Optimization :

- Ideas & Logic
- Gradient based methods
- Gradient free method
- Unconstrained
  - fminsearch
- Constrained optimization
  - fmincon

Optimization :

Minimization of an objective function with respect to constraints ($=, \leq$)

minimize $f(\vec{x})$ } objective

subject to $g(\vec{x}) \leq 0$, $h(\vec{x}) = 0$ } constraints

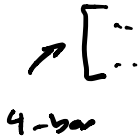
$\vec{x}$ → Design variables, optimizing with respect to

$f(\vec{x})$ → Returns a scalar value as function of design variables

$g(\vec{x}) \leq 0$ inequality constraint on design variables

$h(\vec{x}) = 0$

Can return a vector which corresponds to multiple equations



4-bar

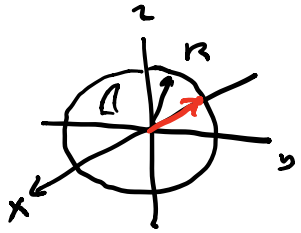
Examples: 1) var: x $\quad f(x) = x^2$

$\nearrow$ $\min f(x) \Rightarrow x = 0$

2) $\vec{x} = [x, y, z]$

$\nearrow$ $f(\vec{x}) = x^3 + y^2 + z^2$

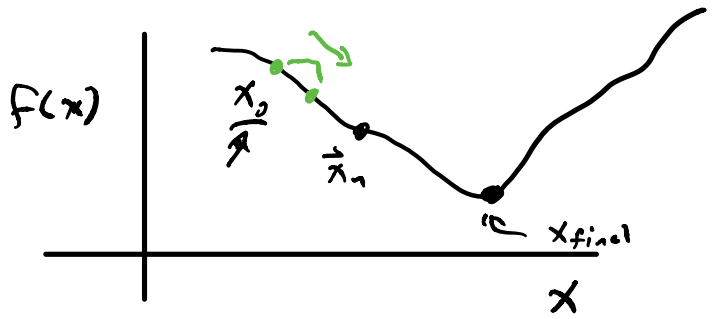
$\min f(\vec{x})$



$h(\vec{x})$: $x^2 + y^2 + z^2 - R^2 = 0$

$g(\vec{x})$: $\begin{bmatrix} z \\ x - R/2 \end{bmatrix} \leq 0$

Gradient descent :

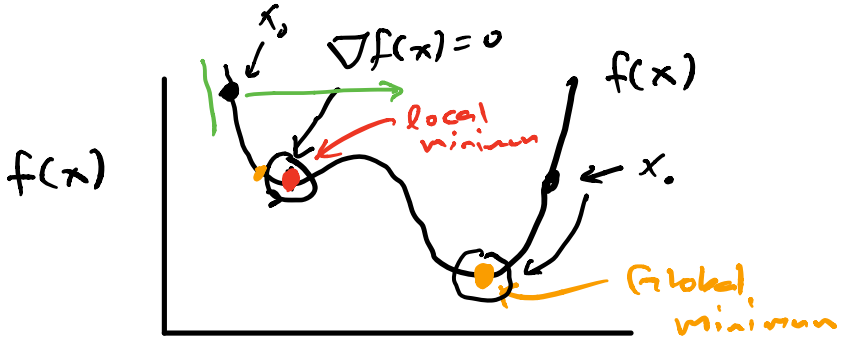


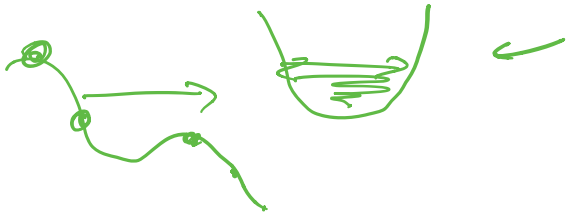
$f(x_{final}) \leq f(x)$

$$\vec{x}_{n+1} = \vec{x}_n - \gamma \nabla f(\vec{x}_n) + b(\vec{x})$$

"step size" and this should be small

An iterative process following line of greatest change in the cost function



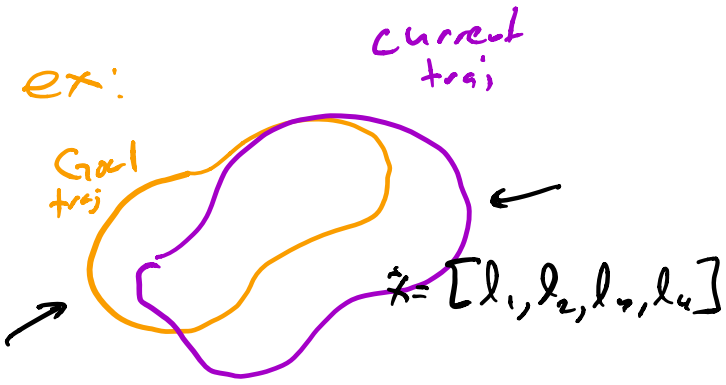


Gradient methods are ubiquitous, but simple forms can suffer from getting trapped in local minima and take a long time to converge.

$$\vec{x}_{n+1} = \vec{x}_n - \gamma \nabla f(\vec{x}_n)$$

What did we assume?

ex:



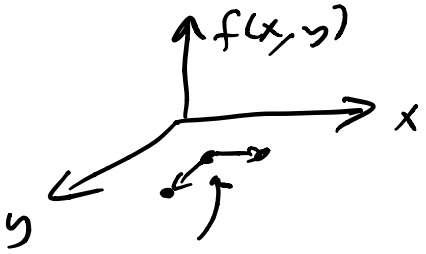
$f(\vec{x}) = \cdots$

Need gradient for gradient descent!!

$$f(\vec{x}) = \sum_i \left(\underset{=}{x}^i_{Goal} - \underset{=}{x}^i_{current}\right)^2 + \sum_i \left(\underset{=}{y}^i_{Goal} - \underset{=}{y}^i_{current}\right)^2$$

Many objective functions are numerical s.t. we don't an analytical expression. So taking gradients are approximated by numerical gradients

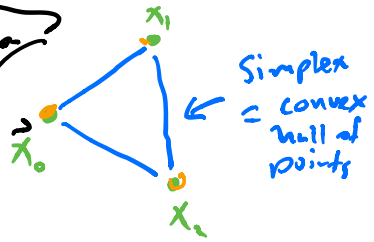
Numerical gradients → Multiple evaluations of cost function



Gradient-free methods:

Nelder-Mead Simplex algorithm

In Two-D optimization:
$\vec{x}_0 = [x_0, y_0]$

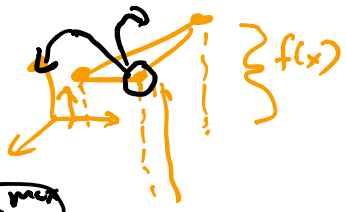


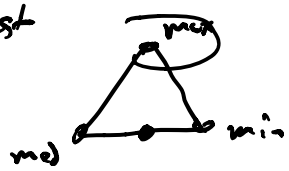
Basic NM:

1) Evaluate $f(\vec{x}_1), f(\vec{x}_2), f(\vec{x}_3)$

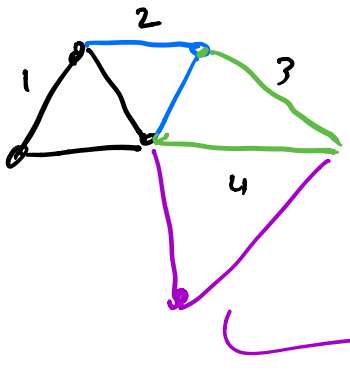
2) Order $f(\vec{x}_2) < f(\vec{x}_3) < f(\vec{x}_1)$

3) Compute centroid of best side





4) Reflection about centroid,
Scaling → expansion/contraction


2
1
3
4

Optimization Decision Table

The following table is designed to help you choose a solver. It does not address multiobjective optimization or equation solving. There are more details on all the solvers in Problems Handled by Optimization Toolbox Functions.

In this table:

- * means relevant solvers are found in Global Optimization Toolbox (Global Optimization Toolbox) functions (licensed separately from Optimization Toolbox™ solvers).
- `fmincon` applies to most smooth objective functions with smooth constraints. It is not listed as a preferred solver for least squares or linear or quadratic programming because the listed solvers are usually more efficient.
- The table has suggested functions, but it is not meant to unduly restrict your choices. For example, `fmincon` can be effective on some nonsmooth problems.
- The Global Optimization Toolbox `ga` function can address mixed-integer programming problems.
- The Statistics and Machine Learning Toolbox™ `bayesopt` function can address low-dimensional deterministic or stochastic optimization problems with combinations of continuous, integer, or categorical variables.

Solvers by Objective and Constraint

Constraint Type	Objective Type				
	Linear	Quadratic	Least Squares	Smooth Nonlinear	Nonsmooth
None	n/a (f = const, or min = $-\infty$)	`quadprog`, Information	`mldivide`, `lsqcurvefit`, `lsqnonlin`, Information	`fminsearch`, `fminunc`, Information	`fminsearch`, *
Bound	`linprog`, Information	`quadprog`, Information	`lsqcurvefit`, `lsqlin`, `lsqnonlin`, `lsqnonneg`, Information	`fminbnd`, `fmincon`, `fseminf`, Information	`fminbnd`, *
Linear	`linprog`, Information	`quadprog`, Information	`lsqlin`, Information	`fmincon`, `fseminf`, Information	*
General Smooth	`fmincon`, Information	`fmincon`, Information	`fmincon`, Information	`fmincon`, `fseminf`, Information	*
Discrete, with Bound or Linear	`intlinprog`, Information	*	*	*	*

fminsearch → unconstrained optimization

fmincon → constrained optimization

fminsearch

Find minimum of unconstrained multivariable function using derivative-free method

Syntax

```
x = fminsearch(fun,x0)
x = fminsearch(fun,x0,options)
x = fminsearch(problem)
[x,fval] = fminsearch(__)
[x,fval,exitflag] = fminsearch(__)
[x,fval,exitflag,output] = fminsearch(__)
```

Description

Nonlinear programming solver. Searches for the minimum of a problem specified by

$$\min_x f(x)$$

$f(x)$ is a function that returns a scalar, and x is a vector or a matrix.

$$f(x) = x^2$$

$$= (x-a)^2$$

In all routines finding min. If want max $f(x) = -\text{cost}$

fitting

$$f(\vec{x}) = \sum (\vec{x}_{\text{target}} - \vec{x})^2$$

$$=$$

fmincon

Find minimum of constrained nonlinear multivariable function

collap

Syntax

```
x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
x = fmincon(problem)
[x,fval] = fmincon(__)
[x,fval,exitflag,output] = fmincon(__)
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(__)
```

fun: objective function

Description

Nonlinear programming solver.

Finds the minimum of a problem specified by

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \le 0 \\ ceq(x) = 0 \\ A \cdot x \le b \\ Aeq \cdot x = beq \\ lb \le x \le ub, \end{cases}$$

nonlcon

$c(x) \le 0$

$c_{eq}(x) = 0$

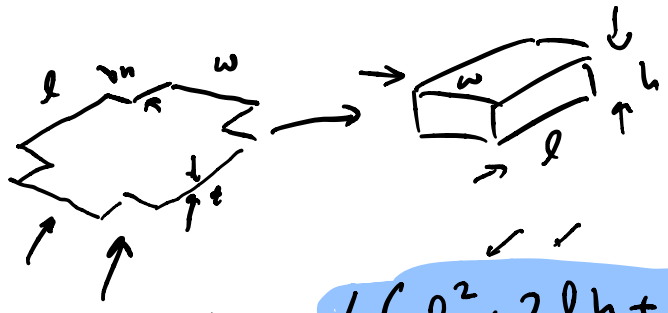
$Ax \le b$

b and beq are vectors, A and Aeq are matrices, $c(x)$ and $ceq(x)$ are functions that return vectors, and $f(x)$ is a function that returns a scalar. $f(x)$, $c(x)$, and $ceq(x)$ can be nonlinear functions.

x, lb, and ub can be passed as vectors or matrices; see Matrix Arguments.

Design of a box that minimizes the weight, but has volume V, and height H

minimize w



$$\text{Weight} = t(l^2 + 2lh + 2wh)$$

$\vec{x} = [l, w, h]$

Constraints:

$l > 0$

$w > 0$

$h - H = 0$

$l \cdot w \cdot h - V = 0$