MCS 471 Project One: Laguerre's Method

The goal of the project is to study the method of Laguerre with Julia programs.

0. Derivation of Laguerre's Method

Let p be a polynomial in one variable x of degree d. Then its highest degree coefficient c_d is nonzero. Assume $c_d = 1$, otherwise, divide every coefficient of p by c_d. By the fundamental theorem of algebra, we may then write p as

$$p(x) = (x - r_1)(x - r_2) \cdots (x - r_d), \quad r_k \in \mathbb{C}, \quad k = 1, 2, \ldots, d, \tag{1}$$

where r_1, r_2, …, r_d are the roots of p. Consider the natural logarithm of p:

$$L(x) = \ln(p(x)) = \ln(x - r_1) + \ln(x - r_2) + \cdots + \ln(x - r_d) \tag{2}$$

and its derivative

$$L'(x) = \frac{1}{x - r_1} + \frac{1}{x - r_2} + \cdots + \frac{1}{x - r_d}. \tag{3}$$

Observe that $L'(x) = p'(x)/p(x)$. Consider the second derivative of L, with minus sign, $M = -L''$,

$$M(x) = -L''(x) = \frac{1}{(x - r_1)^2} + \frac{1}{(x - r_2)^2} + \cdots + \frac{1}{(x - r_d)^2}. \tag{4}$$

Observe that $M(x) = (L'(x))^2 - p''(x)/p(x)$.

Suppose we have a good approximation z_0 for the first root r_1. Denote $\delta = z_0 - r_1$ and assume $z_0 - r_2 \approx \cdots \approx z_0 - r_d$ so we may use the same Δ for all $z_0 - r_k$, $k \neq 1$. With δ and Δ, the expressions for $L'(x)$ and $M(x)$ respectively in (3) and in (4) at $x = z_0$ simplify to

$$L'(z_0) = \frac{1}{\delta} + \frac{d-1}{\Delta} \quad \text{and} \quad M(z_0) = \frac{1}{\delta^2} + \frac{d-1}{\Delta^2}. \tag{5}$$

By the two above observations, the values for $L'(z_0)$ and $M(z_0)$ are computed as $p'(z_0)/p(z_0)$ and $(L'(z_0))^2 - p''(z_0)/p(z_0)$. Knowing δ gives the next approximation for the root r_1 as $z_1 = z_0 - \delta$.

We eliminate Δ using the expression for $L'(z_0)$ in (5):

$$\frac{1}{\Delta} = \frac{1}{d-1}\left(L'(z_0) - \frac{1}{\delta}\right) \quad \Rightarrow \quad M(z_0) = \frac{1}{\delta^2} + \frac{1}{d-1}\left(L'(z_0) - \frac{1}{\delta}\right)^2, \tag{6}$$

which gives a quadratic equation in the unknown δ. The two solutions are

$$\delta = \frac{d}{L'(z_0) \pm \sqrt{(d-1)(dM(z_0) - (L'(z_0))^2)}}, \tag{7}$$

where the sign $\pm$ is chosen to obtain the larger absolute value of the denominator. Equivalently:

$$\delta = \frac{p(z_0)}{p'(z_0)} \cdot \left(\frac{1}{d} + \frac{d-1}{d}\sqrt{1 - \frac{d}{d-1}\frac{p(z_0)p''(z_0)}{p'(z_0)}}\right)^{-1}, \tag{8}$$

where the square root of the complex number is chosen to have a real part that is not negative.

Source: `https://en.wikipedia.org/wiki/Laguerre's_method`

1. Assignments

The answer to the project consists of one single Jupyter notebook.

2.1. Cubic Convergence

Download `laguerremethod.jl` from the course web site. When executed at the command prompt, or included in a Julia session as below, we see two runs of the method of Laguerre, with 256 bits of precision on `Complex{BigFloat}` numbers.

```
julia> include("laguerremethod.jl");
running on x^2 - 3*x + 2 ...
step :        real(root)             imag(root)          |dx|     |p(x)|
  0  : 7.6844767519656987e-01  9.4051500071518701e-01
  1  : 1.0000000000000000e+00  2.5908505665283334e-77  9.69e-01  3.11e-77
succeeded after 1 step(s)
running on a random polynomial ...
step :        real(root)             imag(root)          |dx|     |p(x)|
  0  : 2.8106589510145752e-01  7.9293102916315772e-01
  1  : 3.8195606114724018e-01  1.0558403094632030e+00  2.82e-01  6.12e-01
  2  : 3.5653298955097910e-01  1.0201172610276352e+00  4.38e-02  1.27e-03
  3  : 3.5662939462136947e-01  1.0201705282073562e+00  1.10e-04  2.23e-11
  4  : 3.5662939462020207e-01  1.0201705282088903e+00  1.93e-12  1.19e-34
  5  : 3.5662939462020207e-01  1.0201705282088903e+00  1.03e-35  8.64e-77
succeeded after 5 step(s)
```

Assignment One. Make a Jupyter notebook with the posted program `laguerremethod.jl`.

Apply the method of Laguerre to the square root of a number N, running the method on the polynomial $x^2 - N$, starting at $z_0 = N$. Run experiments at a sequence of increasing precision, starting at 256. Do sufficiently many experiments till you can answer the following question.

How many iterations does the method of Laguerre require to compute the square root of a number N with n bits of precision?

2.2. Computing All Roots

If r is a root of $p(x)$, $q(x) = p(x)/(x - r)$, $p(x) = q(x)(x - r)$. Running the method of Laguerre on the quotient q will give the next root of p. Repeating this as long as $\deg(q) > 2$ will give all roots.

Assignment Two. Write a function to compute all roots of a polynomial p. For the polynomial division, you could write your own function, but you may also use the `Polynomials` package. Demonstrate the correctness by running your function on polynomials with random coefficients.

Is the last root computed as accurately as the first root? Verify with the original p.

2.3. The Wilkinson Polynomial

Consider the Wilkinson polynomial $w_d(x) = (x - 1)(x - 2) \cdots (x - d)$, for increasing degrees, up to degree 20, and increasing precision. Do sufficiently many experiments to answer the following.

Assignment Three. What is the smallest value for the precision to compute roots of $w_d(x)$ with at least 8 decimal places correct? Note that for $d = 2$, this value is smaller than 256 bits.

The third assignment can be solved independently from the second assignment.

2. The Deadline is Friday 5 February 2021, at noon

Some important points.

1. You may (not must) work in pairs, with a partner of your own choosing.

   If you work in a pair, then you must declare your partner in an email to `janv@uic.edu` at the latest before 5PM on Friday 29 January.

2. The solution to this project should be in one single Jupyter notebook.

   - The notebook should run as a program.
     All cells should execute correctly in the order in which they appear.
   - For each assignment, use a separate header in the notebook.
   - Use text cells to answer the questions of the assignments.
   - Write complete, grammatically correct sentences to answer the questions.

   Explanations matter just as much as the numerical results and the code.

3. Submit your project before the deadline to gradescope.

   The name of your notebook should be `laguerreFirstNameLastName` where you replace `FirstName` and `Lastname` by your own first and last name.

   If you work as a pair, submit `laguerreFirstName1LastName1FirstName2LastName2`, as one single submission, where `FirstName1 LastName1` and `FirstName2 LastName2` are the names of the pair members, in alphabetic order according to the last name.

4. There is an automatic extension of the deadline till 5PM on the same day. However, late submissions are penalized with ten points off. Submissions after 5PM will not be graded.

If you have questions about the project, feel free to email `janv@uic.edu` for an online office hour.