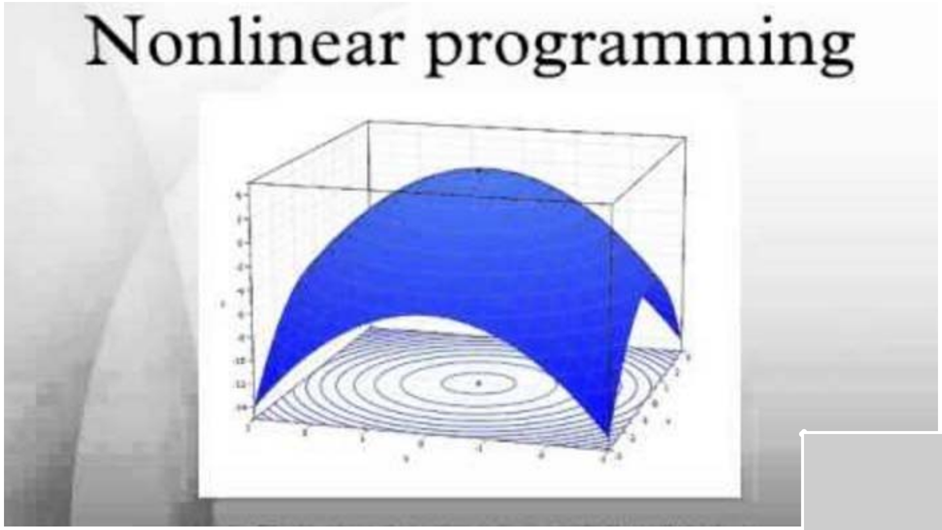
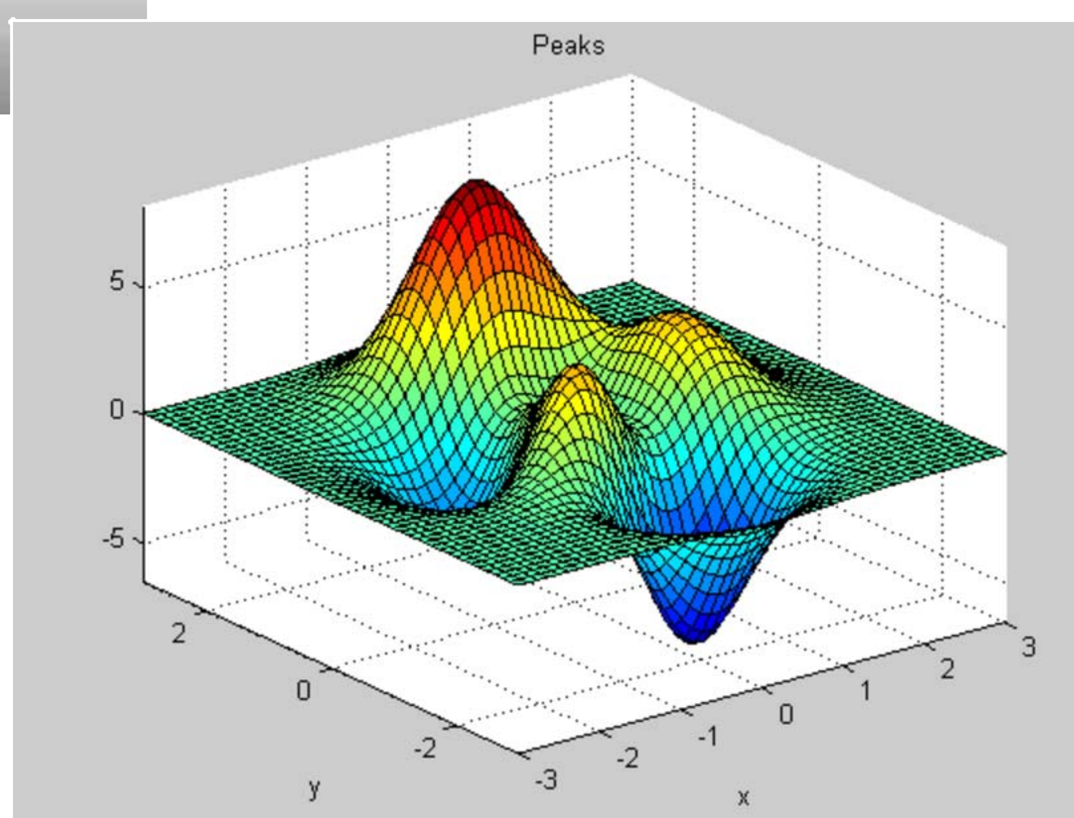
Optimization



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

Find the ***Location*** 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

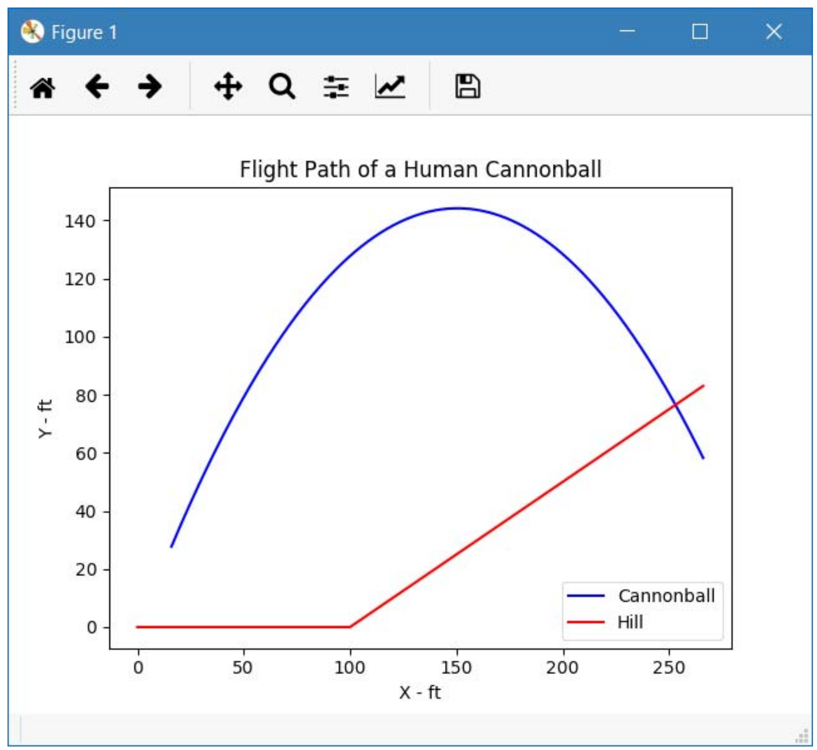
Engineering Optimization

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

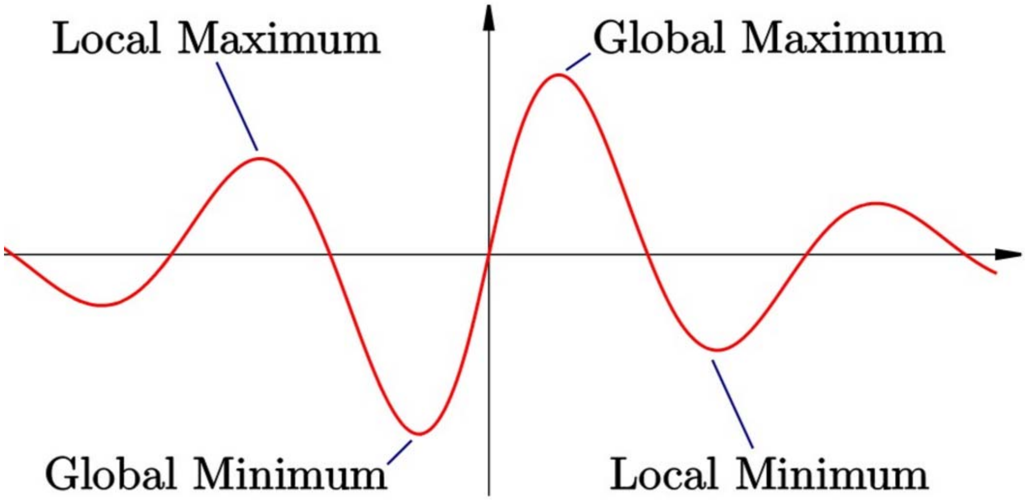
Variable and Functional Constraints

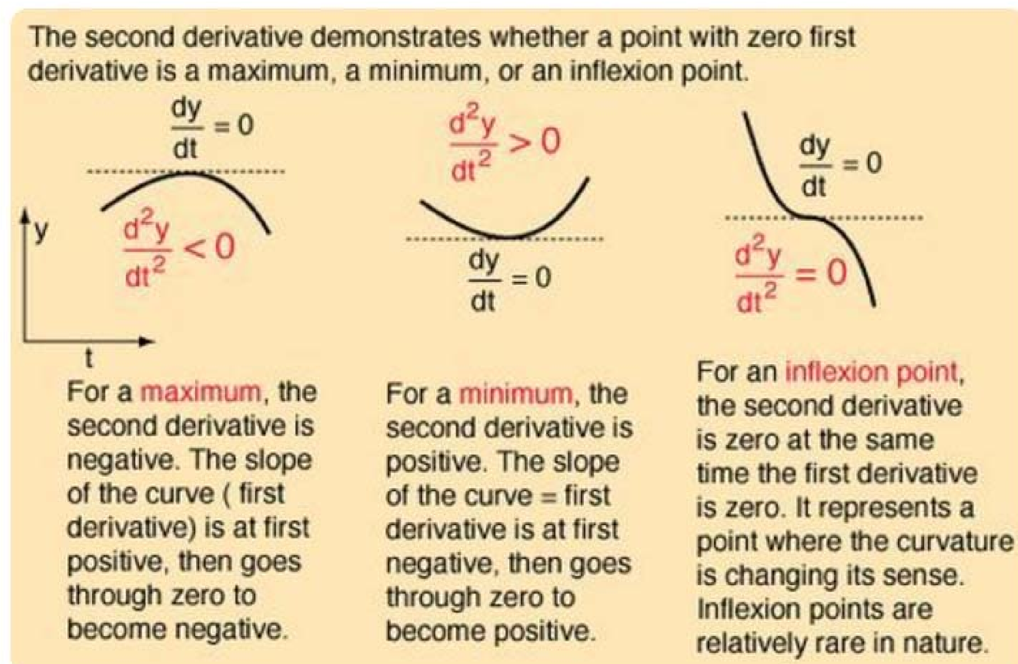
Good and Bad for:

- Airplane
- Race Car
- Commuter Car
- Umbrella
- Camera
- Human Canon



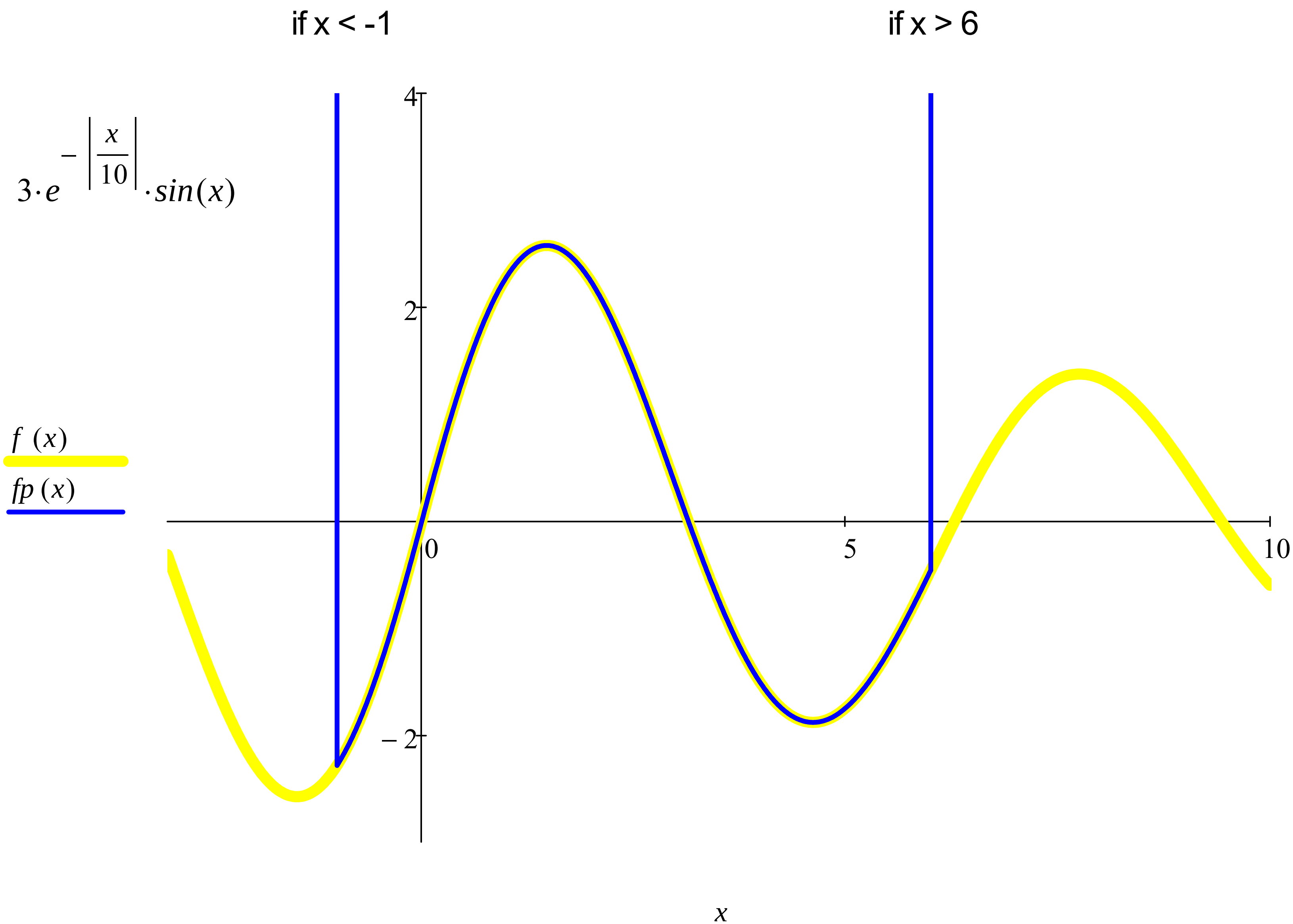
One - Degree of Freedom (dof) Problem
Unconstrained





Constraints - and Local minima and maxima

$$f(x) = 3 \cdot e^{-\left|\frac{x}{10}\right|} \cdot \sin(x)$$



Constraints and Penalty Functions - for Unconstrained MINIMIZERs

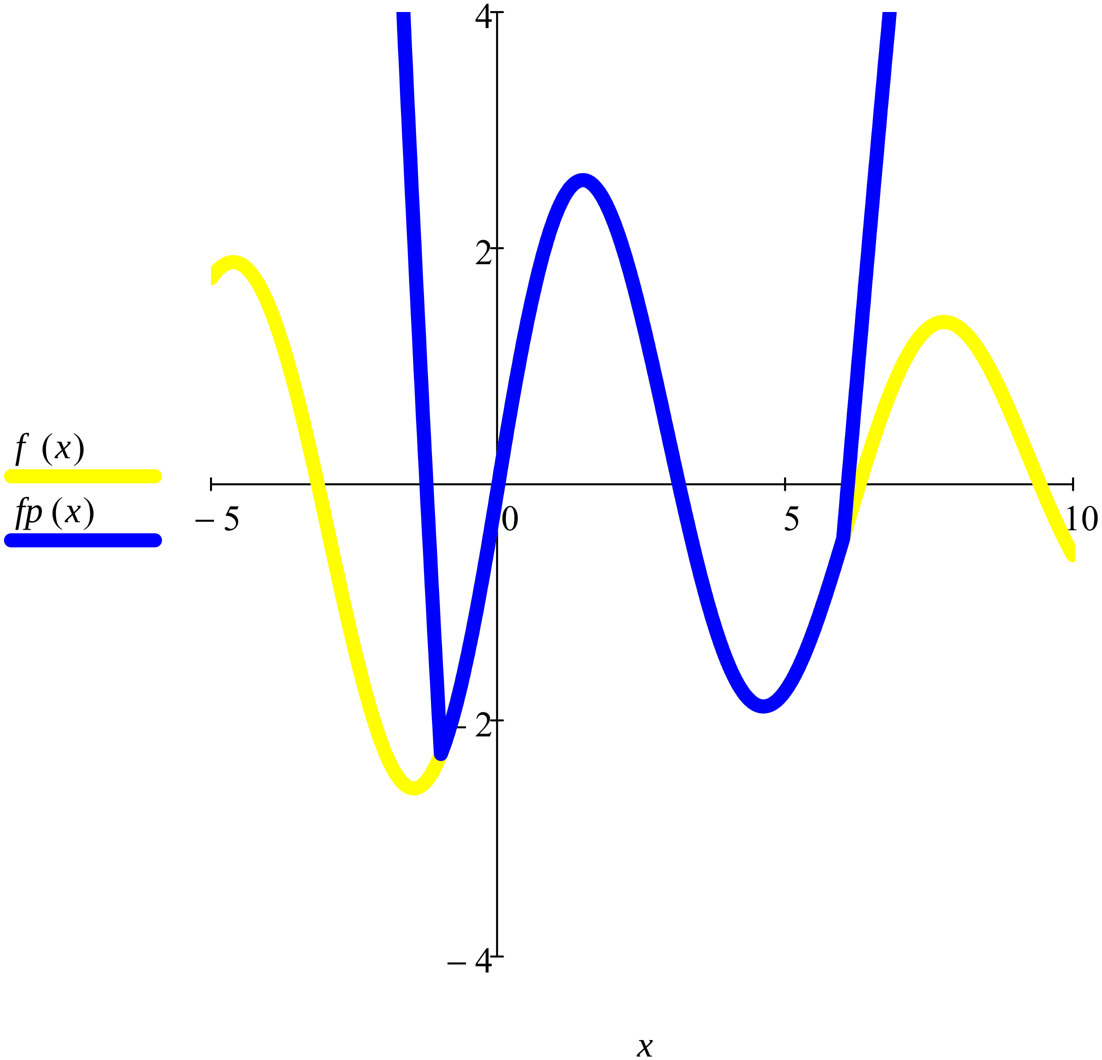
if x < -1

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

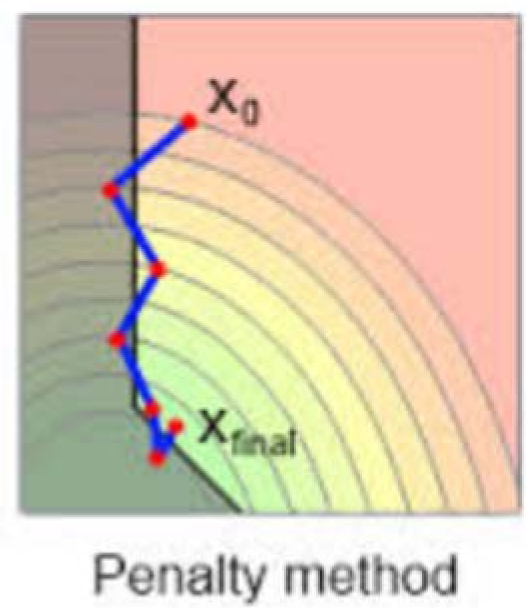
if x > 6

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

$$f(x) = 3\cdot e^{-\left|\frac{x}{10}\right|}\cdot \sin(x)$$

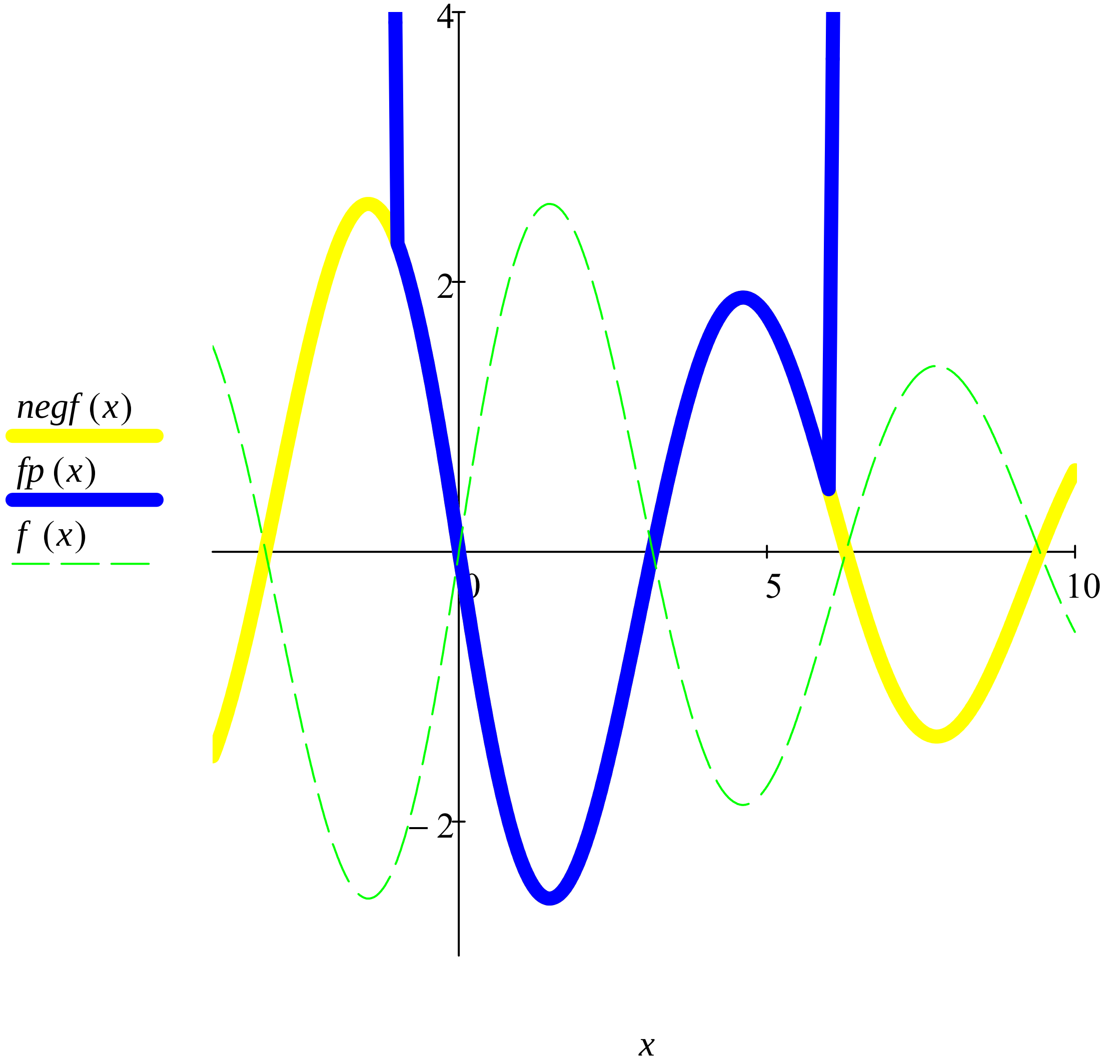


Usually slopes around 10^6



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

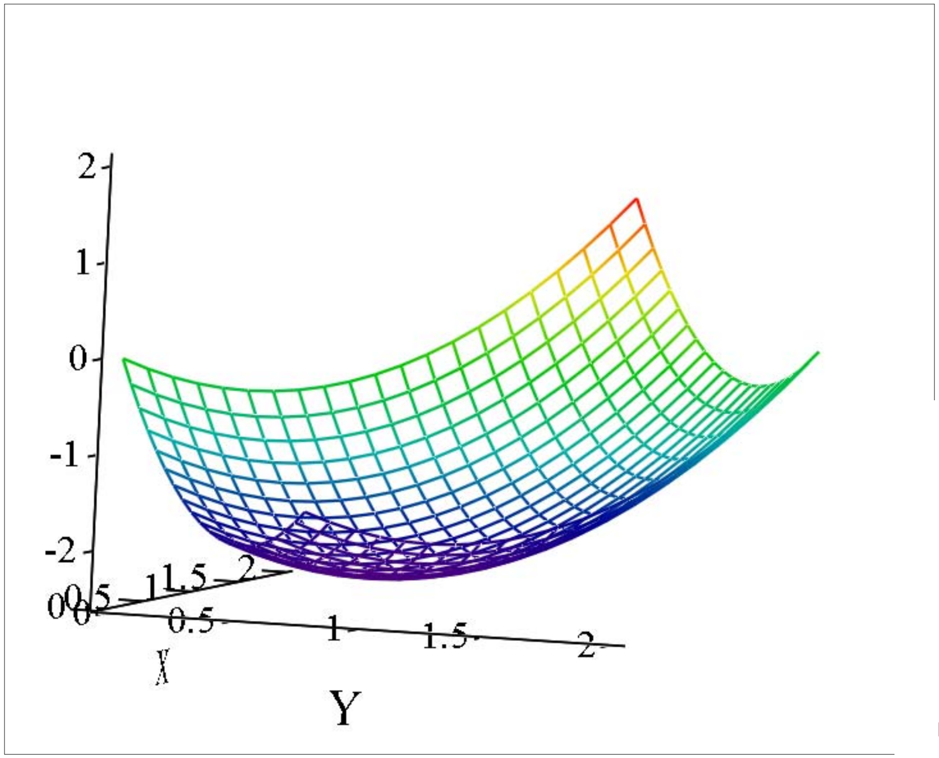
$$negf(x) = (-1)\cdot 3\cdot e^{-\left|\frac{x}{10}\right|}\cdot sin(x)$$



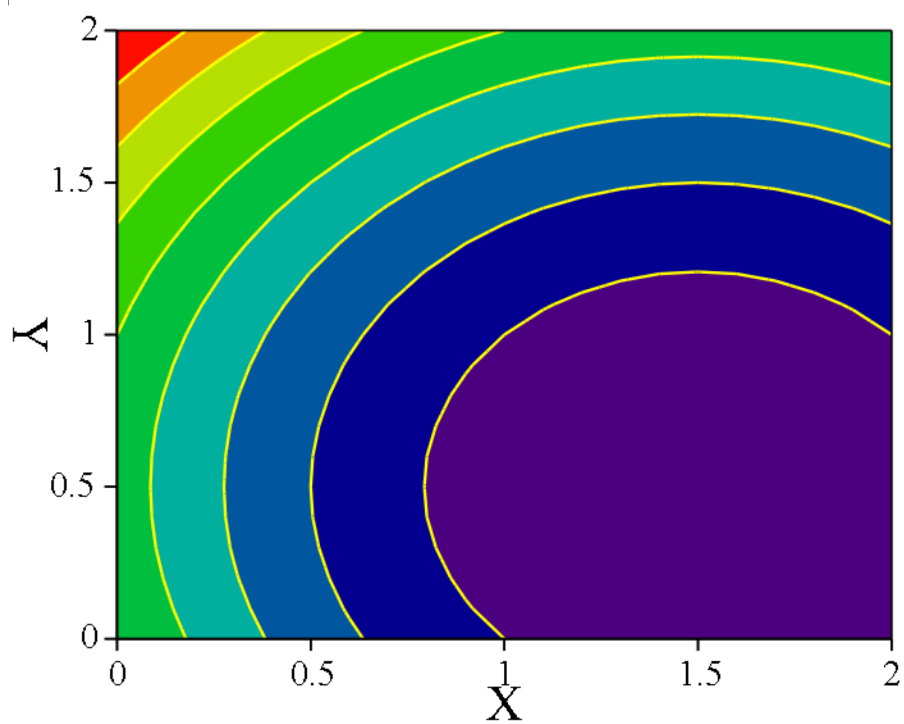
A 2D Example

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

We will play with these in Python

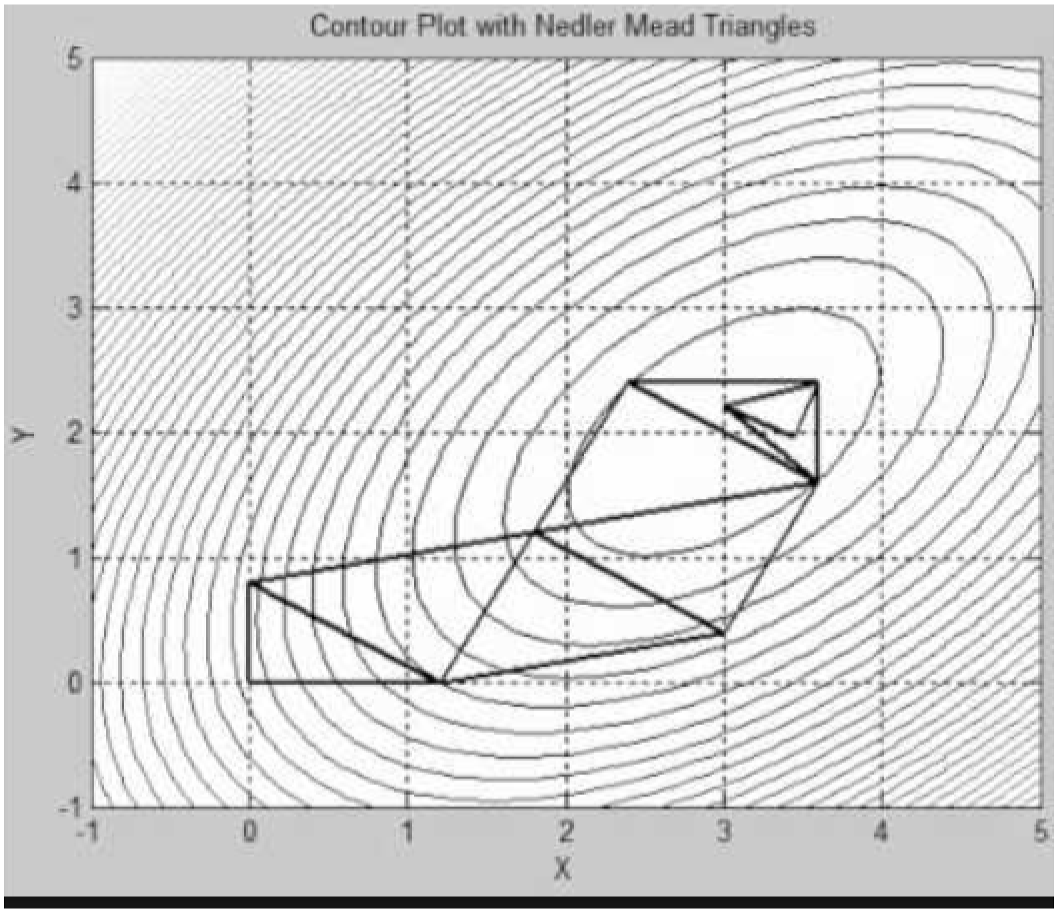


z



z

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

SciPy.org Sponsored By ENTHOUGHT

Scipy.org | Docs | SciPy v0.19.0 Reference Guide | SciPy Tutorial

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

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

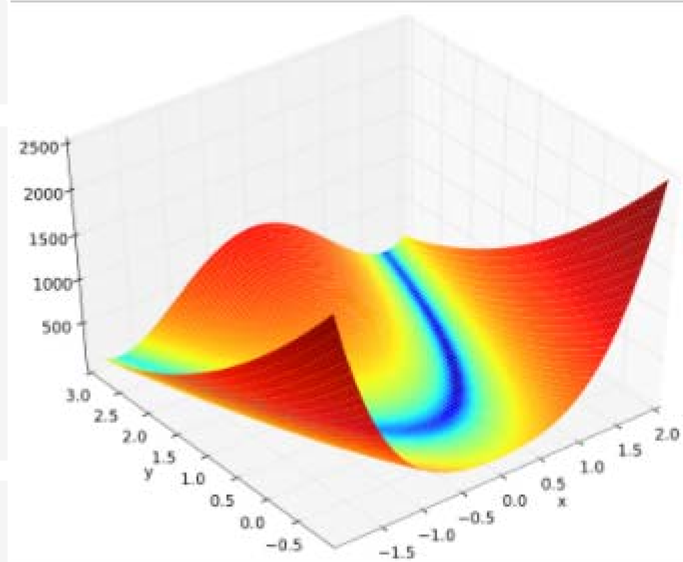
In the example below, the minimize routine is used with the *Nelder-Mead* simplex algorithm

```
>>> 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)
```

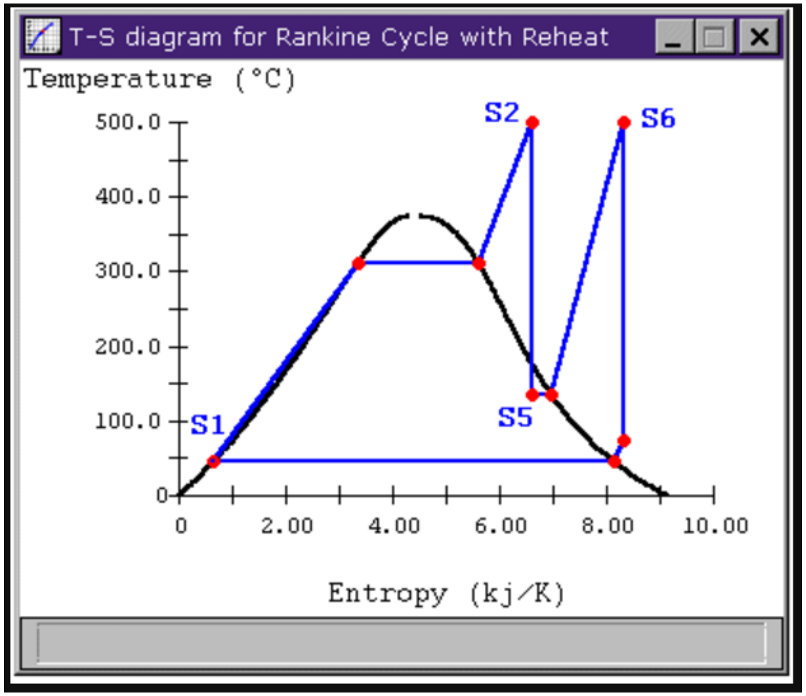
```
>>> x0 = np.array([1.3, 0.7, 0.8, 1.9, 1.2])
>>> res = minimize(rosen, x0, method='nelder-mead',
...                options={'xtol': 1e-8, 'disp': True})
Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 339
         Function evaluations: 571
```




```
>>> 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

```python
def optimize_Rankine(p_high,p_low,t_high,t_mid_max,
                     turbine_efficiency=0.90, pump_efficiency=0.85):
    def eff(vals ):...
    vals=minimize(eff,(p_low*100,t_high),method='Nelder-Mead',
                  options={'fatol':0.01, 'xatol':1.0})
    p_mid=vals.x[0]
    t_mid=vals.x[1]
    eff=vals.fun
    count=vals.nfev

    return p_mid, t_mid, eff, count
```