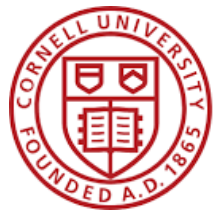
Overview and Recent Advances in Derivative Free Optimization

Katya Scheinberg

Joint work with A. Berahas, J. Blanchet, L. Cao, C. Cartis, A. R. Conn, M. Menickelly, C. Paquette, L. Vicente

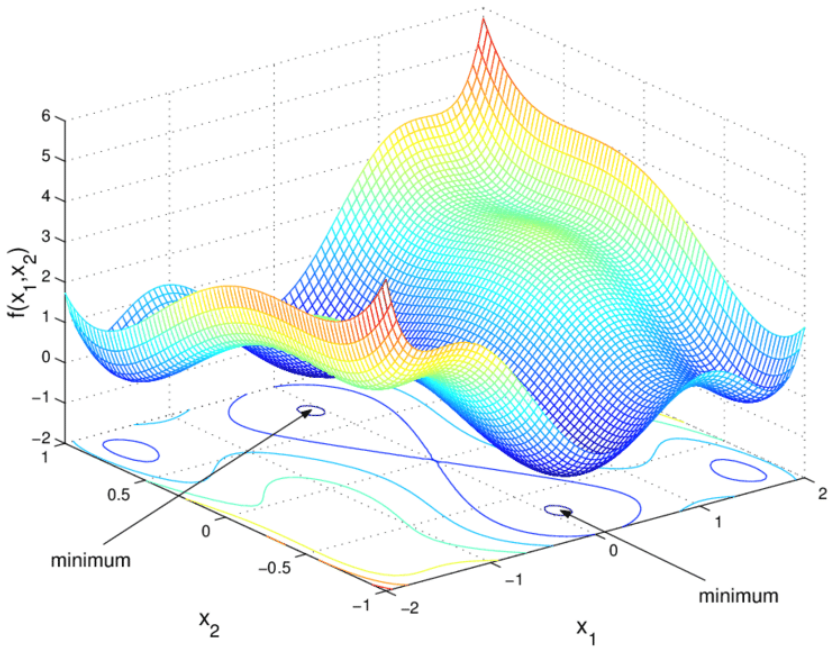
School of Operations Research and Information Engineering



Cornell University

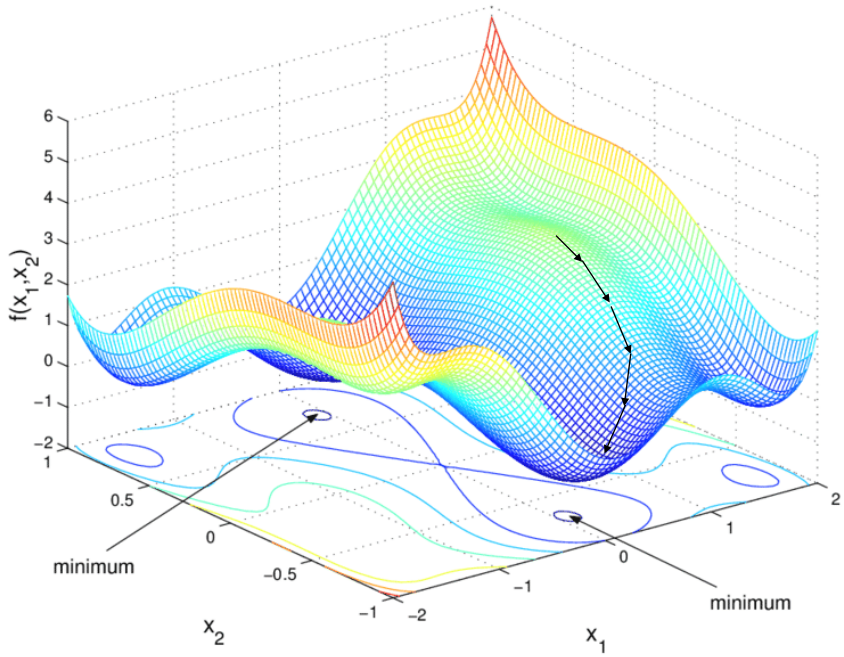
IPAM Workshop: From Passive to Active: Generative and Reinforcement Learning with Physics, Sept 23-27, 2019

Local and Global Optimization



From Roos, Terlaky and DeKlerk, "Nonlinear Optimisation", 2002.

Optimization and gradient descent



Black Box Optimization Problems

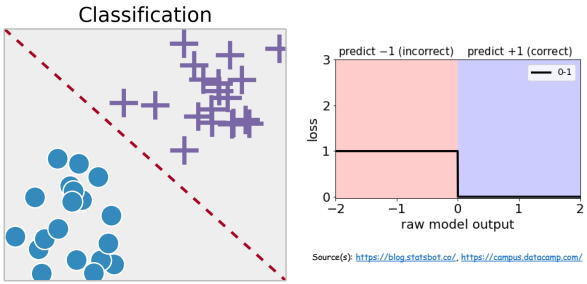
$$\min_{x \in \mathbb{R}^n} f(x)$$

- f nonlinear function; **derivatives of f not available**
- Noisy functions, **stochastic** or **deterministic**

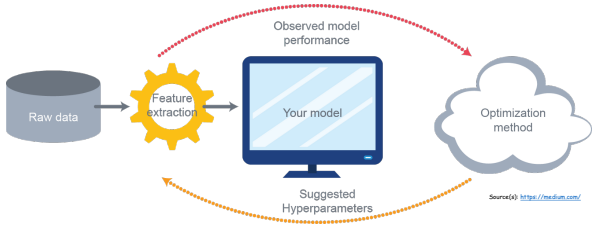
$$\min_{x \in \mathbb{R}^n} f(x) = \phi(x) + \epsilon(x) \qquad \min_{x \in \mathbb{R}^n} f(x) = \phi(x)(1 + \epsilon(x))$$

Motivation

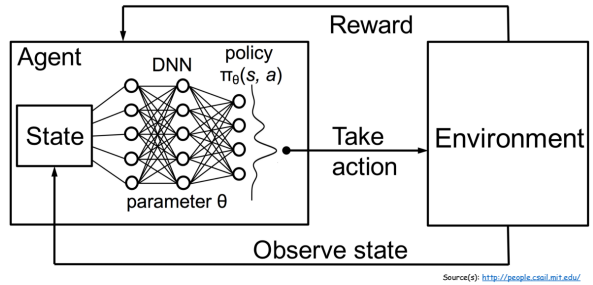
- Machine Learning



- Deep Learning

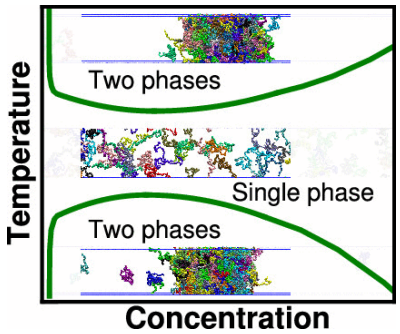


- Reinforcement Learning



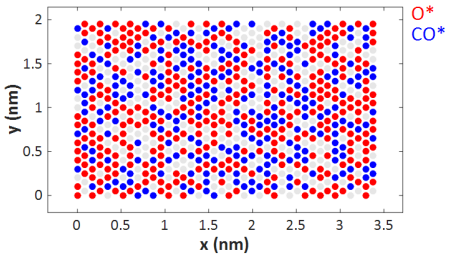
Optimizing properties obtained from expensive simulations or experiments

Critical temperatures from molecular dynamics simulations



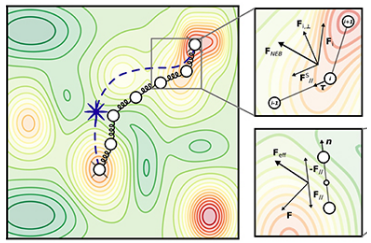
Dignon et al. ACS Cent. Sci., Article ASAP

Reaction rate estimation from kinetic Monte Carlo simulations



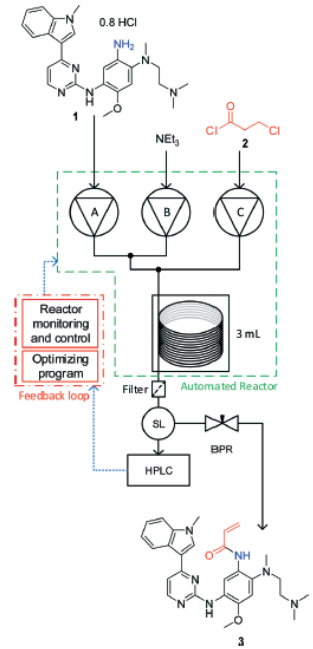
ZACROS (http://zacros.org/tutorials)

Activation barriers from quantum mechanical nudged elastic band calculations



Andersen et al. Front. Chem. 2019

Yield estimation from experimental organic synthesis reactor systems



Holmes et al. React. Chem. Eng., 2016, 1, 36

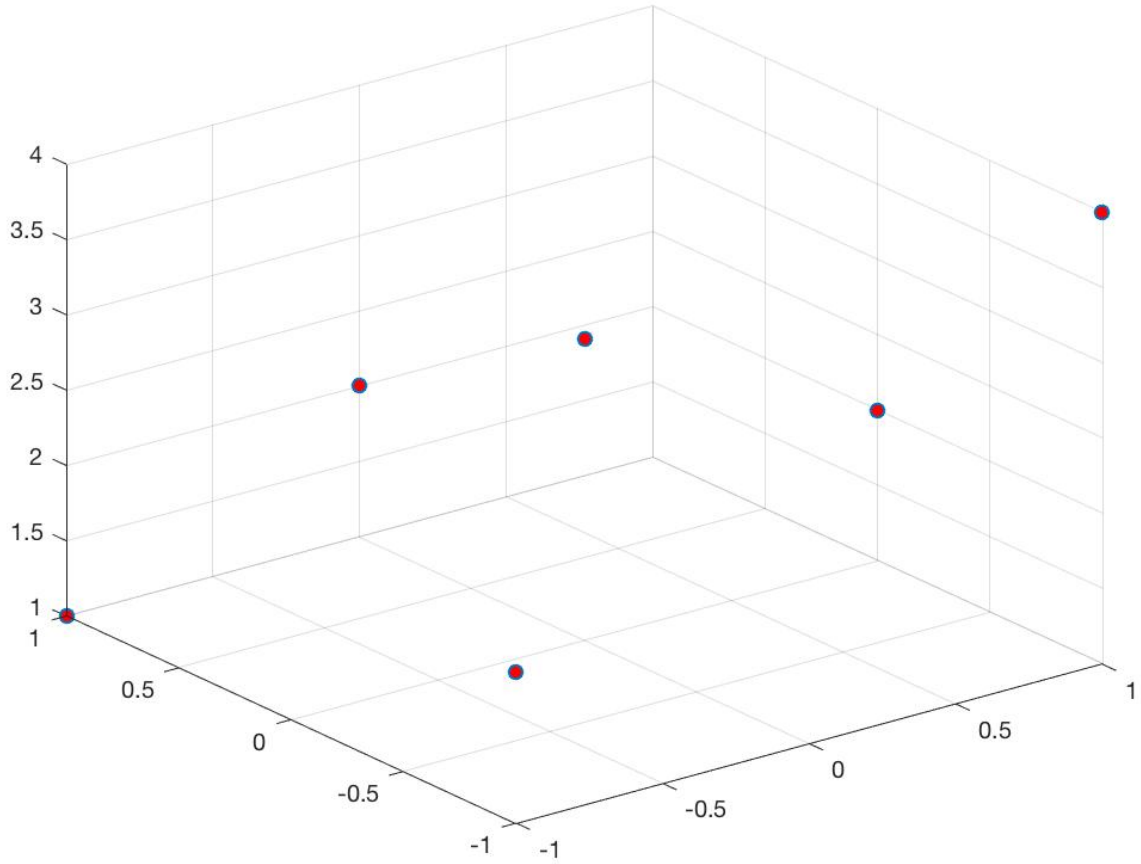
- **Many examples exist in the domain of molecular and materials science where calculating a property requires expensive computations or experiments**
- **In many of these cases, derivatives are not available**

Derivative-free methods: direct and random search

Iterative algorithms that converge to a local optima.

In each iteration:

1. Evaluate a set of sample points around the current iterate;
2. Choose the sample point with the best function value;
3. Make this point the next iterate;

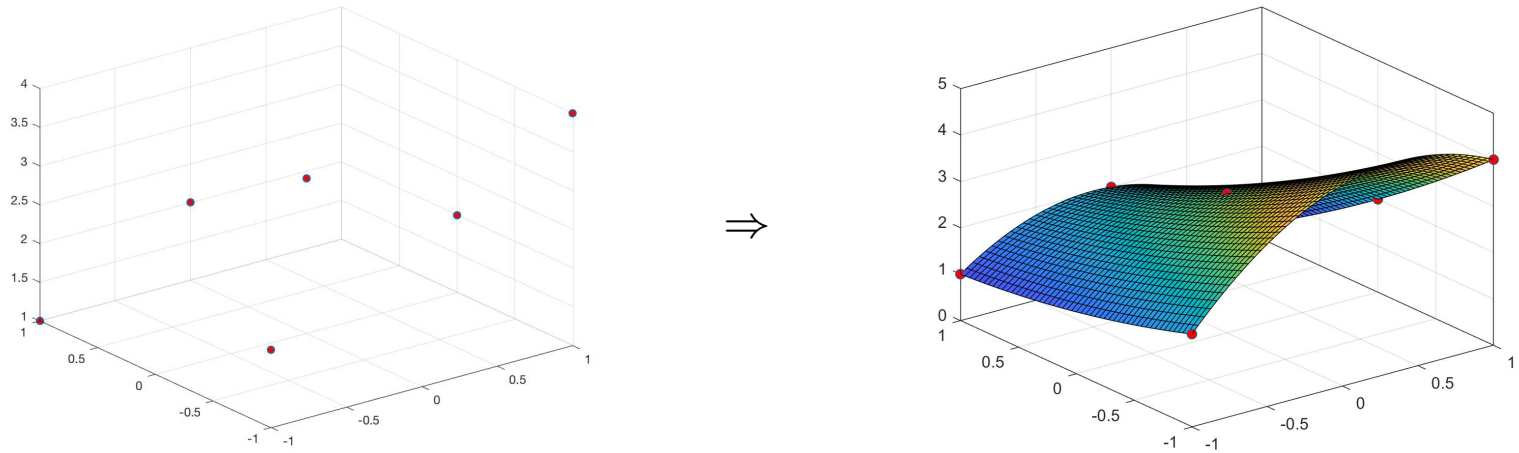


Derivative free methods: model-based

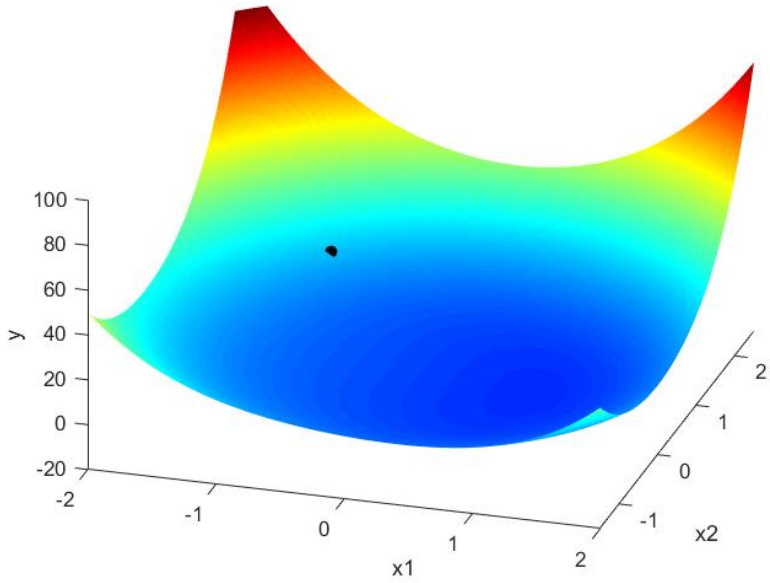
Iterative algorithms that converge to a local optimum.

In each iteration:

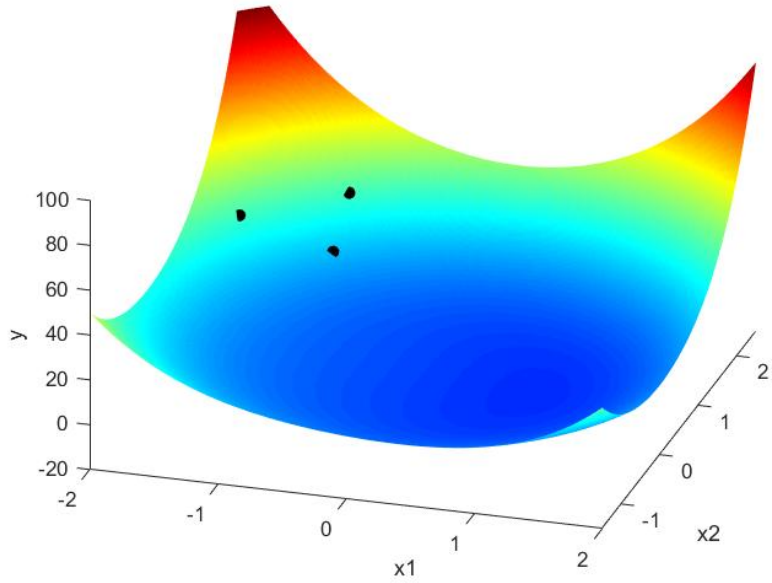
1. Evaluate a set of sample points around the current iterate;
2. Interpolate the sample points with a linear or quadratic model;
3. Use this model to find the next iterate;



Model-Based Trust Region Method (pioneered by M.J.D. Powell)

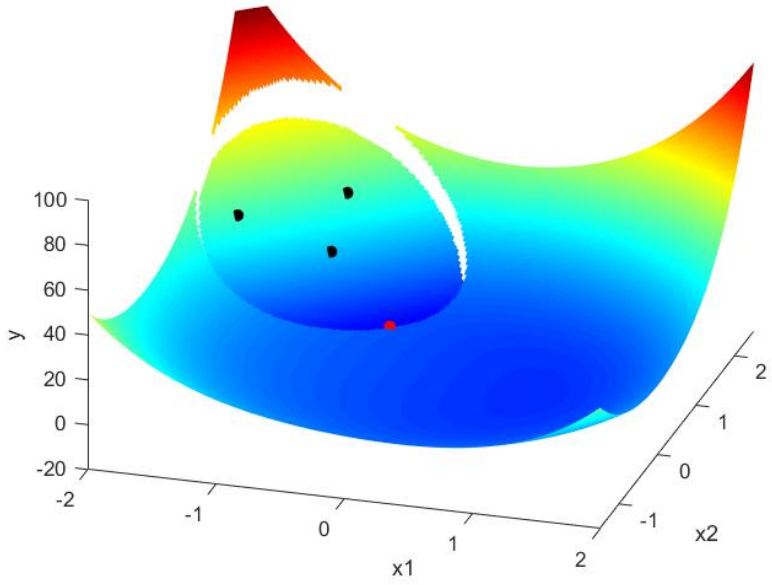


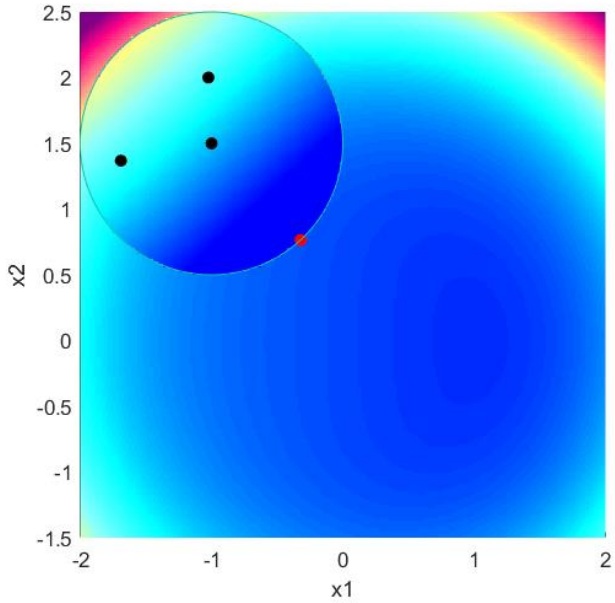
(a) starting point



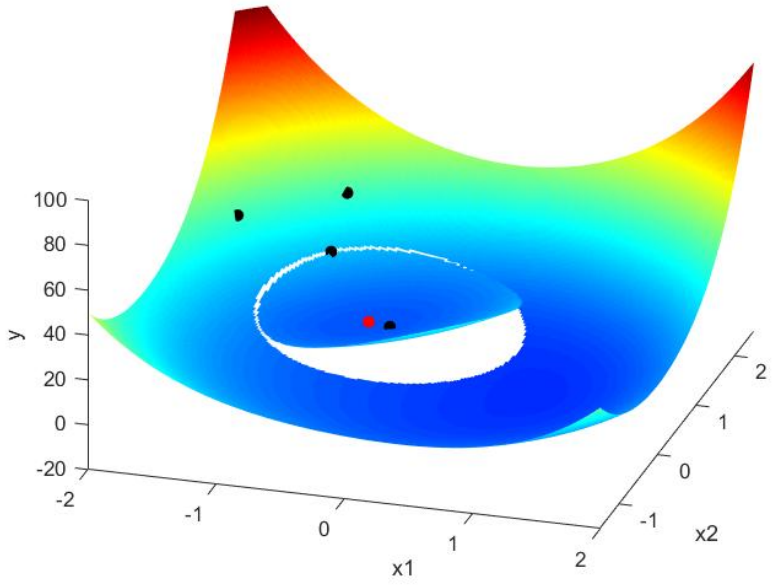
(b) initial sampling

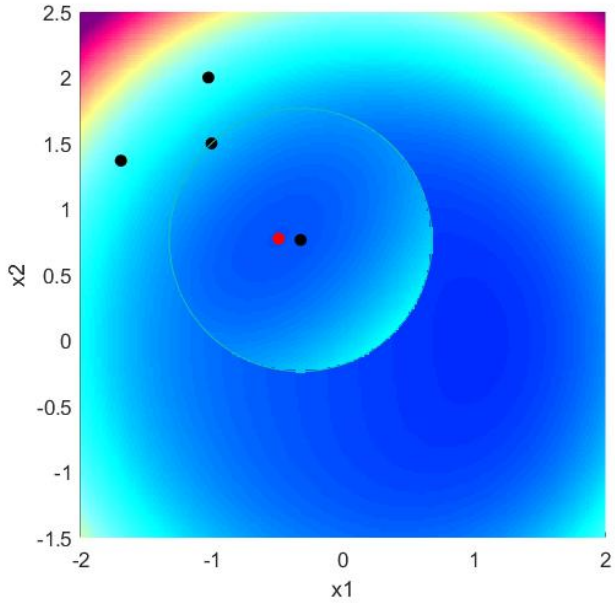
Model-Based Trust Region Method



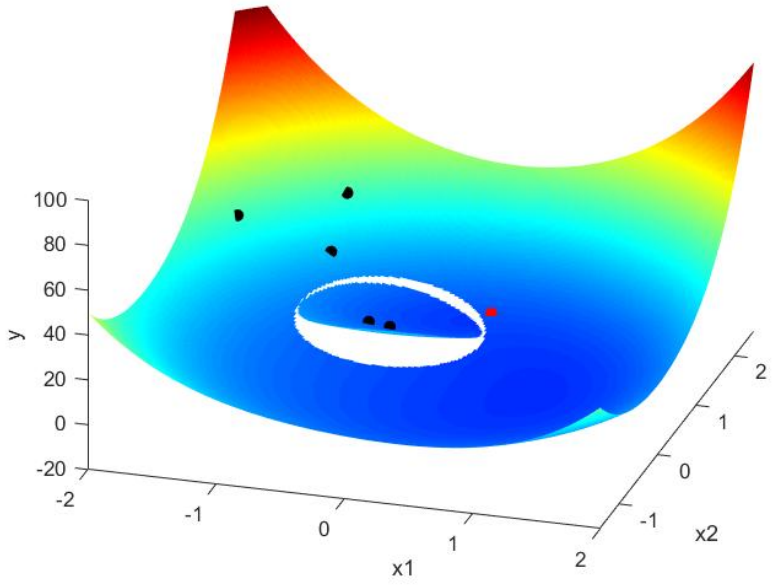


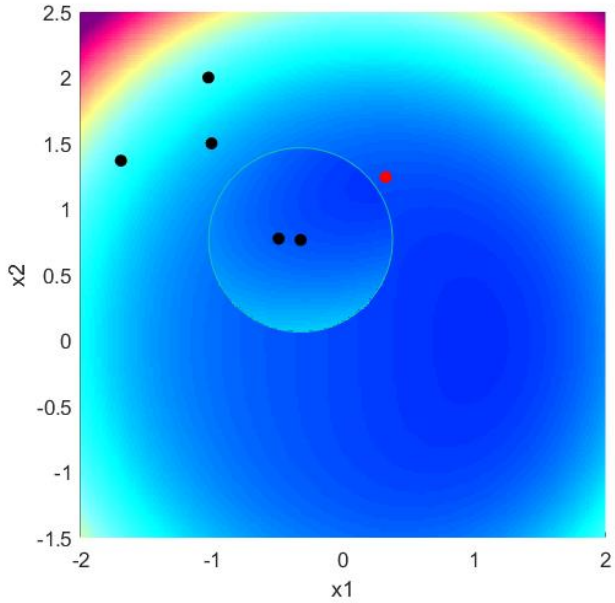
Model-Based Trust Region Method



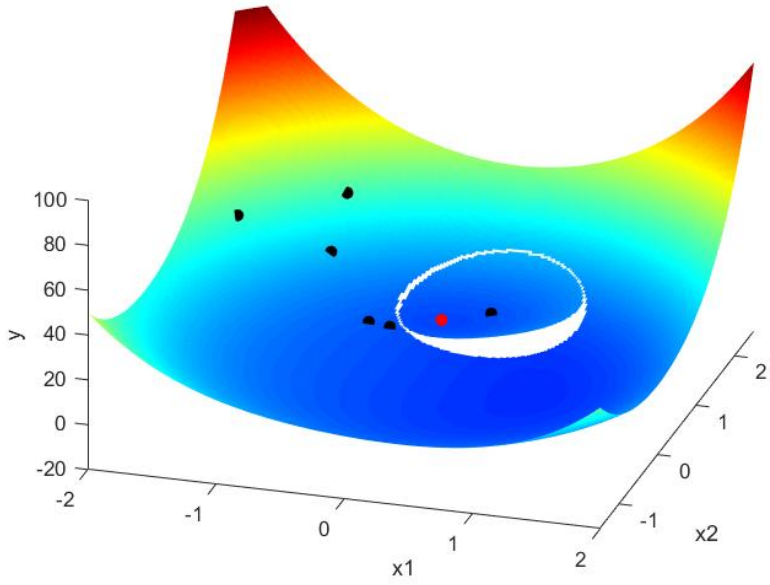


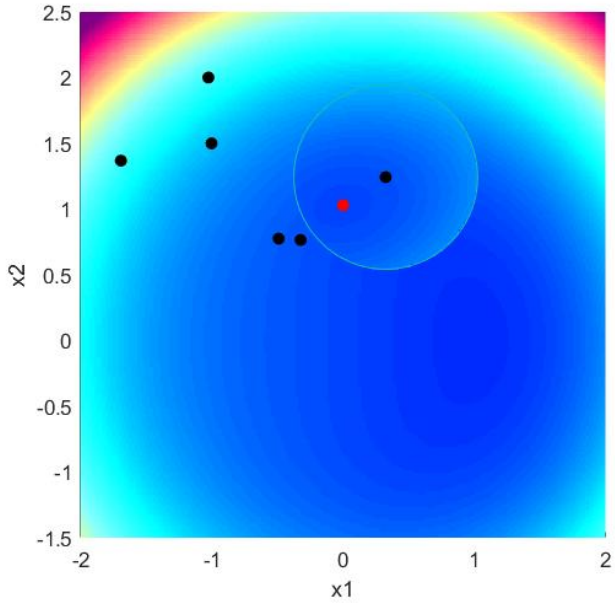
Model-Based Trust Region Method



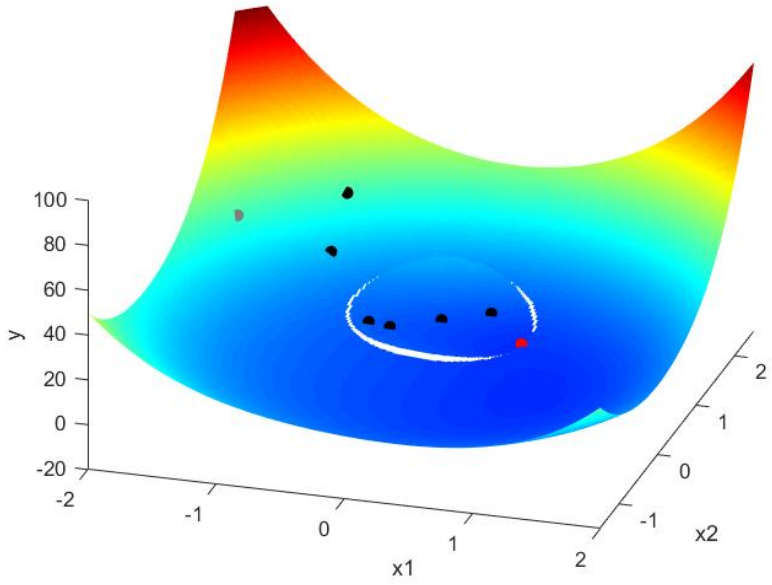


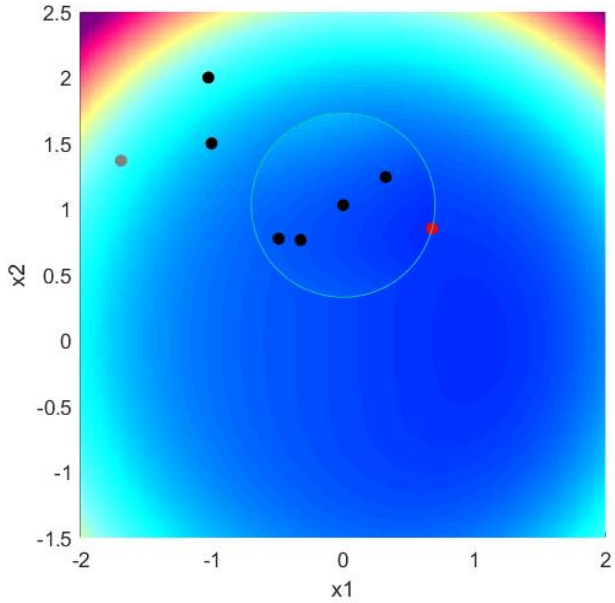
Model-Based Trust Region Method



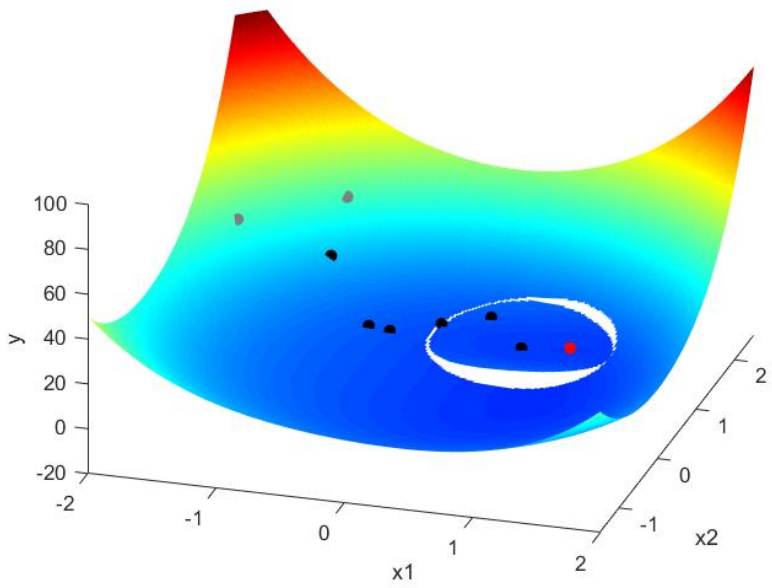


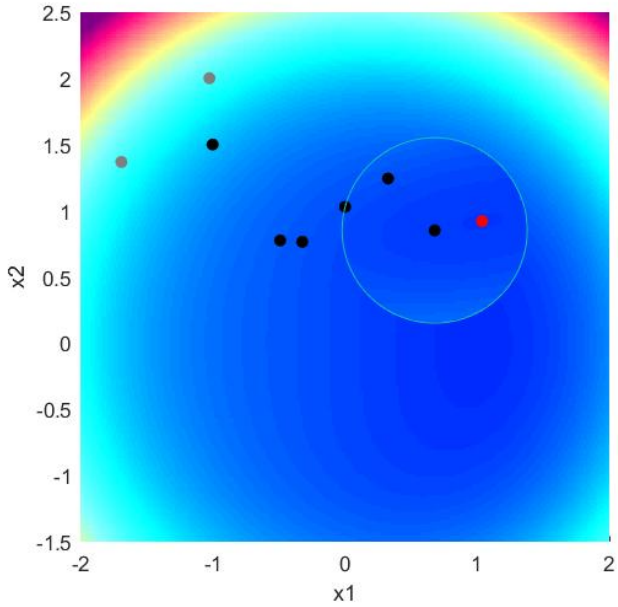
Model-Based Trust Region Method



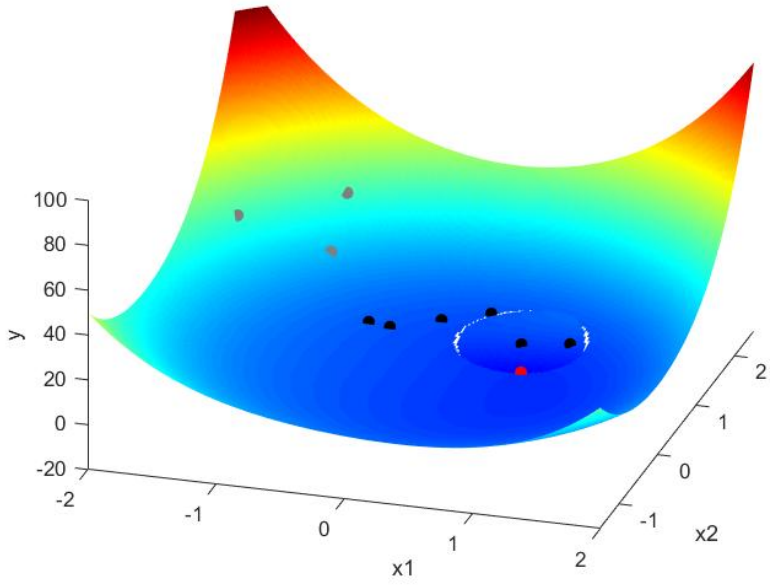


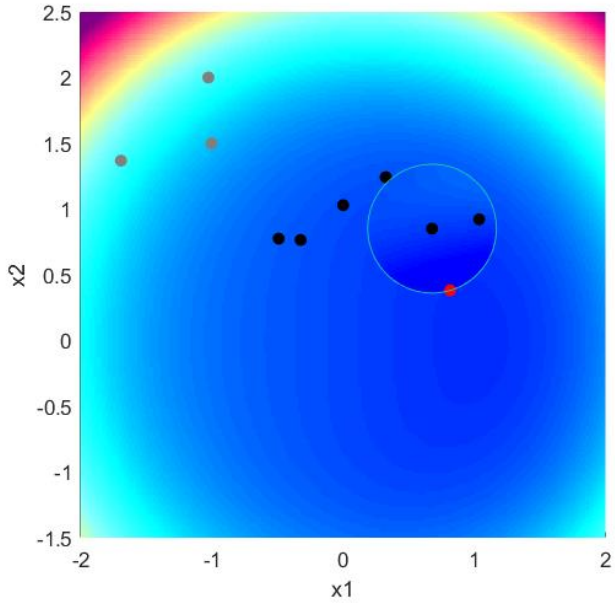
Model-Based Trust Region Method



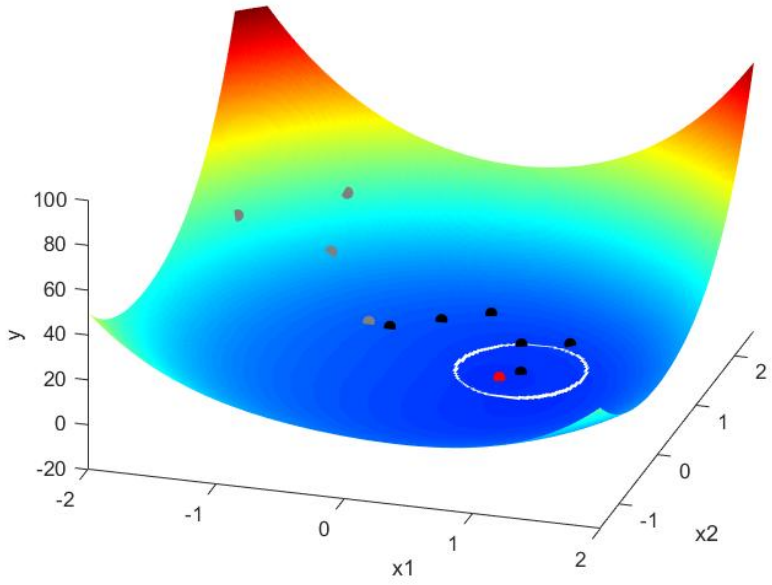


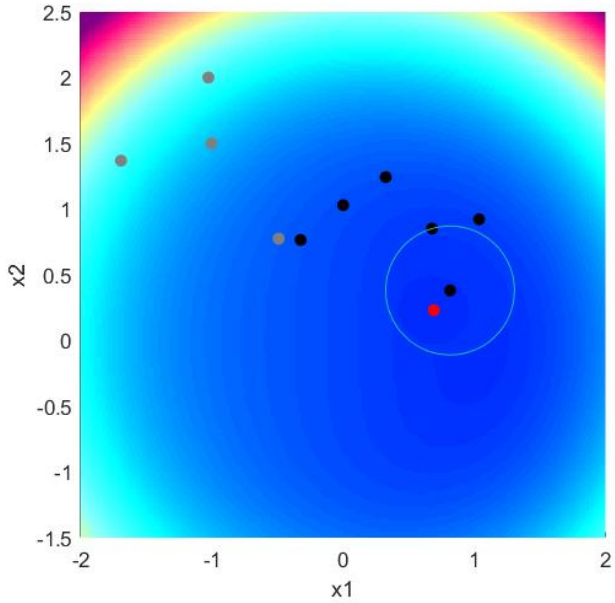
Model-Based Trust Region Method



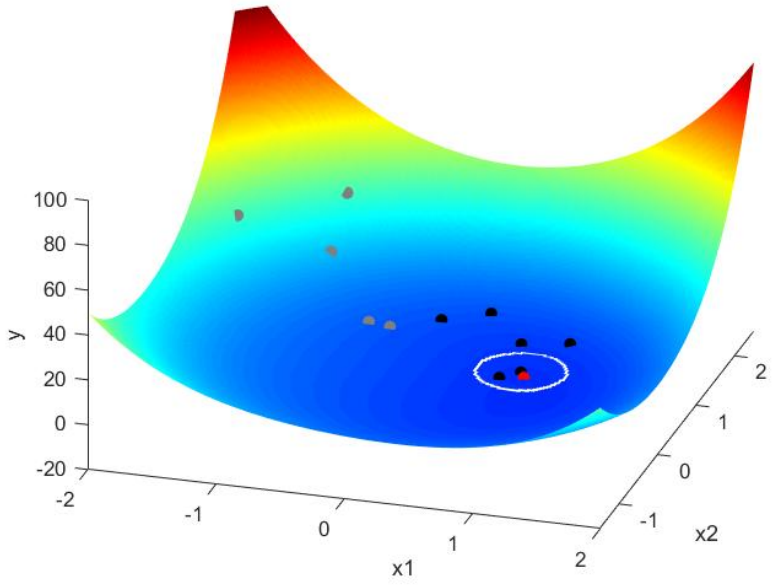


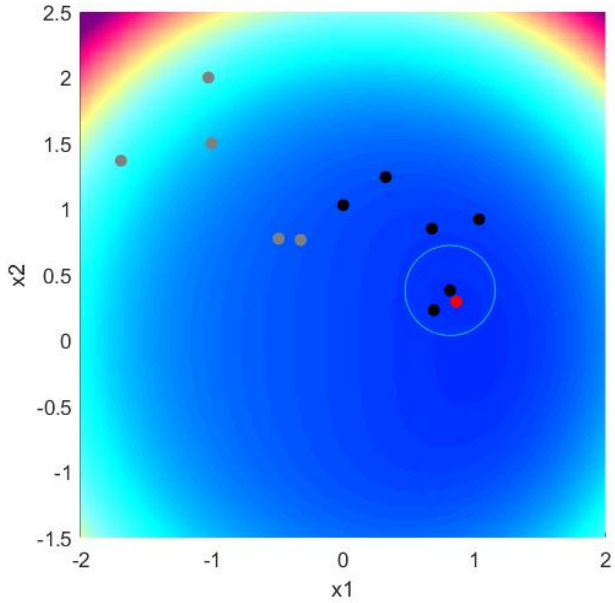
Model-Based Trust Region Method



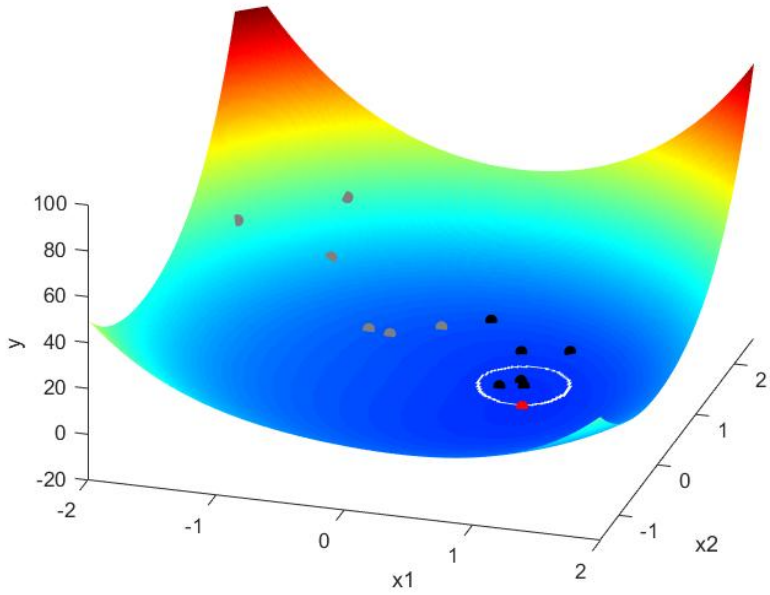


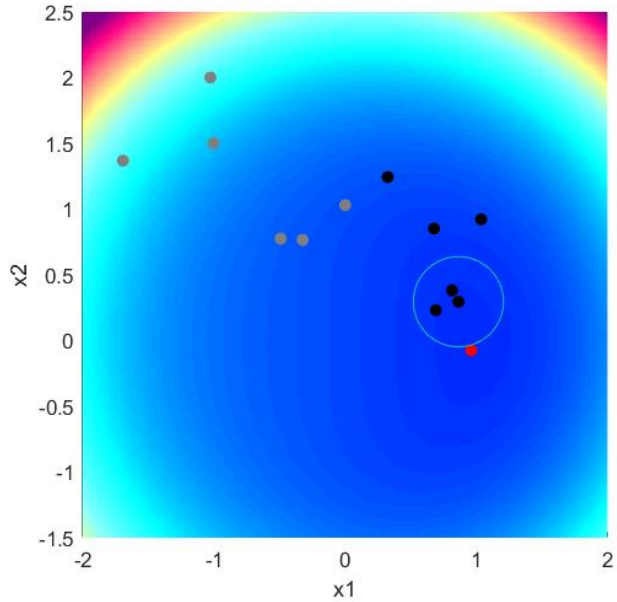
Model-Based Trust Region Method





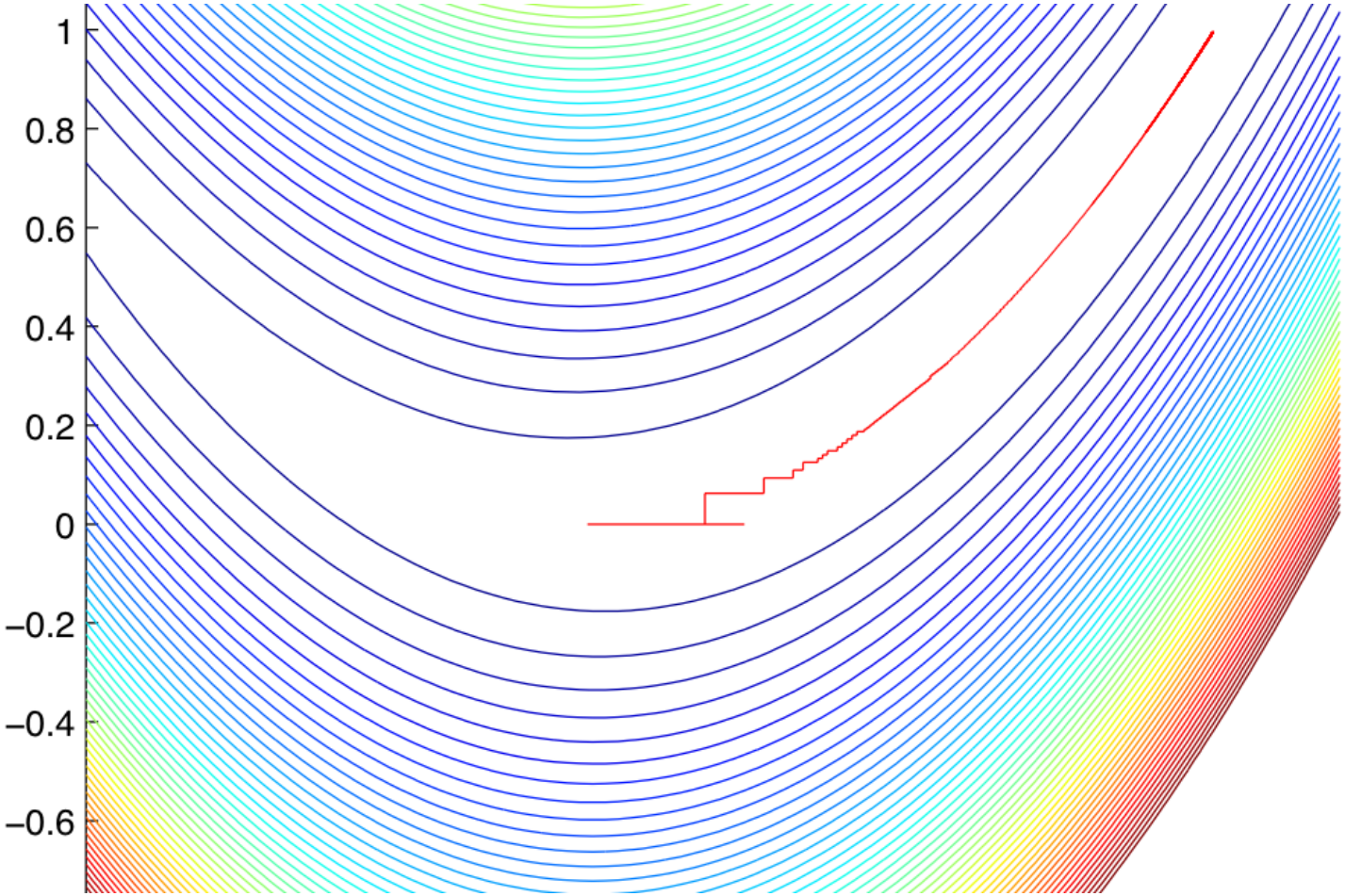
Model-Based Trust Region Method





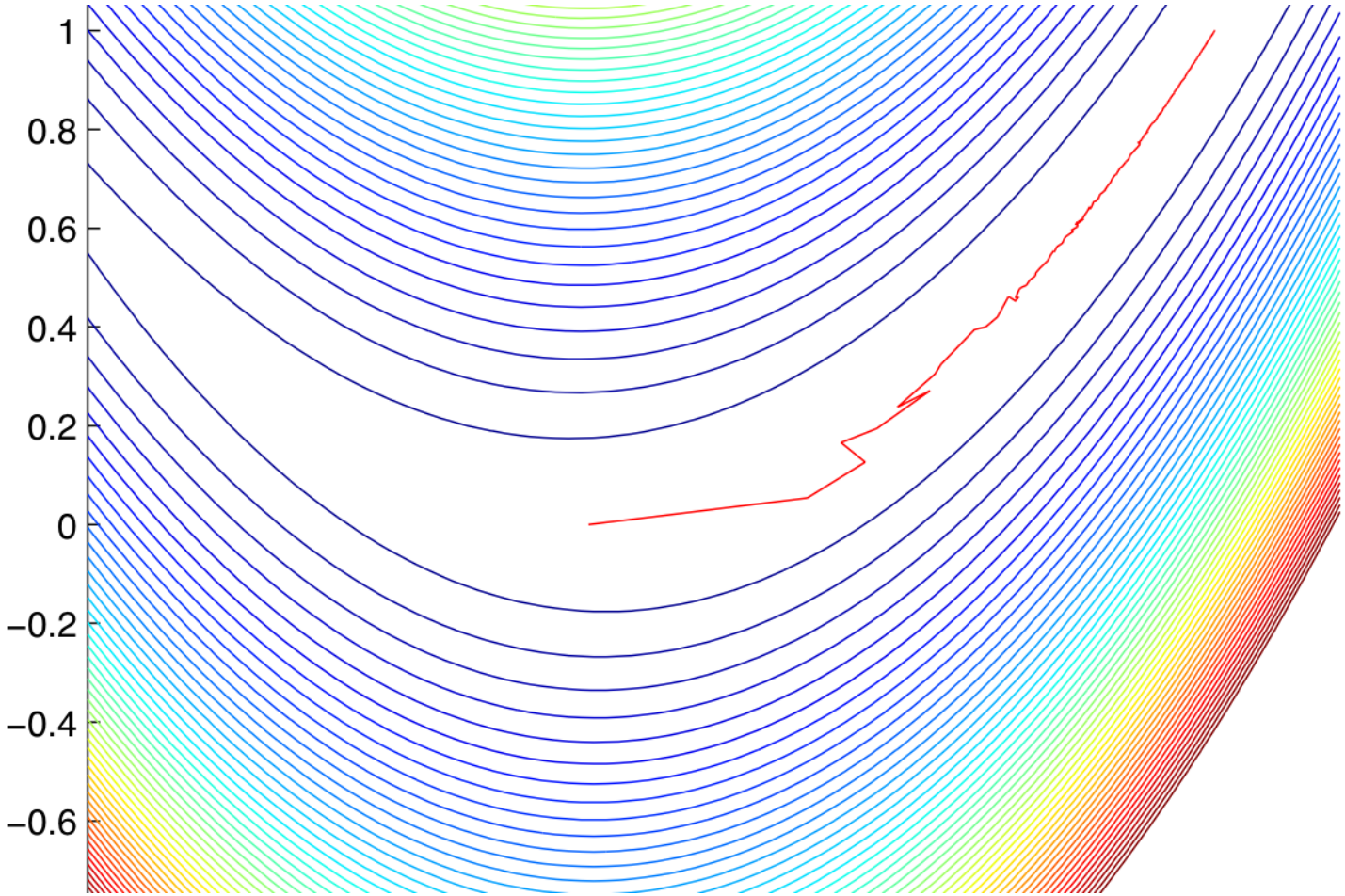
Shrinking and expanding trust region radius, exploiting curvature, efficient in terms of samples

Direct Search



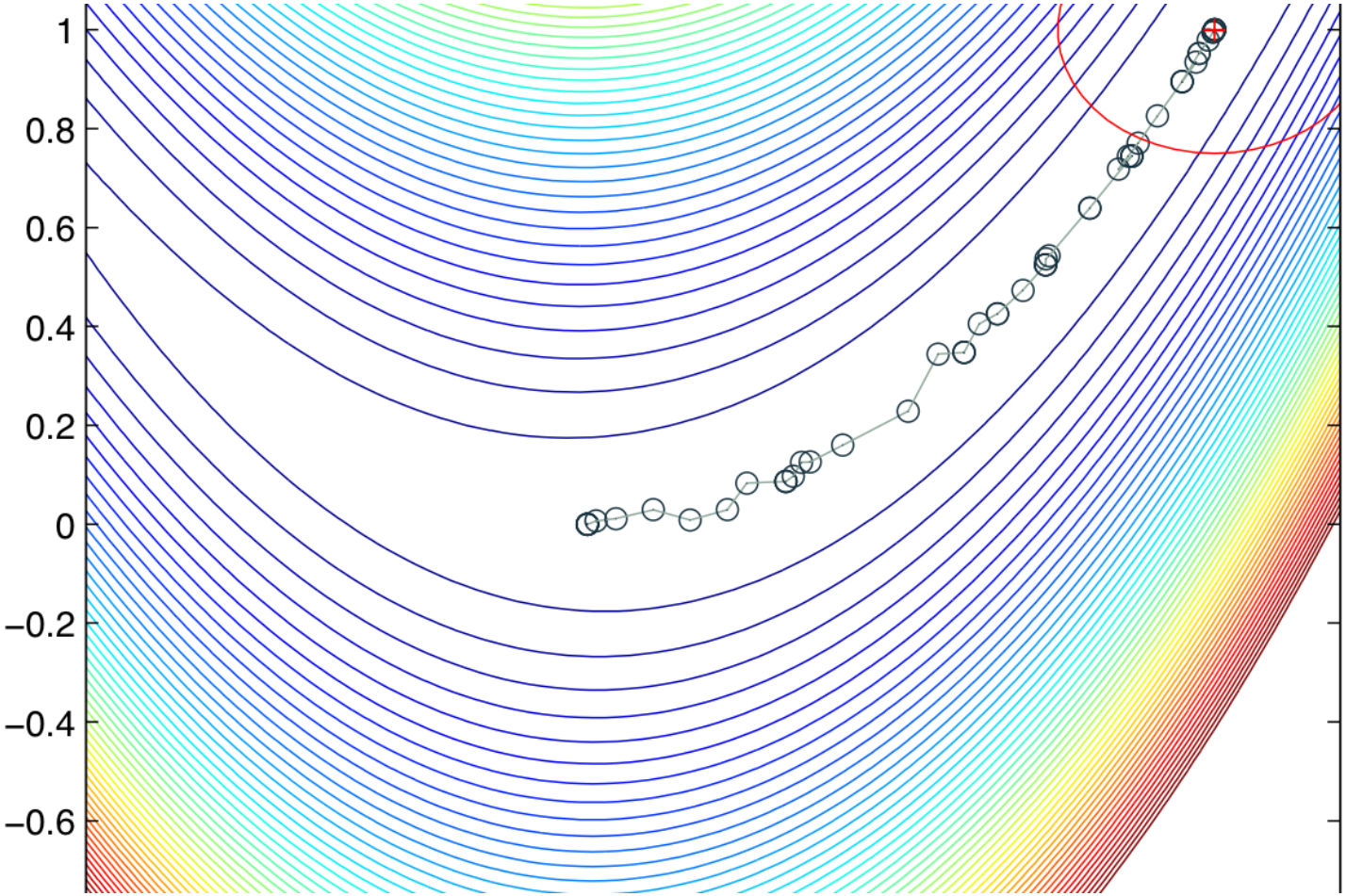
11307 function evaluations

Random Search



3705 function evaluations

Trust Region Method



69 function evaluations

Active learning, generative models and derivative free optimization optimization

What does model-based derivative-free optimization do?

- Using some "labeled" data $(x, f(x))$, build a models $m(x)$. What do we want from that model $m(x)$? Quality? Simplicity?
- Optimize $m(x)$ or "related function", to obtain new potentially interesting data point. What do we optimize?
- Modify model (how?), repeat.
- What do we need for convergence?

Assumptions on models for convergence

For trust region, first-order convergence

$$\|\nabla f(x^k) - \nabla m_k(x^k)\| \quad \leq \quad \mathcal{O}(\Delta_k),$$

For trust region, second-order convergence

$$\begin{aligned} \|\nabla^2 f(x^k) - \nabla^2 m_k(x^k)\| &\leq \mathcal{O}(\Delta_k) \\ \|\nabla f(x^k) - \nabla m_k(x^k)\| &\leq \mathcal{O}(\Delta_k^2) \end{aligned}$$

For line search, first-order converegnce

$$\|\nabla f(x^k) - \nabla m_k(x^k)\| \quad \leq \quad \mathcal{O}(\alpha_k \|\nabla m_k\|)$$

Intuition

In other words, model should have comparable Taylor expansion as the true function w.r.t. the step size.

Assumptions on models for convergence

For trust region, first-order convergence

$$\|\nabla f(x^k) - \nabla m_k(x^k)\| \quad \leq \quad \mathcal{O}(\Delta_k), \quad \text{w.p. } 1-\delta$$

For trust region, second-order convergence

$$\begin{aligned} \|\nabla^2 f(x^k) - \nabla^2 m_k(x^k)\| &\leq \mathcal{O}(\Delta_k) \quad \text{w.p. } 1-\delta \\ \|\nabla f(x^k) - \nabla m_k(x^k)\| &\leq \mathcal{O}(\Delta_k^2) \quad \text{w.p. } 1-\delta \end{aligned}$$

For line search, first-order converegnce

$$\|\nabla f(x^k) - \nabla m_k(x^k)\| \quad \leq \quad \mathcal{O}(\alpha_k \|\nabla m_k\|) \quad \text{w.p. } 1-\delta$$

Intuition

In other words, model should have comparable Taylor expansion as the true function w.r.t. the step size.

Building models via linear interpolation

$$m(y) = f(x) + g(x)^T(y - x): \quad m(y) = f(y), \forall y \in \mathcal{Y}.$$

Building models via linear interpolation

$$m(y) = f(x) + g(x)^T(y - x): \quad m(y) = f(y), \forall y \in \mathcal{Y}.$$

- Let $\mathcal{Y} = \{x + \sigma y_1, ..., x + \sigma y_n\}$, $\sigma > 0$,

$$F_{\mathcal{Y}} = \begin{bmatrix} f(x + \sigma y_1) - f(x) \\ \vdots \\ f(x + \sigma y_n) - f(x) \end{bmatrix} \in \mathbb{R}^n, \quad M_{\mathcal{Y}} = \begin{bmatrix} y_1^T \\ \vdots \\ y_n^T \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Building models via linear interpolation

$$m(y) = f(x) + g(x)^T(y - x): \quad m(y) = f(y), \forall y \in \mathcal{Y}.$$

- Let $\mathcal{Y} = \{x + \sigma y_1, ..., x + \sigma y_n\}$, $\sigma > 0$,

$$F_{\mathcal{Y}} = \begin{bmatrix} f(x + \sigma y_1) - f(x) \\ \vdots \\ f(x + \sigma y_n) - f(x) \end{bmatrix} \in \mathbb{R}^n, \quad M_{\mathcal{Y}} = \begin{bmatrix} y_1^T \\ \vdots \\ y_n^T \end{bmatrix} \in \mathbb{R}^{n \times n}$$

- Model $m(y)$ constructed to satisfy interpolation conditions:

$$\sigma M_{\mathcal{Y}} g = F_{\mathcal{Y}}$$

Building models via linear interpolation

$$m(y) = f(x) + g(x)^T(y - x): \quad m(y) = f(y), \forall y \in \mathcal{Y}.$$

- Let $\mathcal{Y} = \{x + \sigma y_1, ..., x + \sigma y_n\}$, $\sigma > 0$,

$$F_{\mathcal{Y}} = \begin{bmatrix} f(x + \sigma y_1) - f(x) \\ \vdots \\ f(x + \sigma y_n) - f(x) \end{bmatrix} \in \mathbb{R}^n, \quad M_{\mathcal{Y}} = \begin{bmatrix} y_1^T \\ \vdots \\ y_n^T \end{bmatrix} \in \mathbb{R}^{n \times n}$$

- Model $m(y)$ constructed to satisfy interpolation conditions:

$$\sigma M_{\mathcal{Y}} g = F_{\mathcal{Y}}$$

Theorem [Conn, Scheinberg & Vicente, 2008]

Let $\mathcal{Y} = \{x, x + \sigma y_1, \ldots, x + \sigma y_n\}$ be set of interpolation points such that $\max_i \|y_i\| \leq 1$ and that $M_{\mathcal{Y}}$ is nonsingular. Suppose that the function f has L-Lipschitz continuous gradients. Then,

$$\|\nabla m(x) - \nabla f(x)\| \leq \frac{\|M_{\mathcal{Y}}^{-1}\|_2 \sqrt{n} \sigma L}{2}.$$

- Cost: $\mathcal{O}(n^3)$ (reduces to $\mathcal{O}(n^2)$ if $M_{\mathcal{Y}}$ is orthornormal and $\mathcal{O}(n^2)$ if $M_{\mathcal{Y}} = I$)

Quadratic Interpolation Models

$$m(y) = f(x) + g(x)^T(y-x) + \frac{1}{2}(y-x)^T H(x)(y-x): \quad m(y) = f(y),\ \forall y \in \mathcal{Y}.$$

Quadratic Interpolation Models

$$m(y) = f(x) + g(x)^T(y-x) + \frac{1}{2}(y-x)^T H(x)(y-x): \quad m(y) = f(y),\ \forall y \in \mathcal{Y}.$$

- Let $\mathcal{Y} = \{x + \sigma y_1, ..., x + \sigma y_N\}$, $\sigma > 0$,

$$F_{\mathcal{Y}} = \begin{bmatrix} f(x+\sigma y_1) - f(x) \\ \vdots \\ f(x+\sigma y_N) - f(x) \end{bmatrix} \in \mathbb{R}^N, \quad M_{\mathcal{Y}} = \begin{bmatrix} y_1^T & \mathrm{vec}(y_1 y_1^T) \\ \vdots & \vdots \\ y_n^T & \mathrm{vec}(y_n y_n^T) \end{bmatrix} \in \mathbb{R}^{N \times N}$$

Quadratic Interpolation Models

$$m(y) = f(x) + g(x)^T(y-x) + \frac{1}{2}(y-x)^T H(x)(y-x): \quad m(y) = f(y),\ \forall y \in \mathcal{Y}.$$

- Let $\mathcal{Y} = \{x + \sigma y_1, ..., x + \sigma y_N\}$, $\sigma > 0$,

$$F_{\mathcal{Y}} = \begin{bmatrix} f(x+\sigma y_1) - f(x) \\ \vdots \\ f(x+\sigma y_N) - f(x) \end{bmatrix} \in \mathbb{R}^N, \quad M_{\mathcal{Y}} = \begin{bmatrix} y_1^T & \text{vec}(y_1 y_1^T) \\ \vdots & \vdots \\ y_n^T & \text{vec}(y_n y_n^T) \end{bmatrix} \in \mathbb{R}^{N \times N}$$

- Model $m(y)$ constructed to satisfy interpolation conditions:

$$\sigma M_{\mathcal{Y}}(g, \text{vec}(H)) = F_{\mathcal{Y}}$$

Quadratic Interpolation Models

$$m(y) = f(x) + g(x)^T(y-x) + \frac{1}{2}(y-x)^T H(x)(y-x): \quad m(y) = f(y),\ \forall y \in \mathcal{Y}.$$

- Let $\mathcal{Y} = \{x + \sigma y_1, ..., x + \sigma y_N\}$, $\sigma > 0$,

$$F_{\mathcal{Y}} = \begin{bmatrix} f(x+\sigma y_1) - f(x) \\ \vdots \\ f(x+\sigma y_N) - f(x) \end{bmatrix} \in \mathbb{R}^N, \quad M_{\mathcal{Y}} = \begin{bmatrix} y_1^T & \text{vec}(y_1 y_1^T) \\ \vdots & \vdots \\ y_n^T & \text{vec}(y_n y_n^T) \end{bmatrix} \in \mathbb{R}^{N \times N}$$

- Model $m(y)$ constructed to satisfy interpolation conditions:

$$\sigma M_{\mathcal{Y}}(g, \text{vec}(H)) = F_{\mathcal{Y}}$$

Theorem [Conn, Scheinberg & Vicente, 2008]

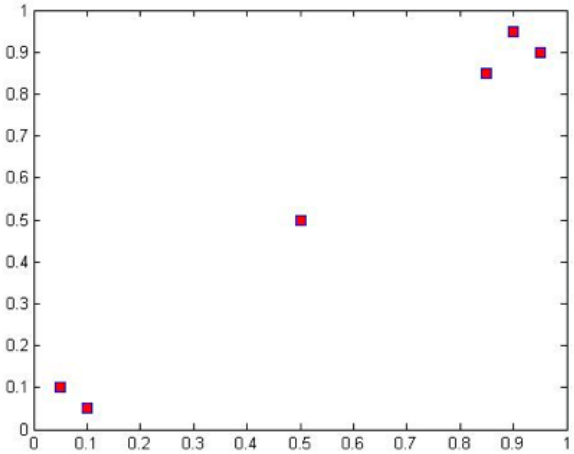
Let $\mathcal{Y} = \{x, x + \sigma y_1, \ldots, x + \sigma y_{n+n(n+1)/2}\}$ be set of interpolation points such that $\max_i \|y_i\| \leq 1$ and that $M_{\mathcal{Y}}$ is nonsingular. Suppose that the function f has L-Lipschitz continuous Hessians. Then,

$$\|\nabla m(x) - \nabla f(x)\| \leq \mathcal{O}\left(\|M_{\mathcal{Y}}^{-1}\|_2 n\sigma^2 L\right).$$

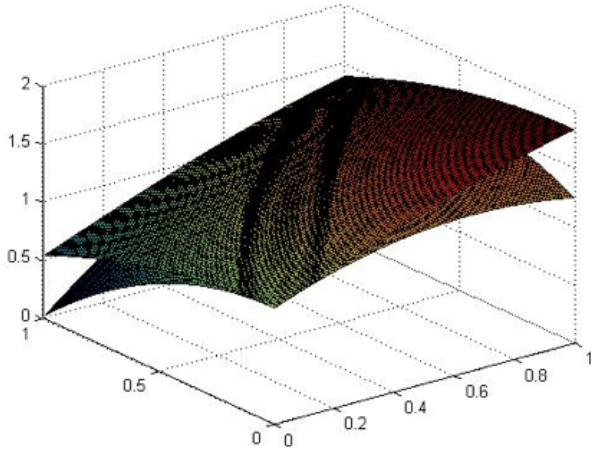
$$\|\nabla^2 m(x) - \nabla^2 f(x)\| \leq \mathcal{O}\left(\|M_{\mathcal{Y}}^{-1}\|_2 n\sigma L\right).$$

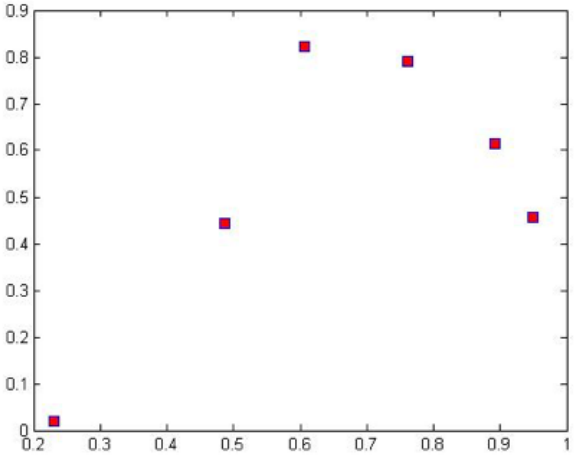
- Cost: $\mathcal{O}(n^6)$

Interpolation model quality

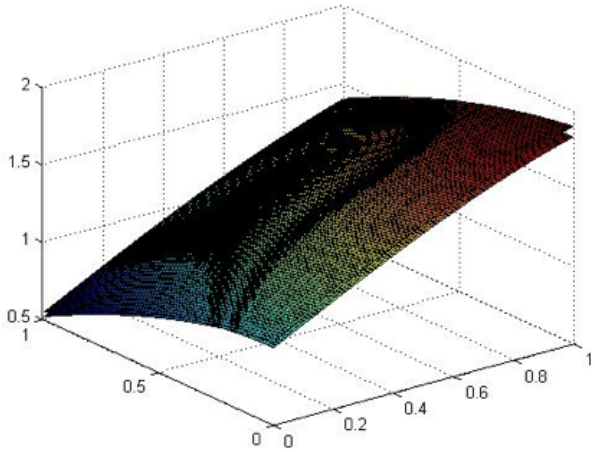


$\Rightarrow$

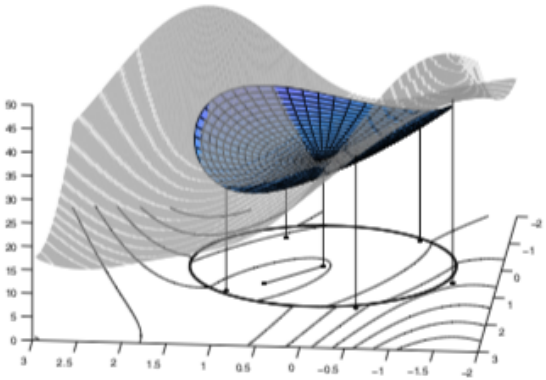


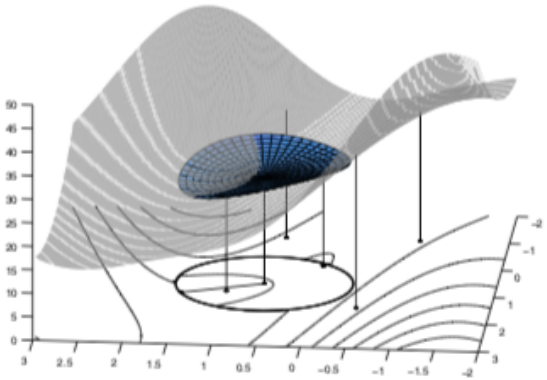


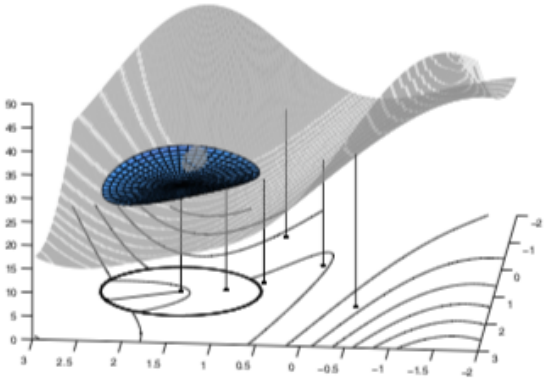
$\Rightarrow$

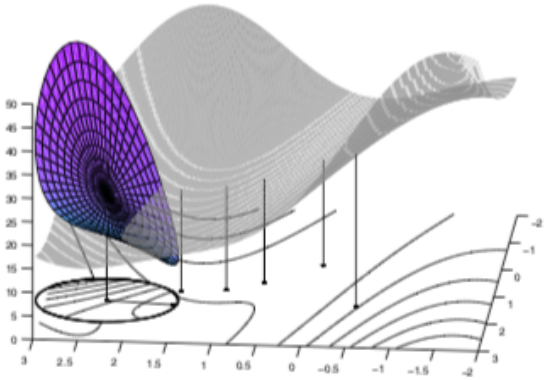


Model deterioration









Some conclusions so far

- Interpolation models allow for old points to be reused and hence are very economical in terms of samples.
- Linear algebra is expensive and more importantly can be ill-conditioned.
- Can improve lin. alg. cost and conditioning by using pre-designed sample sets, but it is more expensive in terms of samples (e.g. FD needs n samples per gradient estimate).
- What alternatives are there?

Gaussian Smoothing

$$F(x) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} f(x + \sigma\epsilon) = \int_{\mathbb{R}^n} f(x + \sigma\epsilon)\pi(\epsilon|0, I)d\epsilon$$

- $\pi(y|x, \Sigma)$ is the pdf of $\mathcal{N}(x, \Sigma)$ evaluated at y
- $F(x)$ is a Gaussian smoothed approximation to $f(x)$

$$\nabla F(x) = \frac{1}{\sigma}\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0},I)} f(x + \sigma\epsilon)\epsilon$$

- Idea: Approximate $\nabla f(x)$ by a sample average approximation of $\nabla F(x)$

$$g(x) = \frac{1}{N\sigma}\sum_{i=1}^{N} f(x + \sigma\epsilon_i)\epsilon_i$$

Gaussian Smoothing

$$F(x) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} f(x + \sigma\epsilon) = \int_{\mathbb{R}^n} f(x + \sigma\epsilon)\pi(\epsilon|0, I)d\epsilon$$

- $\pi(y|x, \Sigma)$ is the pdf of $\mathcal{N}(x, \Sigma)$ evaluated at y
- $F(x)$ is a Gaussian smoothed approximation to $f(x)$

$$\nabla F(x) = \frac{1}{\sigma}\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0},I)} f(x + \sigma\epsilon)\epsilon$$

- Idea: Approximate $\nabla f(x)$ by a sample average approximation of $\nabla F(x)$

$$g(x) = \frac{1}{N\sigma}\sum_{i=1}^{N} f(x + \sigma\epsilon_i)\epsilon_i$$

- Issue: Variance $\to \infty$ as $\sigma \to 0$

Gaussian Smoothing

$$F(x) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} f(x + \sigma\epsilon) = \int_{\mathbb{R}^n} f(x + \sigma\epsilon)\pi(\epsilon|0, I)d\epsilon$$

- $\pi(y|x, \Sigma)$ is the pdf of $\mathcal{N}(x, \Sigma)$ evaluated at y
- $F(x)$ is a Gaussian smoothed approximation to $f(x)$

$$\nabla F(x) = \frac{1}{\sigma}\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0},I)}(f(x + \sigma\epsilon) - f(x))\epsilon$$

- Idea: Approximate $\nabla f(x)$ by a sample average approximation of $\nabla F(x)$

$$g(x) = \frac{1}{N\sigma}\sum_{i=1}^{N}(f(x + \sigma\epsilon_i) - f(x))\epsilon_i$$

Gaussian Smoothing

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

$N = 1$, theoretical analysis of convergence rates for convex problems

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. Technical Report arXiv:1703.03864, 2016.

used in reinforcement learning, no theory, N is large

Alvaro Maggiar, Andreas Wächter, Irina S Dolinskaya, and Jeremy Staum. A derivative-free trust-region algorithm for the optimization of functions smoothed via gaussian convolution using adaptive multiple importance sampling. *SIAM Journal on Optimization*, 28(2):1478–1507, 2018.

uses interpolation on top of sample average approximation

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

uniform distribution on a ball for online learning

Maryam Fazel, Rong Ge, Sham M Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. *arXiv preprint arXiv:1801.05039*, 2018.

uniform distribution on a ball for model-free LQR

Analysis of Variance for Gaussian Smoothing

$$\|g(x) - \nabla f(x)\| \leq \underbrace{\|g(x) - \nabla F(x)\|}_{\text{sample average error}} + \underbrace{\|\nabla F(x) - \nabla f(x)\|}_{\text{smoothing error}}$$
$$\leq r + \sqrt{n}\sigma L$$

Theorem [Berahas, Cao, S., 2019]

Suppose that the function $f(x)$ has L-Lipschitz continuous gradients. Let $g(x)$ denote the GSG approximation to $\nabla f(x)$. If

$$N \geq \frac{1}{\delta r^2}\left(3n\|\nabla f(x)\|^2 + \frac{n(n^2+6n+8)L^2\sigma^2}{4}\right).$$

then,

$$\|g(x) - \nabla f(x)\| \leq r + \sqrt{n}\sigma L.$$

with probability at least $1 - \delta$.

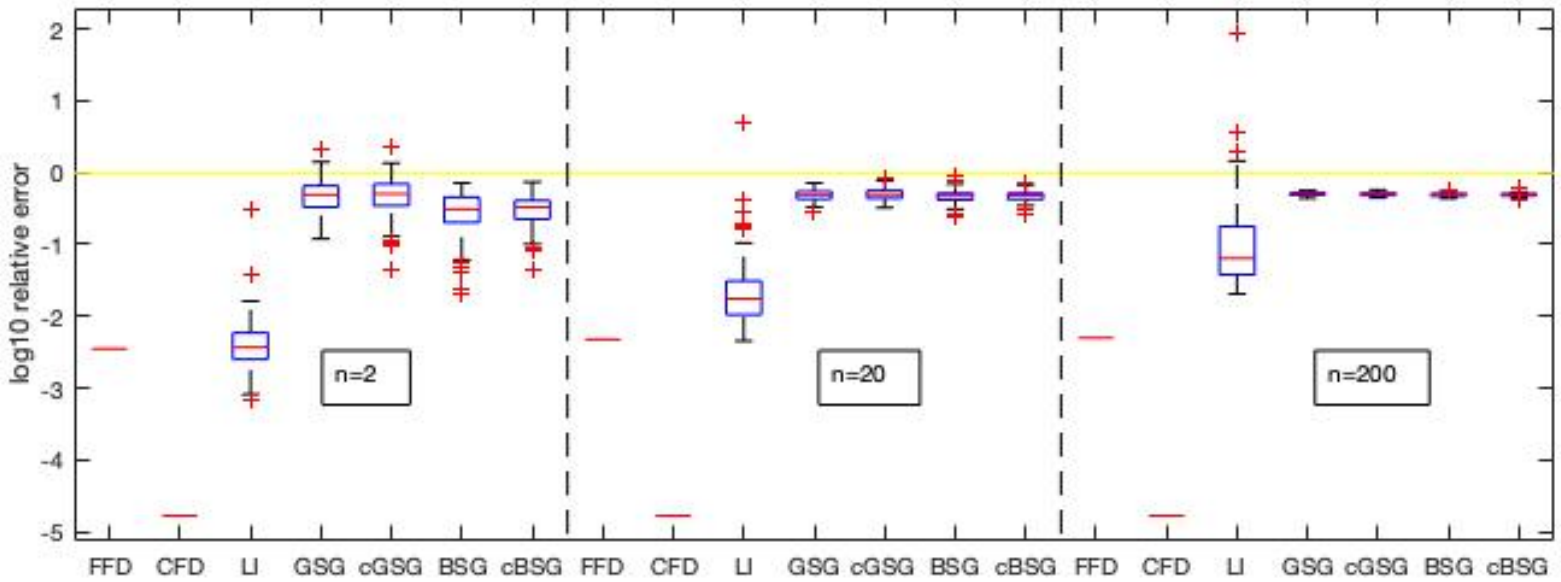
Essentially $N \sim 3n$

Gradient Approximation Accuracy

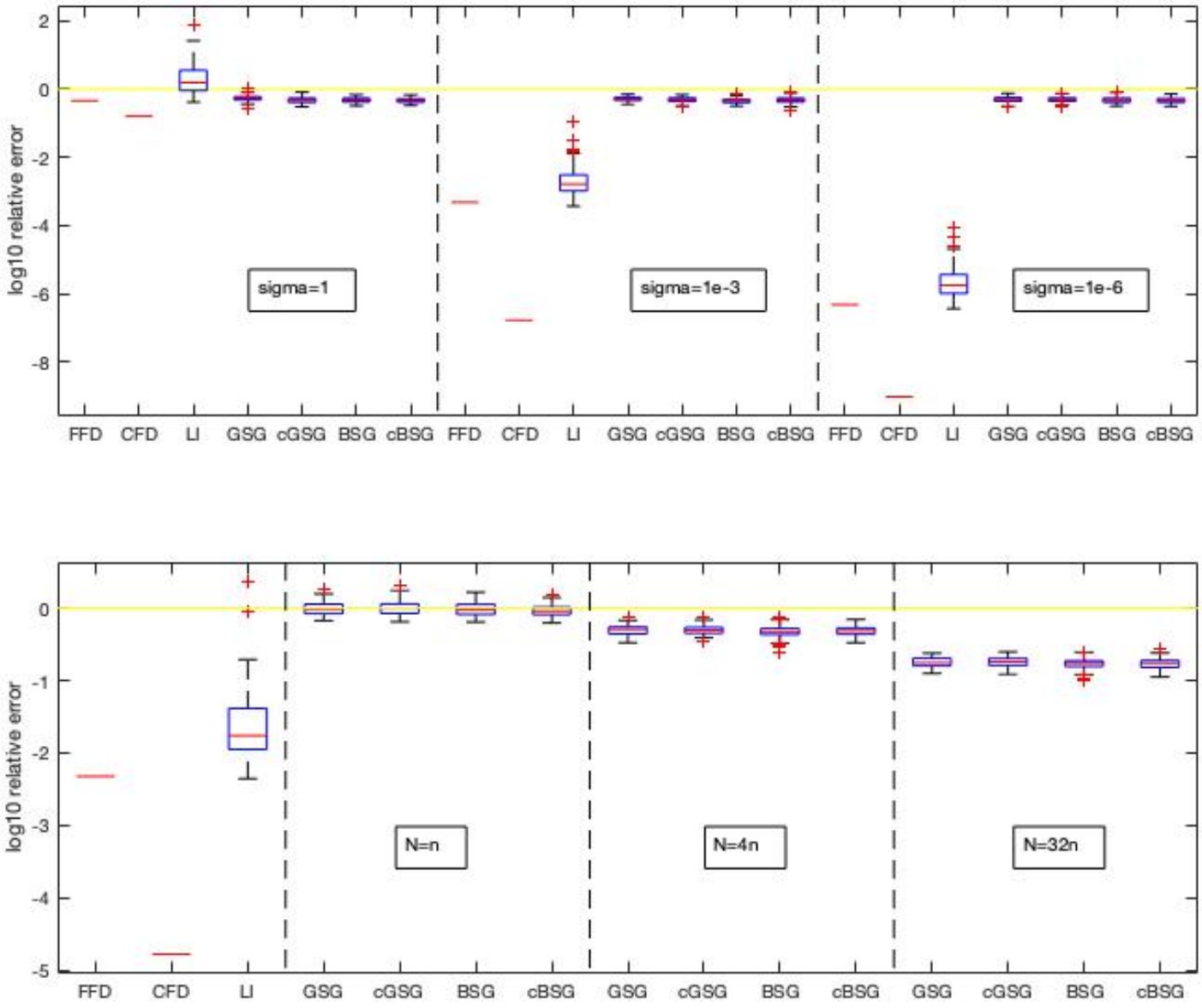
numerical experiment setup and results:

$$f(x) = \left(\sum_{i=1}^{n/2} M \sin(x_{2i-1}) + \cos(x_{2i}) \right) + \frac{L-M}{2n} x^T 1_{n\times n} x,$$

which has $\|\nabla f(0)\| = \sqrt{\frac{n}{2}} M$. We use $n = 20$, $M = 1$, $L = 2$, $\sigma = 0.01$, and $N = 4n$ for the smoothing methods.

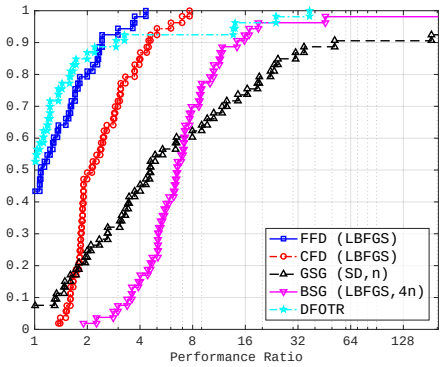


Gradient Approximation Accuracy

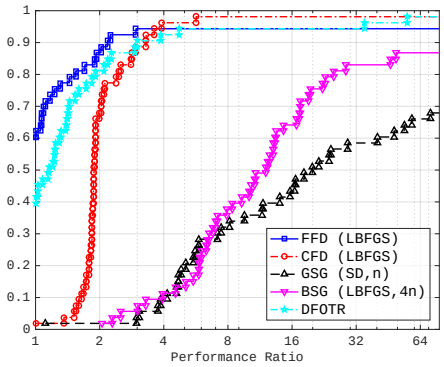


Algorithm Performance

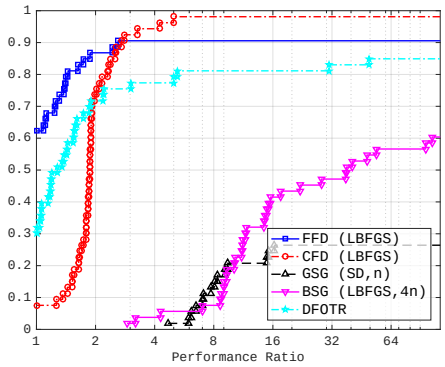
Moré&Wild problems set (53 smooth problems)



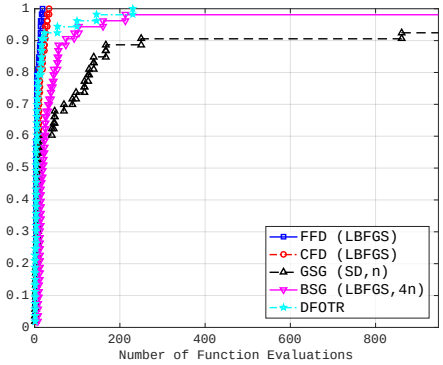
(a) $\tau = 10^{-1}$



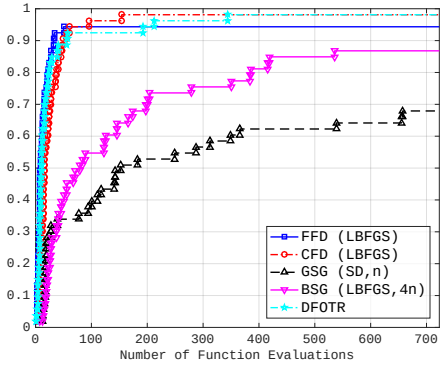
(b) $\tau = 10^{-3}$



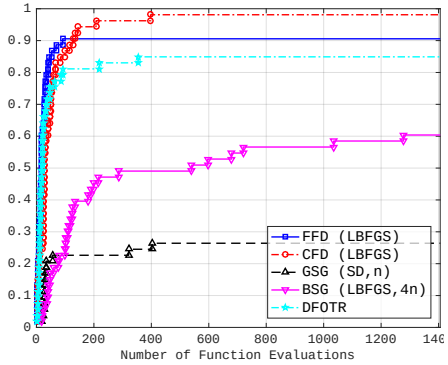
(c) $\tau = 10^{-5}$



(d) $\tau = 10^{-1}$



(e) $\tau = 10^{-3}$

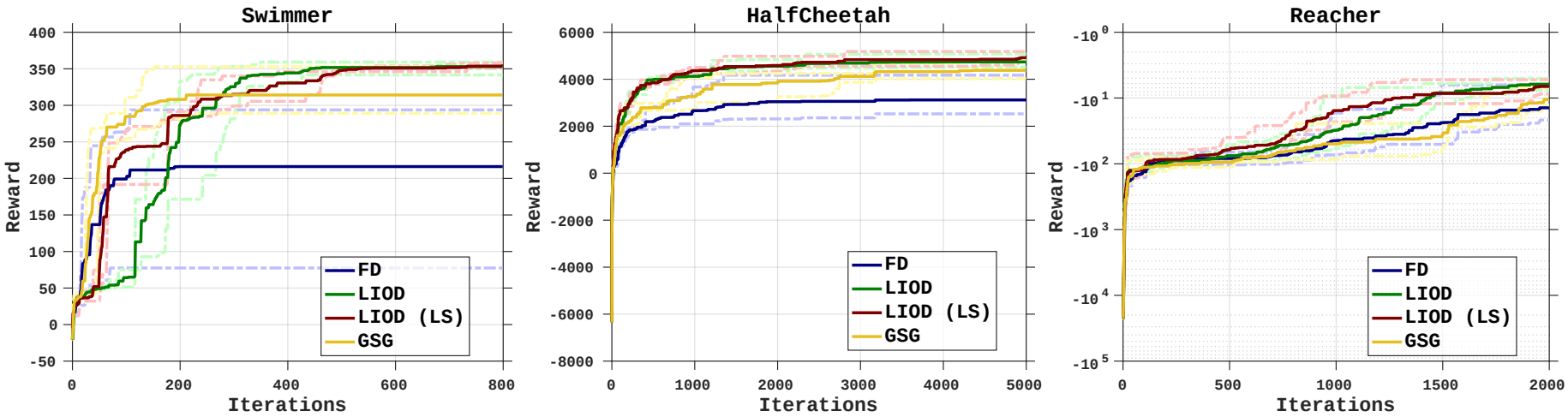


(f) $\tau = 10^{-5}$

Performance and data profiles for best variant of each method. Top row: performance profiles; Bottom row: data profiles.

Algorithm Performance

FD = forward finite difference
LIOD = linear interpolation of orthogonal directions
LS = (backtracking) linear search
GSG = Gaussian smooth gradient



Reinforcement learning tasks: `Swimmer` (left), `HalfCheetah` (center), `Reacher` (right).

Conclusions

- Model based derivative free methods are efficient and theoretically sound
- Select the type of models according to application but make sure theory applies
- Use randomization only when necessary, as it can slow down convergence

Conclusions

- Model based derivative free methods are efficient and theoretically sound
- Select the type of models according to application but make sure theory applies
- Use randomization only when necessary, as it can slow down convergence

- Andrew R Conn, Katya Scheinberg, and Luis N Vicente. **Introduction to Derivative-free Optimization** MPS-SIAM Optimization series. SIAM, Philadelphia, USA, 2008.
- Albert Berahas, Liyuan Cao, Krzyzstof Choromanski, Katya Scheinberg. *A theoretical and empirical comparison of gradient approximations in derivative-free optimization*, arXiv preprint arXiv:1904.11585,1905.01332, 2019.
- Jeffrey Larson, Matt Menickelly, and Stefan M Wild. *Derivative-free optimization methods* arXiv preprint arXiv:1904.11585, 2019.

Thank you!