Notes for MA591U, Spring 2001
(Symbolic Computation)

Squarefree Factorization

DEFINITION: We say that $f \in k[x]$ where $\mathbb{Q} \subset k$ is **squarefree** if

$$f = \prod_{i=1}^{r} p_i = p_1 \cdots p_r$$

where the (nonconstant) p_i are all distinct. We say f has a **squarefree factorization** if

$$f = \prod_{i=1}^{r} Q_i^i$$

where each Q_i is squarefree and $\gcd(Q_i, Q_j) = 1$ when $i \neq j$.

We would like an algorithm for squarefree factorization. Define $f'(x)$ in the usual way:

$$(f(x) = a_0 + a_1 x + \cdots + a_n x^n) \Rightarrow \left(f'(x) = a_1 + \cdots + n a_n x^{n-1}\right).$$

LEMMA: f is squarefree if and only if $\gcd(f, f') = 1$.

PROOF:

Write $f = \prod_{i=1}^{m} p_i^{e_i}$ where p_i is irreducible. Then

$$f' = \sum_{i=1}^{m} \left[\left(e_i p_i^{e_i - 1} p_i'\right) \prod_{j \neq i} p_j \right]$$

(this is easily shown). Then f is squarefree if and only if $e_i = 1$ for all $i \in \{1, \ldots, m\}$, if and only if $(p_i | f)$ and $(p_i \not| f')$, if and only if $\gcd(f, f') = 1$.

NOTE:

(1) It follows from unique factorization that squarefree factorization is unique.

(2) One can find the squarefree factorization without actually factoring.

METHOD TO FIND THE SQUAREFREE FACTORIZATION (WHEN IT EXISTS):

Given f, let $C_1 = \gcd(f, f')$, and $D_1 = f/C_1$.

While $C_i \neq 1$, let $C_{i+1} = \gcd(C_i, C_i')$, $D_{i+1} = C_i/C_{i+1}$, and $Q_{i+1} = D_i/D_{i+1}$.

Because the degree decreases each time, this terminates in m steps. Set $Q_m = D_m$.

Then $f = \prod_{i=1}^{m} Q_i^i$.

PROOF:

Suppose f has a squarefree factorization. Then

$$f = \prod_{i=1}^{m} Q_i^i.$$

It is straightforward to show that

$$C_1 = Q_2 Q_3^2 \cdots Q_m^{m-1}$$

and

$$D_1 = Q_1 \cdots Q_m.$$

Repeating this inductively, we get

$$C_i = Q_{i+1} \cdots Q_m^{m-i}$$

and

$$D_i = Q_i \cdots Q_m.$$

Clearly, setting Q_i as specified by the algorithm gives us the Q_i in the squarefree factorization of f.

EXAMPLE: Let $f(x) = 3x^3 + 4x^2 - x - 2$. We have $f'(x) = 9x^2 + 8x - 1$, $C_1 = x + 1$, and $D_1 = 3x^2 + x - 2$. Since $C_1 \neq 1$, we continue: $C_2(x) = 1$ and $D_2(x) = x + 1$. This gives us $Q_1(x) = 3x - 2$ and $Q_2(x) = x + 1$.

Then

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

EXAMPLE: Let

$$f(x) = x^9 - 4x^8 - 9x^7 + 34x^6 + 47x^5 - 72x^4 - 135x^3 - 54x^2.$$

Then $C_1 = x^5 - 4x^4 - 2x^3 + 12x^2 + 9x$ and $D_1 = x^4 - 7x^2 - 6x$. Since $C_1 \neq 1$, we continue: $C_2 = x^2 - 2x - 3$ and $D_2 = x^3 - 2x^2 - 3x$. Then $Q_1 = x + 2$. Again, $C_2 \neq 1$, so we obtain $C_3 = 1$ and $D_3 = x^2 - 2x - 3$. Then $Q_2 = x$ and $Q_3 = x^2 - 2x - 3$, and we see that

$$f(x) = (x + 2) \cdot x^2 \cdot (x^2 - 2x - 3)^3.$$

A Maple program that implements squarefree factorization:

```
squarefree:=proc(f::polynom,var::name)
   local c_new,c_old,g,d_new,d_old,q,f_squarefree,j:
   c_new:=gcd(f,diff(f,var)):
   d_new:=simplify(f/c_new):
   q:=[]:
   while (c_new<>1) do
     c_old:=c_new:
     d_old:=d_new:
     c_new:=gcd(c_old,diff(c_old,x)):
     d_new:=simplify(c_old/c_new):
     q:=[op(q),simplify(d_old/d_new)]:
   od:
   q:=[op(q),d_new]:
   f_squarefree:=1:
   for j from 1 to nops(q) do
     f_squarefree:=f_squarefree*q[j]^j:
   od:
   RETURN([q,f_squarefree]):
end:
```