MAT360 Section Summary: 2.6

Müller's Method and zeros of Polynomial

Parabolas save the day!

"Hamming's motto, 'the purpose of computing is insight, not numbers', is particularly apt in the area of finding roots. You should repeat this motto aloud whenever your program converges, with ten-digit accuracy, to the wrong root of a problem, or whenever it fails to converge because there is actually no root, or because there is a root but your initial estimate was not sufficiently close to it." From *Numerical Recipes in C*.

Summary

Theorem 2.15 (Fundamental Theorem of Algebra): If $P(x)$ is a polynomial of degree $n \geq 1$ with real or complex coefficients, then $P(x) = 0$ has at least one (possibly complex) root.

(Ironically – embarrassingly, for the algebraists– the easiest proof comes via complex analysis.)

Corollary 2.16 If P is of degree n, then $P(x)$ can be expressed as

$$P(x) = a_n(x - x_1)^{m_1}(x - x_2)^{m_2} \cdots (x - x_k)^{m_k}$$

where the x_i are distinct roots, and

$$\sum_{i=1}^{k} m_i = n$$

Corollary 2.17 If P and Q are polynomials of degree at most n, then if $\{x_1, \ldots, x_k\}$, with $k > n$, are distinct numbers with $P(x_i) = Q(x_i)$ for $i = 1, \ldots, k$, then $P = Q$. In other words, if n^{th} degree polynomials agree on

$n + 1$ points, then they are equal.

Horner's Method: computes values of polynomials efficiently. The idea is pretty simple: we simply evaluate the terms of the nested form of the polynomial P successively, from the inside out, where we've expanded about the point x_0 at which we wish to evaluate P. This is the Taylor series expansion of P about x_0.

Given $P(x) = a_n x^n + \ldots + a_1 x + a_0$, evaluate $P(x_0)$ starting from the Taylor series expansion as follows:

1. P can be written as

$$P(x) = P(x_0) + P'(x_0)(x - x_0) + \ldots + \frac{P^{(n)}(x_0)}{n!}(x - x_0)^n$$

   Factoring a term of $(x - x_0)$, we have that

$$P(x) = P(x_0) + (x - x_0)Q(x) \equiv b_0 + (x - x_0)Q(x) \qquad (1)$$

   where we have defined $b_0 \equiv P(x_0)$. The question is how to compute b_0.

2. If we write

$$\boxed{Q(x) = b_n x^{n-1} + \ldots + b_2 x + b_1}$$

   then we can write (1) as

$$P(x) = b_n x^n + (b_{n-1} - x_0 b_n)x^{n-1} + \ldots + (b_1 - x_0 b_2)x + (b_0 - x_0 b_1)$$

3. Equating coefficients, we have that

$$b_n = a_n$$

   and, solving for b_k in terms of b_{k+1} for $k = n - 1, \ldots, 1, 0$, n terms

$$\boxed{b_k = a_k + b_{k+1} x_0.}$$

   Then $P(x_0) = b_0$.

The cost of Horner's method is n multiplications and n additions.

If we're using Newton's method with for roots of P, then we're in luck, because we can perform the same operation with polynomial Q. Why would we want to evaluate $Q(x_0)$? Because $Q(x_0) = P'(x_0)$! Therefore

$$g(x_0) = x_0 - \frac{P(x_0)}{P'(x_0)} = x_0 - \frac{P(x_0)}{Q(x_0)}$$

1a, p 99 :

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

$b_3 = a_3 + b_4 x_0$

$\boxed{x_0 = 1}$

$(a_4, a_3, a_2, a_1, a_0) = (1, 0, 2, -1, -3)$

$g(x_0) = x_0 - \frac{f(x)}{f'(x_0)}$

$(b_4, b_3, b_2, b_1, b_0) = (1, 1, 3, 2, -1)$

$b_0 = -1 = f(1)$

$f'(x_0) = Q(x_0)$

$$\begin{aligned}(a_3, a_2, a_1, a_0) &\\ = (1, 1, 3, 2) &\\ (b_3, b_2, b_1, b_0) &\\ = (1, 2, 5, 7) &\end{aligned}$$

$\boxed{g(1) = 1 - \frac{-1}{7} = \frac{8}{7}}$

$f'(x_0) = 7$

When we've found an approximate root, r_1, then we will have that

$$P(x) = (x - r_1)Q(x) + b_0 \approx (x - r_1)Q(x)$$

$P'(x) = Q(x) + (x - r_1) Q'(x)$

(since $P(r_1) = b_0 \approx 0$), so that further roots of P could be determined by switching focus to Q. This process is called **deflation** (because we're letting the air out of our n^{th} degree polynomial to get an $(n-1)^{th}$ degree polynomial). As roots are found, continue deflating until you're down to a linear polynomial, whose solution can be written down instantly.

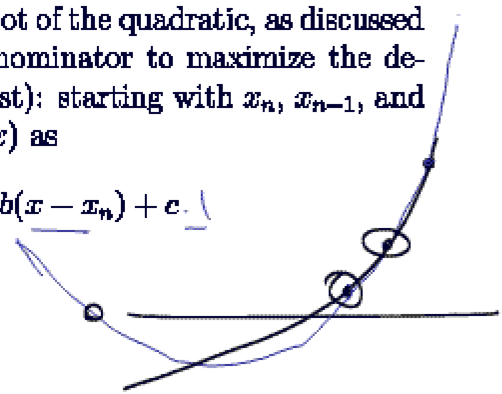
Errors will creep in as this process continues. It's best to start with the small roots, and work up to the big roots, if possible. When all is said and done, you might use a few iterations of Newton's method on the roots with the original polynomial to refine the approximate roots obtained by deflation.

This process of deflation works even when the function f we're dealing with is not a polynomial. The next technique is a good general root finder, with nearly quadratic convergence.

Müller's method: is a generalization of the secant method, where rather than a secant line using two points, we create a parabolic fit to three points. Other than that, there's really no difference. So what's the big deal?

- Well, for one thing, Müller's method will find complex roots for us, starting from real values.
- Slight issue: you will require three initial guesses. (Müller's is pretty robust to bad choices in initial guesses.)
- You have to be careful to choose your root of the quadratic, as discussed previously. Choose the sign in the denominator to maximize the denominator (again, find smaller roots first): starting with x_n, x_{n-1}, and x_{n-2}, and having written quadratic $f(x)$ as

$$f(x) = a(x - x_n)^2 + b(x - x_n) + c.$$



Bob* Linear Algebra!

$$\begin{bmatrix} (x_0-x_2)^2 & (x_0-x_2) & 1 \\ (x_1-x_2)^2 & (x_1-x_2) & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{bmatrix}$$

$x_0, f(x_0)$
$x_1, f(x_1)$
$x_2, f(x_2)$

we can solve for a, b, and c, and then choose x_{n+1} to be the corresponding root of f: if the coefficients of f are real, then

$$x_{n+1} - x_n = \frac{-2c}{\mathrm{signum}(b)(|b| + \sqrt{b^2 - 4ac})}$$

i.e.,

$$x_{n+1} = x_n + \frac{-2c}{\mathrm{signum}(b)(|b| + \sqrt{b^2 - 4ac})}$$

If the coefficients are complex, then you want to choose the sign so as to maximize the modulus of

$$z = b \pm \sqrt{b^2 - 4ac};$$

that is, the size of $|z|$.