Numerical Methods, project B, Number 32

Krzysztof Rudnicki
Student number: 307585
Advisor: dr Adam Krzemieniowski

December 2, 2021

Contents

Chapter 1

Find all zeros of function

1.1 a) False position method

1.1.1 Problem

We have to find zeros of the function

$$f(x) = -2.1 + 0.3x - xe^{-x}$$

In the interval $[-5; 10]$ using false position method.

1.1.2 Theoretical Introduction

False position method also called *regula falsi* in fancier circles is similar to the bisection method, the difference is that the interval we use $[a_n, b_n]$ is divided into two subintervals. We have:

- α - The root

- a_n - 'left' interval

- b_n - 'right' interval

- $f(a_n)$ - Value at left interval

- $f(b_n)$ - Value at right interval

We get:

$$\frac{f(b_n) - f(a_n)}{b_n - a_n} = \frac{f(b_n) - 0}{b_n - c_n}$$

From which we get:

$$c_n = b_n - \frac{f(b_n)(b_n - a_n)}{f(b_n) - f(a_n)} = \frac{a_n f(b_n) - b_n f(a_n)}{f(b_n) - f(a_n)}$$

Then we choose next interval as in the bisection method so: we calculate products of function values at a_n and b_n and that subinterval is selected for the next iteration of *false position* method. This subinterval corresponds to the negative product value.

Properties of *false position method*

This method is always convergent, simillary to bisection method, since it will always choose and shorten the interval which contains the root. If the function is continous and differentiable the method is linearly convergent. That being said the convergence may become sluggish. It can happen if for example one of the endpoints of the intervals will remain the same and the iterating will not shorten the interval to 0. One of the examples of functions that lead to that are barrier functions used in constrained optimization methods.

Improvement to the method In order to improve the formula and avoid aforementioned situation we can take smaller value of the function for the value that does not change. For right end:

$$c_n = \frac{a_n \frac{f(b_n)}{2} - b_n f(a_n)}{\frac{f(b_n)}{2} - f(a_n)}$$

And for left end:

$$c_n = \frac{a_n f(b_n) - b_n \frac{f(a_n)}{2}}{f(b_n) - \frac{f(a_n)}{2}}$$

This is called *modified regula falsi* or *Illinois algorithm*. It is superlinearly convergent, globally convergent and length of intervals we get in each iterations converges to zero.

1.1.3 Results

1.2 b) the Newton's method

1.2.1 Problem

We have to find zeros of the function

$$f(x) = -2.1 + 0.3x - xe^{-x}$$

In the interval $[-5; 10]$
    using the Newton's method

1.2.2 Theoretical Introduction

The Newton's method also called *the tangent method* relies on first order part
of its expansion into Taylor series for a given current approximation of root.

$$f(x) \approx f(x_n) + f'(x_n)(x - x_n)$$

Then we obtain the next point x_{x+1} by finding root of linear function:

$$f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0$$

From this we get formula for x_{n+1}:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

This method as opposed to *regula falsi* method is locally convergent,
should we choose initial point too far from the root (area which is close
enough to root is called set of attraction) then we can get a divergence. On
the other side if the Newton's method will converge then it is quite rapid
with convergence of order p $= 2$ - quadratic convergence.

Newton's method is also effective if the function derrivative is far from
zero, so the slope of the function is steep, conversely if the derrivative is close
to zero the method is not recommended.

1.2.3 Results

Chapter 2

Find real and complex roots of the polynomial

2.1 Problem

We have to find all real and complex roots of the polynomial:

$$f(x) = a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

where:

$$[a_4 \ a_3 \ a_2 \ a_1 \ a_0] = [-2 \ 12 \ 4 \ 1 \ 3]$$

So our polynomial looks like this:

$$f(x) = -2x^4 + 12x^3 + 4x^2 + 1x + 3$$

Using the Müller's method. We have to implement both MM1 and MM2 versions. We also need to find real roots using the Newton's method and compare these results with what we got from MM2 version of the Müller's method.

2.2 Theoretical Introduction

Müller's method revoles around the idea of approximating the polynomial locally close to the root by a quadratic function. Based on three different points we can use quadratic interpolation and develop our method. This means that we can treat it as a generalization of secant method. That being

said we can also realize it in an efficient way if we use just one point. We can use for this case values of polynomial, and its first and second derrivative at current point.

Accordingly there are two versions of Müller's method: **MM1** and **MM2**.

2.2.1   MM1

Given three points: $x_0; x_1; x_2$ and their polynomial values: $f(x_0), f(x_1), f(x_2)$ we construct a (quadratic) function passing through these points. Then we find roots of this parabola and we choose one of these rots for the approximation of the result.

For example: Assume that x_2 is the approximation of the root. Let's introduce variable z such that:

$$z = x - x_2$$

And differences:

$$z_0 = x_0 - x_2$$
$$z_1 = x_1 - x_2$$

We have quadratic function:

$$y(z) = az^2 + bz + c$$

Using three points from above we get:

$$y(z_0) = az_0^2 + bz_0 + c = f(x_0)$$

$$y(z_1) = az_1^2 + bz_1 + c = f(x_1)$$

$$y(z_2) = c = f(x_2)$$

And then we get system of equation that we can solve to find a and b:

$$az_0^2 + bz_0 = f(x_0) - f(x_2)$$

$$az_1^2 + bz_1 = f(x_1) - f(x_2)$$

Roots are equal to:

$$z_+ = \frac{-2c}{b + \sqrt{b^2 - 4ac}}$$

$$z_- = \frac{-2c}{b - \sqrt{b^2 - 4ac}}$$

We choose a root with smaller absolute value for next iteration:

$$z_{min} = \min |z_+, z_-|$$

$$x_3 = x_2 + z_{min}$$

Then we choose new point x_3 and two selected from x_0, x_1, x_2 which were closer to x_3.

This method should also work for $\Delta < 0$

2.2.2   MM2

This method being numerically more effective is usually recommended. We calculate values of a polynomial and its first and second derrivatives at one point.

from definition of quadratic function:

$$y(z) = az^2 + bz + c$$

we can get:

$$z = x - x_k$$

If $z = 0$ then:

$$y(0) = c = f(x_k)$$
$$y'(0) = b = f'(x_k)$$
$$y''(0) = 2a = f''(x_k)$$

We can derive from that formula for roots:

$$z_\pm = \frac{-2f(x_k)}{f'(x_k) \pm \sqrt{(f'(x_k))^2 - 2f(x_k)f''(x_k)}}$$

Then we choose root with smaller absolute value for next iteration:

$$x_{k+1} = x_k + z_{min}$$

Again this method should be implemented in complex number arithmetic. This method is locally convergent with order of convergence equal to 1.84. It is locally more effective that secant method and it is almost as fast as Newton's method while being capable of finding complex roots. It can be used to find roots of polynomials or another nonlinear functions.

2.3   Results

2.3.1   Comparison of results between MM1 and MM2

2.3.2   Comparison of results between Newton's method and MM2

Chapter 3

Find real and complex roots of the polynomial using Laguerre's method

3.1 Problem

We have to find all (real and complex) roots of the polynomial from previous exercise:
$$f(x) = -2x^4 + 12x^3 + 4x^2 + 1x + 3$$

Using the Laguerre's method. Then we should compare those results with the MM2 version of the Müller's method.

3.2 Theoretical Introduction

Laguerre's method is defined by a single formula:

$$x_{k+1} = x_k - \frac{nf(x_k)}{f'(x_k) \pm \sqrt{(n-1)[(n-1)((f'(x_k))^2 - nf(x_k)f''(x_k))]}}$$

Where: n - order of the polynomial

This formula is similar to the one from MM2 but also takes order of the polynomial into consideration. In general this method is better. For polynomials with real roots it is globally convergent. It does not have formal

analysis for complex roots but it usually shows good numerical properties, although divergence may happen.

3.3   Results

3.3.1   Comparison of results between MM1 and MM2

Chapter 4

Code appendix

Bibliography

[1] Piotr Tatjewski (2014) *Numerical Methods*, Oficyna Wydawnicza Politechniki Warszawskiej