Introduction to Programming and Computational Physics

Lecture 9

Root finding in one dimension:

The bisection method

The Newton-Raphson method

Bibliography

W. H. Press, B. P. Flannery, S.A. Teukolsky, W. V. Wetterling
Numerical recipes in C, the art of scientific computing
http://www.nrbook.com/a/bookcpdf.php

Chapter 9: root finding

Chapter 4: integration

Chapter 16: differential equations

W. Benz
Numerical Methods – Lecture notes (old course)

General problem

Finding roots of a given function means finding those values for which we have f(x) = 0. We will treat cases where the analytical way is not possible

We will focus on real roots. There could be no roots at all, one or more or an infinite number of roots.

We will see how to evaluate an approximate value of the root.

Example:

Let's calculate the root of the function

$$f(x) = e^{-x} - x$$

It's quite easy to draw the graph of the function, given that

$$f(0) = 1 \qquad f(1) = \frac{1}{e} - 1 \cong -0.63$$

$$\lim_{x \to -\infty} f(x) = \lim_{x \to -\infty} e^{-x}$$

$$\lim_{x \to +\infty} f(x) = \lim_{x \to +\infty} -x$$

You could use the software "Root" to draw it

```
void fun_draw()
{
   TF1 *f1 = new TF1("f1","exp(-x)-x",-3,3);
   f1->Draw();
}
```

$f(0) > 0 \quad f(1) < 0$ and the function is continuous

There must be a root somewhere in the interval (0,1)

Preliminary remarks

Root finding involves always some form of iteration

Convergence is not always guaranteed

Multiple or very close roots can fool a program easily

The initial guess(es) and a good prior knowledge of the function are important

There are several methods. They differ essentially by the rate at which they converge toward the solution

The bisection method

If f(x) is continuous in (a,b) and $f(a) \bullet f(b) < 0$

there must be a root in the interval (a,b)

Let's consider the mean value $c = \frac{a+b}{2}$

and evaluate f(c). If f(c) has the same sign of f(a) then the root will be in the interval (c,b), so we replace a with c. Otherwise the root will be in the interval (a,c) and we replace b with c.

In both cases the interval is reduced by a factor 2.

Iterating the process we will obtain the root with the desired precision. Initial precision will be indeed ε_0 = (b-a) and at each step we will have

$$\varepsilon_{n+1} = \frac{\varepsilon_n}{2}$$

Let's go back to our function $f(x) = e^{-x} - x$

As $f(0) > 0$ and $f(1) < 0$ $\varepsilon_0 = 1$

Step 1: $f(1/2) > 0$ new interval (1/2, 1) $\varepsilon_1 = \frac{1}{2}$

Step 2: $f(3/4) < 0$ new interval (1/2, 3/4) $\varepsilon_2 = \frac{1}{4}$

Step 3: $f(5/8) < 0$ new interval (1/2, 5/8) $\varepsilon_3 = \frac{1}{8}$

Step 4: $f(9/16) > 0$ new interval (9/16, 5/8) $\varepsilon_4 = \frac{1}{16}$

And so on… until we reach the desired precision

Precision of the bisection method

If we start from a generic interval (a,b) the precision after

First step will be $\frac{(b-a)}{2}$

Second step $\frac{(b-a)}{2} \bullet \frac{1}{2} = \frac{(b-a)}{4}$

N-th step $\frac{(b-a)}{2^N}$

In other words, if we want to obtain a precision ε the number of iteration needed will be

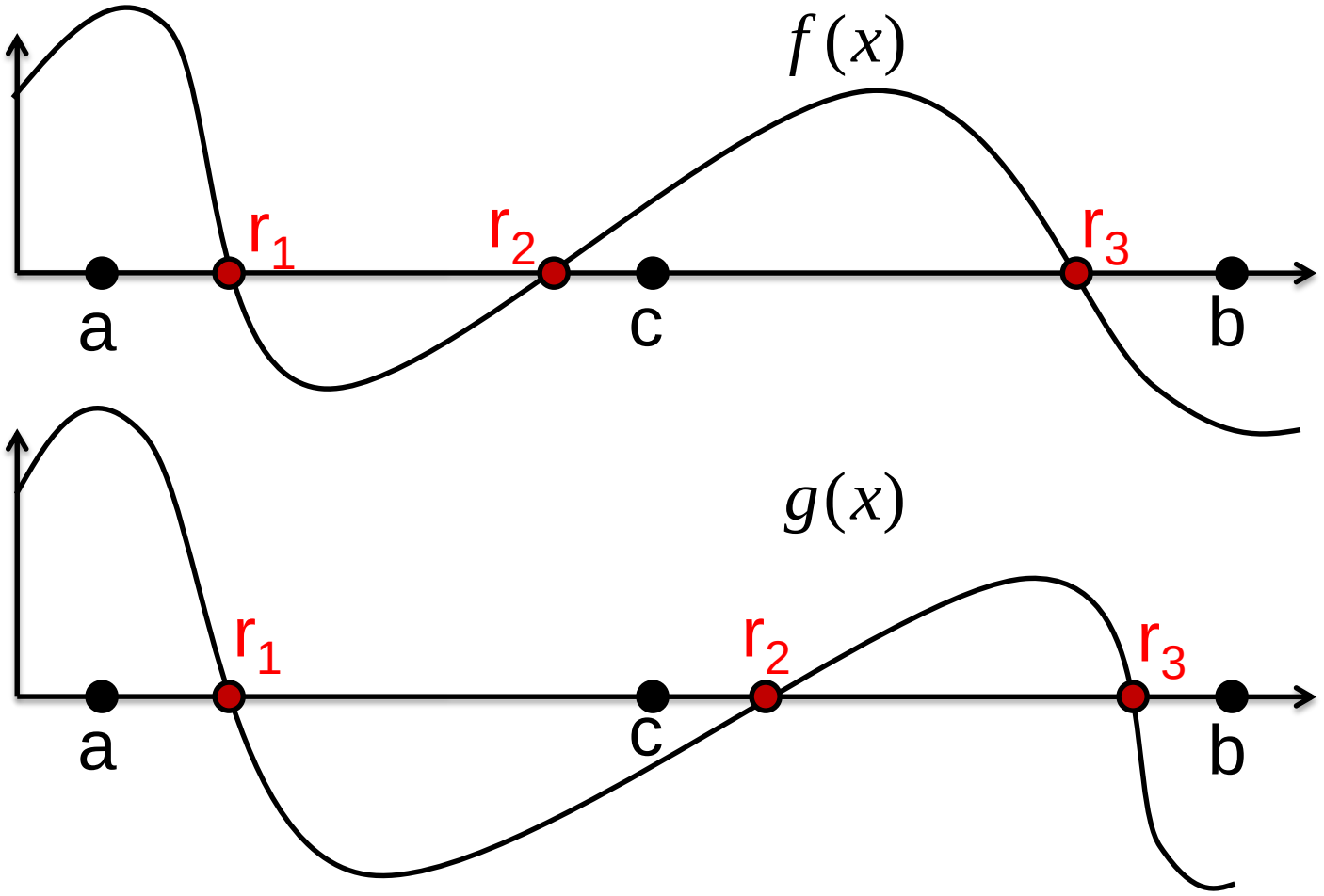
$$N = \log_2 \frac{(b-a)}{\varepsilon}$$

Remarks

Provided that the function is continuous, the bisection method always converges to the solution, but very slowly.

If the function is easy to evaluate, this is not a problem. When we deal with a function which is complicated to evaluate, the process could require a significant amount of CPU time.

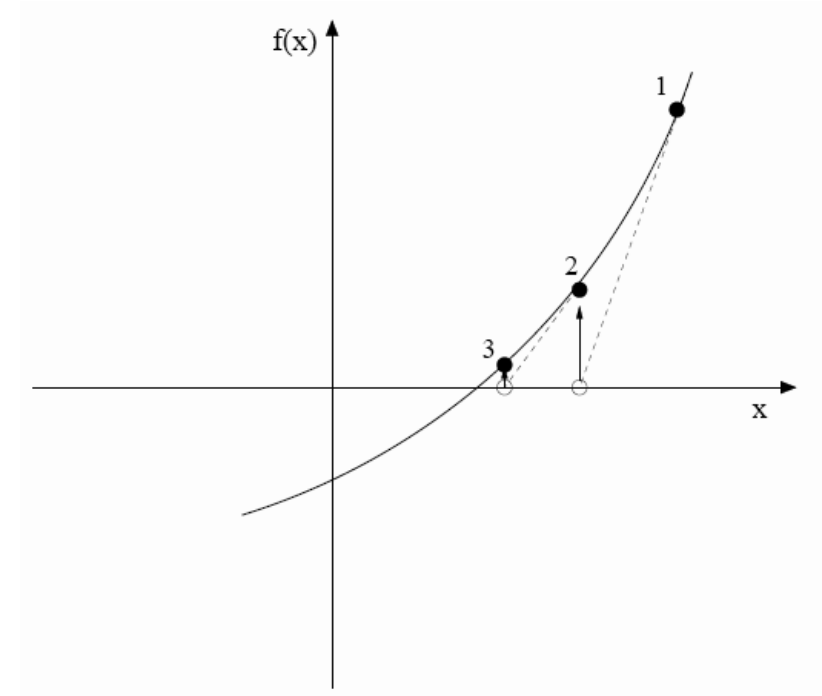
Multiple roots



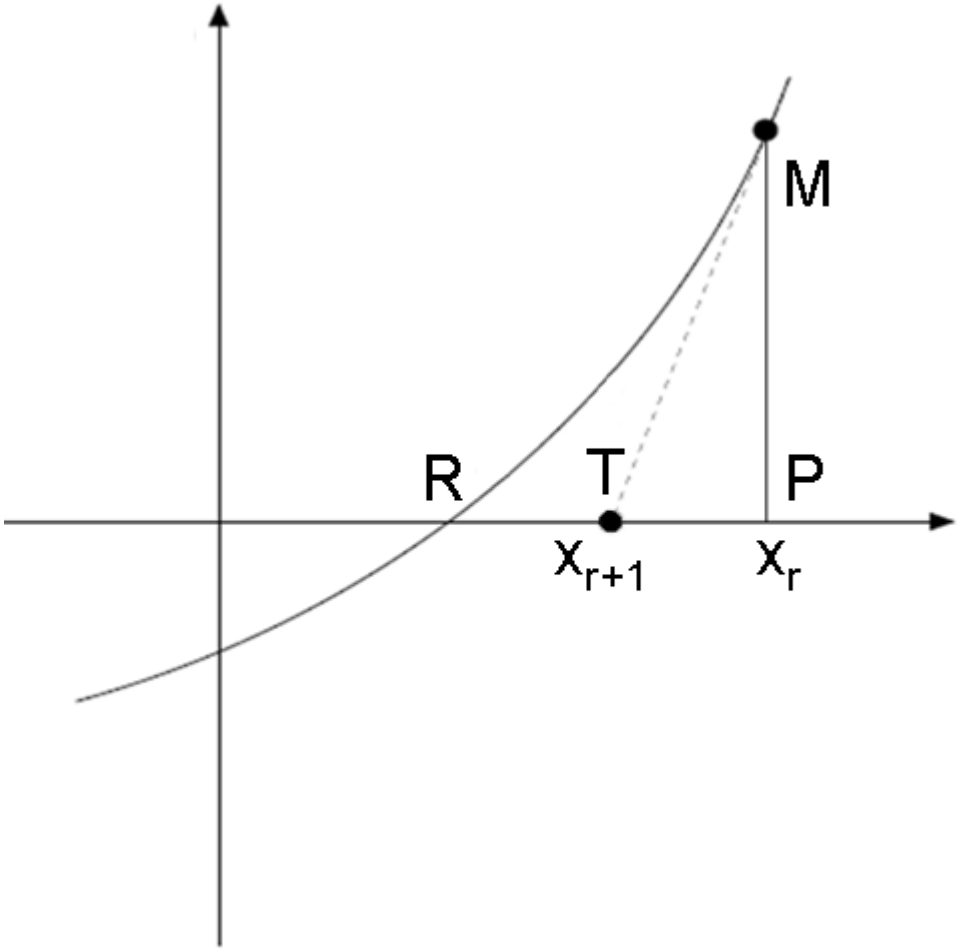
Starting from a,b the method will converge to r_3 for f(x) and to r_1 for g(x). What if we want to find all the roots?

The Newton-Raphson method

It improves the convergence of the root finding method in case not only the function f(x) but also its derivative f’(x) can be evaluated at arbitrary locations.



The idea is to extend the tangent line at the current point to find where it crosses the abscissa and to use this new point as the start of the next iteration



$$\overline{PM} = f(x_r)$$

$$\tan \hat{PTM} = f'(x_r)$$

$$\overline{TP} = \frac{f(x_r)}{f'(x_r)}$$

And the formula to be used for Newton-Raphson method is:

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)}$$

Precision of the Newton-Raphson method

Let's assume $x_r = \alpha + \varepsilon_r$ where α is the exact value of the root

Taylor expansion:

$$f(\alpha) = f(x_r - \varepsilon_r) = f(x_r) - \varepsilon_r f'(x_r) + \tfrac{1}{2}\varepsilon_r^2 f''(x_r) - \ldots$$

But $f(\alpha) = 0$ (α is a root)

$$f(x_r) = \varepsilon_r f'(x_r) - \tfrac{1}{2}\varepsilon_r^2 f''(x_r) + \ldots$$

The Newton-Raphson's formula is $x_{r+1} = x_r - \dfrac{f(x_r)}{f'(x_r)}$

$$x_{r+1} = (\alpha + \varepsilon_r) - \varepsilon_r \frac{f'(x_r)}{f'(x_r)} + \frac{1}{2}\varepsilon_r^2 \frac{f''(x_r)}{f'(x_r)} - \ldots$$

$$x_{r+1} = \alpha + \frac{1}{2}\varepsilon_r^2 \frac{f''(x_r)}{f'(x_r)} - \ldots$$

So that $\varepsilon_{r+1} = x_{r+1} - \alpha \cong \frac{1}{2}\varepsilon_r^2 \frac{f''(x_r)}{f'(x_r)} \cong \frac{1}{2}\varepsilon_r^2 \frac{f''(\alpha)}{f'(\alpha)}$

If α e' is enough close to x_r

The Newton-Raphson method converges quadratically and the bisection method linearly

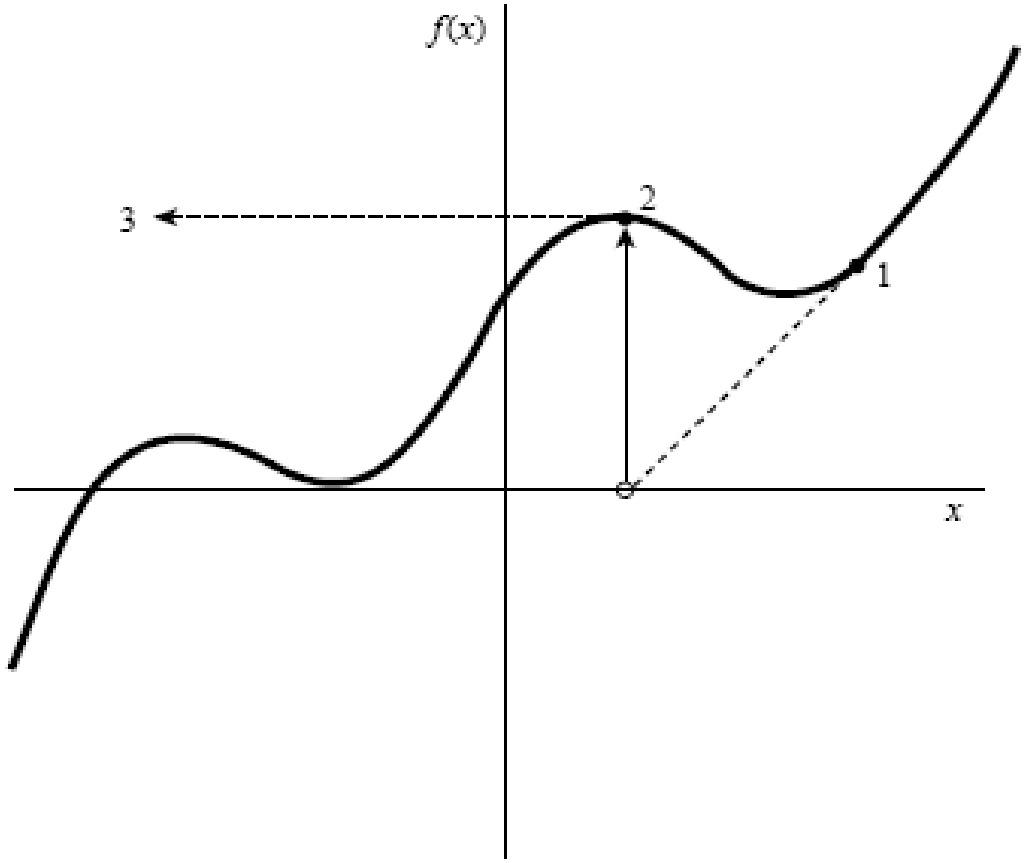
Newton-Raphson is second order process $\varepsilon_{r+1} \cong \frac{1}{2}\varepsilon_r^2 \frac{f''(\alpha)}{f'(\alpha)}$

Bisection is a first order process $\varepsilon_{r+1} = \frac{1}{2}\varepsilon_r$

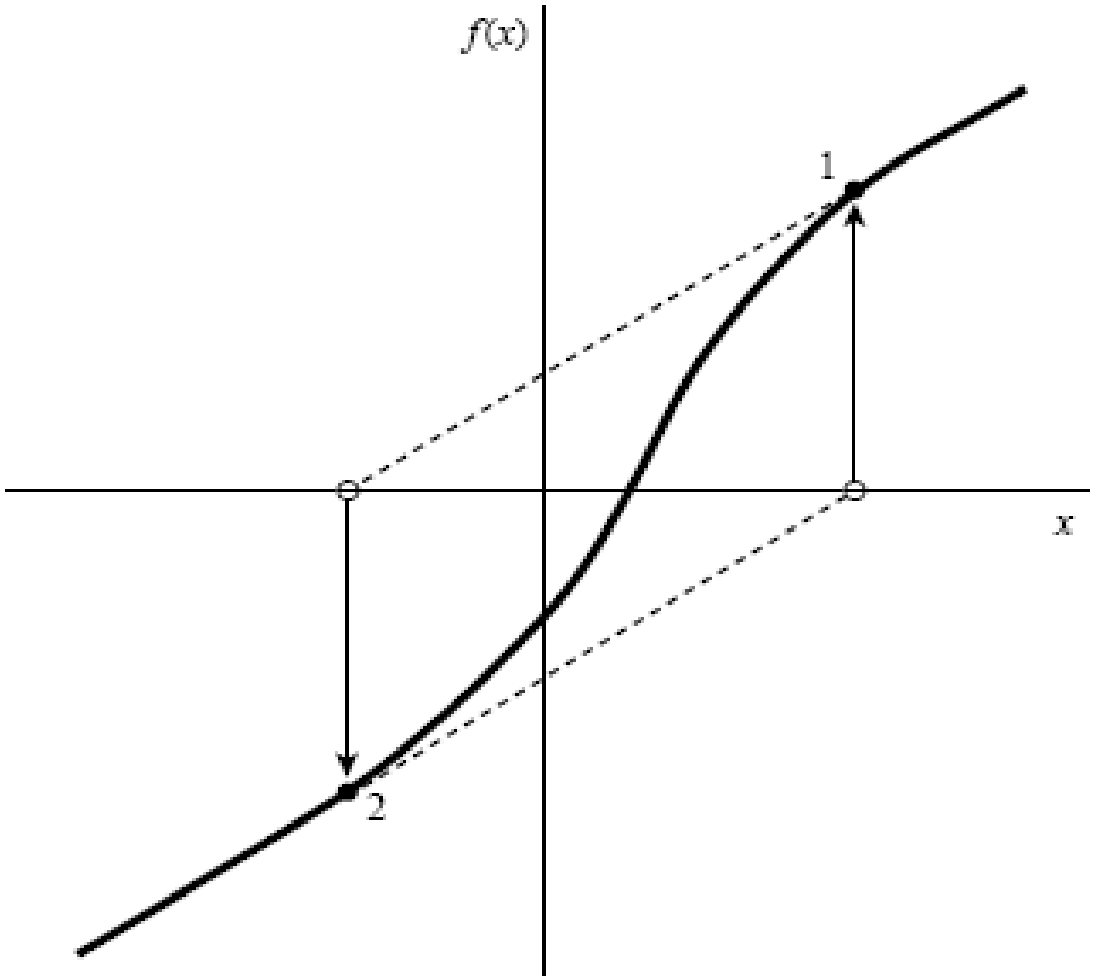
For the bisection method, the error after N iterations can be precisely calculated. For the Newton-Raphson method this is instead not possible.

The precision is evaluated comparing the current value x_{r+1} with the previous value x_r

If derivative is null your program will crash… (division by zero, segmentation fault…). Be careful…



Or if you have a inflexion point your program could fall in an infinite loop and it will never give a result…



Once again… a good prior knowledge of the function can help for the choice of the method and of the initial guess