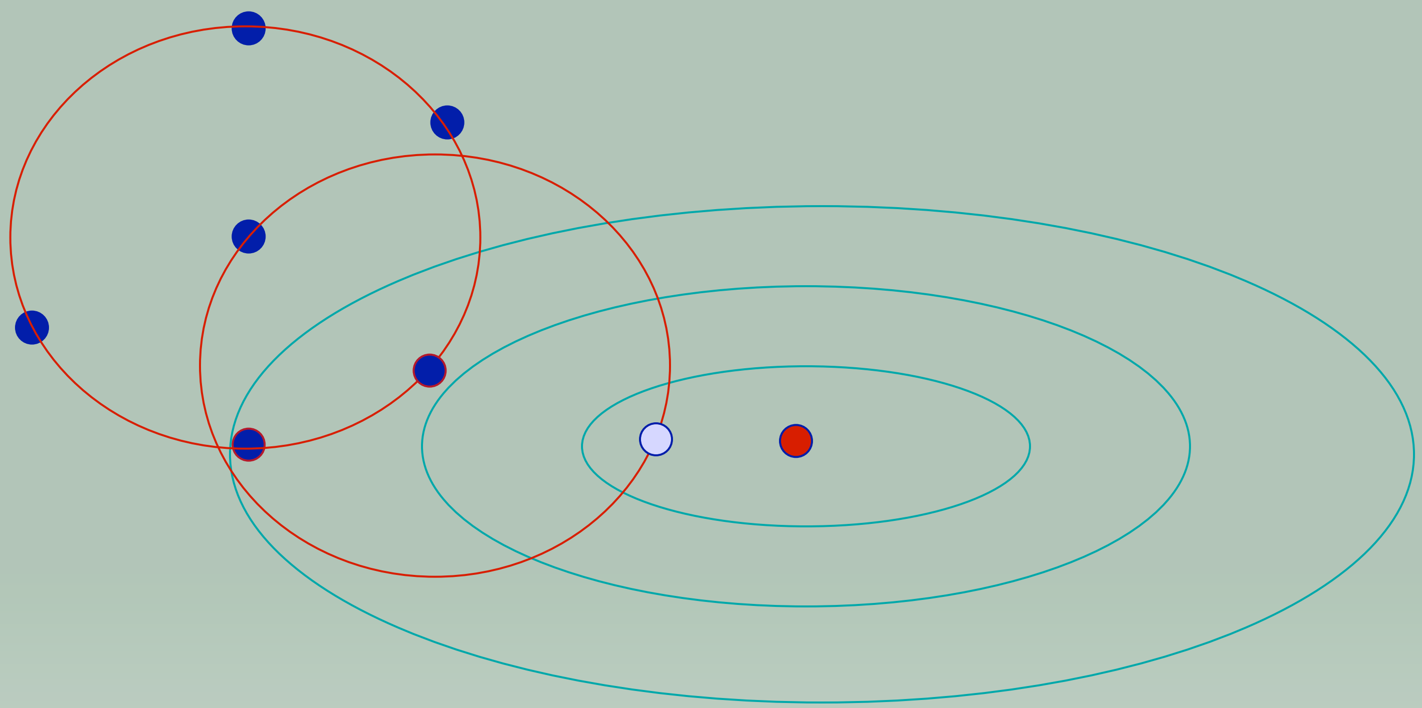
Better progress, but very sensitive to step size choices

Model based trust region methods



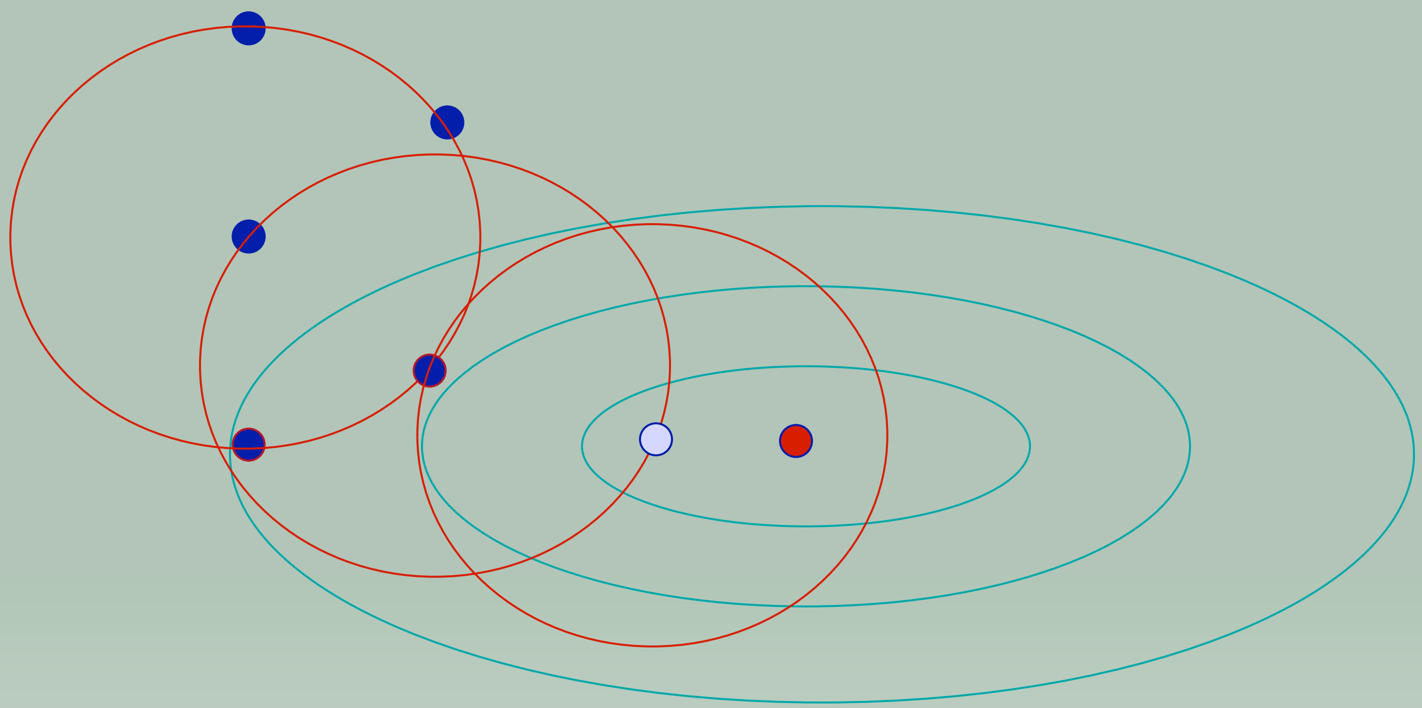
Powell, Conn, S. Toint,
Vicente, Wild, etc.

Model based trust region methods



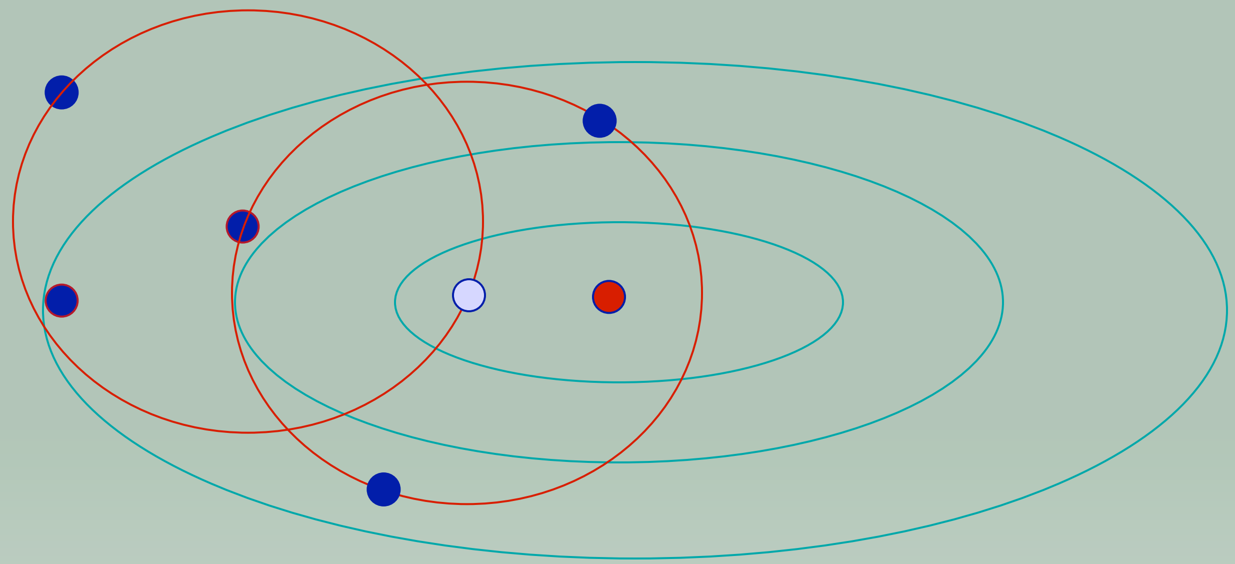
Powell, Conn, S. Toint,
Vicente, Wild, etc.

Model based trust region methods



Powell, Conn, S. Toint,
Vicente, Wild, etc.

Model Based trust region methods



Exploits curvature, flexible efficient steps, uses second order models.

Second order model based TR method on Rosenbrock

Moral:

- Building and using models is a good idea.
- Randomness may offer speed up.
- Can we combine randomization and models successfully and what would we gain?

Polynomial models

Linear Interpolation

Any linear polynomial $m(x)$ can be expressed as

$$m(x) \;=\; \alpha_0 + \sum_{k=1}^{n} \alpha_k x_k$$

Given an interpolation set $Y = \{y^0, \ldots, y^n\}$ the interpolation conditions are

$$m(y^i) \;=\; \alpha_0 + \sum_{k=1}^{n} \alpha_k y_k^i = f(y^i) \quad \forall i = 0, \ldots, n.$$

We have a system of linear equations

$$M(Y)\alpha = f(Y) \qquad M(Y) = \begin{bmatrix} 1 & y_1^0 & y_2^0 & \cdots & y_n^0 \\ 1 & y_1^1 & y_2^1 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^n & y_2^n & \cdots & y_n^n \end{bmatrix}$$

Good vs. bad linear Interpolation

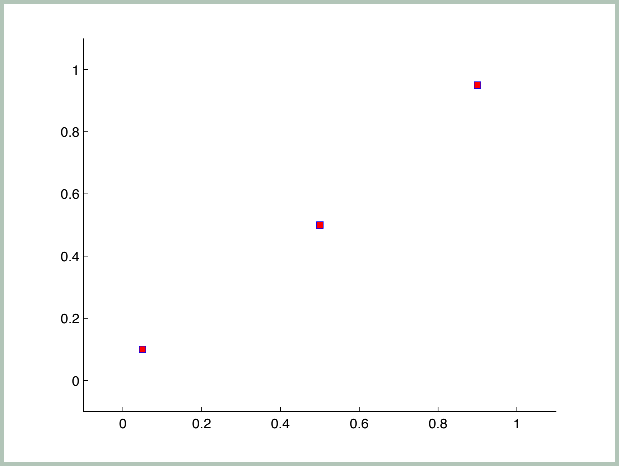
If $M(Y) = \begin{bmatrix} 1 & y_1^0 & y_2^0 & \cdots & y_n^0 \\ 1 & y_1^1 & y_2^1 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^n & y_2^n & \cdots & y_n^n \end{bmatrix}$ is nonsingular

then linear model exists for any *f(x)*

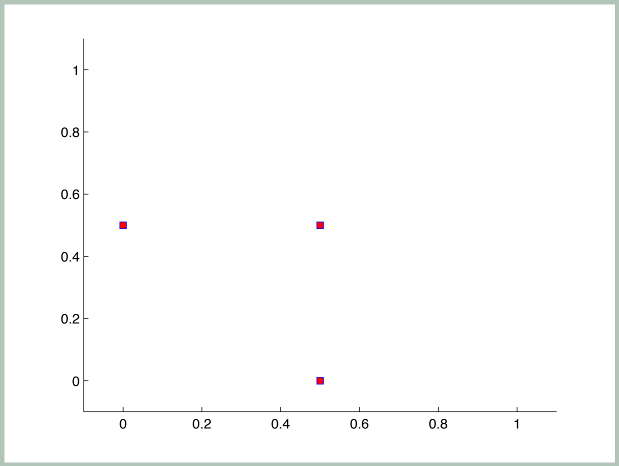
Better conditioned M => better models

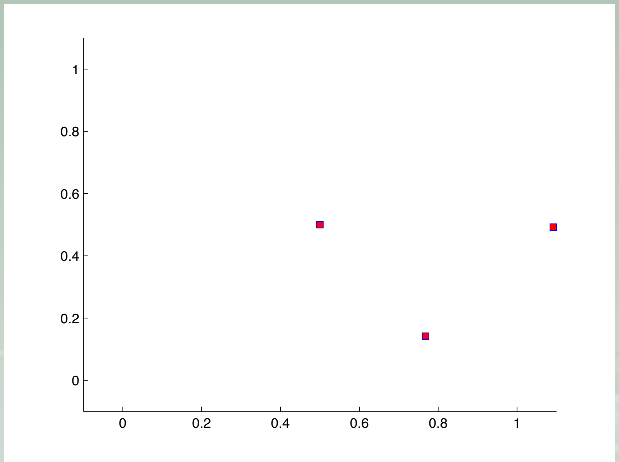
Examples of sample sets for linear interpolation

Badly poised set



Finite difference sample set





Random sample set

Polynomial Interpolation

Given a polynomial basis $\phi = (\phi_1(x), \ldots, \phi_q(x))$ any polynomial $m(x)$ is expressed as

$$m(x) = \sum_{k=1}^{q} \alpha_k \phi_k(x)$$

Given an interpolation set $Y = \{y^1, \ldots, y^p\}$ the interpolation conditions are

$$m(y^i) = \sum_{k=1}^{q} \alpha_k \phi_k(y^i) = f(y^i) \quad \forall i = 1, \ldots, p.$$

The coefficient matrix of the system is:

$$M(\phi, Y) = \begin{bmatrix} \phi_1(y^1) & \phi_2(y^1) & \cdots & \phi_q(y^1) \\ \phi_1(y^2) & \phi_2(y^2) & \cdots & \phi_q(y^2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(y^p) & \phi_2(y^p) & \cdots & \phi_q(y^p) \end{bmatrix} \qquad (p = q).$$

Specifically for quadratic interpolation

Specifically for $\bar{\phi} = \{1, x_1, \cdots, x_n, \frac{1}{2}x_1^2, x_1 x_2, \cdots, \frac{1}{2}x_n^2\}$

$$M(\bar{\phi}, Y) = M = \begin{bmatrix} 1 & y_1^1 & \cdots & y_n^1 & \frac{1}{2}(y_1^1)^2 & y_1^1 y_2^1 & \cdots & \frac{1}{2}(y_n^1)^2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & y_1^p & \cdots & y_n^p & \frac{1}{2}(y_1^p)^2 & y_1^p y_2^p & \cdots & \frac{1}{2}(y_n^p)^2 \end{bmatrix}$$

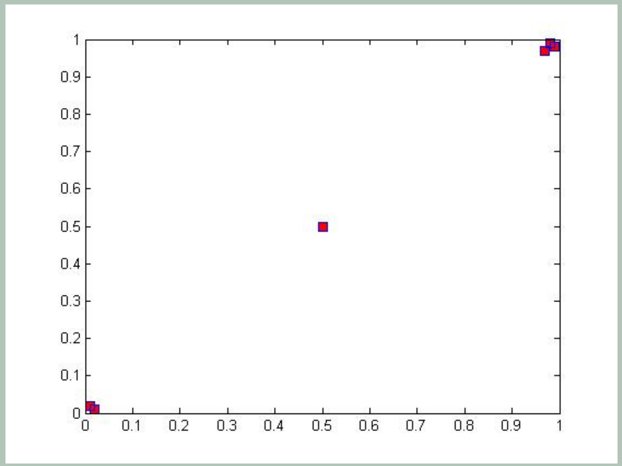
Interpolation model:

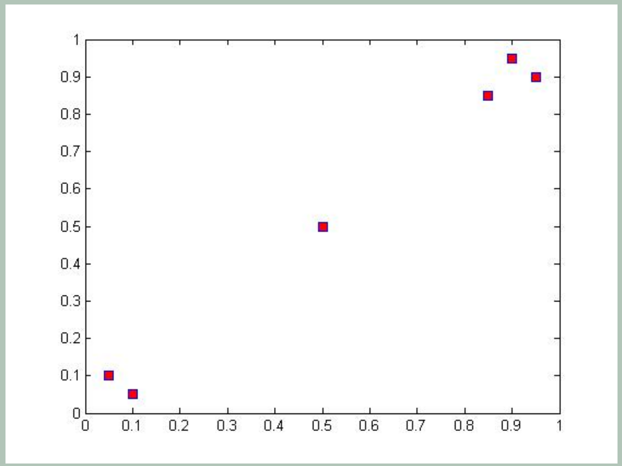
find $\alpha : \ M\alpha = f(Y)$

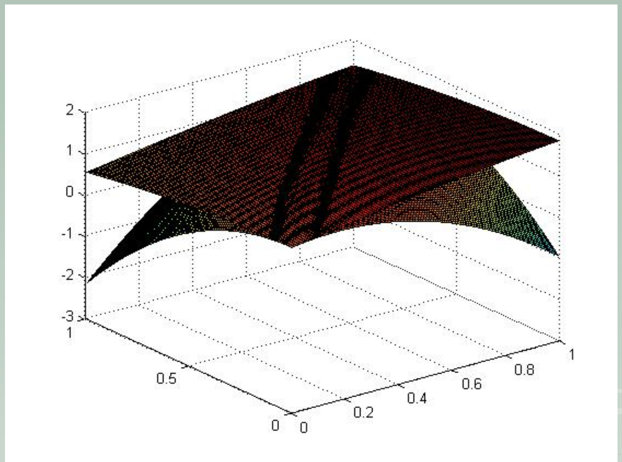
$$m(x) = \sum_{i=1}^{q} \alpha_i \bar{\phi}_i(x) = \frac{1}{2} x^\top H x + g^\top x + \kappa$$

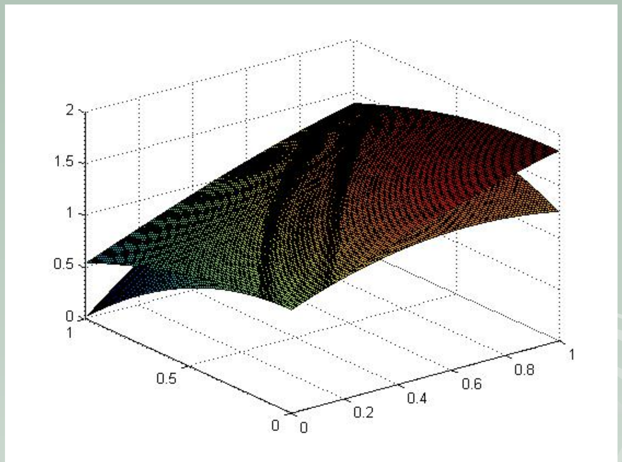
- $\kappa = \alpha_1$
- $g = (\alpha_2, \ldots, \alpha_{n+1})$
- $H_{ij} = \alpha_{n+(i-1)*n+j+1}$

Sample sets and models for f(x)=cos(x)+sin(y)

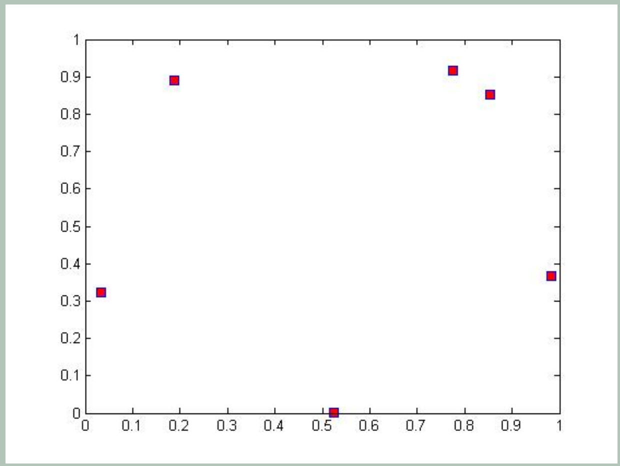


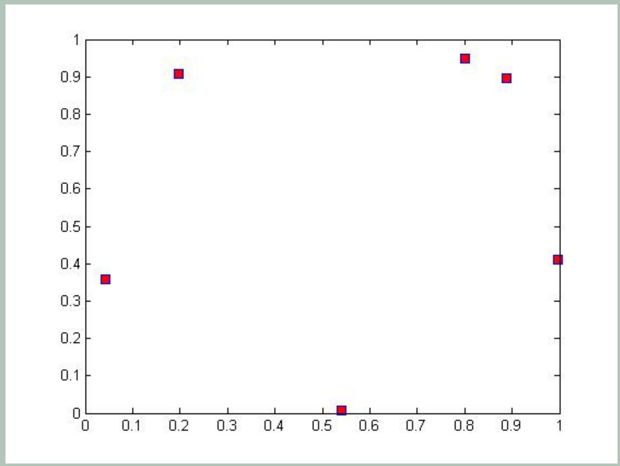


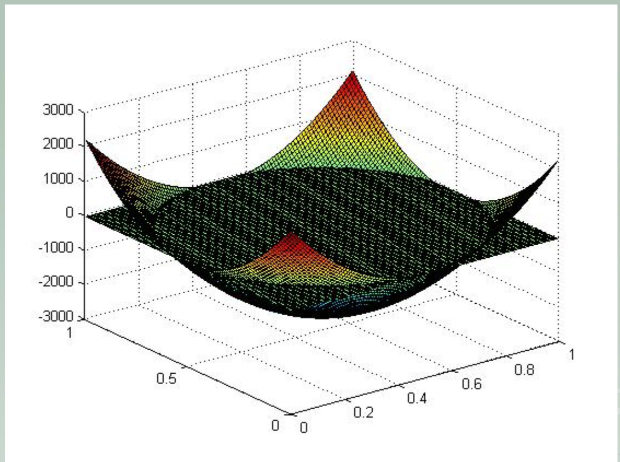


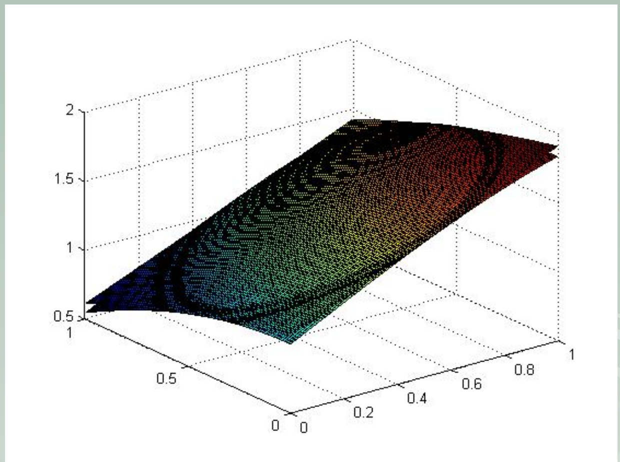


Sample sets and models for f(x)=cos(x)+sin(y)

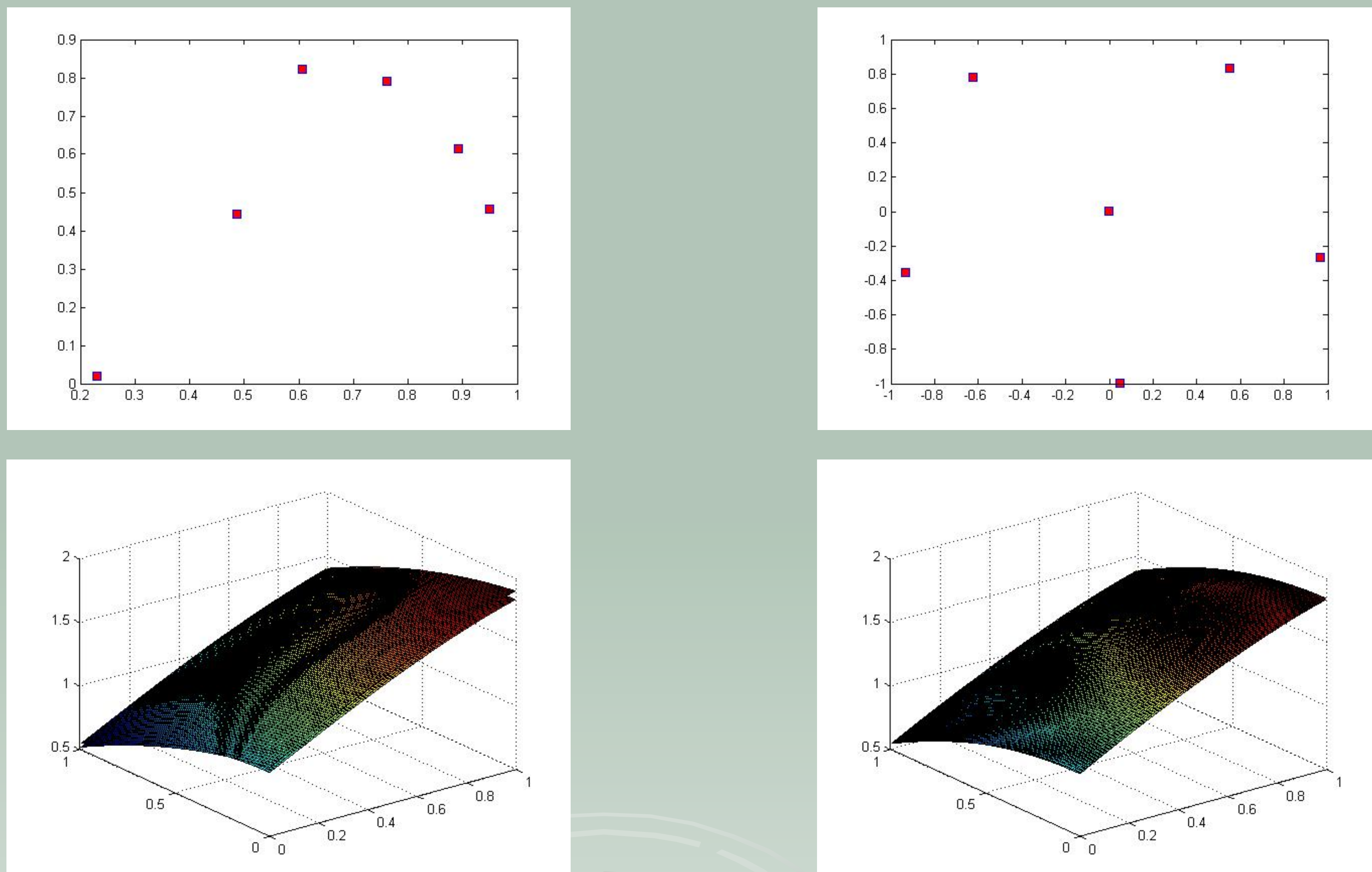






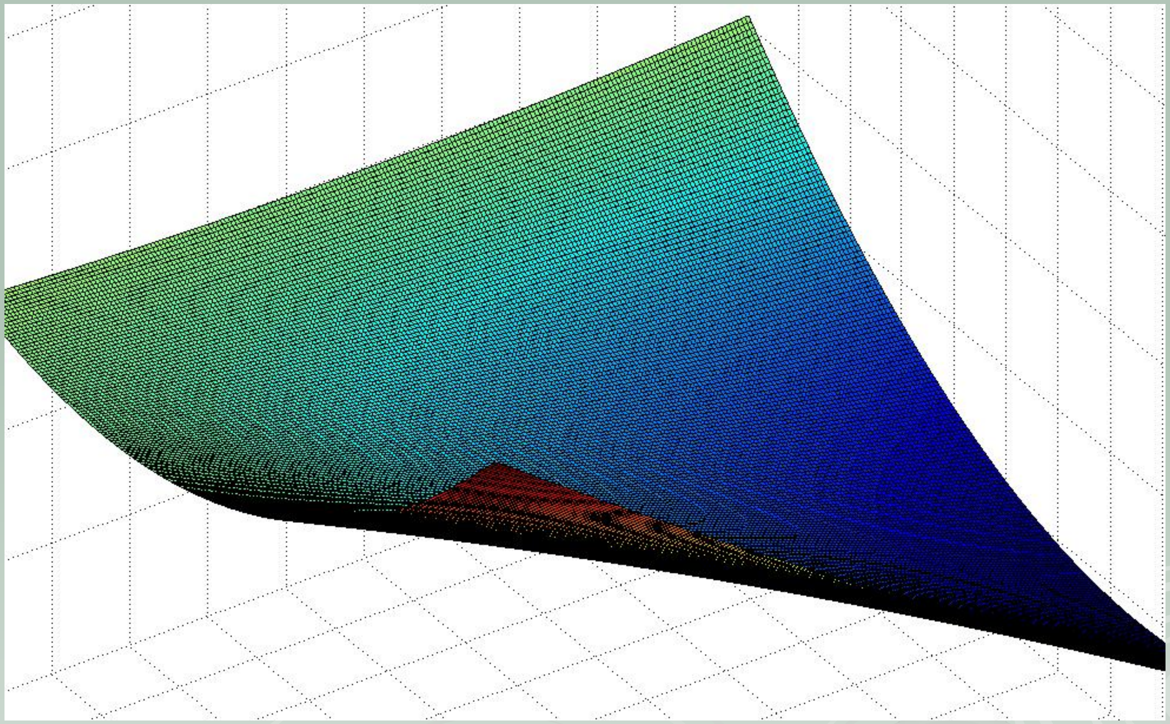


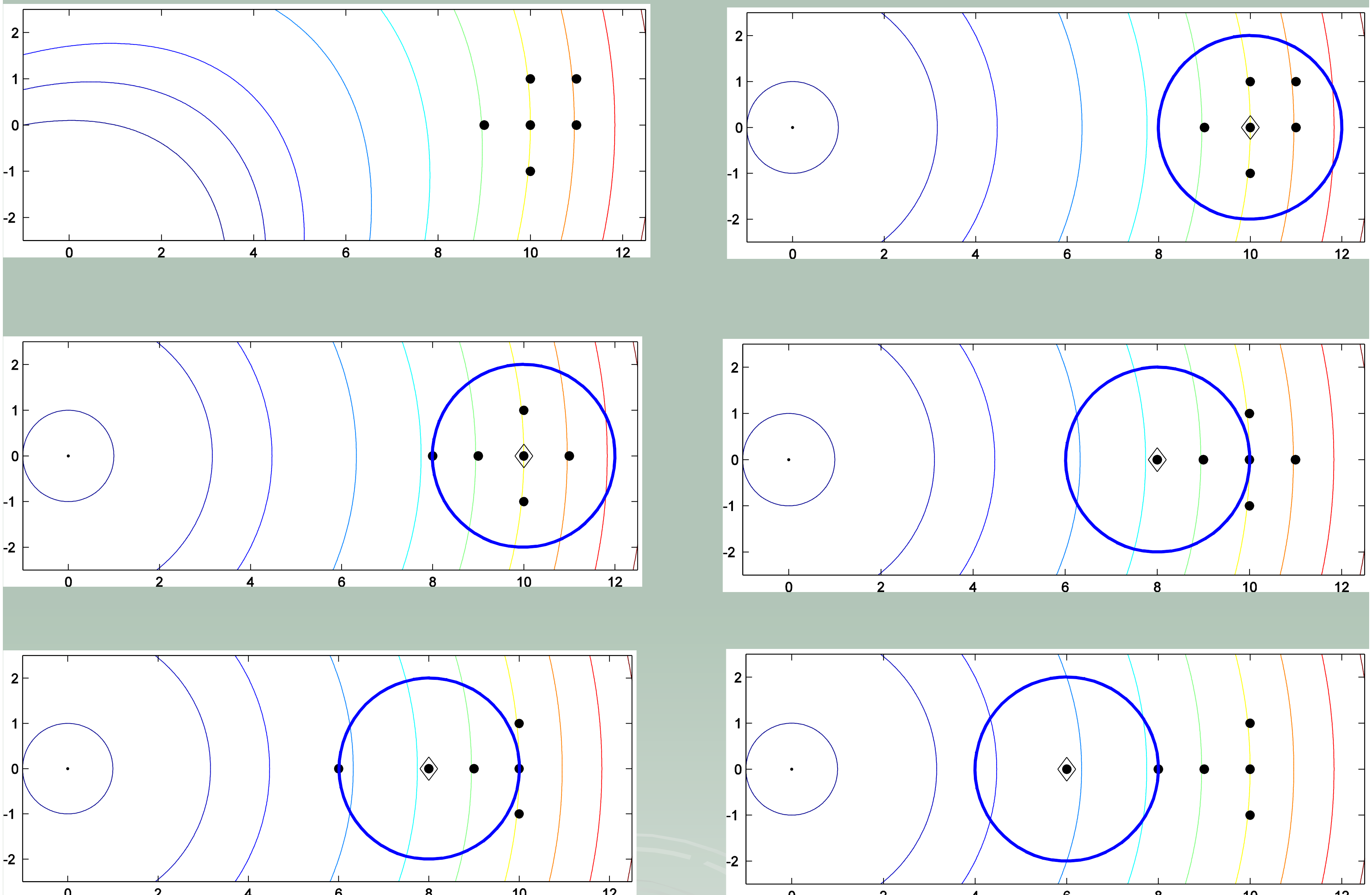
Sample sets and models for f(x)=cos(x)+sin(y)

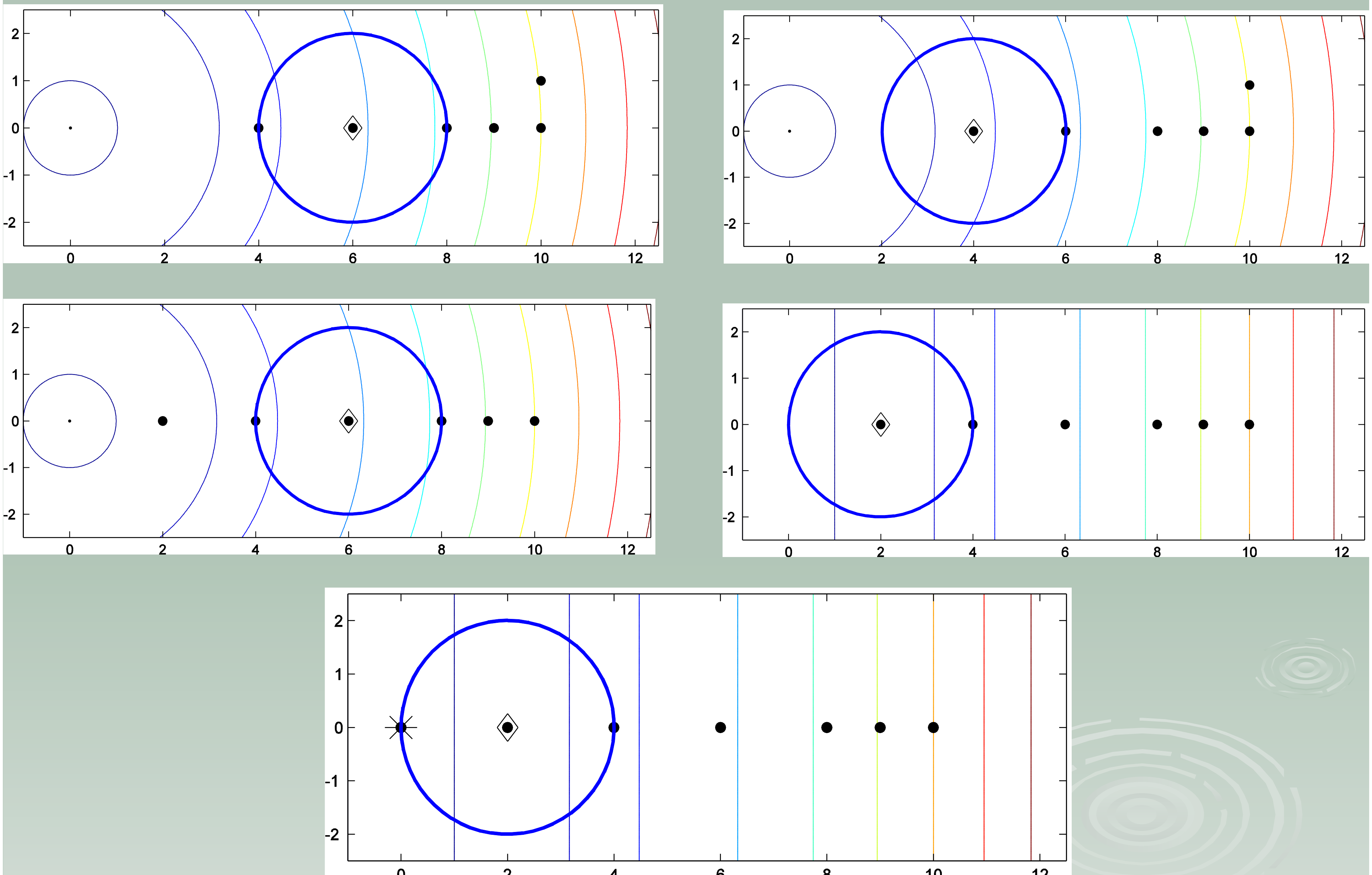


Example that shows that we need to maintain the quality of the sample set

$$f(x) = \begin{cases} x_1^2 + \alpha(x_2^2 + (10 - x_1)x_2) & \text{if } x_1 < 10; \\ x_1^2 + \alpha x_2^2 & \text{if } x_1 \geq 10, \end{cases}$$







2
1
0
-1
-2
0
2
4
6
8
10
12

Observations:

- Building and maintaining good models is needed.
- But it requires computational and implementation effort and many function evaluations.
- Random sample sets usually produce good models, the only effort required is computing the function values.
- This can be done in parallel and random sample sets can produce good models with fewer points.

How?

“sparse” black box optimization

$x=(x_1, x_2, x_3, \ldots, x_n)$

$v=f(x_S)$
$S \subset \{1..n\}$

v

Sparse linear Interpolation

Given an interpolation set $Y = \{y^0, \ldots, y^p\}$ find

$$m(x) \;=\; \alpha_0 + \sum_{k=1}^{n} \alpha_k x_k$$

with sparse coefficient vector α such that

$$m(y^i) \;=\; \alpha_0 + \sum_{k=1}^{n} \alpha_k y_k^i = f(y^i) \quad \forall i = 0, \ldots, p$$

Sparse linear Interpolation

We have an (underdetermined) system of linear equations with a sparse solution

$$M(Y)\alpha = f(Y) \qquad M(Y) = \begin{bmatrix} 1 & y_1^0 & y_2^0 & \cdots & y_n^0 \\ 1 & y_1^1 & y_2^1 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^p & y_2^p & \cdots & y_n^p \end{bmatrix}$$

Can we find correct sparse α using less than n+1 sample points in Y?

Using celebrated compressed sensing results (Candes&Tao, Donoho, etc)

By solving $\min \|\alpha\|_1 : M(Y)\alpha = f(Y)$

Whenever $M(Y) = \begin{bmatrix} 1 & y_1^0 & y_2^0 & \cdots & y_n^0 \\ 1 & y_1^1 & y_2^1 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^p & y_2^p & \cdots & y_n^p \end{bmatrix}$ has **RIP**

Using celebrated compressed sensing results and random matrix theory

(Candes&Tao, Donoho, Rauhut, etc)

Does $$M(Y) = \begin{bmatrix} 1 & y_1^0 & y_2^0 & \cdots & y_n^0 \\ 1 & y_1^1 & y_2^1 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^p & y_2^p & \cdots & y_n^p \end{bmatrix}$$ **have RIP?**

Yes, with high prob., when *Y* is random and *p=O(|S|log n)*

Note: *O(|S|log n)<<n*

Quadratic interpolation models

$$M(\bar{\phi}, Y) = M = \begin{bmatrix} 1 & y_1^1 & \cdots & y_n^1 & \frac{1}{2}(y_1^1)^2 & y_1^1 y_2^1 & \cdots & \frac{1}{2}(y_n^1)^2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & y_1^p & \cdots & y_n^p & \frac{1}{2}(y_1^p)^2 & y_1^p y_2^p & \cdots & \frac{1}{2}(y_n^p)^2 \end{bmatrix}$$

Need p=(n+1)(n+2)/2 sample points!!!

Interpolation model:

$$\text{find } \alpha : \;\; M\alpha = f(Y)$$

$$m(x) = \sum_{i=1}^{q} \alpha_i \bar{\phi}_i(x) = \frac{1}{2} x^\top H x + g^\top x + \kappa$$

- $\kappa = \alpha_1$
- $g = (\alpha_2, \ldots, \alpha_{n+1})$
- $H_{ij} = \alpha_{n+(i-1)*n+j+1}$

Example of a model with sparse Hessian

Colson, Toint

$$\min f(x) = \sum_{i}^{n} ((x_i^2 - x_n^2)^2 - 4x_i)$$

$$\nabla^2_{ij} f(x) = 0, \ \forall i \neq j, j \neq n$$

α has only 2n+n nonzeros

Can we recover the sparse α using less than O(n) points?

Sparse quadratic interpolation models

$$M(\bar{\phi}, Y) = M = \begin{array}{c} \\ \end{array} \left[\underbrace{\begin{array}{cccc} 1 & y_1^1 & \cdots & y_n^1 \\ \vdots & \vdots & & \vdots \\ 1 & y_1^p & \cdots & y_n^p \end{array}}_{M_L} \;\; \underbrace{\begin{array}{cccc} \frac{1}{2}(y_1^1)^2 & y_1^1 y_2^1 & \cdots & \frac{1}{2}(y_n^1)^2 \\ \vdots & \vdots & & \vdots \\ \frac{1}{2}(y_1^p)^2 & y_1^p y_2^p & \cdots & \frac{1}{2}(y_n^p)^2 \end{array}}_{M_Q} \right]$$

Recover sparse α

$$\min_{\alpha} \quad \|\alpha_Q\|_1$$

$$\text{s.t.} \quad M_L \alpha_L + M_Q \alpha_Q = f(Y)$$

$$m(x) = \tfrac{1}{2} x^\top H x + g^\top x + \kappa$$

- $\alpha_L \to (k, g)$
- $\alpha_Q \to H$

Does RIP hold for this matrix?

$$M(\bar{\phi}, Y) = M = \left[\begin{array}{cccc|cccc} 1 & y_1^1 & \cdots & y_n^1 & \frac{1}{2}(y_1^1)^2 & y_1^1 y_2^1 & \cdots & \frac{1}{2}(y_n^1)^2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & y_1^p & \cdots & y_n^p & \frac{1}{2}(y_1^p)^2 & y_1^p y_2^p & \cdots & \frac{1}{2}(y_n^p)^2 \end{array} \right]$$

$$\underbrace{}_{M_L} \quad \underbrace{\phantom{\tfrac{1}{2}(y_1)^2 \; y_1 y_2 \cdots \tfrac{1}{2}(y_n)^2}}_{M_Q}$$

Does RIP hold for this matrix?

$$M(\bar{\phi}, Y) = M = \left[\begin{array}{cccc|cccc} \multicolumn{4}{c}{\overbrace{\hphantom{1 \quad y_1^1 \quad \cdots \quad y_n^1}}^{\boldsymbol{M_L}}} & \multicolumn{4}{c}{\overbrace{\hphantom{\frac{1}{2}(y_1^1)^2 \quad y_1^1 y_2^1 \quad \cdots \quad \frac{1}{2}(y_n^1)^2}}^{\boldsymbol{M_Q}}} \\ 1 & y_1^1 & \cdots & y_n^1 & \frac{1}{2}(y_1^1)^2 & y_1^1 y_2^1 & \cdots & \frac{1}{2}(y_n^1)^2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & y_1^p & \cdots & y_n^p & \frac{1}{2}(y_1^p)^2 & y_1^p y_2^p & \cdots & \frac{1}{2}(y_n^p)^2 \end{array} \right]$$

Actually we need RIP for M_Q and some other property on M_L

Using results from random matrix theory

(Rauhut, Bandeira, S. & Vincente)

$$M(\bar{\phi}, Y) = M = \left[\begin{array}{cccc|cccc} \multicolumn{4}{c}{\overbrace{\hphantom{1 \quad y_1^1 \quad \cdots \quad y_n^1}}^{M_L}} & \multicolumn{4}{c}{\overbrace{\hphantom{\frac{1}{2}(y_1^1)^2 \quad y_1^1 y_2^1 \quad \cdots \quad \frac{1}{2}(y_n^1)^2}}^{M_Q}} \\ 1 & y_1^1 & \cdots & y_n^1 & \frac{1}{2}(y_1^1)^2 & y_1^1 y_2^1 & \cdots & \frac{1}{2}(y_n^1)^2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & y_1^p & \cdots & y_n^p & \frac{1}{2}(y_1^p)^2 & y_1^p y_2^p & \cdots & \frac{1}{2}(y_n^p)^2 \end{array}\right]$$

Yes, with high probability, when Y is random and ***$p=O((n+s)(\log n)^4)$***

Note: ***$p=O((n+s)(\log n)^4) \ll n^2$*** (sometimes)

For more detailed analysis
see Afonso Bandeira's talk
Tue 15:15 - 16:45, room: H 3503

Model-based method on 2-dimensional Rosenbrock function lifted into 10 dimensional space

Consider $f(x_1, x_2, \ldots, x_{10})$=*Rosenbrock*$(x_1, x_2)$

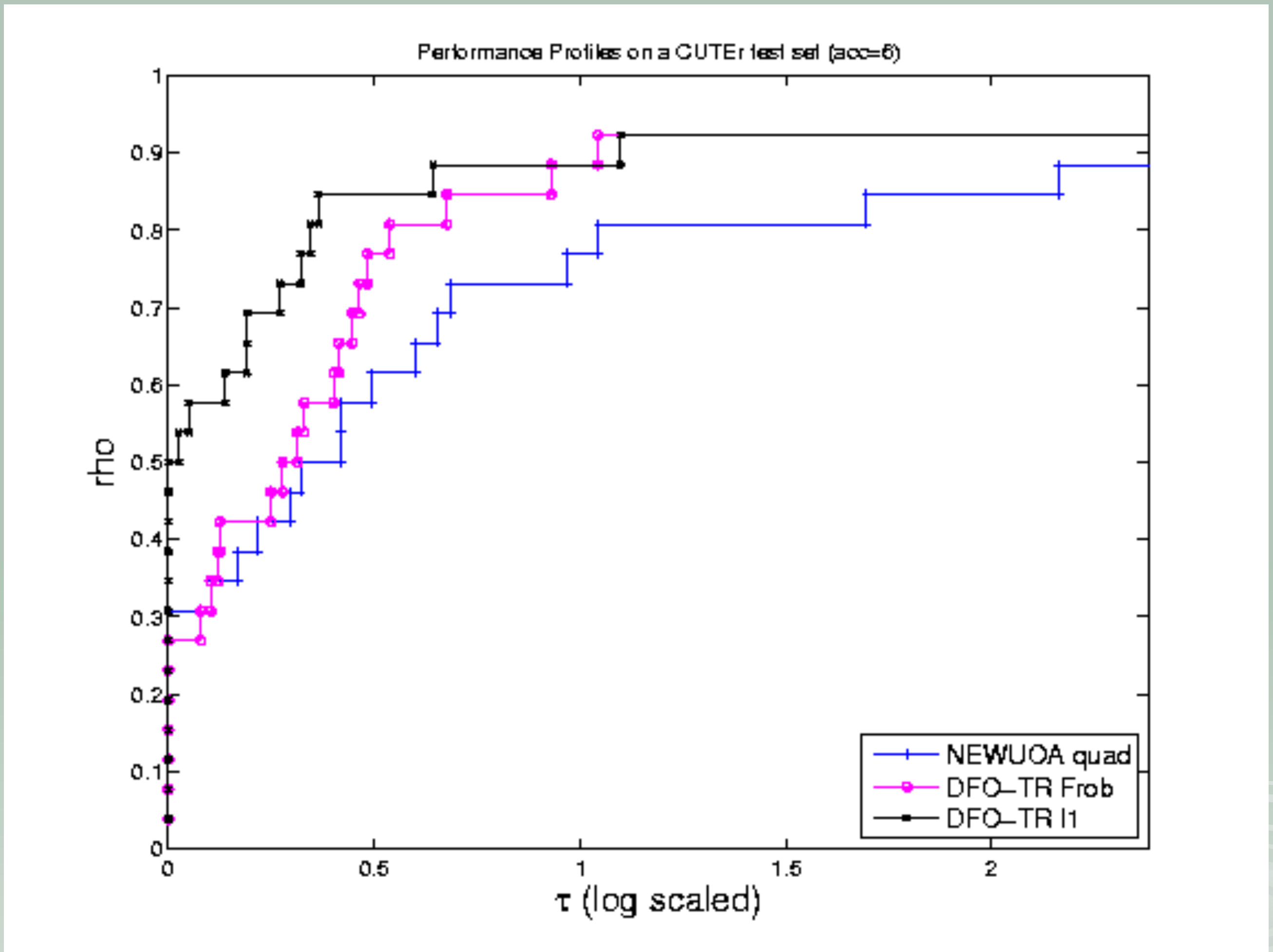
To build full quadratic interpolation we need 66 points. We test two methods:

1. *Deterministic model-based TR method: builds a model using whatever points it has on hand up to 66 in the neighborhood of the current iterate, using MFN Hessian models (standard reliable good approach).*
2. *Random model based TR method: builds sparse models using 31 randomly sampled points.*

Deterministic MFN model based method

Random sparse model based method

Comparison of sparse vs MFN models (no randomness) within TR on CUTER problems



Algorithms based on random models

- We now forget about sample sets and how we build the models.
- We focus on properties of the models that are essential for convergence.
- Ensure that those properties are satisfied by models we just discussed.

What do we need from a deterministic model for convergence?

We need Taylor-like behavior of first-order models

A model is called κ-fully-linear in $B(x,\Delta)$, for $\kappa = (\kappa_{ef}, \kappa_{eg})$ if

$$\|\nabla f(x+s) - \nabla m(x+s)\| \ \le \ \kappa_{eg}\,\Delta, \quad \forall s \in B(0;\Delta),$$

$$|f(x+s) - m(x+s)| \ \le \ \kappa_{ef}\,\Delta^2, \quad \forall s \in B(0;\Delta),$$

What do we need from a model to explore the curvature?

We may want Taylor-like behavior of second-order models

A model is called κ-fully-quadratic in $B(x, \Delta)$ for $\kappa = (\kappa_{ef}, \kappa_{eg}, \kappa_{eh})$ if

$$\|\nabla^2 f(x+s) - \nabla^2 m(x+s)\| \leq \kappa_{eh}\, \Delta, \quad \forall s \in B(0; \Delta),$$

$$\|\nabla f(x+s) - \nabla m(x+s)\| \leq \kappa_{eg}\, \Delta^2, \quad \forall s \in B(0; \Delta),$$

$$|f(x+s) - m(x+s)| \leq \kappa_{ef}\, \Delta^3, \quad \forall s \in B(0; \Delta),$$

What do we need from a random model for convergence?

We need likely Taylor-like behavior of first-order models

A random model is called (κ, δ)-fully-linear in $B(x, \Delta)$ if

$$\|\nabla f(x+s) - \nabla m(x+s)\| \;\leq\; \kappa_{eg}\,\Delta, \quad \forall s \in B(0;\Delta),$$

$$|f(x+s) - m(x+s)| \;\leq\; \kappa_{ef}\,\Delta^2, \quad \forall s \in B(0;\Delta),$$

with probability at least $1-\delta$.

What do we need from a random model to explore curvature?

We need likely Taylor-like behavior of second order models

A random model is called (κ, δ)-fully-quadratic in $B(x, \Delta)$ if

$$\|\nabla^2 f(x+s) - \nabla^2 m(x+s)\| \leq \kappa_{eh}\,\Delta, \quad \forall s \in B(0;\Delta),$$

$$\|\nabla f(x+s) - \nabla m(x+s)\| \leq \kappa_{eg}\,\Delta^2, \quad \forall s \in B(0;\Delta),$$

$$|f(x+s) - m(x+s)| \leq \kappa_{ef}\,\Delta^3, \quad \forall s \in B(0;\Delta),$$

with probability at least $1-\delta$.

What random models have such properties?

- Linear interpolation and regression models based on random sample sets of n+1 points are (κ, δ)-fully-linear.
- Quadratic interpolation and regression models based on random sample sets of (n+1)(n+1)/2 points are (κ, δ)-fully-quadratic.
- Sparse linear interpolation and reg. models based on smaller random sample sets are (κ, δ)-fully-linear.
- Sparse quadratic interpolation and reg. models based on smaller random sample sets are (κ, δ)-fully-quadratic.
- Taylor models based on finite difference derivative evaluations with asynchronous faulty parallel function evaluations are (κ, δ)-FL or FQ.
- Gradient sampling models? Other examples?

Basic Trust Region Algorithm

Model selection

Pick a random model $m_k(x)$ which is κ-fully-linear in $B(x_k, \Delta_k)$ w.p. $1-\delta$.

Compute potential step

Compute a point x^+ which minimizes (reduces) $m(x)$ in $B(x_k, \Delta_k)$.
Compute $f(x^+)$ and check if f is reduced comparably to m by x^+.

Successful step

If yes and if the radius Δ_k is not too big compared to $\nabla m_k(x_k)$ then we take the step and increase Δ_k by a constant factor.

Unsuccessful step

Otherwise, decrease Δ_k by the constant factor and repeat the iteration.

Convergence results for the basic TR framework

If models are fully linear with prob. $1-\delta > 0.5$ then with probability *one* $\lim \|\nabla f(x_k)\| = 0$

If models are fully quadratic w. p. $1-\delta > 0.5$ then with probability *one*
$\liminf \max \{\|\nabla f(x_k)\|, \lambda_{min}(\nabla^2 f(x_k))\} = 0$

For *lim* result δ need to decrease occasionally

For details see Afonso Bandeira's talk on Tue 15:15 - 16:45, room: H 3503

Intuition behind the analysis shown through line search ideas

When *m(x)* is linear ~ line search instead of Δ_k use $\alpha_k \|\nabla m_k(x_k)\|$

Model selection step

Pick a random model $m_k(x) = f(x_k) + g_k^\top(x - x_k)$
κ-fully-linear in $B(x_k, \alpha_k\|g_k\|)$ w.p. $1 - \delta$.

Compute Step

$x^+ = x_k - \alpha_k g_k$. Check if f is sufficiently reduced an x^+.

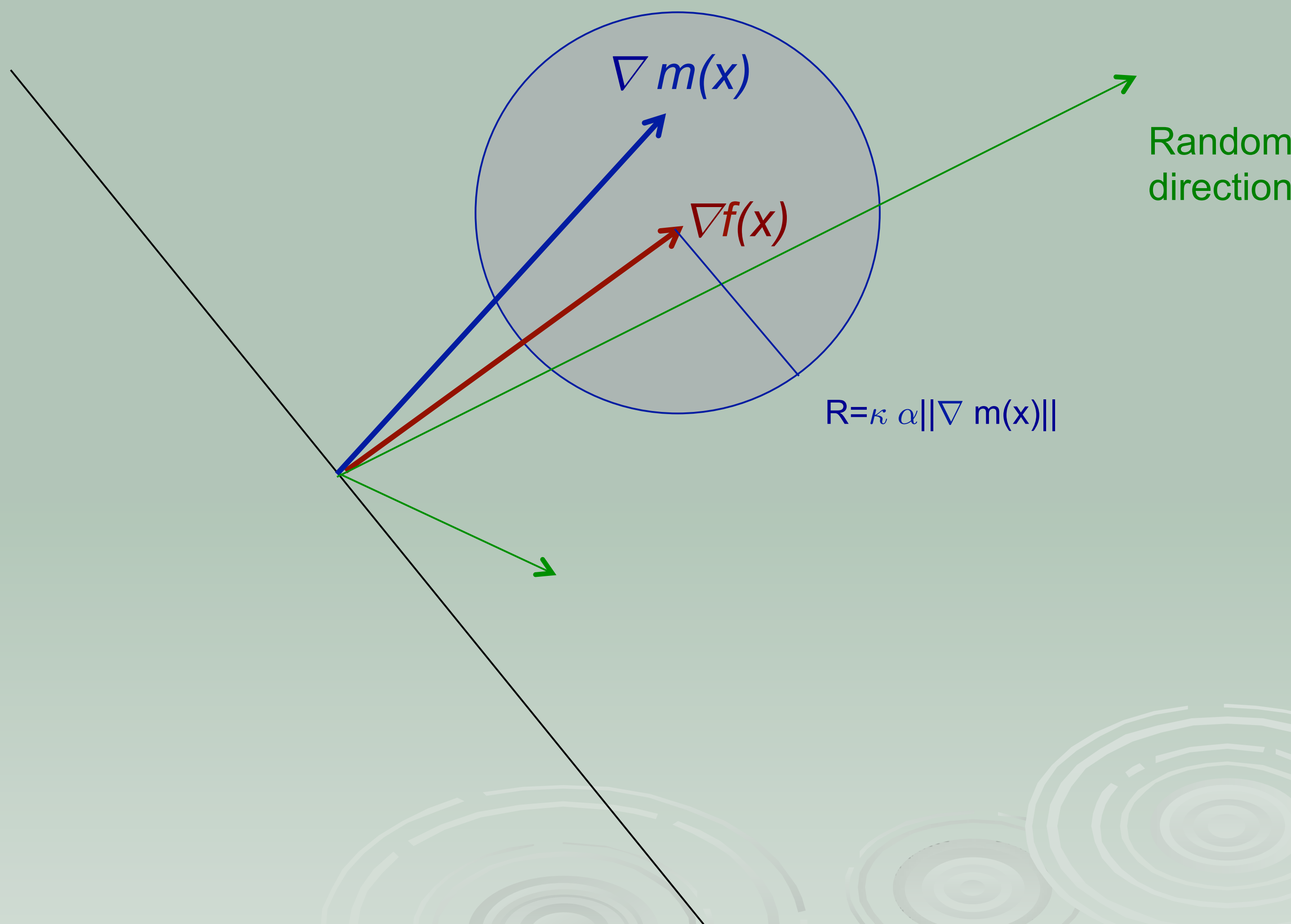
Successful step

If yes accept x^+ as the new iterate.
Increase α_k by a constant factor if not too large.

Unsuccessful step

Otherwise decrease α_k by the constant factor.
Repeat the iteration.

Random directions vs. random fully linear model gradients



Key observation for line search convergence

If m_k is κ-fully linear and ∇f is L-Lipschitz continuous

then when α_k is small enough (i.e. $\alpha_k \leq (1-\theta)/(L/2+\kappa)$)

$$f(x^+) = f(x_k - \alpha_k g_k) \leq f(x_k) - \alpha_k \theta \|g_k\|^2$$

Successful step!

Analysis of line search convergence

Assume m_k is always κ-fully linear

$\Downarrow$

$$\alpha_k \geq C \;\forall k$$

and

$$\text{if } \|\nabla f(x_k)\| \geq \epsilon \;\text{ then } \|g_k\| \geq \epsilon/2$$

$\Downarrow$

$$f(x_k) - f(x_{k+1}) \geq \frac{C\theta\epsilon^2}{4}$$

C is a constant depending on κ, θ, *L*, etc

Convergence!!

Analysis of line search convergence

Assume m_k is ~~always~~ κ-fully linear w.p. $\geq$ *1-*δ

$$\Downarrow$$

$$\alpha_k \geq \cancel{C \; \forall k}$$

and

if $\|\nabla f(x_k)\| \geq \epsilon$ then $\|g_k\| \geq \epsilon/2$ w.p. $\geq$ *1-*δ

$$\Downarrow$$

success $\quad f(x_k) - f(x_{k+1}) \geq \frac{\alpha_k \theta \epsilon^2}{4}$ w.p. $\geq$ *1-*δ

$$\alpha_{k+1} = \gamma \alpha_k$$

no success $\quad \alpha_{k+1} = \gamma^{-1} \alpha_k$ w.p. $\leq \delta$

Analysis via martingales

Analyze two stochastic processes: X_k *and* Y_k:

$$X_{k+1} = \begin{cases} \min\{C, \gamma X_k\} & \text{w.p. } 1-\delta \\ \gamma^{-1} X_k & \text{w.p. } \delta \end{cases}$$

$$Y_{k+1} = \begin{cases} Y_k + X_k \theta \epsilon^2/4 & \text{w.p. } 1-\delta \\ Y_k & \text{w.p. } \delta \end{cases}$$

We observe that

$$\alpha_k \geq X_k$$

$$f(x_0) - f(x_k) \geq Y_k$$

If random models are independent of the past, then X_k and Y_k are random walks, otherwise they are submartingales if $\delta \leq 1/2$.

Analysis via martingales

Analyze two stochastic processes: X_k *and* Y_k:

$$X_{k+1} = \begin{cases} \min\{C, \gamma X_k\} & \text{w.p. } 1-\delta \\ \gamma^{-1} X_k & \text{w.p. } \delta \end{cases}$$

$$Y_{k+1} = \begin{cases} Y_k + X_k \theta \epsilon^2/4 & \text{w.p. } 1-\delta \\ Y_k & \text{w.p. } \delta \end{cases}$$

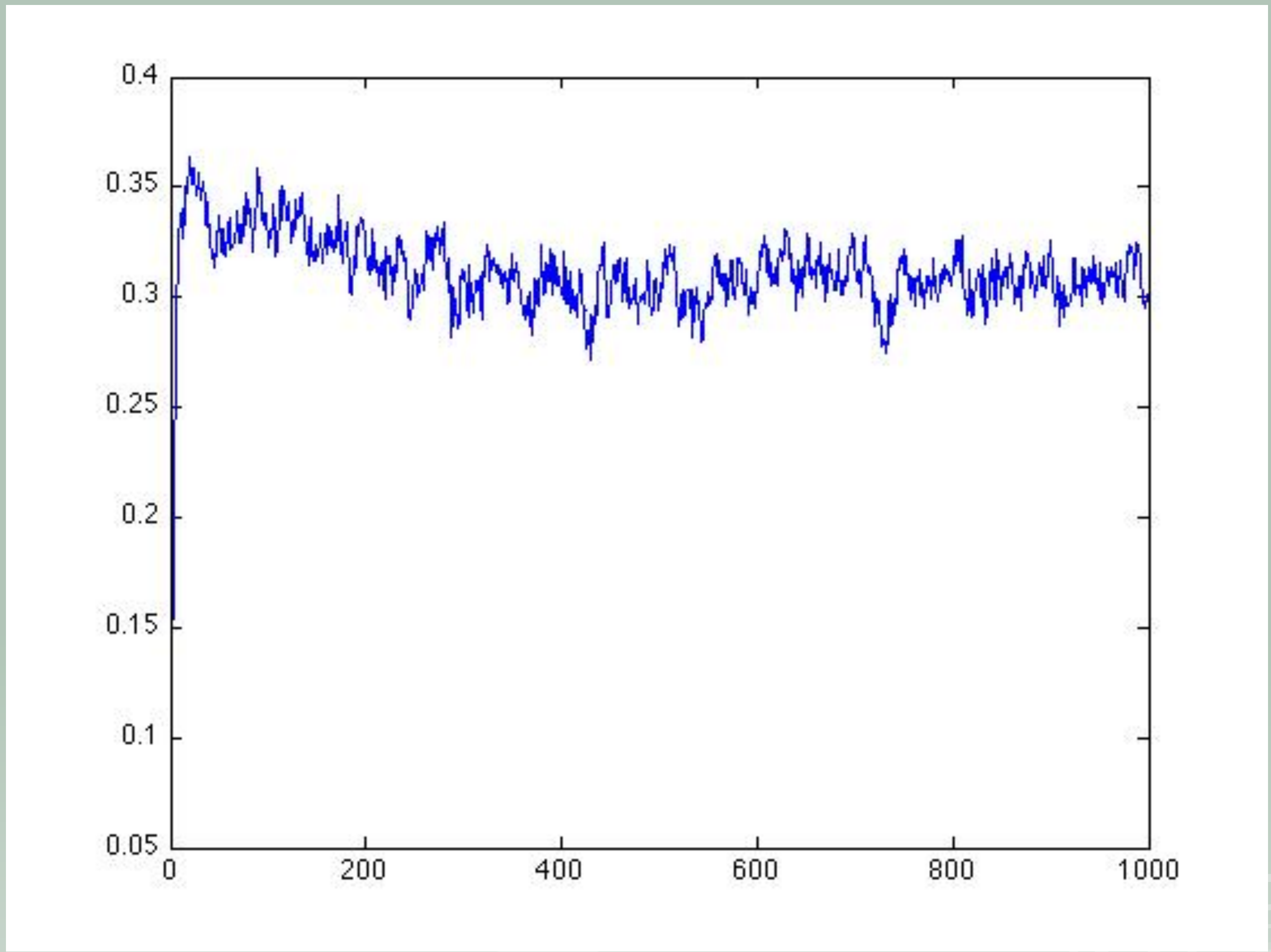
We observe that

$$\alpha_k \geq X_k$$

$$f(x_0) - f(x_k) \geq Y_k$$

X_k does not converge to 0 w.p. 1 => algorithm converges
Expectations of Y_k and X_k will facilitate convergence rates.

Behavior of X_k for γ=2, C=1 and δ=0.45



X_k

k

Future work

- Convergence rates theory based on random models.
- Extend algorithmic random model frameworks.
- Extending to new types of models.
- Recovering different types of function structure.
- Efficient implementations.

Thank you!

Analysis of line search convergence

If m_k is κ-fully linear

$$\|g_k - \nabla f(x_k)\| \leq \kappa \Delta_k = \kappa \alpha_k \|g_k\|$$

If ∇f is L-Lipschitz continuous and $\alpha_k \leq (1-\theta)/(L/2+\kappa)$

$$f(x_k - \alpha_k * g_k) \leq f(x_k) - \alpha_k \theta \|g_k\|^2$$

If $\|\nabla f(x_k)\| \geq \epsilon$ then $\|g_k\| \geq \epsilon/2$ and

$$f(x_k) - f(x_{k+1}) \geq \frac{\alpha_k \theta \epsilon^2}{4}$$

Hence only so many line search steps are needed to get a small gradient

Analysis of line search convergence

If m_k is κ-fully linear

$$\|g_k - \nabla f(x_k)\| \leq \kappa \Delta_k = \kappa \alpha_k \|g_k\|$$

If ∇f is L-Lipschitz continuous and $\alpha_k \leq (1-\theta)/(L/2+\kappa)$

$$f(x_k - \alpha_k * g_k) \leq f(x_k) - \alpha_k \theta \|g_k\|^2$$

If $\|\nabla f(x_k)\| \geq \epsilon$ then $\|g_k\| \geq \epsilon/2$ and

$$f(x_k) - f(x_{k+1}) \geq \frac{\alpha_k \theta \epsilon^2}{4}$$

We assumed that $m_k(x)$ is κ-fully-linear every time.