Using random models and random steps in unconstrained optimization

Katya Scheinberg

Lehigh University

(based on work with A. Bandeira and L.N. Vicente, X. Tang and also with C. Cartis)

Difficult unconstrained optimization

- Unconstrained optimization problem

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

- Function f can be
  - computed by a black box or
  - very large scale or
  - stochastic
- $f \in C^1$ or C^2 for now.
- Derivatives are often inaccurate or impossible/ expensive to compute.

Trust region framework

Original trust region framework

1. Compute a potential step

$m_k(x) = f(x^k) + g_k^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top H_k (x - x_k)$ in $B(x_k, \Delta_k)$.
(g_k, H_k are some gradient and Hessian approximations at x^k.)
Compute a point x^+ which minimizes (reduces) $m(x)$ in $B(x^k, \Delta_k)$.

2. Check decrease

Compute $f(x^+)$ and check if f is reduced comparably to m by x^+.

3. Successful step

If yes $x^{k+1} := x^+$, set $\Delta_{k+1} \geq \Delta_k$.
Generate new g_{k+1}, H_{k+1}.

4. Unsuccessful step

Otherwise, $x_{k+1} = x_k$, decrease Δ_k by the constant factor.
$g_{k+1} = g_k$, $H_{k+1} = H_k$.

Trust region framework in DFO

1. Compute a potential step

$m_k(x) = f(x^k) + g_k^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top H_k (x - x_k)$ in $B(x_k, \Delta_k)$.

(g_k, H_k are some gradient and Hessian approximations at x^k.)

Compute a point x^+ which minimizes (reduces) $m(x)$ in $B(x^k, \Delta_k)$.

2. Check decrease

Compute $f(x^+)$ and check if f is reduced comparably to m by x^+.

3. Successful step

If yes $x^{k+1} := x^+$, set $\Delta_{k+1} \geq \Delta_k$.

Generate new g_{k+1}, H_{k+1}.

4. Unsuccessful step

Otherwise, $x_{k+1} := x_k$.

Possibly decrease Δ_k by the constant factor.

Generate new g_{k+1}, H_{k+1}.

Trust region framework in DFO

1. Compute a potential step

$m_k(x) = f(x^k) + g_k^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top H_k (x - x_k)$ in $B(x_k, \Delta_k)$.

(g_k, H_k are some gradient and Hessian approximations at x^k.)

Compute a point x^+ which minimizes (reduces) $m(x)$ in $B(x^k, \Delta_k)$.

2. Check decrease

Compute $f(x^+)$ and check if f is reduced comparably to m by x^+.

3. Successful step

If yes $x^{k+1} := x^+$, set $\Delta_{k+1} \geq \Delta_k$.

Generate new g_{k+1}, H_{k+1}.

4. Unsuccessful step

Otherwise, $x_{k+1} := x_k$.

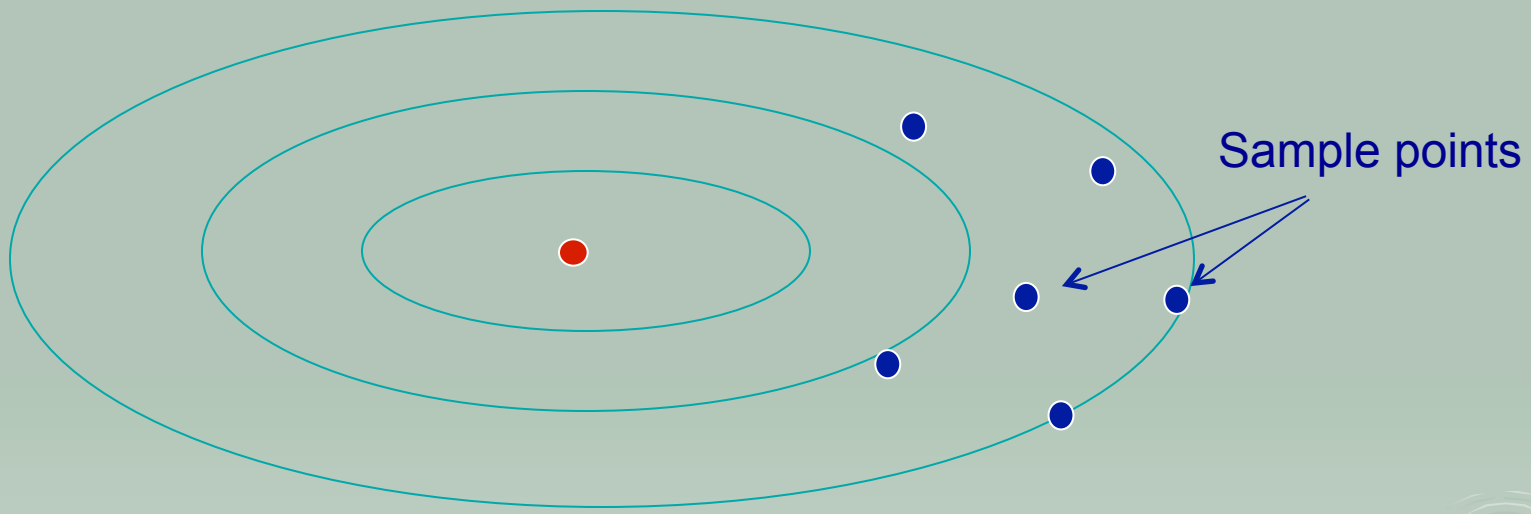
Possibly decrease Δ_k by the constant factor.

Generate new g_{k+1}, H_{k+1}.

"Refreshing" models at each iteration allows the occasional use of really bad models.

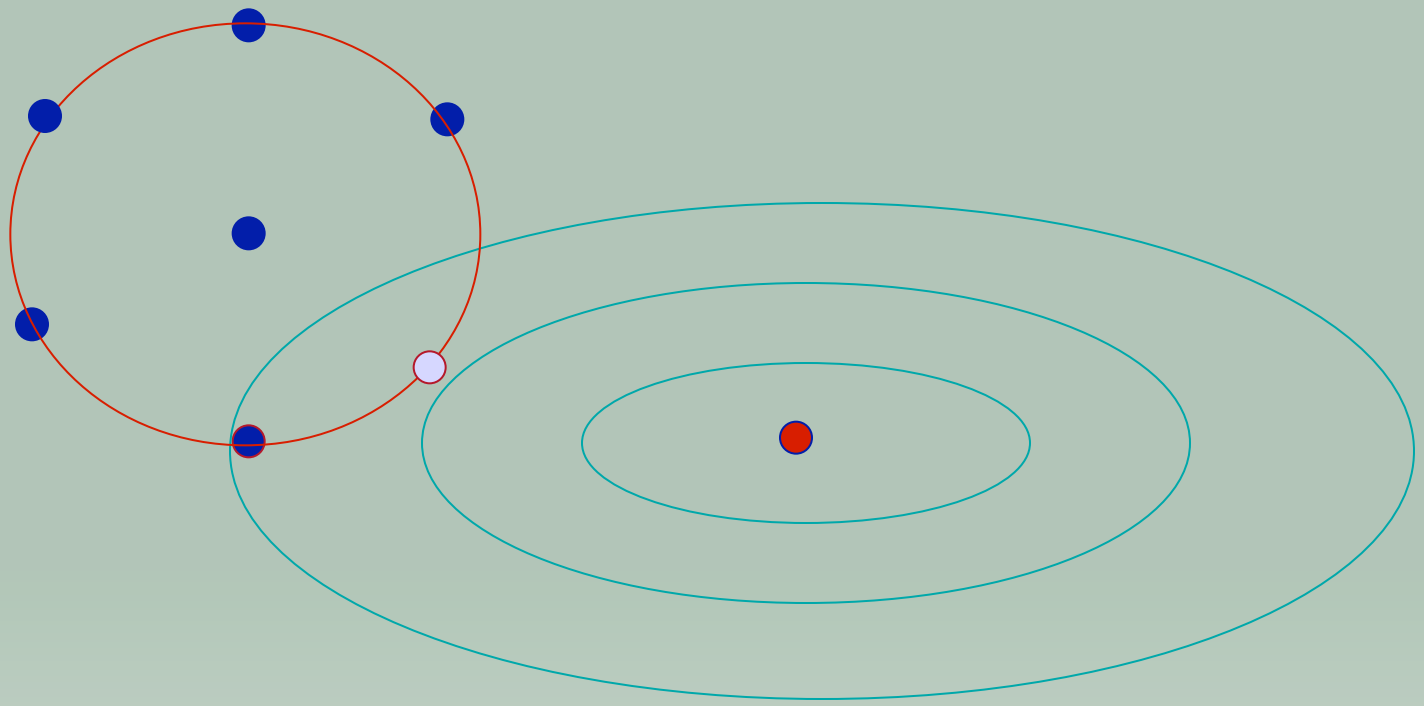
Motivating example with interpolation or regression models

Sampling the black box function



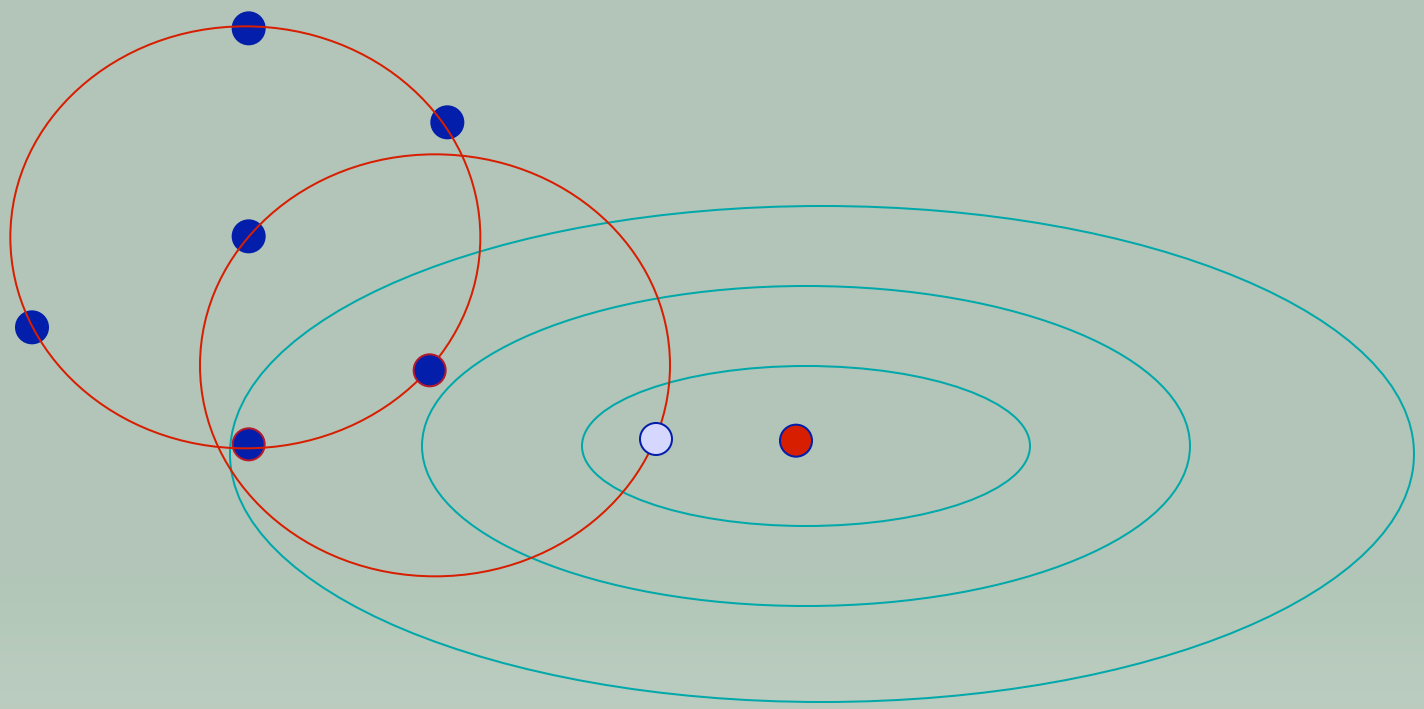
How to choose and to use the sample points and the functions values defines different DFO methods

Model based trust region methods



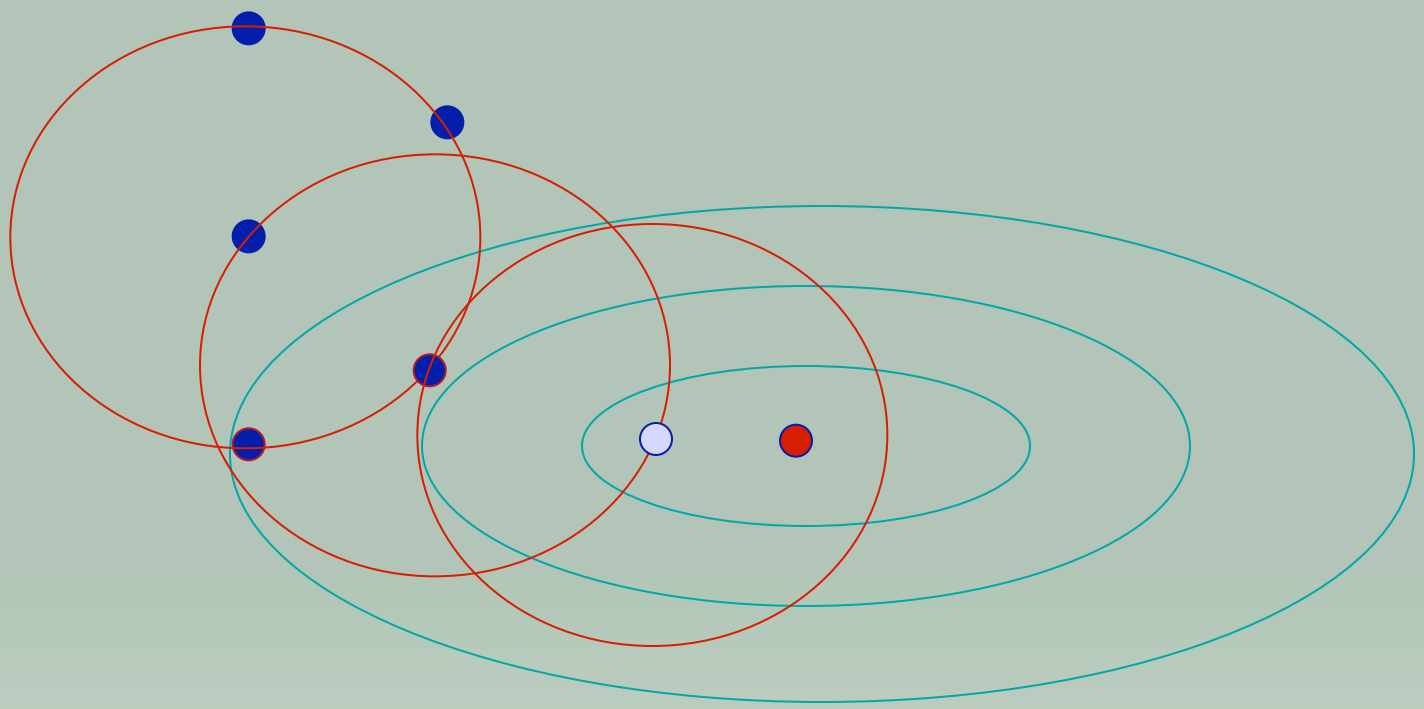
Powell, Conn, S. Toint, Vicente, Wild, etc.

Model based trust region methods



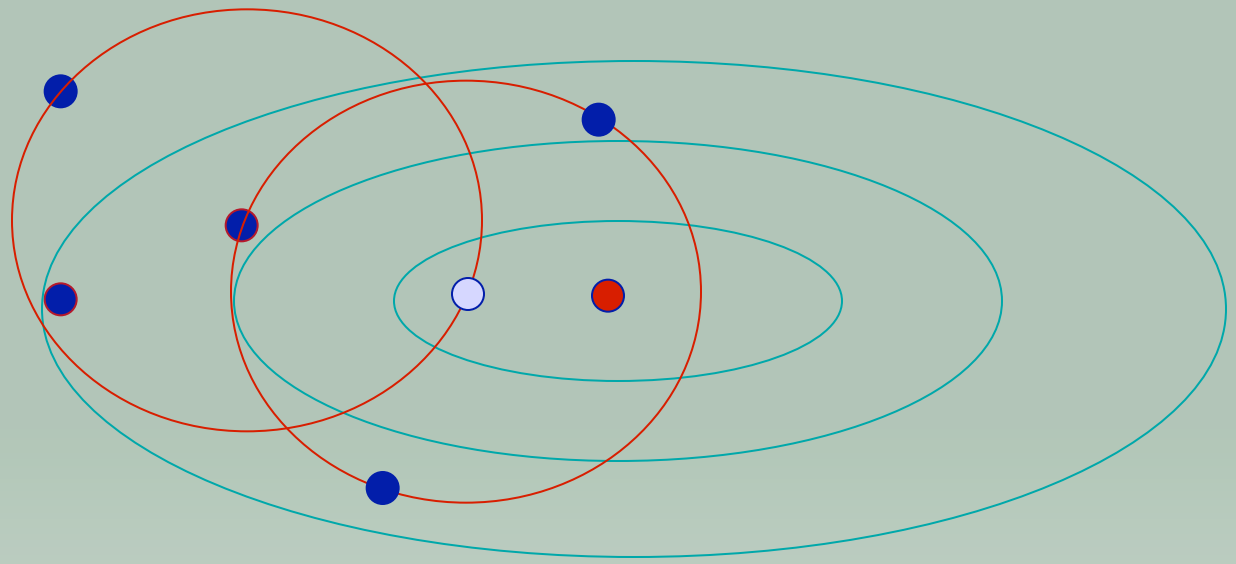
Powell, Conn, S. Toint, Vicente, Wild, etc.

Model based trust region methods



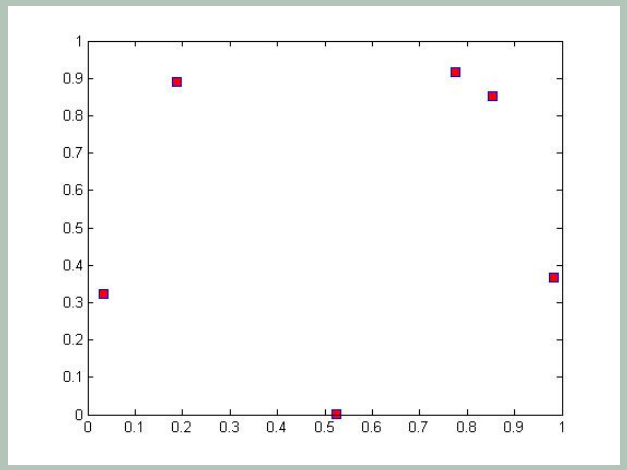
Powell, Conn, S. Toint, Vicente, Wild, etc.

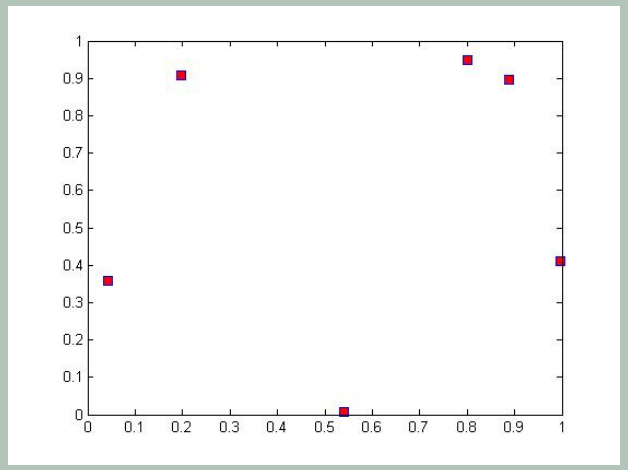
Model Based trust region methods

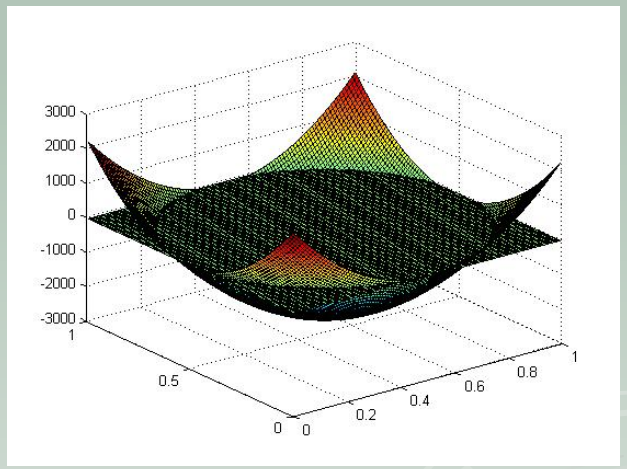


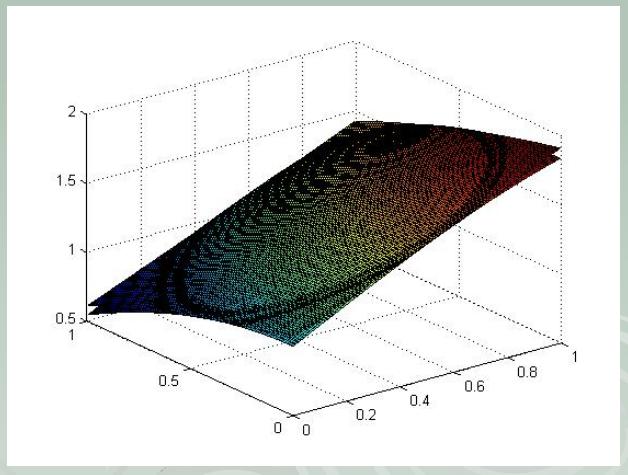
At each iteration a model is defined by the current sample set of points.

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









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

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

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

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

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

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

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

For deterministic methods convergence theory requires the model to be “good” (i.e. κ-fully linear) before trust region size is reduced.

For deterministic methods convergence theory requires the model to be "good" (i.e. κ-fully linear) before trust region size is reduced.

This means that some model quality checks or guarantees are needed, which can be expensive.

Line search framework

Traditional line search algorithm

1.Computing direction

$m_k(x) = f(x^k) + g_k^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top H_k (x - x_k)$

(g_k, H_k are approximations of gradient and Hessian (p.d).)

Compute a direction $d^k \approx -H_k^{-1} g_k$.

2.Compute the step and check decrease

Given a step size α_k, compute $f(x^k + \alpha_k d^k)$ and check if f is reduced sufficiently at $x^k + \alpha_k d^k$.

3. Successful step

If yes $x^{k+1} := x^k + \alpha_k d^k$ and $\alpha_{k+1} \geq \alpha_k$.

Gerate new g_{k+1} and H_{k+1}.

4.Unsuccessful step

Otherwise, $x^{k+1} := x^k$, $\alpha_{k+1} = \gamma \alpha_k$, $\gamma < 1$

$g_{k+1} = g_k$, $H_{k+1} = H_k$, $d^{k+1} = d^k$.

Modified "line search" algorithm

1.Computing direction

$m_k(x) = f(x^k) + g_k^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top H_k (x - x_k)$

(g_k, H_k are approximations of gradient and Hessian (p.d).)

Compute a direction $d^k \approx -H_k^{-1} g_k$.

2.Compute the step and check decrease

Given a step size α_k, compute $f(x^k + \alpha_k d^k)$ and check if f is reduced sufficiently at $x^k + \alpha_k d^k$.

3. Successful step

If yes $x^{k+1} := x^k + \alpha_k d^k$ and $\alpha_{k+1} \geq \alpha_k$.

Generate new g_{k+1} and H_{k+1}.

4.Unsuccessful step

Otherwise, $x^{k+1} := x^k$, $\alpha_{k+1} = \gamma \alpha_k$, $\gamma < 1$

Generate new g_{k+1} and H_{k+1} and/or d^{k+1}.

Modified "line search" algorithm

1.Computing direction

$m_k(x) = f(x^k) + g_k^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top H_k (x - x_k)$

(g_k, H_k are approximations of gradient and Hessian (p.d).)

Compute a direction $d^k \approx -H_k^{-1} g_k$.

2.Compute the step and check decrease

Given a step size α_k, compute $f(x^k + \alpha_k d^k)$ and check if f is reduced sufficiently at $x^k + \alpha_k d^k$.

3. Successful step

If yes $x^{k+1} := x^k + \alpha_k d^k$ and $\alpha_{k+1} \geq \alpha_k$.

Generate new g_{k+1} and H_{k+1}.

4.Unsuccessful step

Otherwise, $x^{k+1} := x^k$, $\alpha_{k+1} = \gamma \alpha_k$, $\gamma < 1$

Generate new g_{k+1} and H_{k+1} and/or d^{k+1}.

Possibly more work, but allows for occasionally bad directions/models.

Motivating example with large scale sparse optimization

Motivation from large scale sparse optimization

$$\min_{x\in\Omega} f(x) + \lambda\|x\|_1$$

- f(x) is smooth, convex (not quadratic) and very large scale, often with dense Hessian.
- Very common examples: logistic regression and sparse inverse covariance selection. (Hsieh et al., Byrd et al., etc)
- Latest most efficient approaches build Lasso models:

$$m(x) = f(x^k) + \nabla f(x^k)^\top (y - x^k) + \frac{1}{2}(y - x^k)^\top H_k (y - x^k) + \lambda\|x\|_1$$

- Optimize the resulting Lasso subproblem approximately by coordinate descent of a first order method.

Approximate proximal Newton method:

1. Build a model of the objective by approximating the smooth part, *f(x)*, by a quadratic function around current iterate x^k.

$$f(x^k) + \nabla f(x^k)^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top H_k (x - x^k) + \lambda \|x\|$$

1. Approximately (by coordinate descent, FISTA or another method) optimize the resulting model to obtain a direction d^k

2. Evaluate the objective function $f(x+\alpha_k d^k)$.
   - If sufficient decrease has been achieved, *accept* as the new iterate, $x^{k+1}=x+\alpha_k d^k$
   - Otherwise, reduce α_k and repeat.

3. Return to the Step 1.

For convergence need to make sure that Lasso subproblem produces sufficiently accurate d^+ => expensive checking of KKT conditions.

Algorithms based on random models and directions

- We focus on properties of the models that are essential for convergence.
- We will assume that these properties are satisfied often enough in a random fashion, but we do not know when.
- Ensure that those properties are satisfied by models of interest.

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 $1-\delta$.

Random models/steps with good properties

- Interpolation and regression models based on random sample sets of points are (κ, δ)-fully-linear (quadratic).
- Sparse interpolation and reg. models based on smaller random sample sets are (κ, δ)-fully-linear (quadratic).
- Interpolation and reg. models of stochastic functions based on larger random sample sets are (κ, δ)-fully-linear (quadratic).
- Taylor models based on finite difference derivative evaluations with asynchronous faulty parallel function evaluations can be (κ, δ)-fully-linear (quadratic).
- Steps computed by optimizing Lasso subproblem using *k* steps of randomized coordinate descent are "good" with some probability (using results of Nesterov (2010) Richtarik and Takac (2011)).

Trust region framework with random models

1. Compute a potential step

$m_k(x) = f(x^k) + g_k^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top H_k (x - x_k)$ in $B(x_k, \Delta_k)$.

(g_k, H_k are such that $m_k(x)$ is (κ, δ)-fully-linear.)

Compute a point x^+ which minimizes (reduces) $m(x)$ in $B(x^k, \Delta_k)$.

2. Check decrease

Compute $f(x^+)$ and check if f is reduced comparably to m by x^+.

3. Successful step

If yes $x^{k+1} := x^+$,

$\Delta_{k+1} := \Delta_k/\gamma$, $\gamma < 1$, if Δ_k is small compared to $\|g^k\|$.

Generate new g_{k+1}, H_{k+1} such that $m_k(x)$ is (κ, δ)-fully-linear.

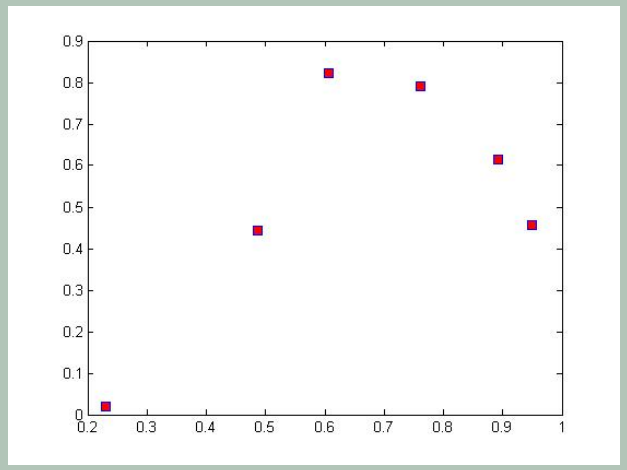
4. Unsuccessful step

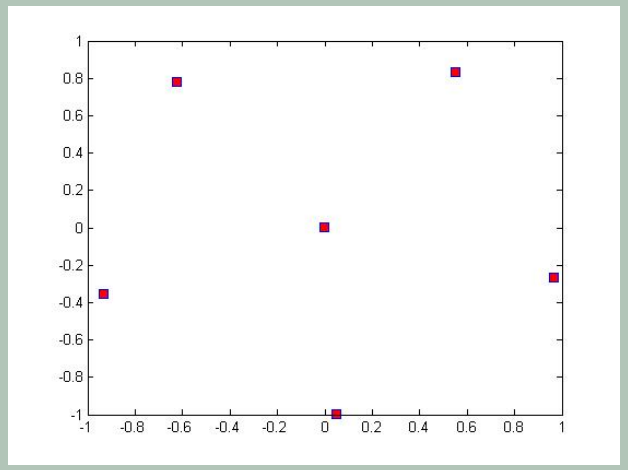
Otherwise, $x_{k+1} := x_k$.

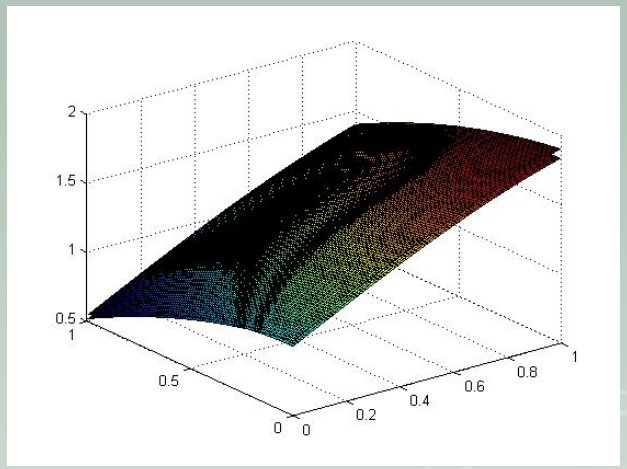
$\Delta_{k+1} := \gamma \Delta_k$.

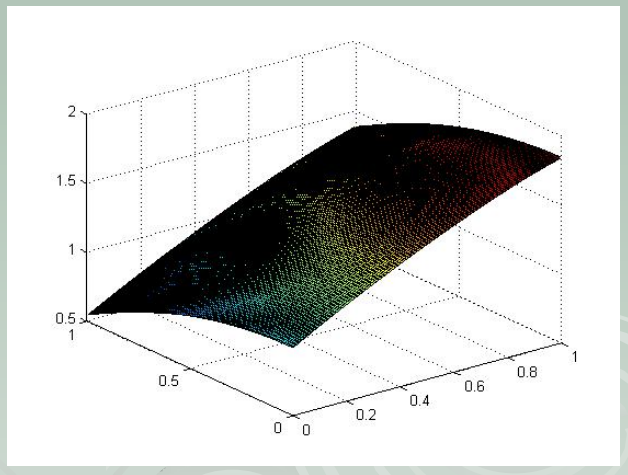
Generate new g_{k+1}, H_{k+1}. such that $m_{k+1}(x)$ is (κ, δ)-fully-linear.

Quality of the “best” model vs. random model









Convergence results for the basic TR framework

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

If models are fully quadratic w. p. 1-δ > 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

Bandeira, S. and Vicente, 2013

"Line search" algorithm for random models/directions

1.Computing direction

$m_k(x) = f(x^k) + g_k^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top H_k (x - x_k)$

(g_k, H_k are approximations of gradient and Hessian (p.d).)

Compute a direction $d^k \approx H_k^{-1} g_k$, (given α_k)

such that $\|d^k - H_k^{-1}\nabla f(x^k)\| \leq \kappa\alpha_k\|d^k\|$ with prob $1 - \delta$.

2.Compute the step and check decrease

Compute $f(x^k + \alpha_k d^k)$ and check if

f is reduced sufficiently at $x^k + \alpha_k d^k$.

3. Successful step

If yes $x^{k+1} := x^k + \alpha_k d^k$ and $\alpha_{k+1} = \alpha_k/\gamma$, $\gamma < 1$..

Generate new g_{k+1} and H_{k+1}.

4.Unsuccessful step

Otherwise, $x^{k+1} = x^k$, $\alpha_{k+1} = \gamma\alpha_k$, $\gamma < 1$

Generate new g_{k+1} and H_{k+1} and/or d^{k+1}

such that $\|d^k - H_k^{-1}\nabla f(x^k)\| \leq \kappa\alpha_k\|d^k\|$ with prob $1 - \delta$.

Approximate Newton method with randomized coordinate descent:

1. Build a model of the objective by approximating the smooth part, *f(x)*, by a quadratic function around current iterate x^k.

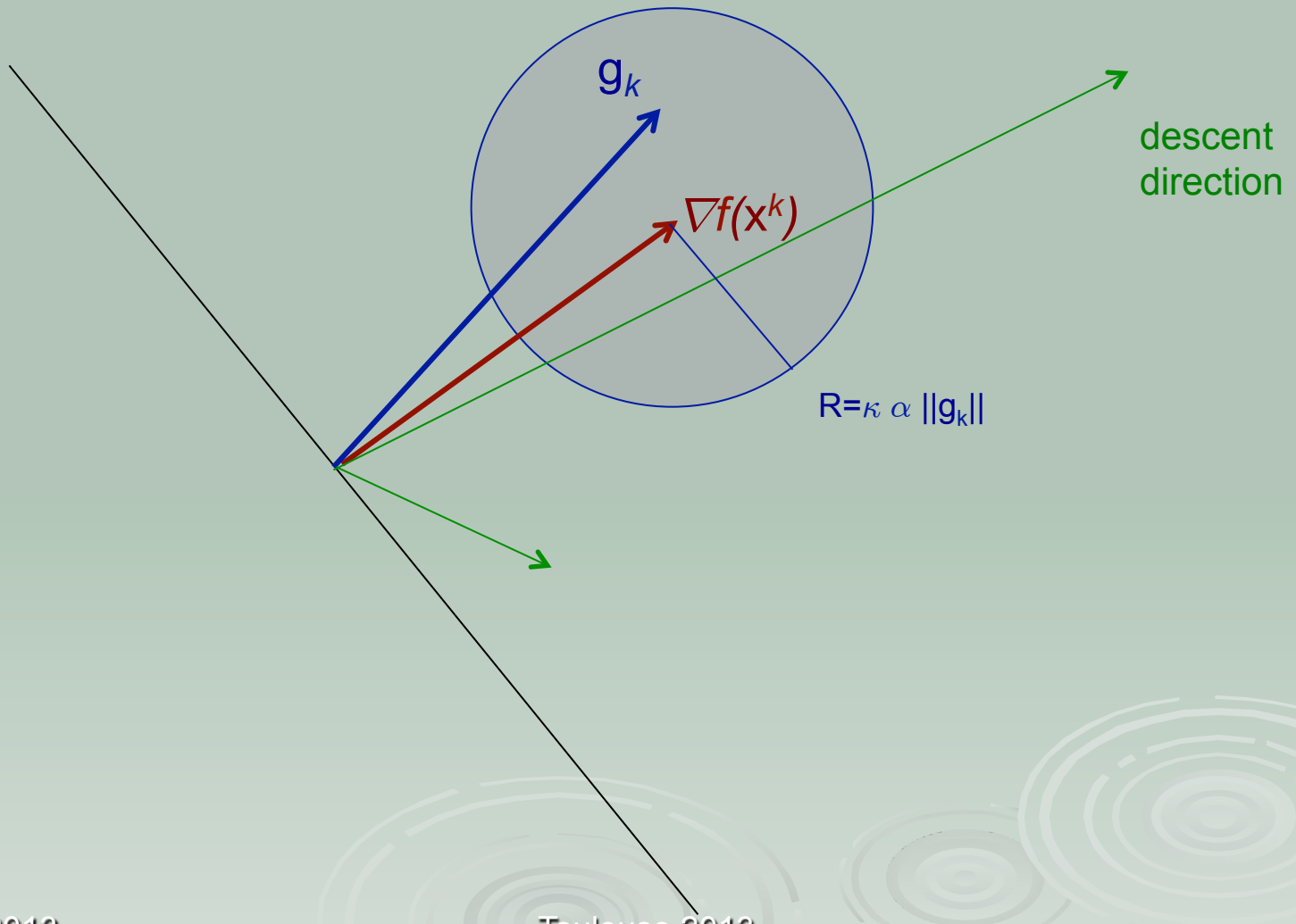
$$f(x^k) + \nabla f(x^k)^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top H_k (x - x^k) + \lambda \|x\|_1$$

1. Approximately (with sufficient accuracy, with probability p, by randomized coordinate descent) optimize the resulting model to obtain a direction d^k.

2. Evaluate the objective function $f(x+\alpha_k d^k)$.
   - If sufficient decrease has been achieved, *accept* as the new iterate, $x^{k+1}=x+\alpha_k d^k$
   - Otherwise, reduce α_k and repeat.

3. Return to the Step 1.

Random directions vs. random fully linear model gradients



Key observation for line search convergence

If g_k is a "good" direction 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 g_k is always a "good direction"

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

and

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

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

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

Convergence!!

Analysis of line search convergence

Assume g_k is ~~always~~ a "good direction" w.p. $\geq$ *1-δ*

$\Downarrow$

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

and

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

$\Downarrow$

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

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

no success $\alpha_{k+1} = \gamma^{-1} \alpha_k$ 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*:*

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

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

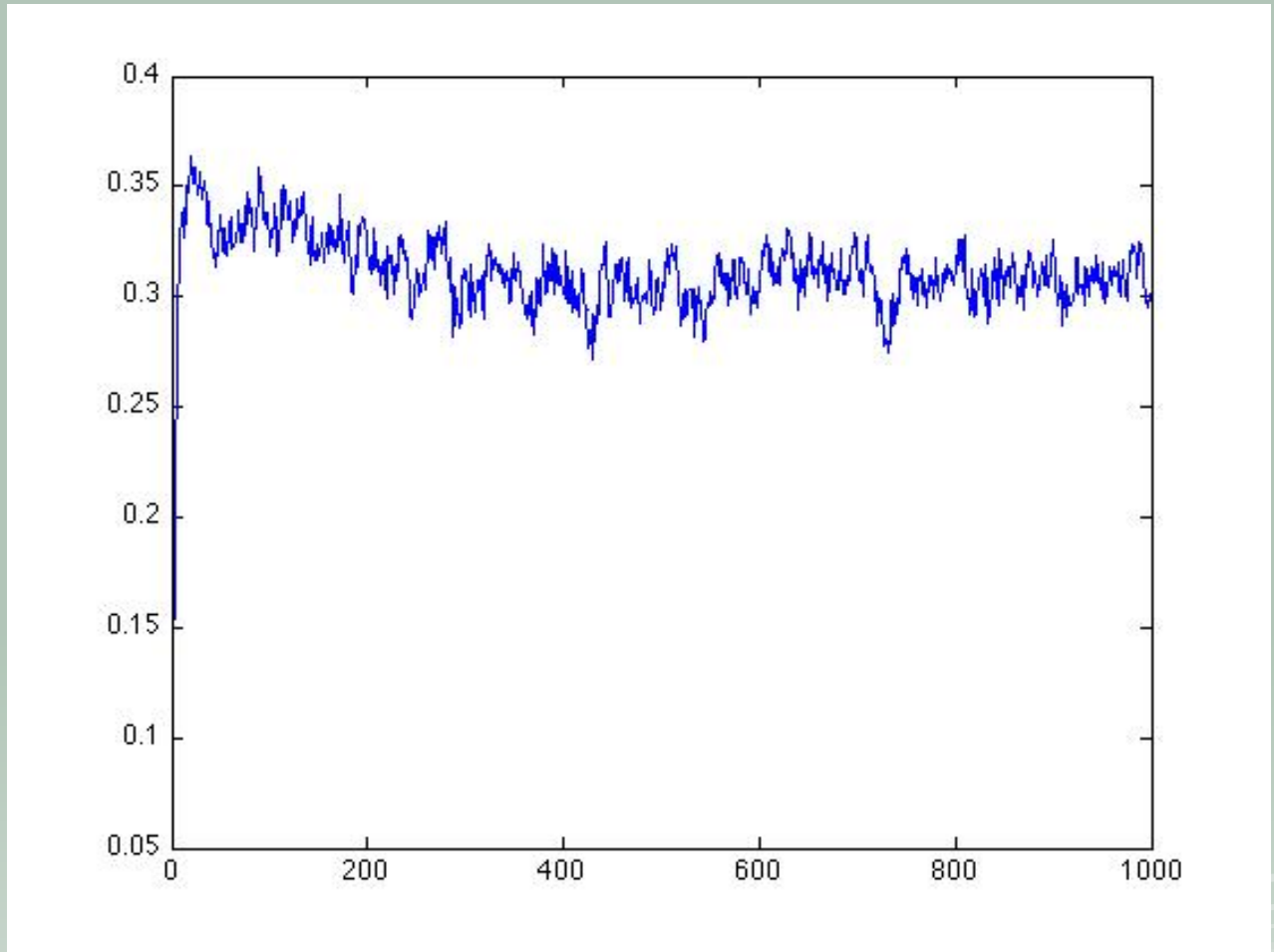
We observe that

$$\alpha_k \geq X_k$$

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

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.
- Convex optimization cases.
- Extend to composite optimization.
- Extend to stochastic programming.

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.

Polynomial models