Today

I. Root Finding in Python

   I. The goal of root finding is to solve non-linear equations that we can't solve analytically.

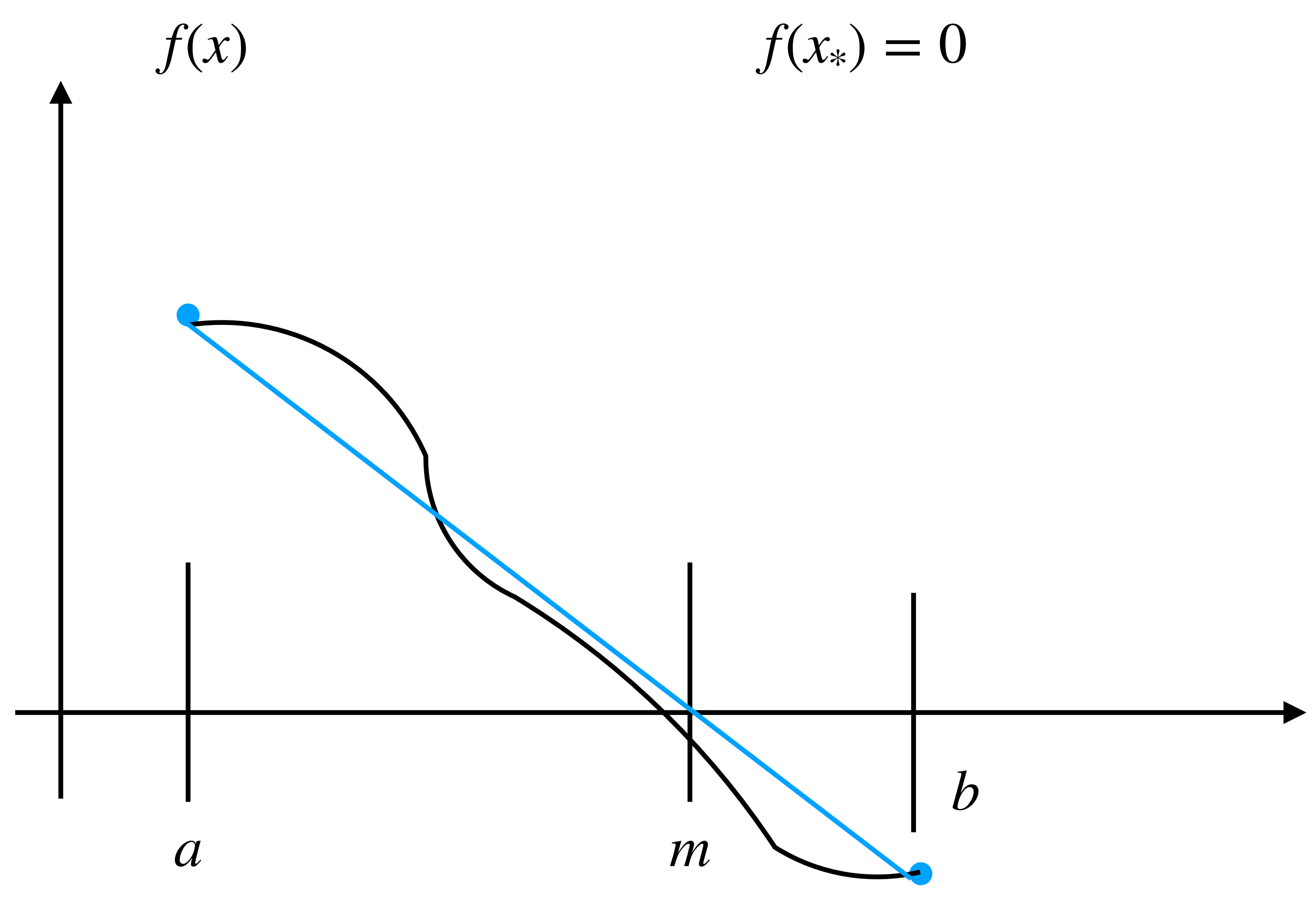
What's a nonlinear equation you can solve:

$$ax^2 + bx + c = 0 \qquad \implies \qquad x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Already when you get to quintic equations, you can prove that there's no analytic formula for the roots.

How do you use python to find numerical answers for the roots? (Note that for the algorithm below to work, you need to wrap the whole thing in a function definition. Alternatively, you can change the return statements to print statements.) First method: bisection.

II. Intermediate value theorem.



If f(a)*f(b)>=0

print(“Bisection method fails.”)

Return None

a_n = a

b_n= b

II. Intermediate value theorem.

```
a_n = a, N=10
b_n= b
for n in range(1,N+1):
  m_n = (a_n+b_n)/2
  fatm_n = f(m_n)
  if f(a_n)*fatm_n <0:
     a_n=a_n
     b_n=m_n
  elif f(b_n)*fatm_n < 0:
     a_n=m_n
     b_n=b_n
  elif fatm_n==0:
    print("Found exact solution.")
    return m_n
```

```
for n in range(1,N+1):
  m_n = (a_n+b_n)/2
  fatm_n = f(m_n)
  if f(a_n)*fatm_n <0:
     a_n=a_n
     b_n=m_n
  elif f(b_n)*fatm_n < 0:
     a_n=m_n
     b_n=b_n
  elif fatm_n==0:
    print("Found exact solution.")
    return m_n
  Else:
    print("Bisection method fails.")
    return None
  Return (a_n+b_n)/2
```

Let's try it on a fun quadratic

$$f(x) = x^2 - x - 1$$

```
def f(x):
    return x**2-x-1
```