Chapter 7

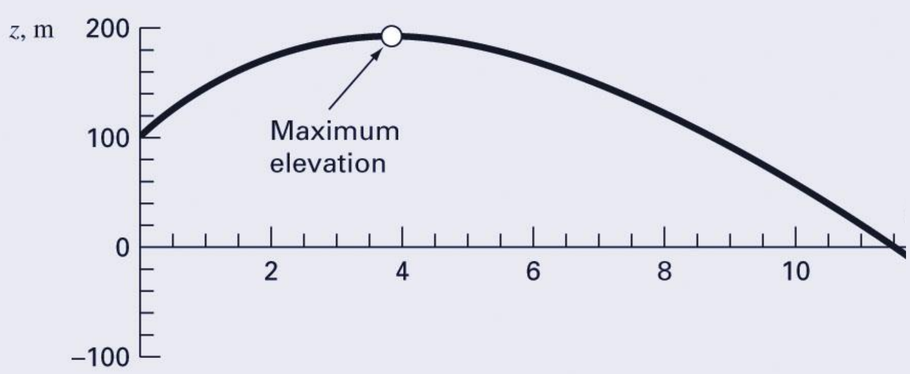
Optimization

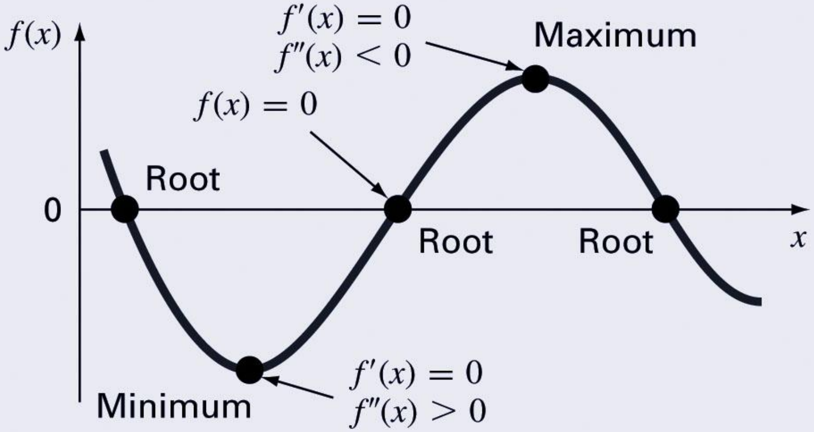
Chapter Objectives

- Distinguish between global and local optima.
- Recast a maximization problem so that it can be solved with a minimizing algorithm.
- Use the golden ratio to make one-dimensional optimization efficient.
- Knowing how to apply the Python minimize_scalar function to determine the minimum of a one-dimensional function.
- Knowing how to apply the Python minimize function to determine the minimum of a multidimensional function.

Optimization

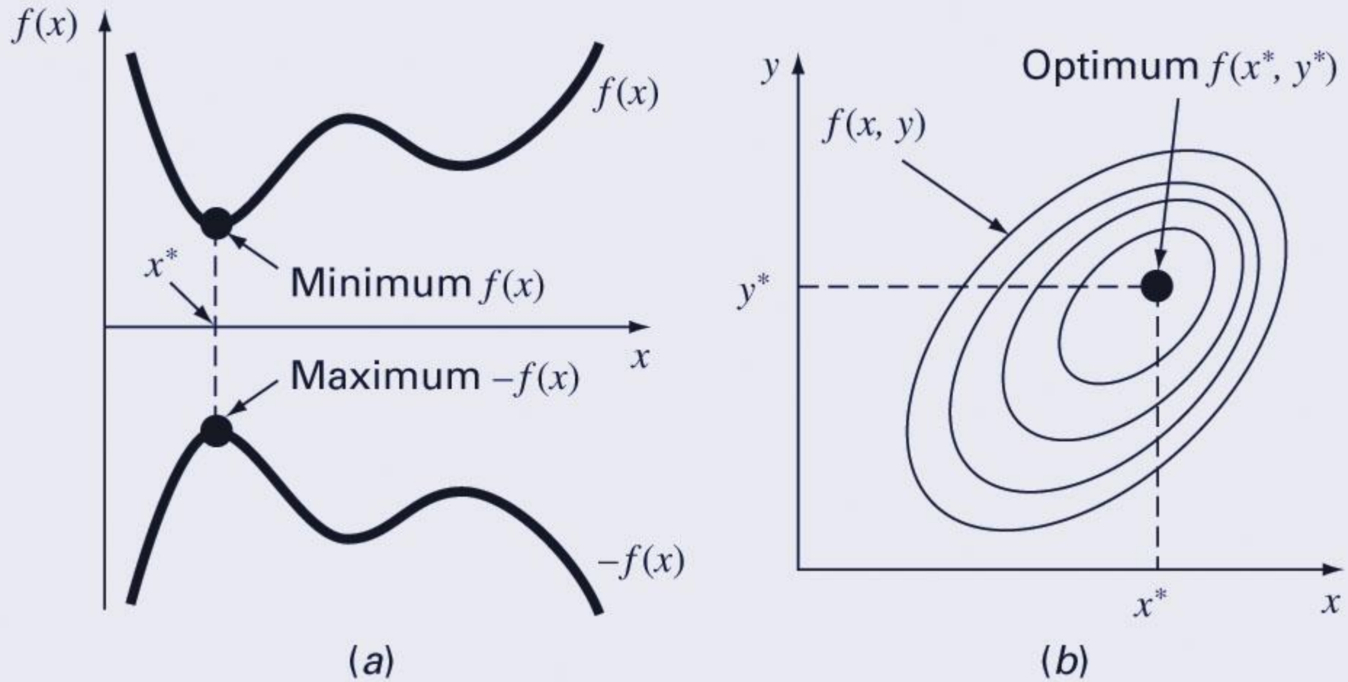
From a mathematical perspective, optimization deals with finding the maxima and minima of a function that depends on one or more variables.





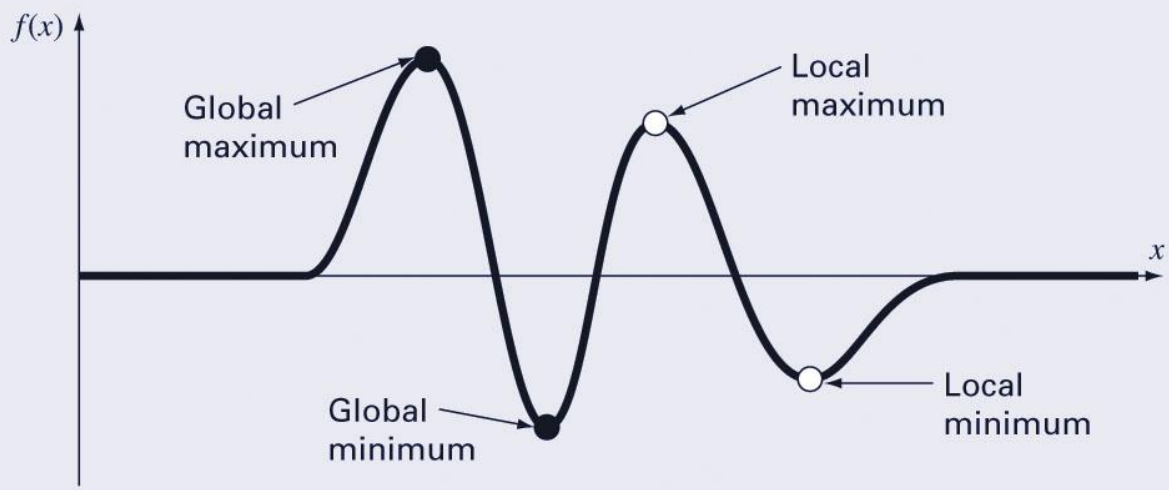
Multidimensional Optimization

- (Single) Multidimensional problems involve functions that depend on (one) two or more dependent variables - for example, (f(x)) $f(x,y)$



Global vs. Local

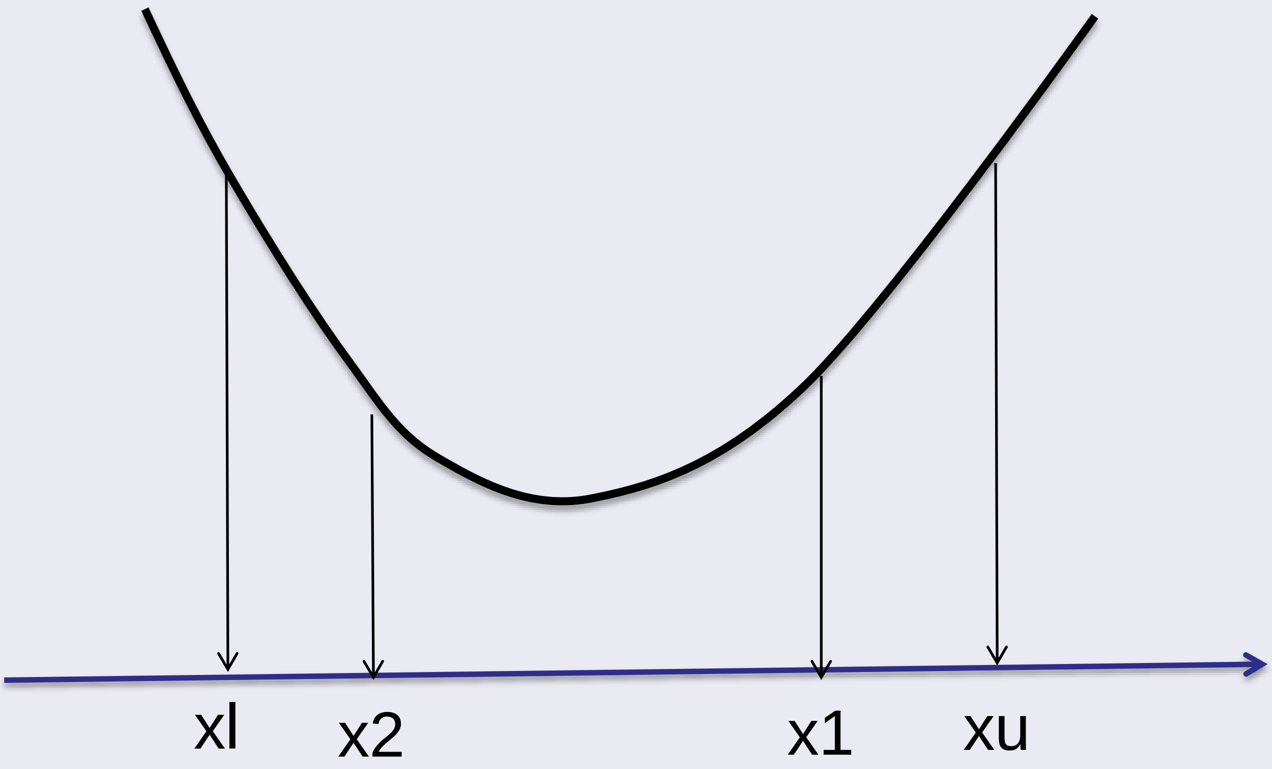
- A *global optimum* represents the very best solution while a *local optimum* is better than its immediate neighbors. Cases that include local optima are called *multimodal*.
- Generally desire to find the global optimum.



Minimization Search

The idea behind searching for a minimum is as follows:

- Suppose a minimum exist between two points xl and xu.
- Divide the interval into two unequal intervals by selecting an x1 in between xl and xu.
- Choose another point x2 in the interval xl-xu such that xl<x2<x1<xu. This leads to two intervals [xl x2 x1] and [x2 x1 xu]
- Evaluate the function at x1 and x2.


xl
x2
x1
xu

Minimization Search

- If f(x2) < f(x1) then the minimum is in the interval [xl x2 x1] and we reject the other interval.
- To proceed further we take the a new interval with

xl=xl

xu=x1

x1=x2

Repeat the previous steps all over (calculate new x2 only)

Minimization Search

- If $f(x1)<f(x2)$ then the minimum is in the interval [x2 x1 xu]
- Hence set the new interval as

$$xl = x2$$

$$x2 = x1$$

$$xu = xu$$

Calculate the new x1

- Repeat the whole process until $|x1-x2|<$ Tol

Golden Ratio

Divide a line into two unequal parts such that the longer section divided by the shorter section is equal to the whole line divided by the longer section

a/b = (a+b)/a = $\phi = \frac{1+\sqrt{5}}{2}$ = 1.618…

Solve the equation for a/b

Golden ratio appears in nature in many ways. For example the bellybutton divide the human body as golden ratio. The distance from bellybutton to feet is longer. In the human hand fingers, the joints are multiples of ϕ apart.

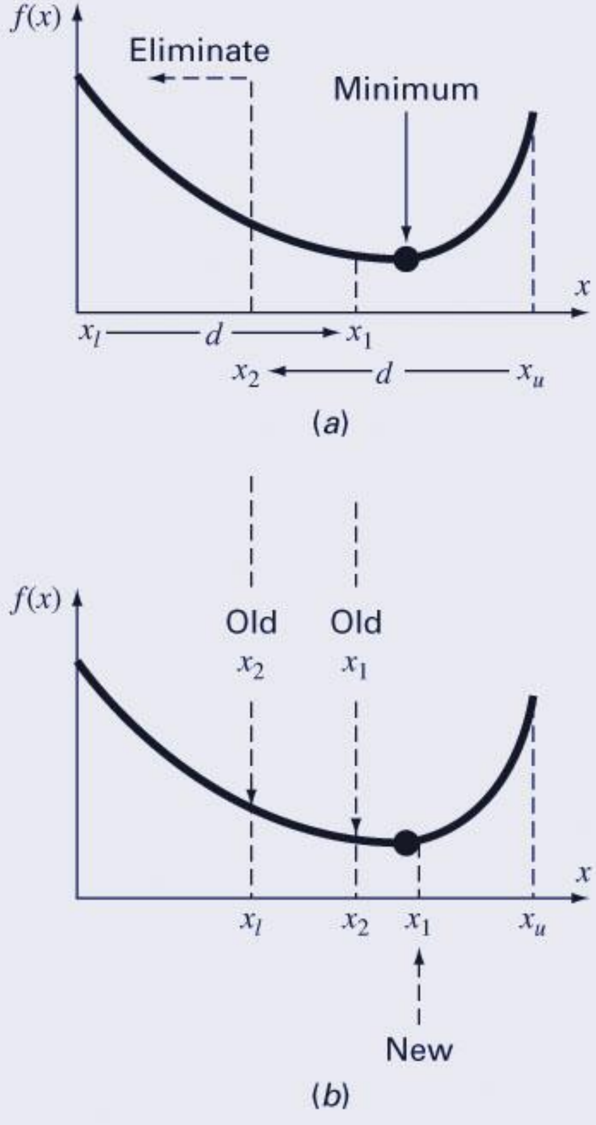
Golden-Section Search

- Search algorithm for finding a minimum on an interval $[x_l \;\; x_u]$ with a single minimum
- Uses the golden ratio $\phi = \frac{\sqrt{5}+1}{2} = 1.618..$ to determine location of two interior points $x_1, \;\; x_2$ by using golden ratio. One of the interior points can be re-used in the next iteration.

Golden-Section Search (cont)

$$d = (\phi - 1)(x_u - x_l)$$
$$x_1 = x_l + d$$
$$x_2 = x_u - d$$

- If $f(x_1)<f(x_2)$, x_2 becomes the new lower limit and x_1 becomes the new x_2 (as in figure).
- If $f(x_2)<f(x_1)$, x_1 becomes the new upper limit and x_2 becomes the new x_1.
- In either case, only one new interior point is needed and the function is only evaluated one more time.



Error in golden-section search

The error is estimated from the largest interval.

For the interval $[x_2, x_1]$, the error is

$$\Delta x = (x_1 - x_2) = (2\phi - 3)(x_u - x_l)$$
$$= 0.236(x_u - x_l)$$

For the interval $[x_1, x_u]$, the error is

$$\Delta x = (x_u - x_1) = (2 - \phi)(x_u - x_l) = 0.382(x_u - x_l)$$

Note that each iteration reduce the interval by $1/\phi$.

Hence the interval after n iteration is $\Delta x_n = \Delta x_0/\phi^n$.

From above, the error after n iterations is

$$E_a = (2 - \phi)\Delta x_n = (2 - \phi)\Delta x_0/\phi^n$$

Hence number of iterations is

$$n = \log[(2 - \phi)\Delta x_0)\,/E_a]/\log(\phi)$$

To show reduction by $1/\phi$

Initial interval $\Delta x_0 = (x_u - x_l)$

$$d = (\phi - 1)(x_u - x_l)$$

Recall $\phi^2 - \phi - 1 = 0 \Rightarrow \phi - 1 = \frac{1}{\phi}$

Hence $d = \frac{1}{\phi}\Delta x_0$, New interval is

$$\Delta x_1 = (x_u - x_2) = d = \frac{1}{\phi}\Delta x_0$$

OR

$$\Delta x_1 = (x_1 - x_l) = d = \frac{1}{\phi}\Delta x_0$$

```python
import numpy as np
def goldmin(f,xl,xu,Ea=1.e-7,maxit=30):
    """
    use the golden-section search to find the minimum of f(x)
    input:
    f = name of the function
    xl = lower initial guess
    xu = upper initial guess
    Ea = absolute relative error criterion (default = 1.e-7)
    maxit = maximum number of iterations (default = 30)
    output:
    xopt = location of the minimum
    f(xopt) = function value at the minimum
    ea = absolute relative error achieved
    i+1 = number of iterations required
    """
    phi = (1+np.sqrt(5))/2
    d = (phi - 1)*(xu-xl)
    x1 = xl + d ; f1 = f(x1)
    x2 = xu - d ; f2 = f(x2)
    for i in range(maxit):
        xint = xu - xl
        if f1 < f2:
            xopt = x1
            xl = x2
            x2 = x1
            f2 = f1
            x1 = xl + (phi-1)*(xu-xl)
            f1 = f(x1)
        else:
            xopt = x2
            xu = x1
            x1 = x2
            f1 = f2
            x2 = xu - (phi-1)*(xu-xl)
            f2 = f(x2)
        if xopt != 0:
            ea = (2-phi)*abs(xint/xopt)
            if ea <= Ea: break
    return xopt,f(xopt),ea,i+1
```

Example

Find the maximum of the function $x\ \sin(x)$ using goldmin

```python
import math
def example_goldmin():
    # Define the function to minimize
    f = lambda x: -x * math.sin(x)

    # Set the lower and upper bounds
    xl = 0
    xu = math.pi

    # Call the golden section search to find the minimum
    xmax, fx, ea, iter = goldmin(f, xl, xu)

    # Print results
    print(f"Maximum x: {xmax}")
    print(f"Function value at max x: {fx}")
    print(f"Approximate relative error: {ea}%")
    print(f"Number of iterations: {iter}")

# Run the example
example_goldmin()
```

```
Maximum x: 2.028758190742389
Function value at max x: -1.819705741159485
Approximate relative error: 5.144021069215409e-07%
Number of iterations: 30
```

- minimize_scalar function in the optimize submodule of Python’s SciPy module.

- The function minimize_scalar is intended for functions of a single variable

```
from scipy.optimize import minimize_scalar
result = minimize_scalar(funx)
xmin = result.x
```

The minimize function is the “big brother” of the minimize_scalar function in the optimize submodule of SciPy. The syntax for using the function is

```
from scipy.optimize import minimize
result = minimize(f,x0)
xmin = result.x
```

The initial guess is x0. For multidimensional optimization, x0 is an array of initial guesses and xmin is an array of independent variable values for the minimum.

Example `minimize`

- Use `minimize` to find the maximum of

  $Z = 2 + X - Y + 2*X**2 + 2*X*Y + Y**2$

- Plot the function in the interval

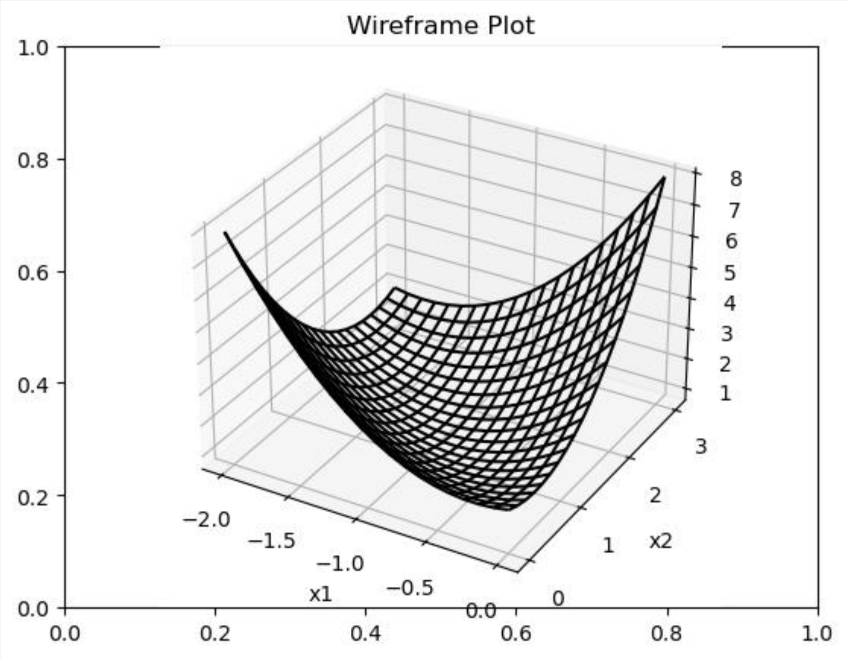
```
from scipy.optimize import minimize
def f(x):
    x1 = x[0]
    x2 = x[1]
    return 2+x1-x2+2*x1**2+2*x1*x2+x2**2

x0 = [-0.5,0.5]
result = minimize(f,x0,method='Nelder-
Mead',options={'disp':False})
xval = result.x
print(xval)
```

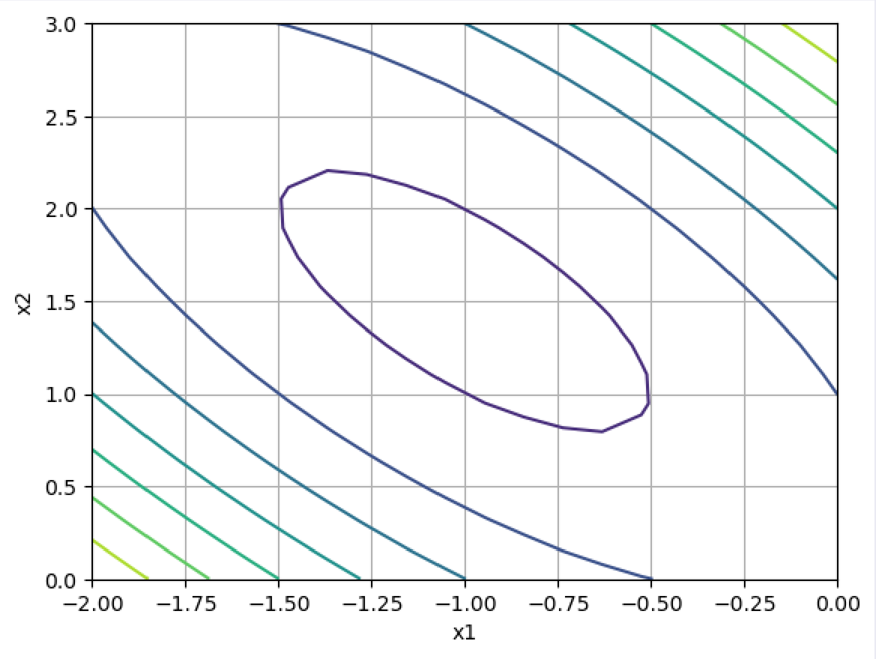
```
[-0.99996784 1.49997544]
```

```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
x1 = np.linspace(-2,0,20)
x2 = np.linspace(0,3,20)
X,Y = np.meshgrid(x1,x2)
Z = 2 + X - Y + 2*X**2 + 2*X*Y + Y**2
fig = plt.figure()
ax = plt.subplot(111)
ax = fig.add_subplot(111,projection='3d')
ax.plot_wireframe(X,Y,Z,color='k')
ax.set_xticks([-2,-1.5,-1,-0.5,0])
ax.set_yticks([0,1,2,3])
ax.set_xlabel('x1')
ax.set_ylabel('x2')
ax.set_title('Wireframe Plot')
plt.show()
fig2 = plt.figure()
ax2 = fig2.add_subplot(111)
ax2.contour(X,Y,Z)
ax2.set_xlabel('x1')
ax2.set_ylabel('x2')
ax2.grid()
plt.show()
```

Multidimensional Visualization

Functions of two-dimensions may be visualized using contour or surface/mesh plots.





Homework

- Problem 7.23 (Vertical motion including air resistance)
- In addition to the numerical solution, determine the analytic solution and compare the two values.
- Determine the time of flight (use a root finding method)
- Plot y vs t during the time of flight.

2^{nd} Homework

- Use `minimize` to find the maximum of $x \sin(x)$ in the interval $[0, \pi]$.

- Plot the function in the interval $[0, \pi]$.