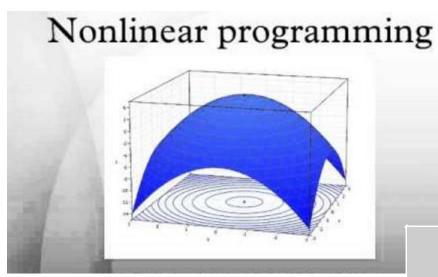
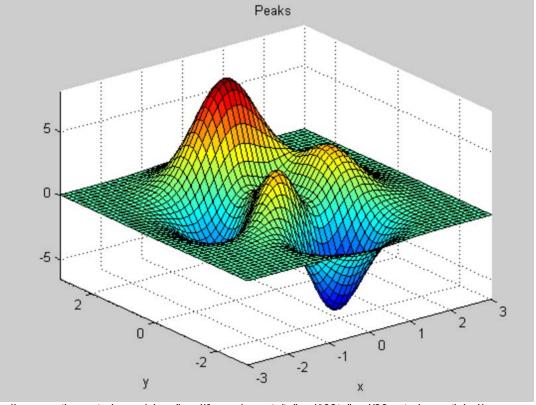
### **Optimization**



https://ocw.mit.edu/courses/sloan-school-of-management/15-0 84j-nonlinear-programming-spring-2004/15-084jf04.jpg

Find the <u>Location</u> of the Maximum or Minimum Value



http://www.math.uwaterloo.ca/~hwolkowi//henry/reports/talks.d/t09talks.d/09waterloomatlab.d/optimTipsWebinar/html/optimTipsTricksWalkthrough\_06.png

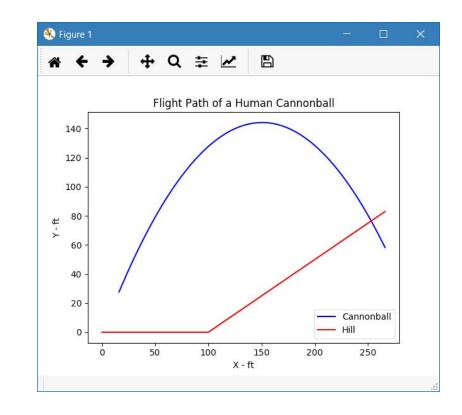
## **Engineering Optimization**

Choose the values of Design Parameters or Operating Parameters to achieve maximum GOOD or minimum BAD, subject to Constraints

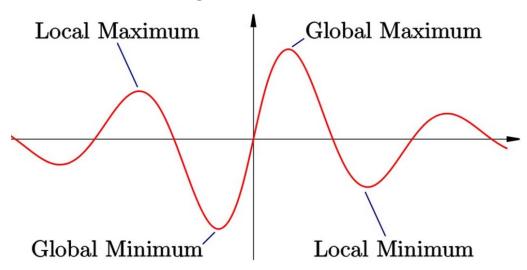
Variable and Functional Constraints

### Good and Bad for:

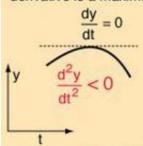
Airplane
Race Car
Commuter Car
Umbrella
Camera
Human Canon



One - Degree of Freedom (dof) Problem Unconstrained



The second derivative demonstrates whether a point with zero first derivative is a maximum, a minimum, or an inflexion point.



For a maximum, the second derivative is negative. The slope of the curve ( first derivative) is at first positive, then goes through zero to become negative.

$$\frac{\frac{d^2y}{dt^2} > 0}{\frac{dy}{dt} = 0}$$

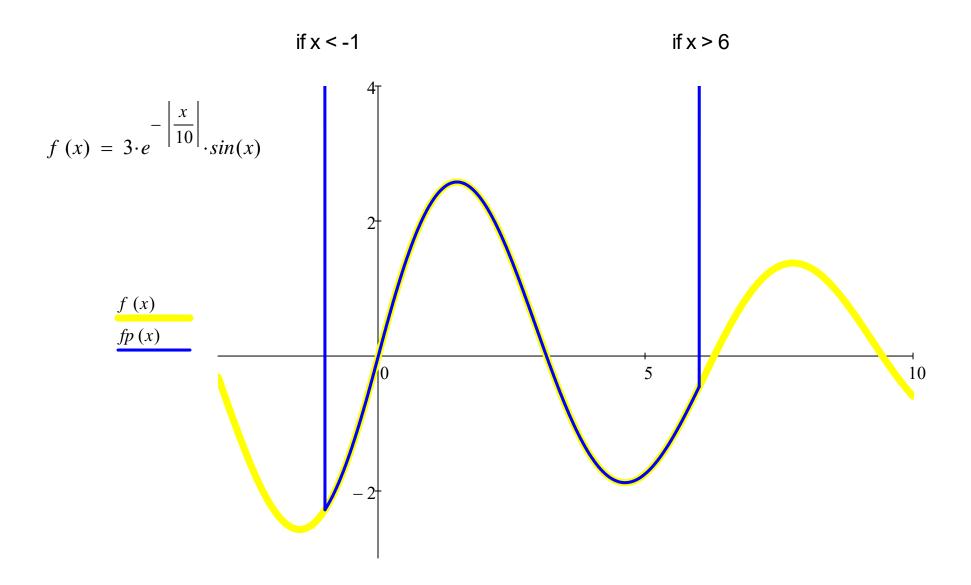
For a minimum, the second derivative is positive. The slope of the curve = first derivative is at first negative, then goes through zero to become positive.

$$\frac{dy}{dt} = 0$$

$$\frac{d^2y}{dt^2} = 0$$

For an inflexion point, the second derivative is zero at the same time the first derivative is zero. It represents a point where the curvature is changing its sense. Inflexion points are relatively rare in nature.

### Constraints - and Local minima and maxima



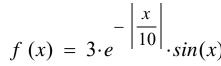
Constraints and Penalty Functions - for Unconstrained MINIMIZERs

if 
$$x < -1$$

if 
$$x > 6$$

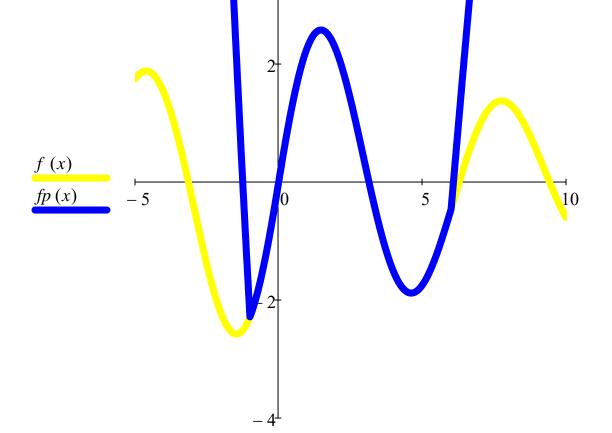
$$p = 10 \cdot (-1 - x)$$
  $p = 4 \cdot (x - 6)$ 

$$p = 4 \cdot (x - 6)$$

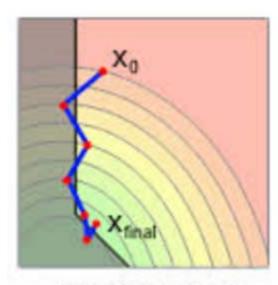


Usually slopes around

 $10^6$ 

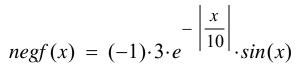


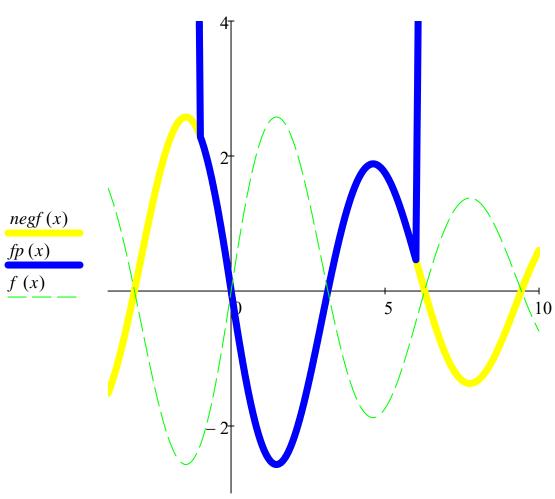
 $\boldsymbol{x}$ 



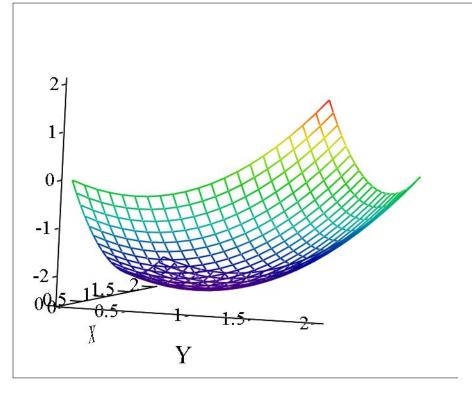
Penalty method

To Maximize a function, using a MINIMIZER, use the NEGATIVE of the function, and use POSITIVE penalties





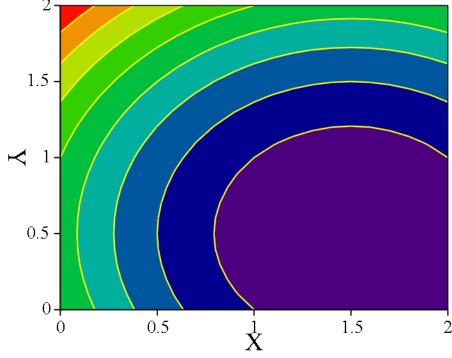
A 2D Example



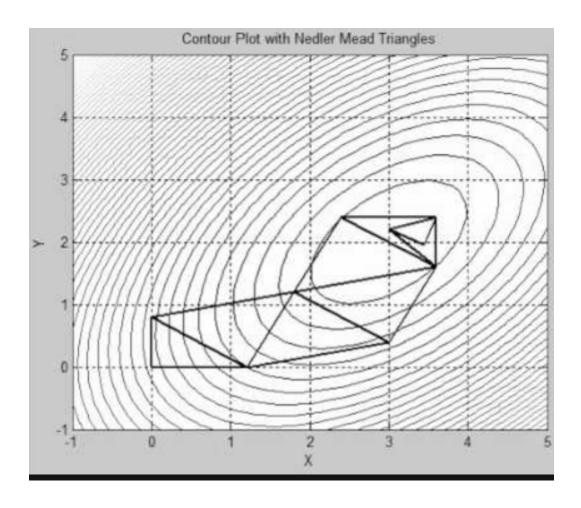
Z

$$z(x,y) = y^2 - y + x^2 - 3x$$

We will play with these in Python



# Nelder - Mead Simplex Method - an Unconstrained Minimizer



http://www.jakubkonka.com/images/nelder-mead/no-minimum.png

#### SciPy Documentation - A really poor example for Beginners



$$f(\mathbf{x}) = \sum_{i=1}^{N-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$$

Scipy.org

Docs

SciPy v0.19.0 Reference Guide

SciPy Tutorial

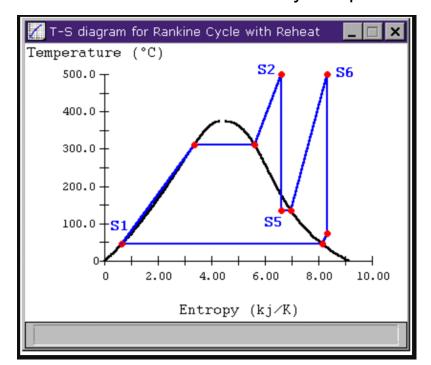
#### Nelder-Mead Simplex algorithm (method='Nelder-Mead')

In the example below, the minimize routine is used with the Nelder-Mead simplex algorithm

```
>>> import numpy as np
>>> from scipy.optimize import minimize
                                                                               f(x,y) = (a-x)^2 + b(y-x^2)^2
>>> def rosen(x):
        """The Rosenbrock function"""
       return sum(100.0*(x[1:]-x[:-1]**2.0)**2.0 + (1-x[:-1])**2.0)
                                                                          2500
>>> x0 = np.array([1.3, 0.7, 0.8, 1.9, 1.2])
                                                                           20001
>>> res = minimize(rosen, x0, method='nelder-mead',
                                                                           1500
                   options={'xtol': 1e-8, 'disp': True})
                                                                           1000
Optimization terminated successfully.
                                                                            500
         Current function value: 0.000000
         Iterations: 339
                                                                                                  -1.5 -0.5 0.0 x 1.0 1.5 2.0
                                                                               2.0
1.5
y 1.0
         Function evaluations: 571
>>> print(res.x)
[1. 1. 1. 1. 1.]
```

https://docs.scipy.org/doc/scipy/reference/tutorial/optimize.html

#### Rankine Cycle Optimization - Reheat Pressure and Temperature



http://www.qrg.northwestern.edu/thermo/design-library/reheat/rhtTs.gif

```
def optimize Rankine(p high,p low,t high,t mid max,
 5
                            turbine efficiency=0.90, pump efficiency=0.85):
 6
           def eff(vals ):...
           vals=minimize(eff,(p_low*100,t_high),method='Nelder-Mead',
21
                      options={'fatol':0.01, 'xatol':1.0})
22
           p mid=vals.x[0]
23
           t mid=vals.x[1]
24
           eff=vals.fun
25
           count=vals.nfev
26
27
28
           return p_mid, t_mid, eff, count
20
```