Polynomial Evaluation

Suppose we have a real polynomial p in standard form, that is, we know its coefficients in the standard ordered basis $1, x, x^2, \ldots, x^n$, that is, we know the a_i in

$$p(x) = \sum_{i=0}^{n} a_i x^i.$$

If $s \in \mathbb{R}$, then how many operations are needed to compute $p(s)$? Think about it for a while before you read on. It turns out that we can do it with n multiplications and n additions. Here is an example:

$$7s^4 + 2s^3 - 5s^2 + 4s - 3 = (((7s + 2)s - 5)s + 4)s - 3.$$

The general iteration below gives $b_0 = p(s)$:

$$b_n = a_n; \qquad b_{j-1} = sb_j + a_{j-1}, \quad j = n, n-1, \ldots, 2, 1.$$

It's called Horner's method, nested multiplication, or synthetic division. There are faster algorithms for evaluating $p(s)$ if s is complex, or if s is a matrix, or if we want to evaluate p at several places at the same time, etc., but this is an optimal algorithm for evaluating a real polynomial at a single real number.

It is called synthetic division because of a division procedure (known in China at least 500 years before Horner) which gives the b_j as auxilliary quantities. We can see the division by forming the polynomial $q(x) = b_1 + b_2 x + \cdots + b_n x^{n-1}$. Then

$$p(x) = (x - s)q(x) + b_0,$$

giving the quotient q and the remainder b_0 in $p(x)/(x - s)$.

As a little bonus, this gives us a formula for $p'(s)$:

$$p'(x) = (x - s)q'(x) + q(x),$$

so that $p'(s) = q(s)$...

So the best way to evaluate a polynomial at s is to divide it by $(x - s)$. This theory presents another opportunity for numerical analysts. Suppose we have one root, say s, of a polynomial p of degree n. If we want to compute all n roots of p we might now divide it by $(x - s)$ to get the remainder polynomial q as above. Since (presumably) s is a root we have

$$p(s) = (x - s)q(x) + 0,$$

and the $n - 1$ roots of p which we still desire are precisely the $n - 1$ roots of q. Now we can try to find a root of q... This process of dividing out a root to get a smaller problem of the same type is called *deflation*. While we need to take care in its implementation, deflation is one of the fundamental tools for modern scientific computation.