Mathematics 392 – Seminar in Computational Commutative Algebra
Lab 1 – The Euclidean Algorithm in $k[x]$
September 11-13, 2006

Background

Recall from high school algebra the "polynomial long division" process that takes polynomial $f(x)$, divides by $g(x)$ and finds a (unique) quotient $q(x)$ and remainder $r(x)$ with

$$f(x) = q(x)g(x) + r(x),$$

and either $r(x)$ is the zero polynomial, or $\deg r(x) < \deg g(x)$. In $k[x]$, the fact that we have a *division algorithm* has many strong consequences. Our goals in this lab are

- to start to get comfortable working on the Math/CS department linux network,
- to introduce built-in Maple commands for polynomial arithmetic, including division,
- to implement the Euclidean algorithm for the GCD of two polynomials in a Maple procedure (and test it against Maple's built-in polynomial GCD routine),
- to generalize the Euclidean Algorithm so that we can compute the polynomials $A(x)$ and $B(x)$ in an equation

$$\mathrm{GCD}(f_1(x), f_2(x)) = A(x)f_1(x) + B(x)f_2(x).$$

Some Maple 10 Information

To use Maple 10, in SW 219,

- From the "red hat" menu, select HC Applications/Maple
- Maple 10 has a much more flexible user interface than the previous versions you have used before (much easier to create documents with mathematical formulas, etc. even as inputs to Maple). For now, though, to make it act and look like the old Maple:
- To start a new worksheet use File/New/Worksheet Mode, *and*
- *At each new input prompt*, hit the F5 key to toggle to "1-D Math input".

If you do this, input commands are entered exactly the same way as in "old" Maple.

Maple's Built-in Polynomial Arithmetic

Maple implements polynomial division in *two separate* procedures:

- `quo` for the quotient
- `rem` for the remainder

Both have similar format

```
quo(f,g,x);        rem(f,g,x);
```

(Note, the third input is the variable in the polynomials, which must be specified.) For instance, enter the following commands to compute the quotient and remainder on division of $f = x^6 - 6x^4 - 96x^2 + 5x^5 + 40x^3 + 80x - 224$ by $g = x^4 - 5x^3 + 10x^2 - 20x + 24$. (Note the assignment of results to names each time – makes it easier to reuse the results of computations later!)

```
f:=x^6-6*x^4-96*x^2+5*x^5+40*x^3+80*x-224;
g:=x^4-5*x^3+10*x^2-20*x+24;
q:=quo(f,g,x);
r:=sort(rem(f,g,x),x);
expand(q*g + r);
```

To see why I put the `sort` in the command computing the remainder r, try removing it! Also, what is the final output?
Maple also has factoring and polynomial GCD's built-in in commands:

- `factor(f);`
- `gcd(f,g);` – this one *doesn't* include the name of the variable, go figure! It uses the Euclidean algorithm, of course. Compute $\mathrm{GCD}(f, g)$ with f, g as above, then factor f, g, and the GCD and convince yourself the GCD is correct!

The Euclidean Algorithm, "From the Ground Up"

A consequence of the fact that we have the division algorithm in $k[x]$ is the following important statement:

- Every ideal I in $k[x]$ is *principal*, that is, every $I = \langle g(x) \rangle$ for some polynomial $g(x)$.

In fact, if $I \neq \{0\}$ and we require that $g(x)$ have the form

$$g(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_0$$

(leading coefficient $= 1$), then $g(x)$ is *unique*. This means in particular that if we take an ideal generated by several polynomials, say $I = \langle f_1(x), f_2(x) \rangle$, then there is some single $g(x)$ that generates *the same* ideal I. The generator for $\langle f_1(x), f_2(x) \rangle$ is the GCD:

$$\langle f_1(x), f_2(x) \rangle = \langle \mathrm{GCD}(f_1(x), f_2(x)) \rangle.$$

From Algebraic Structures (MATH 243), you should recall the *Euclidean algorithm* for computing the GCD of two integers. The way you probably recall the process is something like this. To find the GCD of m, n where $m < n$, first divide m into n yielding remainder

r_1. Then divide r_1 into m yielding remainder r_2, and so forth. You stop when you obtain a zero remainder, and *the last non-zero remainder is the GCD*:

$$
\begin{aligned}
n &= q_1 m + r_1 \\
m &= q_2 r_1 + r_2 \\
r_1 &= q_3 r_2 + r_3 \\
&\vdots \\
r_{k-2} &= q_k r_{k-1} + r_k \\
r_{k-1} &= q_{k+1} r_k + 0
\end{aligned}
\tag{1}
$$

Then assuming $r_1, r_2, \ldots, r_k$ are all nonzero, $\mathrm{GCD}(m, n) = r_k$.

Since we also have the division algorithm for polynomials now, we can do the same process with $f(x), g(x) \in k[x]$, and this is called the Euclidean algorithm too. What we will do next is see how to implement an algorithm like the Euclidean algorithm in a Maple procedure. The "MATH 243 version" of the Euclidean algorithm from equation (1) above *could* be turned into a procedure, but it has a couple of defects:

1) It's rarely apparent exactly how many division steps will be needed before a zero remainder will be obtained,
2) It's not actually necessary to remember *all* the previous remainders since each division involves only the previous two of them.

These criticisms lead to the following ideas – we only need to remember the previous two remainders and "shift them" after each division step to get ready for the next one (those are the `h,s` in the following). When we get a zero remainder for the first time, it's the previous remainder we want to output as the GCD. Here is a Maple procedure that does this, following the outline given in the text on page 41. Note: `rem` is the name of the Maple remainder procedure, so we call the variable for the remainder `r` instead.

```
Euc := proc(f,g,x)
local h,s,r;
h:=f;
s:=g;
while s <> 0 do
   r:=rem(h,s,x);
   h:=s;
   s:=r;
end do;
return h;
end proc:
```

Enter these in your worksheet after one input prompt, using Shift + Return to start new lines. This defines a new procedure called `Euc` "on top of" the basic Maple commands.

After you have done the above, to compute a GCD by this procedure you would just enter a command like

```
Euc(f,g,x);
```

and the output will be the GCD. Test this out on the f, g from before (do you get exactly the same result? is your result correct?!), and on several other examples of your own choice.

Lab Assignment – Due: Monday, September 18

Do problem 10 from Section 1.5 in the text, but implemented as a second Maple procedure that returns the GCD, plus the polynomials $A(x), B(x)$ such that

$$\mathrm{GCD}(f, g) = A(x)f(x) + B(x)g(x).$$

You will not want to delete anything in the `Euc` procedure above, only *add* the necessary commands to compute A, B as above. The Hint contains all the ideas you will need!

Turn in a paper copy of a worksheet with your "generalized" Euclidean algorithm procedure, plus the output from several runs checking that it is working correctly.