Lecture 23

Derivative free optimization

- Fitting quadratic models.
- Nelder-mead method

General nonlinear optimization

- Nonlinear constraints
- Penalty methods
- Augmented Lagrangian methods

(Useful for Optimization on the Hypersphere)

Project

* Your reports should be independent of code...

Derivative Free Optimization

Computational Methods in Optimization

CS 520, Purdue

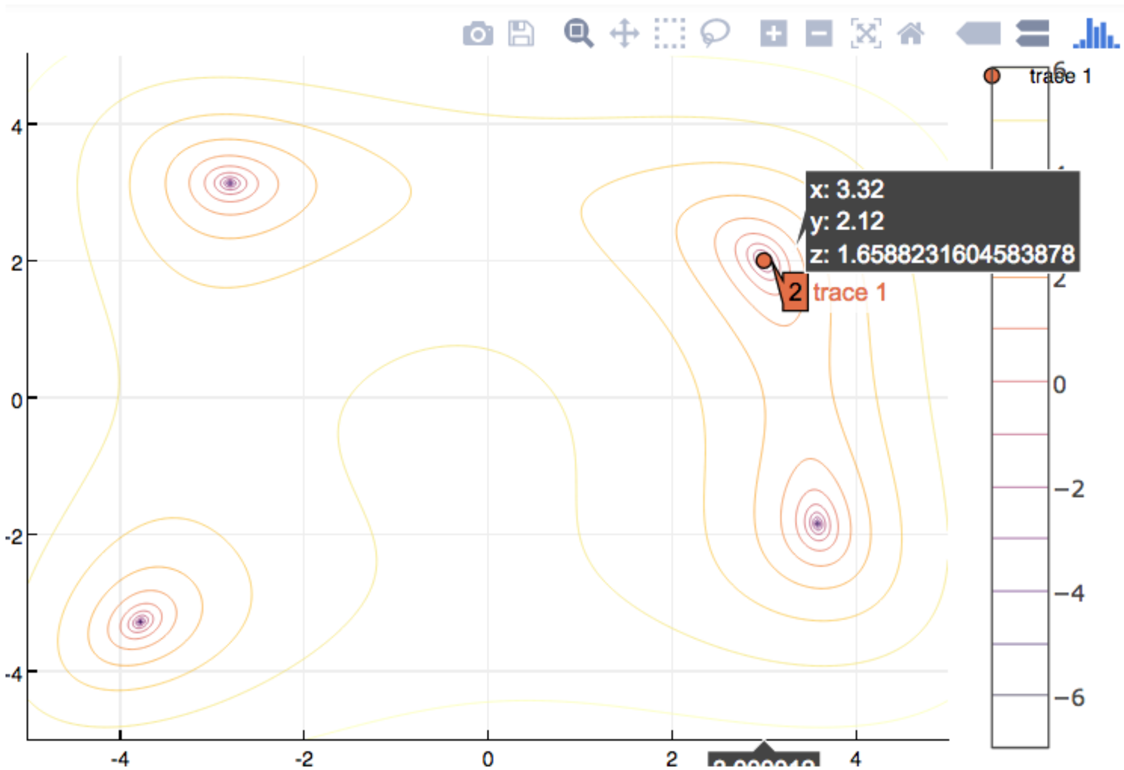
David F. Gleich

Purdue University

2025-04-08

A demo

```
f = x -> (x[1].^2 + x[2] - 11).^2 +
     (x[1] + x[2].^2 - 7).^2
sol = optimize(f, [0.0;0.0], NelderMead())
xs = sol.minimizer
```



Derivative-free optimization (DFO)

Chapter 9

Question

How would you do optimization without derivatives?

Solution 1

Use finite differences

$$f'(x) \approx \frac{1}{\gamma}(f(x+\gamma) - f(x))$$

Solution 1

Rough $\sqrt{\varepsilon}$.
Maybe scaled by
$f(x)$.
There is some literature
on picking this
adaptively.

Use finite differences

$$f'(x) \approx \frac{1}{\gamma}(f(x+\gamma) - f(x))$$

How to pick gamma?

How much work? → $O(n)$ function evaluations.

How to fit?

- least Squares

↳ Sample $\geq L$ parameters.

↳ min $\| m_k(y_\ell) - f(y_\ell) \|_2$.

Solution 2

Interpolate and update a quadratic model

(Section 9.2)

How many parameters

$$m_k(\mathbf{x}_k + \mathbf{p}) = c + \mathbf{q}^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T\mathbf{G}\mathbf{p}$$

$1 + dn + \frac{(n+1)(n)}{2} = L$

$$m_k(\mathbf{y}_\ell) = f(\mathbf{y}_\ell) \quad 1 \leq \ell \leq \text{total parameters}$$

Then use a trust-region method.

Solution 2

- How to find *c*, *q*, and *G*?

  $O(n^2)$ parameters

  How to choose the point set y_l?

- How to update *c*, *q*, and *G*?

  Details of interpolation methods. See the book, or references.

Solution 2

- How to find *c*, *q*, and *G*? $O(n^2)$ parameters

Use interpolation condition to form an n^2 by n^2 linear system

$O(n^6)$ to solve

$O(n^4)$ to update

How to choose the point set y_l?

Solution 3

Fix a sequence of search directions that span R^n, and cycle among them

"p = " $\mathbf{e}_1, -\mathbf{e}_1, \ldots, \mathbf{e}_n, -\mathbf{e}_n, \mathbf{e}_1, -\mathbf{e}_1 \ldots, \mathbf{e}_n, \ldots$

$\pm\mathbf{e}_1, \ldots, \pm\mathbf{e}_n, \pm\mathbf{e}_{n-1}, \ldots, \pm\mathbf{e}_1, \ldots$

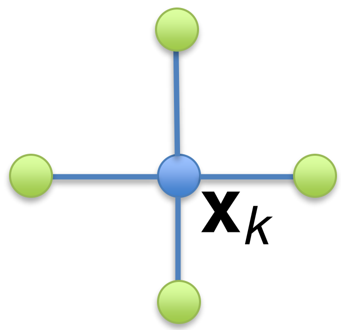
brutally slow in general

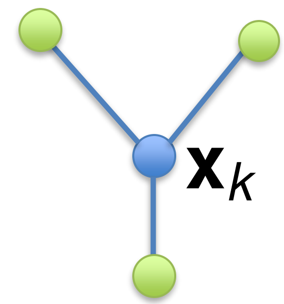
wickedly fast when applicable

(like a scalpel)

Solution 4

Pick a stencil around the current point





Example stencils in R^2

 $= \mathbf{x}_k + \gamma_k \mathbf{p}_k, \mathbf{p}_k \in \mathcal{D}_k$

Move to the best point if "good enough" (sufficient decrease)

Otherwise, reduce gamma and revaluate

Solution 4

Example stencils in R^2

$$= \mathbf{x}_k + \gamma_k \mathbf{p}_k, \mathbf{p}_k \in \mathcal{D}_k$$

We need $\beta_{\min} \leq \mathbf{p} \leq \beta_{\max}$

$$\min_{\mathbf{v} \in \mathbb{R}^n} \max_{\mathbf{p} \in \mathcal{D}_k} \frac{\mathbf{v}^T \mathbf{p}}{\|\mathbf{p}\| \, \|\mathbf{v}\|} \geq \delta$$

This ensures we can always get at least a delta projection on any gradient.

for $\mathbf{p} \in \mathcal{D}_k$ to satisfy Zoutendijk

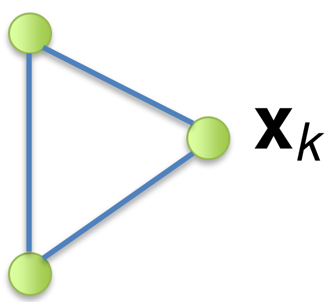
Solution 5 – Nelder-Mead

Consider a simplex of points

A simplex consists of n+1 non-colinear points

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \ldots \leq f(\mathbf{x}_{n+1})$$

We order the vertices by decreasing function value.

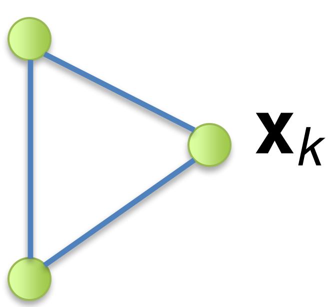


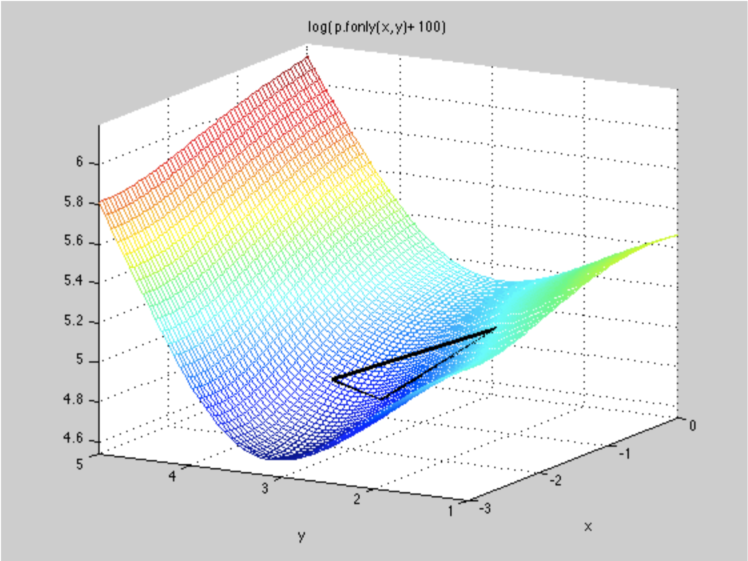
Such a simplex gives us a local "linear" model of our function!

Solution 5 – Nelder-Mead

Use the "slope" of the simplex to find a good direction

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \ldots \leq f(\mathbf{x}_{n+1})$$

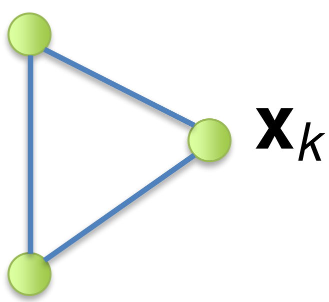




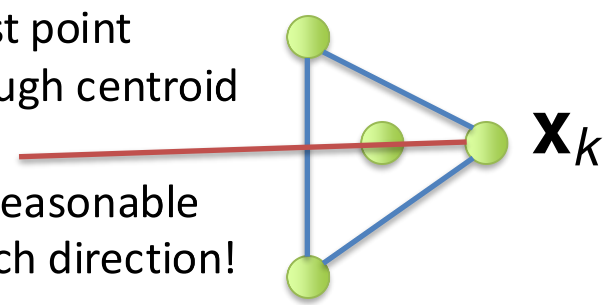
Solution 5 – Nelder-Mead

Use the "slope" of the simplex to find a good direction

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \ldots \leq f(\mathbf{x}_{n+1})$$



The line from the worst point through centroid

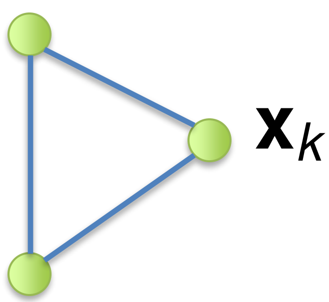


is a reasonable search direction!

Solution 5 – Nelder-Mead

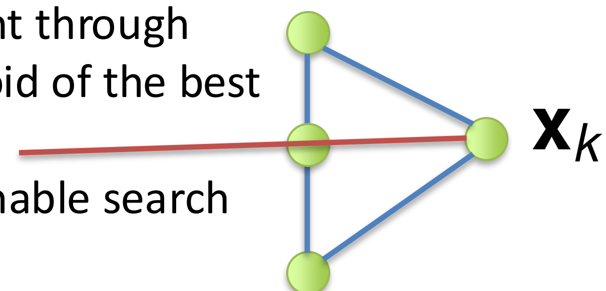
Use the "slope" of the simplex to find a good direction

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \ldots \leq f(\mathbf{x}_{n+1})$$



The line from the worst point through the centroid of the best

is a reasonable search direction!



Why can't we just use this direction and be done?

Nelder-Mead

Because we need a simplex at the next step too!

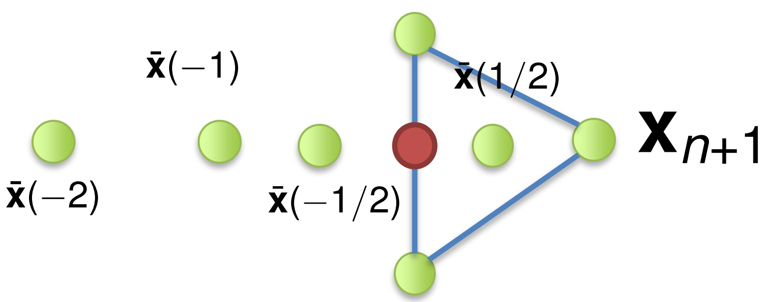
Can’t be too big.

Can’t be too small.

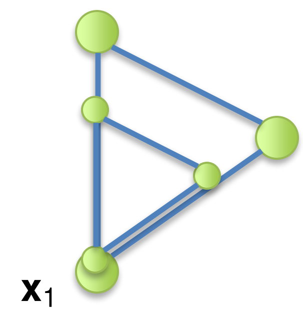
Solution 5 – Nelder-Mead

Use the slope of the simplex to find a good direction.

Reflecting the *worst point* in the simplex around the centroid of what's left to find a better point



Or shrink to the best point

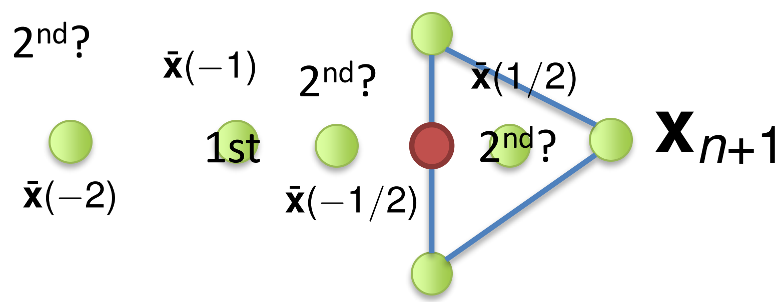


$$\bar{\mathbf{x}}(t) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i + t\left(\mathbf{x}_{n+1} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\right)$$

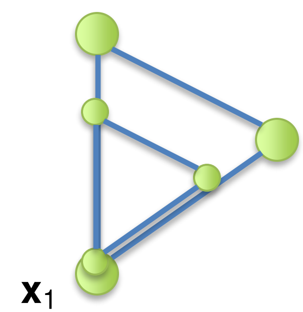
Solution 5 – Nelder-Mead

Use the slope of the simplex to find a good direction.

Reflecting the *worst point* in the simplex around the centroid of what's left to find a better point



Or shrink to the best point if no point is better



$$\bar{\mathbf{x}}(t) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i + t\left(\mathbf{x}_{n+1} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\right)$$

Quiz

Why is this better than pattern-search?

We get to reuse

Nonlinear programming

Computational Methods in Optimization
CS 520, Purdue

David F. Gleich
Purdue University
2025-04-08

Problems

Equality constrained

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \end{array}$$

Inequality constrained

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{d}(\mathbf{x}) \geq 0 \end{array}$$

General optimization

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \\ & \mathbf{d}(\mathbf{x}) \geq 0 \end{array}$$

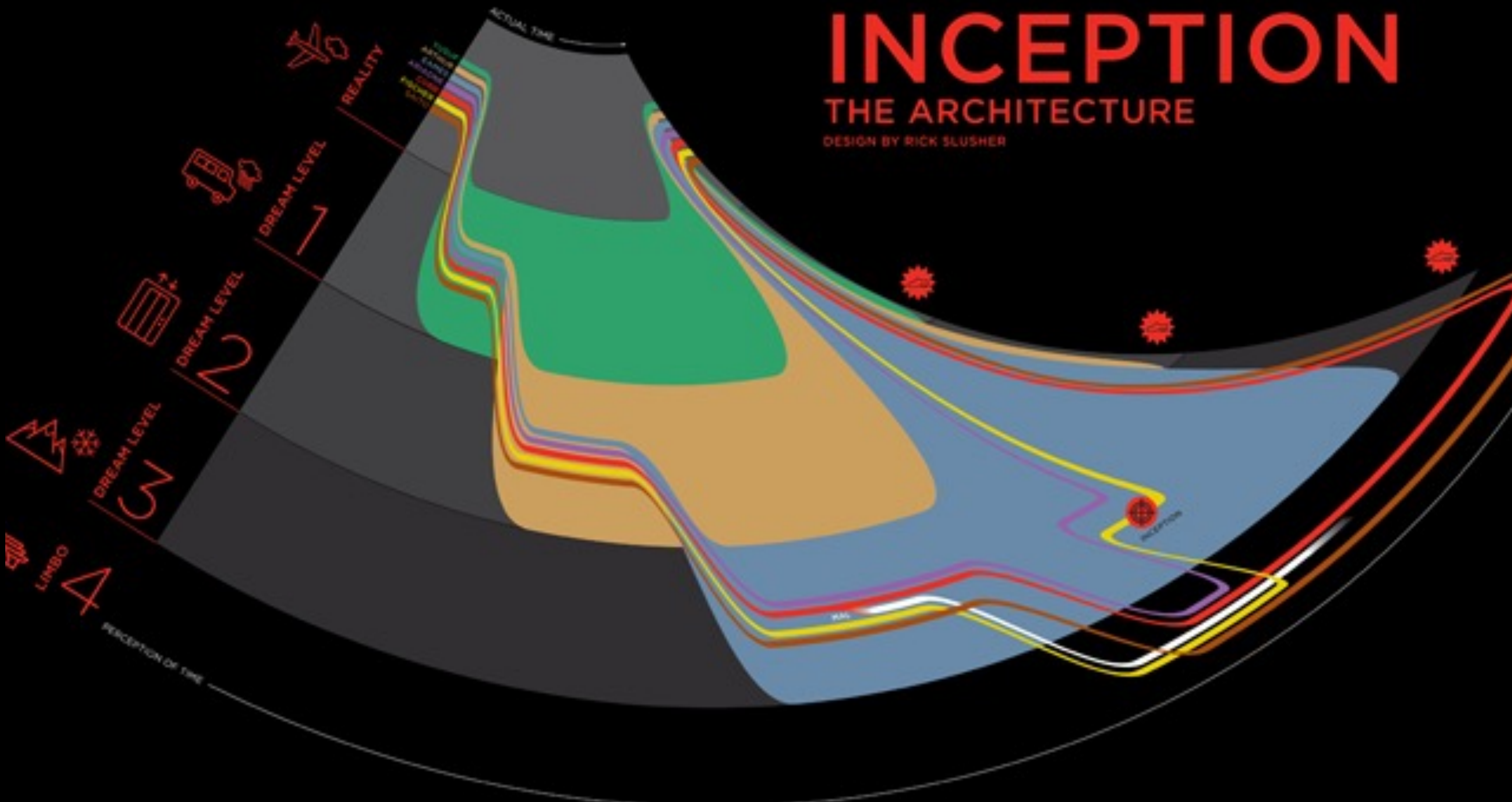
Overarching idea

Approximate these problems by something

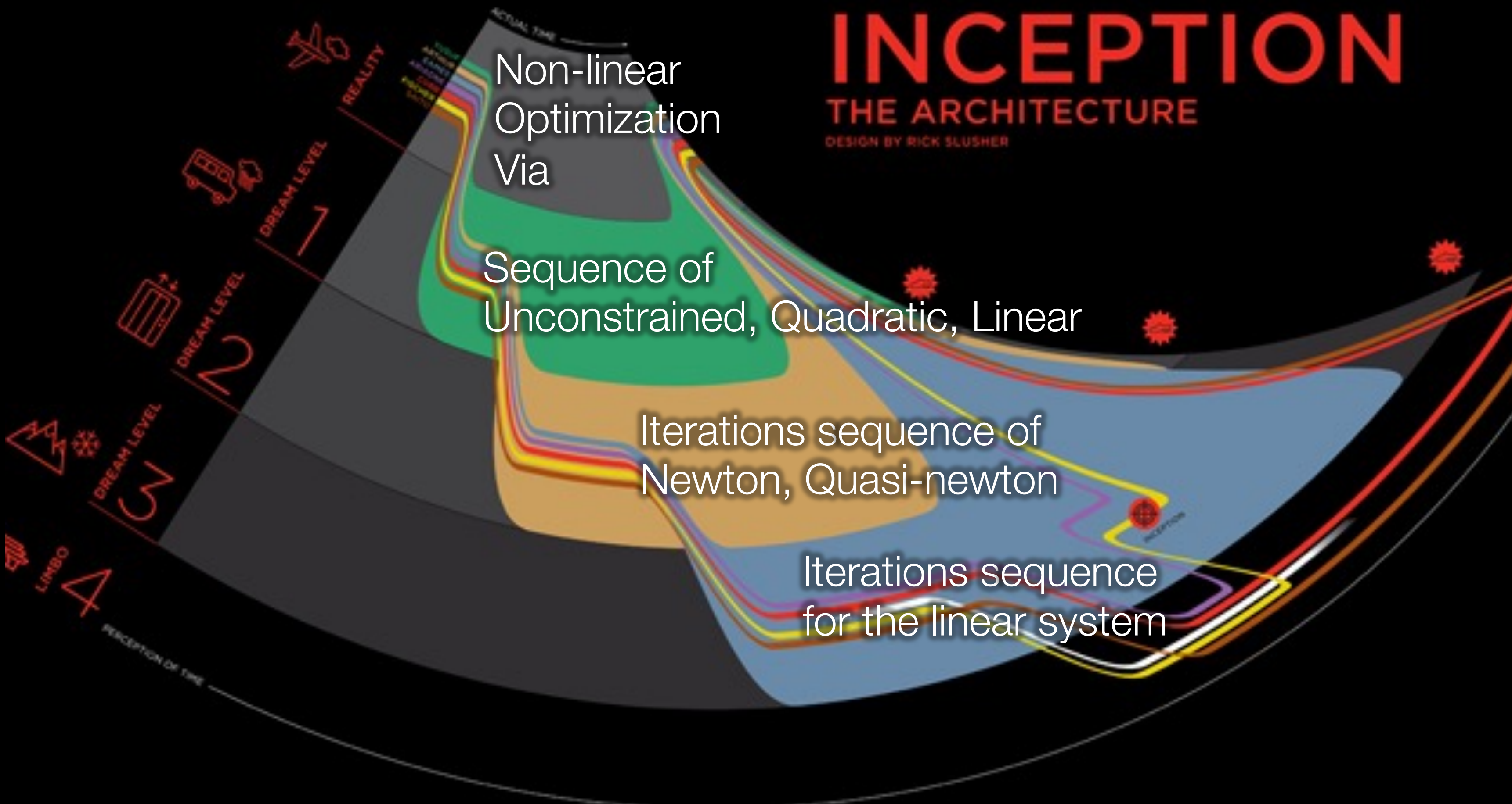
- easier, or
- more simple

And then solve a sequence of optimization problems.

Inception


INCEPTION
THE ARCHITECTURE
DESIGN BY RICK SLUSHER
ACTUAL TIME
REALITY
DREAM LEVEL 1
DREAM LEVEL 2
DREAM LEVEL 3
LIMBO 4
PERCEPTION OF TIME

Inception



INCEPTION
THE ARCHITECTURE
DESIGN BY RICK SLUSHER
REALITY
DREAM LEVEL 1
DREAM LEVEL 2
DREAM LEVEL 3
LIMBO 4
Non-linear
Optimization
Via
Sequence of
Unconstrained, Quadratic, Linear
Iterations sequence of
Newton, Quasi-newton
Iterations sequence
for the linear system

The equality constrained problem

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \end{array}$$

Chapter 17, Nocedal & Wright

Penalty Methods and
Augmented Lagrangians

A penalty method

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \end{array}$$

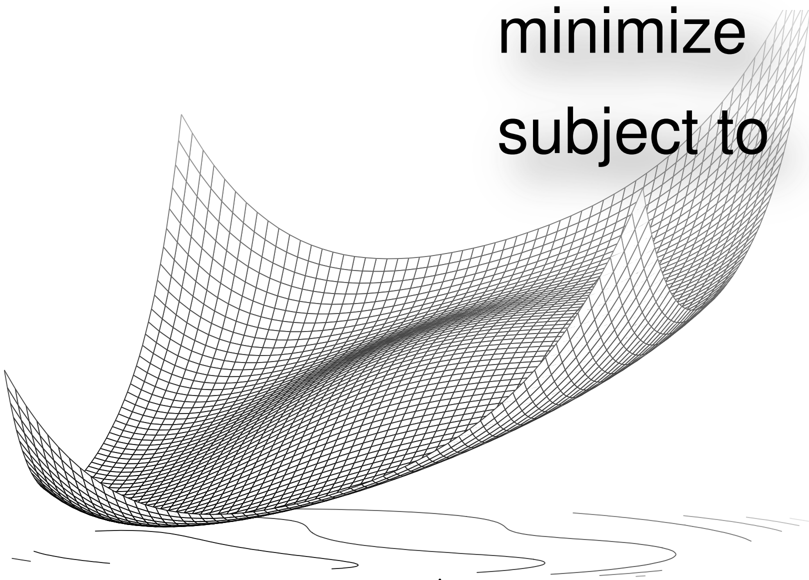
$$\text{minimize} \quad f(\mathbf{x}) + \mu \textstyle\sum_i c_i(\mathbf{x})^2 \quad \leftrightarrow \quad \text{minimize} \quad f(\mathbf{x}) + \mu \mathbf{c}(\mathbf{x})^T \mathbf{c}(\mathbf{x})$$

A penalty method

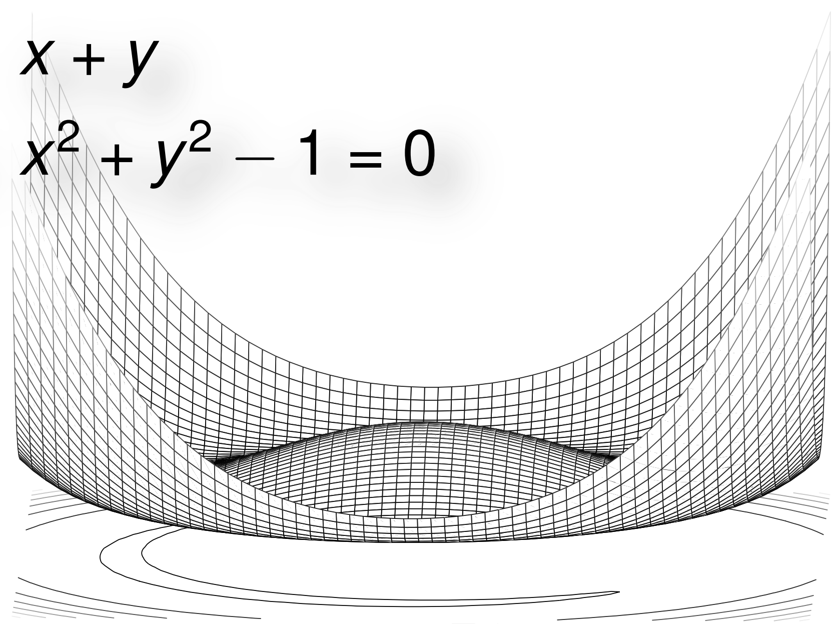
$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \end{array}$$

$$\text{minimize} \quad f(\mathbf{x}) + \mu \textstyle\sum_i c_i(\mathbf{x})^2 \;\leftrightarrow\; \text{minimize} \quad f(\mathbf{x}) + \mu \mathbf{c}(\mathbf{x})^T \mathbf{c}(\mathbf{x})$$

$$\begin{array}{ll} \text{minimize} & x + y \\ \text{subject to} & x^2 + y^2 - 1 = 0 \end{array}$$



$\mu = 1/2$



$\mu = 50$

A penalty method

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \end{array}$$

$$\text{minimize} \quad f(\mathbf{x}) + \mu \sum_i c_i(\mathbf{x})^2 \quad \leftrightarrow \quad \text{minimize} \quad f(\mathbf{x}) + \mu \mathbf{c}(\mathbf{x})^T \mathbf{c}(\mathbf{x})$$

Let $\{\tau_k\} \to 0$, $\{\mu_k\} \to \infty$.
While $\|\mathbf{c}(\mathbf{x}_k)\| \geq \text{tol}$

Gradient norm

Solve $\text{minimize} \quad f(\mathbf{x}) + \mu_k/2\mathbf{c}(\mathbf{x})^T\mathbf{c}(\mathbf{x})$ to tolerance τ_k

Set $\mathbf{x}^{(k+1)}$ to be the solution.

Convergence of the penalty method

If we'll be able to prove this convergences, we'll need a strong condition.

Why?

Convergence of the penalty method

If we'll be able to prove this convergences, we'll need a strong condition.

$$\mathbf{c}(\mathbf{x})^T \mathbf{c}(\mathbf{x}) \text{ vs. } \mathbf{c}(\mathbf{x}) = 0$$

Convergence of the penalty method

If we'll be able to prove this convergences, we'll need a strong condition.

$$\mathbf{c}(\mathbf{x})^T\mathbf{c}(\mathbf{x}) \text{ vs. } \mathbf{c}(\mathbf{x}) = 0$$

Theorem 17.1 (Paraphrased)

If we use the global minimizer of each subproblem, then we solve the problem in the $\mu_k \to \infty$ limit

Convergence of the penalty method

Theorem 17.2 (Paraphrased)

If we approximately minimize each problem to a point where

$$\|\mathbf{g}(\mathbf{x}_k)\| \leq \tau_k$$

Then either a limit point of the sequence is either (infeasible) and a stationary point of $\|\mathbf{c}(\mathbf{x})\|^2$

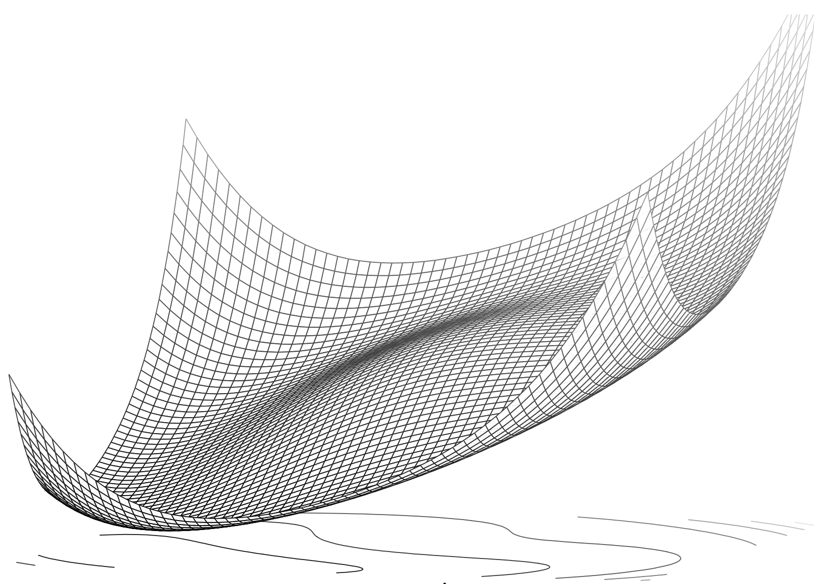
Or

It's a KKT point of the original problem

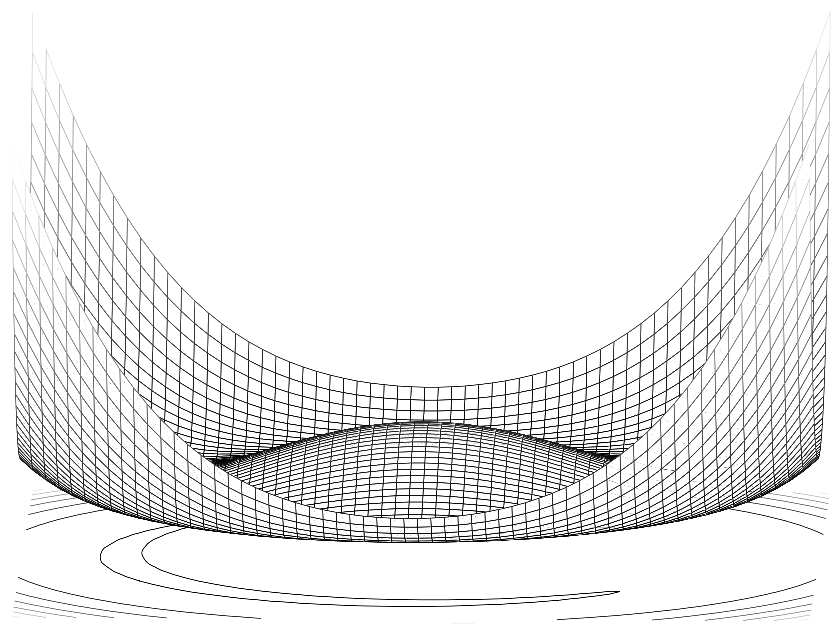
Weaknesses

Highly "ill-conditioned" as $\mu_k \to \infty$

Convergence only about limit points



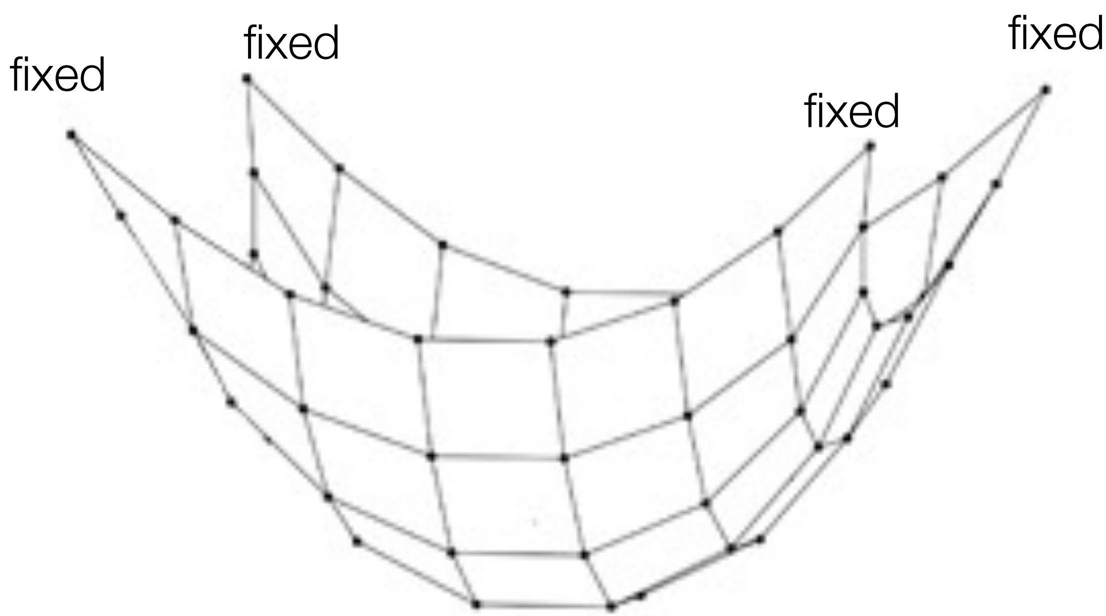
$\mu = 1/2$



$\mu = 50$

Problems with penalty methods

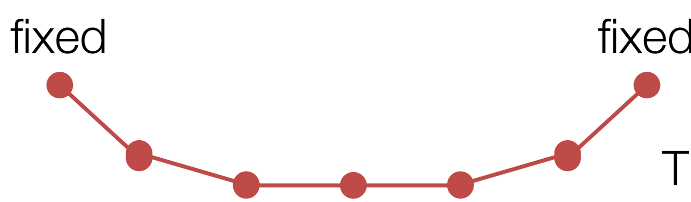
The hanging net problem



minimize the energy in the system
subject to using “steel” links

Problems with penalty methods

The hanging net problem



These links would stretch!
(Small constraint violation gives a large change in the energy objective)
As $\mu_k \to \infty$ we'd violate these constraints "more" than others.

minimize the energy in the system
subject to using "steel" links

Problems with penalty methods

All constraints are not equal!

Augmented Lagrangian Methods

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \end{array}$$

$$\mathcal{L}(\mathbf{x}; \lambda, \mu) = f(\mathbf{x}) - \lambda^T \mathbf{c}(\mathbf{x}) + \mu/2 \|\mathbf{c}(\mathbf{x})\|^2.$$

Augmented Lagrangian Methods

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \end{array}$$

$$\mathcal{L}(\mathbf{x}; \lambda, \mu) = f(\mathbf{x}) - \lambda^T \mathbf{c}(\mathbf{x}) + \mu/2 \|\mathbf{c}(\mathbf{x})\|^2.$$

If we minimize in x alone

$$\nabla \mathcal{L}(\mathbf{x}) = \mathbf{g}_f(\mathbf{x}) - \mathbf{J}_c(\mathbf{x})^T (\lambda - \mu \mathbf{c}(\mathbf{x})) = 0$$

One of the KKT conditions of the non-linear program is

$$\mathbf{g}_f(\mathbf{x}^*) - \mathbf{J}_c(\mathbf{x}^*)^T \lambda^* = 0$$

Augmented Lagrangian Methods

$$\nabla \mathcal{L}(\mathbf{x}) = \mathbf{g}_f(\mathbf{x}) - \mathbf{J}_c(\mathbf{x})^T(\lambda - \mu \mathbf{c}(\mathbf{x})) = 0$$

$$\mathbf{g}_f(\mathbf{x}^*) - \mathbf{J}_c(\mathbf{x}^*)^T \lambda^* = 0$$

So in an algorithm, we use

$$\lambda_{k+1} = \lambda_k - \mu_k \mathbf{c}(\mathbf{x})$$

$\underset{\mathbf{x}}{\text{minimize}} \quad \mathcal{L}(\mathbf{x}; \lambda_k, \mu_k)$ to tolerance τ_k starting from $\mathbf{x}_k$

if $\|\mathbf{c}(\mathbf{x})\|$ is small, stop!

else set $\lambda_{k+1} = \lambda_k - \mu_k \mathbf{c}(\mathbf{x}_k)$

set $\mu_{k+1} \geq \mu_k$

Convergence of AL methods

See theorem 17.5 and 17.6

See Alg 17.4 for the method used in LANCELOT with bound-constraints.

Barrier methods

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{d}(\mathbf{x}) \geq 0 \end{array}$$

$$\text{minimize} \quad f(\mathbf{x}) - \mu \mathbf{e}^T \log(\mathbf{d}(\mathbf{x}))$$

Chapter 16, Griva, Sofer & Nash

The inequality constrained problem

$$
\begin{array}{ll}
\text{minimize} & f(\mathbf{x}) \\
\text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \\
& \mathbf{d}(\mathbf{x}) \geq 0
\end{array}
$$

can be transformed into

$$
\begin{array}{ll}
\text{minimize} & f(\mathbf{x}) \\
\text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \\
& \mathbf{d}(\mathbf{x}) - \mathbf{s} = 0 \\
& \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{s} \geq 0
\end{array}
$$

So handling equality, and bounds suffices!

The bound constrained problem

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) = 0 \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{array}$$

See algorithms in
Algorithm 17.4, Chapter 18
Nocedal & Wright

LANCELOT
Sequential quadratic programming
Gradient projection