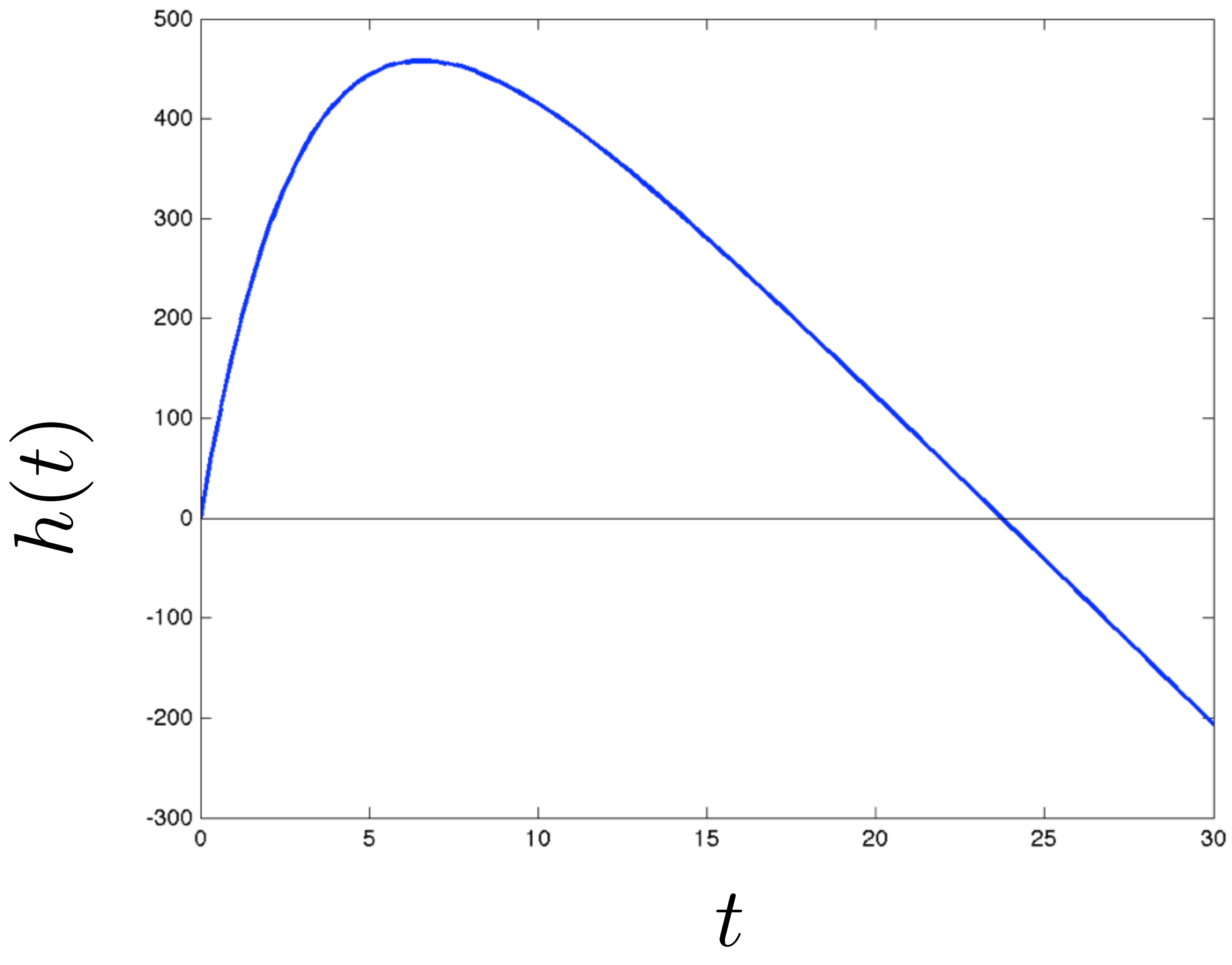
Numerical Root Finding

- The Bisection Method

Motion with air resistance

Vertical height of a rocket with air resistance



$$h(t) = -33t + 784(1 - e^{-0.3t})$$

Finding the maximum height

It is easy (for this problem at least) to find the maximum height reached by the rocket.

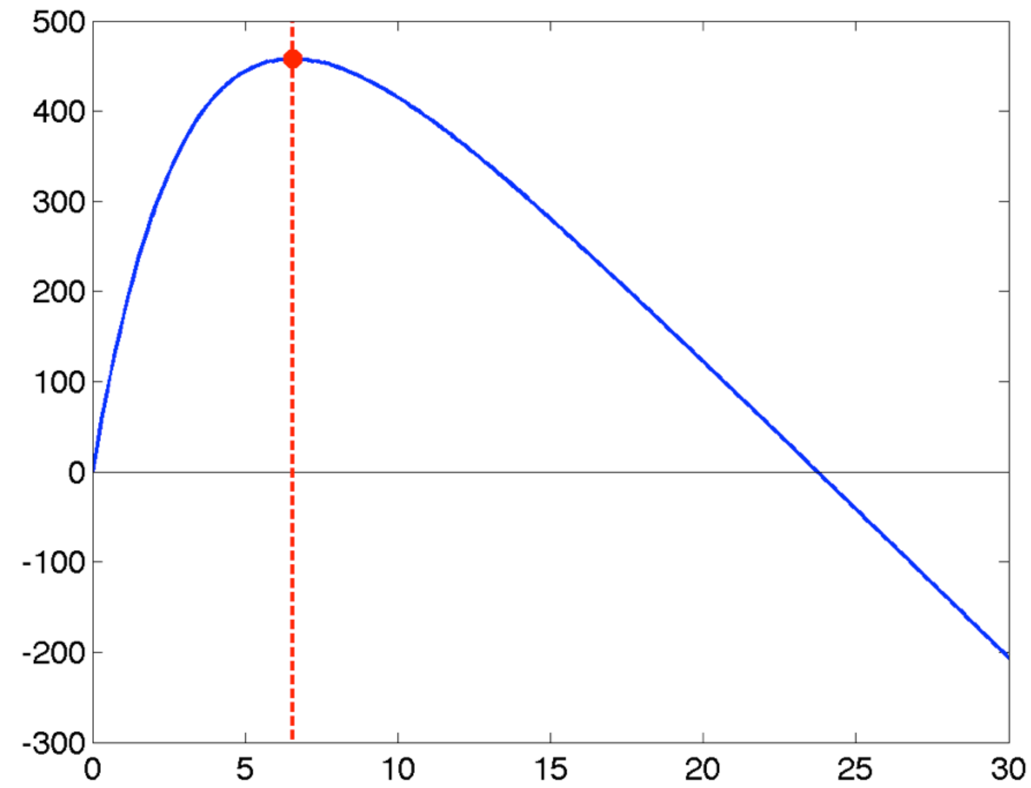
Set the derivative $h'(t)$ equal to zero and solve for t :

$$h'(t) = -33 + (0.3)(784)e^{-0.3t} = 0$$

$$\rightarrow \quad e^{-0.3t} = \frac{33}{(0.3)(784)}$$

$$\rightarrow \quad -0.3t = \log\left(\frac{33}{0.3 \cdot 784}\right)$$

$$\rightarrow \quad t \approx 6.546428\ldots$$



This was easy to solve for analytically because we have convenient inverse functions (i.e. log)

Time spent in the air

It is harder (for this problem) to find the time at which the rocket hits the ground.

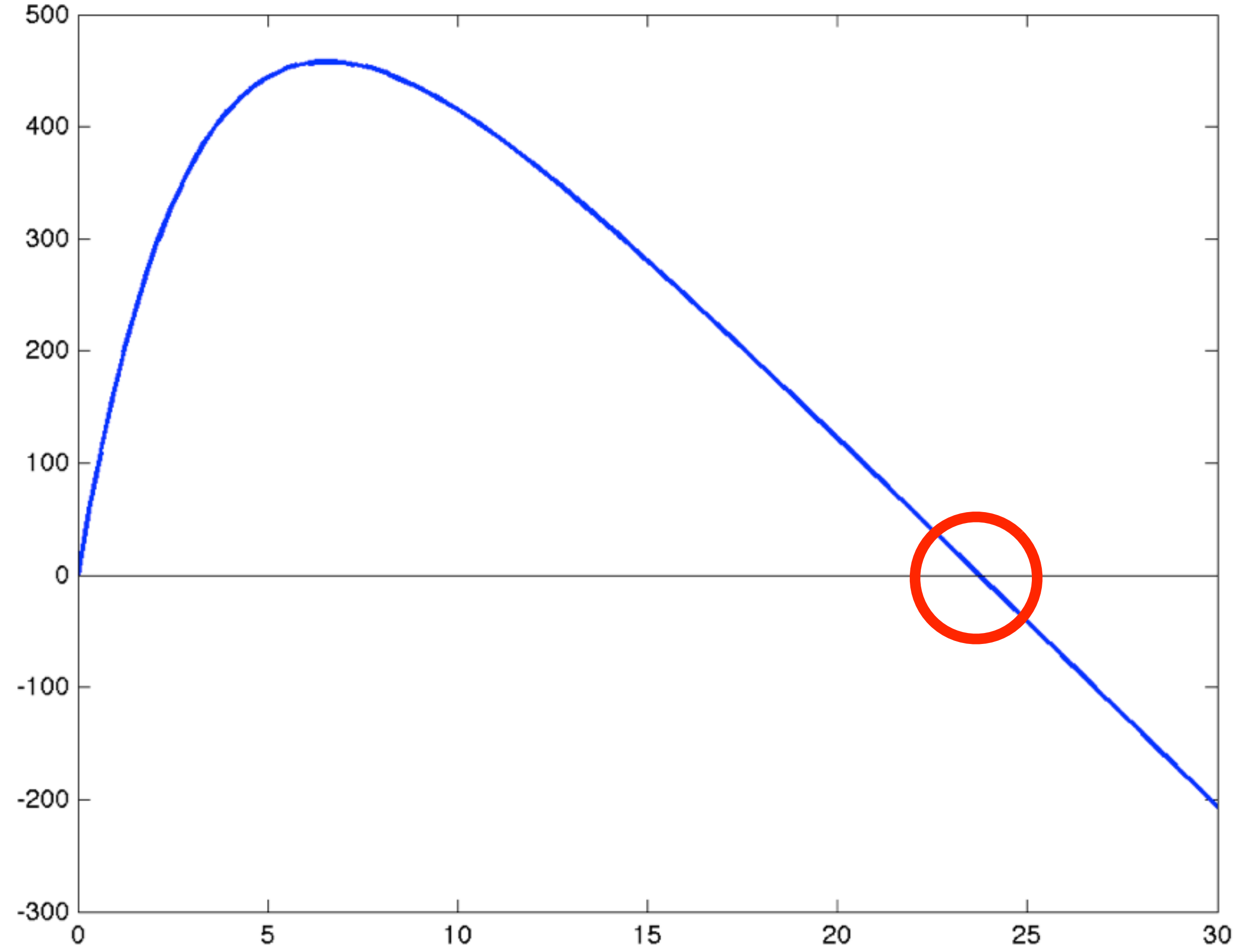
We have to set $h(t) = 0$ and solve for t :

$$h(t) = -33t + 784(1 - e^{-0.3t}) = 0$$
$$\rightarrow \quad \frac{33}{784}t + e^{-0.3t} = 1$$

It does not seem that we can solve this with known elementary functions and their inverses.

So we will attempt to find the root of h(t) numerically.

Numerical root finding



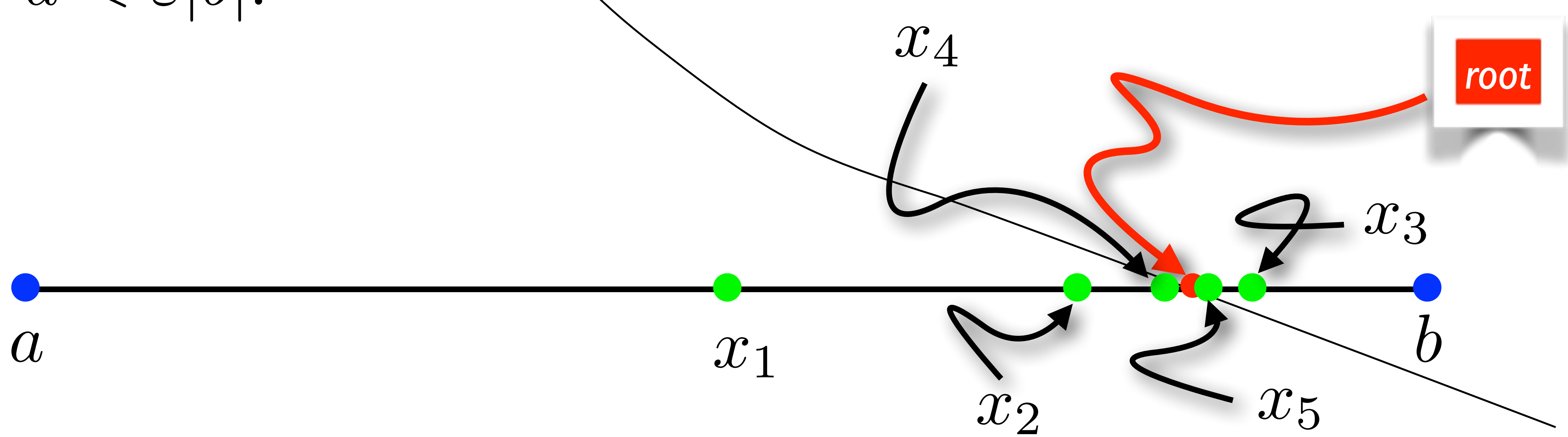
$$h(t) = -33t + 784(1 - e^{-0.3t})$$

= 134

Bisection algorithm

The first algorithm we might consider is the *Bisection* algorithm.

We first find an interval $[a, b]$ over which we know the function changes sign. Assuming the function is continuous, we then know that our function contains a zero in the interval. We attempt to isolate the zero by successively cutting the interval $[a, b]$ in half until $b - a < \varepsilon|b|$.



Convergence of Bisection

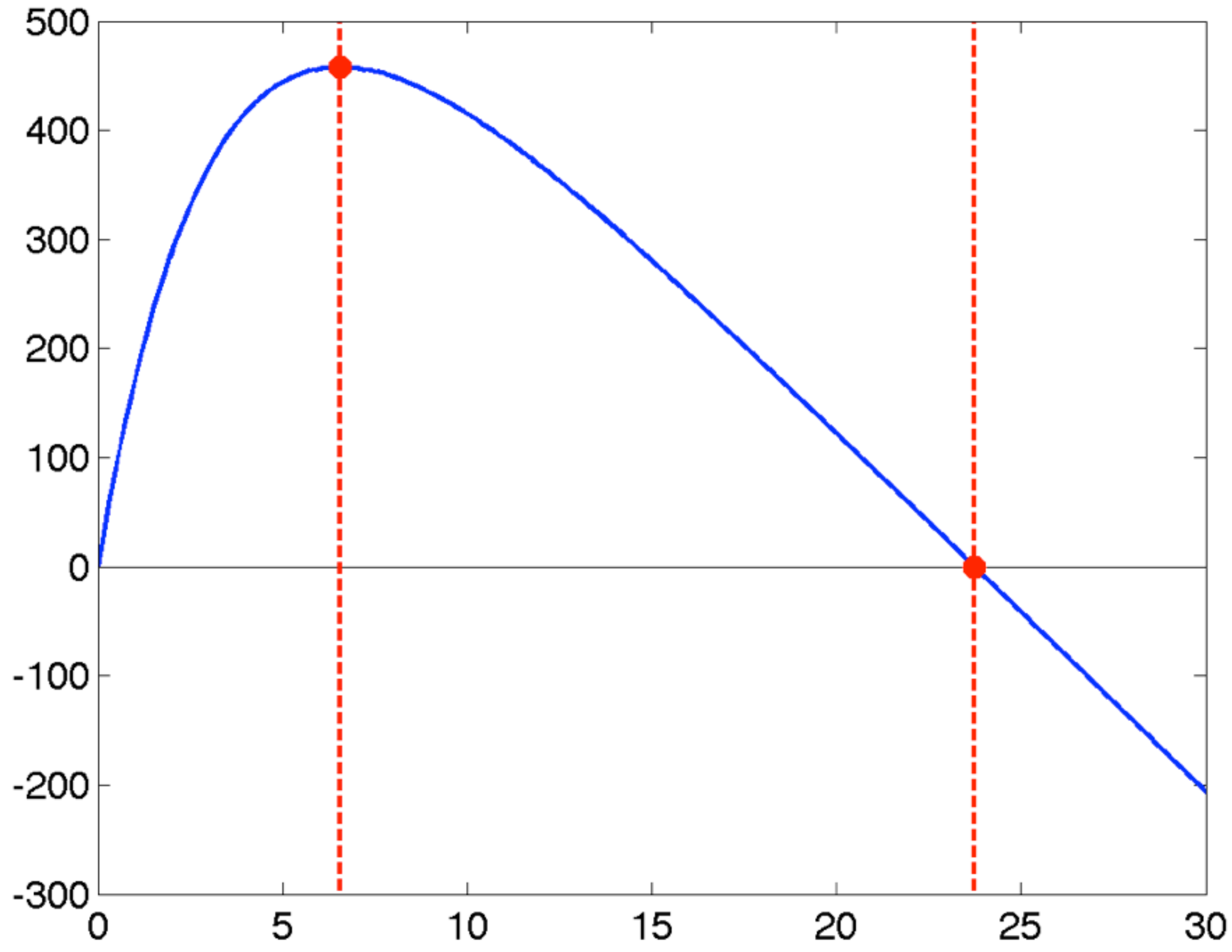
The bisection algorithm is guaranteed to converge.
The number of steps it needs to subdivide the interval down to machine epsilon can be computed as follows :

$$2^{-n}|b_0 - a_0| \leq \varepsilon_0$$

$$\rightarrow \quad n \geq \frac{\log\left(\frac{b_0 - a_0}{\varepsilon_0}\right)}{\log(2)}$$

where n is the number of steps it takes to converge, $[a_0, b_0]$ is the initial interval, and ε_0 is the value of machine epsilon near the b_0, i.e. the smallest floating point value such that $b_0 + \varepsilon_0 \neq b_0$.

Time in the air - result



At approximate time $t = 23.738$, the rocket hits the ground.

Bisection - advantages

What are the advantages of the bisection algorithm?

- Easy to program
- We will always get an answer provided we can supply an initial interval
- We can estimate how many iterations we need for convergence
- Has minimal requirements - we only need a continuous function

Bisection is what we like to call a "robust" algorithm.

Bisection - drawbacks?

Are there any disadvantages to the bisection algorithm?

- Need to find an interval containing a root
- Multiple roots?
- Can we do better than just halving the size of the interval?
- Does not take into account any additional information we may have about our algorithm.
- Stopping criteria?