Polynomial Interpolation

George E. Collins

October 14, 1998

The Chinese remainder theorem can be used to reduce problems involving large integers to a number of similar problems involving smaller integers. Likewise polynomial interpolation can be sued to reduce polynomial problems in several variables to some similar problems in fewer variables. Application of the Chinese remainder theorem utilizes modular homomorphisms H_m from Z onto Z_m. Application of interpolation utilizes evaluation homomorphisms E_a from a polynomial domain $D[x]$ onto D. If $A(x) \in D[x]$ and $a \in D$, then E_a is defined by $E_a(A(x)) = A(a)$, the value of A at a. It is easy to verify that E_a is a homomorphism of $D[x]$ onto D. In order to perform interpolation, D must be a field. Interpolation fits into the scheme of the Chinese remainder theorem by the observation that $A(x) \equiv A(a)$ (modulo $(x-a)$), that is, $(x-a)|(A(x)-A(a))$. In particular, Garner's method is applicable, and in SACLIB we again wish to apply it only for the case of two moduli for reasons similar to those given for the case of integer moduli. The method is implemented only for the case that D is a finite field Z_p and the program is MPINT. MPINT interpolates a polynomial $A(x_1, \ldots, x_r)$ in several variables over Z_p, but the interpolation is with respect to x_1. If $r = 2$, for example, this just means that $A(x_1, x_2)$ is a polynomial in x_2 whose coefficients are elements of $Z[x_1]$ and that each of these coefficients is interpolated. So, in addition to the five inputs of MIDCRA, there are two other inputs, p and r. The evaluation homomorphism is performed by the program MPEMV, which applies Horner's method just as in IPEMV. Consider the case $r = 1$, which is covered by Step 1 of MPINT. The polynomial $B(x_1) \in Z_p[x_1]$ is multipled by $c \in Z_p$ by the subprogram MPMDP. The result is added to $A(x_1) \in Z[x_1]$. Each operation is performed in time dominated by n, the degree of $B(x_1)$, if $n > 0$. Thus the maximum computing time of MPINT for $r = 1$ is dominated by n. Notice that when $r > 1$ the number of recursive applications of MPINT on univariate polynomials will be bounded above by $\prod_{i=2}^{r}(n_i + 1)$, where n_i is the maximum of the degree of A in x_i and the degree of A_1 in x_i, so the computing time of MPINT will be dominated by $n \prod_{i=2}^{r}(n_i + 1)$.