PREMUR 2007 Seminar
Week 1 Computer Laboratory Exercises

Background and Goals

A part of our work this summer will be carrying out calculations of Gröbner bases, resultants, and other computations within the computer algebra systems *Mathematica*, **Singular**, and others. This week, we will want to get familiar doing mathematics in this context.

General Information about Mathematica

Launching Mathematica

To get into Mathematica, you will need to:

1) If necessary, turn the computer and monitor on.
2) Log on the network with the username and password that were provided for you. When see the desktop with "icons" ("shortcuts") for Mathematica, etc. double click on the Mathematica icon.
4) After a few more seconds you should see a new Mathematica window, with a "sub-window" marked Untitled (1) opened inside it. The Untitled (1) window is a blank *Mathematica notebook*.

Mathematica Notebooks

Notebooks are integrated text/graphics/mathematics documents where any or all of the following can be done:

1) you can type in commands from the keyboard to ask Mathematica to perform many different kinds of calculations, read data from external files, store work in files, etc.
2) output generated by Mathematica from your input commands (numerical values, symbolic formulas, and graphics) will be displayed,
3) you can modify commands, generating new output, store your results for use later, etc.
4) you can enter text to annotate and explain the results of computations.

Some of your interactive work with a notebook in progress will take place through the File, Edit, View, Insert, etc. pull-down menus across the top the "scroll bar" on the right that you can use to move around within the worksheet, to see previous input and output lines, etc.

Input and Output in a Notebook

A new notebook window will be labeled something like Untitled (1) window. It looks blank to start with no input prompt. To enter a command, you just type it in anywhere in the notebook, then press Shift-Return to execute the command. (When this is done, you will see that `In[1]:=` is generated to the left of what you typed in. The number will be increased automatically each time you execute a new command. (This can be helpful if you need to reconstruct the sequence in which several commands were executed). If you command generates any output, it will be displayed underneath in the notebook, with the notation `Out[1]:=`. (Not all commands generate such output lines, though.) You can then continue and enter new commands *anywhere within the notebook*.

As we mentioned before, Mathematica notebooks can contain text as well as commands and output. After you have generated the output you want, you can simply enter text anywhere in the notebook to annotate your work.

Saving and Reloading your Mathematica Notebooks

When you begin working on a worksheet, you will usually want to *save your work* every once in a while in case a computer problem develops, or in case you need more than one lab session to complete the work you are doing. This can be done most directly by saving to your network drive. following these directions:

1) In Mathematica, select the SAVE option from the FILE pull-down menu.
2) If you are saving your work for the first time, you will see a SAVE AS dialog box. In the Drives box, highlight network drive as appropriate, then go to the File Name box, and type in a name for the file where the worksheet will be stored (any string of letters and digits – no spaces). The file name you chose will appear with the "extension" *nb* for *Mathematica Notebook*. For instance, a good choice might be something like `Lab1`. Then click the OK button to save the worksheet. (You will only need to type the filename in once in a session. Subsequent saves just update the file.)
3) When you have the worksheet saved as you want it, you can exit Mathematica.
4) To update the worksheet further in a later session, you can read the worksheet back into Mathematica using the OPEN option from the FILE Menu, or the "open" toolbar button. Get into Mathematica as above, and select OPEN from the FILE Menu. Mathematica will prompt you as above for the name of the worksheet with a dialog box very like the SAVE AS box described above. Highlight the network drive as appropriate, then the worksheet you want, and click OK. IMPORTANT NOTE: When you read a previously created notebook into a new Mathematica session, you *have not* actually executed any of the commands in it in the new session. If you want to make use of any results in that worksheet, you will need to execute the commands again. In the pull-down menus, you will see an EXECUTE option that lets you recompute an entire notebook or a section of one.

Printing

With the worksheet you want to print open and highlighted, use FILE/PRINT, and press OK on the PRINT dialog box (the settings should be set up correctly to print automatically). Your output should appear shortly on the printer.

Getting Out

When you leave, quit the Mathematica window (FILE/EXIT) and log off the network.

Days 1, 2: First Sample Mathematica Sessions, Plotting Commands

Let's get right down to work and walk through a first sample session! First, you will need to log on and get into Mathematica as described in the General Information section above.

What is a Mathematica Command?

It may help to think of a command as a *program* or *function of one or more variables* that Mathematica knows how to compute, given "inputs" from you. (You must then keep straight the distinction between these functions Mathematica knows how to execute, and the functions or formulas you enter for Mathematica to work with!)

Here is some more information about the *syntax rules* that Mathematica uses to decide if what you have typed in is a well-formed command it can execute, and what the *semantics* or meaning of the commands are.

Almost all of the built-in Mathmetica commands have a syntax whose format is

`Name[values]` or `Name[values];`

In this general description, `Name` is the name of the command or function, there is a matching set of open and close square brackets following the name, and they surround the `values` – the input from you the command needs to do its job (like the formula of the function to be plotted and the range of x-values you want). The values are listed separated by commas within the parentheses. The two forms of the command above tell Mathematica to execute the command when you press ENTER on that input line. In the first case – you are instructing Mathematica to display the output. With semicolon (;), Mathematica will execute the command but not display the output (this is useful sometimes for intermediate steps in a big computation where you don't want or don't need to see the output).

Getting Help On-Line

The exact format Mathematica is expecting for each type of command is specified in the programming of the Mathematica system. Usually, there is *little or no freedom* in the ordering of what has to go where and in what format in the list of values. Much useful information on this for all the built-in commands, and LOTS of instructive examples are contained in the Mathematica on-line HELP facility. For example try looking at the on-line help page for the `Plot` command. (In the Help Browser, type in the word `Plot` click on the listing for `Plot` at the right.) Note the example plot commands at the bottom and the links to related help pages. We'll see some other examples in a moment.

Mathematica Expressions

Formulas or expressions are entered in something like usual mathematical notation:

1) The symbols for addition, subtraction, and division are $+, -, /$ respectively.
2) The caret (^) is the symbol for raising to a power.
3) Multiplication can be indicated by "juxtaposing" the factors (putting them side by side as in usual mathematical notation) One thing to be aware of: If you want the product $x \times y$, you will need to leave a space between the x and the y. (Otherwise, Mathematica will take xy as another variable with a two-letter name.)

Everything must be entered in one linear string of characters, so you will need to use *parentheses* to group terms to get the expressions you want. The rule to keep in mind is: *Mathematica always evaluates expressions by doing powers first, then products and quotients, then sums and differences, all left to right, unless parentheses are used to override these built-in rules*. For example, the expression `a + b c^2/d + e` (note the space between the `b` and the `c`) is the same as the mathematical formula:

$$a + \frac{bc^2}{d} + e.$$

If you really want

$$\frac{a + bc^2}{d + e}$$

you will need to enter the expression `(a + b c^2)/(d + e)`. What if you really wanted:

$$\frac{(a + bc)^2}{d + e}?$$

4) Mathematica "knows" all the usual elementary functions from calculus. The names of the most common ones are `Sin, Cos, Tan, Exp, Log, Sqrt`. To use one of these functions in a Mathematica formula, you put the name, followed by the "argument" (that is, the constant or expression you are applying the function to) *in square brackets*.

Some practice

How would you enter each of the following formulas as Mathematica expressions? Try typing each in as an input line, and compare with the output they generate. Type your formula, and press Shift-ENTER. (Until you press Shift-ENTER, Mathematica will not do anything with your command. Also, if a command you want to enter doesn't all fit on one line, just keep typing, but don't press ENTER until you are finished. Mathematica automatically wraps around to a new line if you need it.)

If you make a typing mistake, Mathematica will let you know about it (!) One possible error in entering Mathematica commands is unmatched brackets, or curly braces ({ , }), or parentheses, etc. The Mathematica notebook interface highlights the "opening" characters until the matching "closing" characters are entered, so it should be clear if you have something unbalanced in your input line.

Fortunately, if this happens, the whole command does NOT need to be re-entered. Just move the cursor arrow to the place on the input line you want to change, press the left mouse button, and edit the input as needed. Typing from the keyboard will *insert* new stuff at the cursor location; the DELETE and BACKSPACE keys will remove stuff (DELETE removes the character in front of the "insert point"; BACKSPACE removes the character in back). You can also move around on the input line with the ARROW keys if more than one thing needs to be changed. When you think it's OK, press Shift-ENTER again to have Mathematica execute the command again. (Pressing shift-ENTER with the cursor anywhere on an input line executes the whole input line.)

1)
$$4xy^2 - 2x^3 + 3y - 7$$

2)
$$1 + \frac{1}{x^2} - \frac{x+2}{x^4}$$

3) (Note: `Sqrt[x]` is the Mathematica syntax for $\sqrt{x}$.)
$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

When all goes well you will see the output displayed under the input command in the worksheet. You will probably also notice the slightly odd-looking *brackets on the right of the notebook window* at this point. This is Mathematica's way of grouping input and output lines together. You can highlight these to delete, or cut and paste, or modify regions of the notebook file in other ways. They are not shown when you print.

2D Plotting

The most basic Mathematica command for 2D plotting, used for graphs of the form $y = f(x)$, is called `Plot`. The format is

```
Plot[function,range,options];
```

where
1) `function` is the function to be plotted – the simplest way to specify one is via a formula for $f(x)$ (an *expression* in Mathematica)
2) `range` is the range of x-values you want to see plotted, entered in the form

```
{ x, low, high }
```

3) `options` can be used to control the form of the plot if desired. No options need be specified however if you don't want to. More on this later.

Example 1

To plot $y = x^4 - 2x^3 + x - 5\sin(x^2)$ for $-2 \le x \le 1$, you could use the Mathematica `Plot` command with no options:

```
Plot[x^4 - 2 x^3 + x - 5 Sin[x^2],{x,-2,1}]
```

(If you have used other computer algebra systems such as Maple, you will probably be noticing that Mathematica tends to use square brackets ([,]) in many situations where Maple would use parentheses. Also, the names of the builtin commands and functions (like `Plot`, `Sin`) are capitalized consistently. Finally, unlike Maple, in Mathematica, it is possible to indicate multiplication by juxtaposing the factors – `5 Sin[x^2]` for instance.)

From the formula $y = x^4 - 2x^3 + x - 5\sin(x^2)$, you might guess that there is at least one other x-intercept for this graph for $x > 2$ (why?). To see that part of the graph as well, edit your previous command line to change the right hand endpoint of the interval of x values (do not retype the whole command). Press Shift-ENTER on that input line to have Mathematica execute the command again. Experiment until you are sure that your plot shows all the x-intercepts of this graph. (You can repeat this process of editing a command and re-running it as often as you want; the previous output is replaced by the new output each time, and the `In[ ], Out[ ]` numbers are updated each time.)

You can manipulate the graphics output *in place* within the worksheet in several ways. For instance:

- If you click the left mouse button once over the graphics output, you will see a black box with eight "tabs" displayed at the corners and the midpoints of the edges of the box. If you place the cursor on one of the tabs, hold down the left mouse button, drag the cursor and release, you can *resize the graphics*. Try it!

Example 2

Next, let's move to another 2D plotting command, to see the parametric curve $\alpha(t) = (t^2 - 1, t^3 - t)$. The basic format for parametric curves $\alpha(t) = (x(t), y(t))$ is

```
ParametricPlot[x-comp,y-comp,t-range,options]
```

where

1) `x-comp` and `y-comp` are the x- and y-component functions of the curve to be plotted (each of these can be an *expression* involving the variable (parameter) t,
2) `t-range` is the range of t-values you want to see plotted, and
3) `options` can be used to control the form of the plot if desired. No options need be specified however if you don't want to.

Use this `ParametricPlot` command to display a plot of the curve $\alpha(t) = (t^2 - 1, t^3 - t)$ for $t \in [-3, 3]$:

```
ParametricPlot[{t^2 - 1, t^3 - 1},{t,-3,3}]
```

Example 3

Frequently, it's the relationship between two or more different graphs that you want to understand by looking at a plot. You can also use the `Plot` command to generate plots showing several graphs with the same coordinate axes. You do this by plotting a *list* of functions. You do this by putting the formulas for the functions you want, separated by *commas*, between *curly braces* – {, } – where the single formula went before. For example, to plot $y = \sin(x)$ together with $y = \sin(x + 2)$ for $0 \le x \le 4\pi$, you could use:

```
Plot[{Sin[x],Sin[x+2]},{x,0,4 Pi}]
```

Try it. What is the general relationship between the graphs $y = f(x)$ and $y = f(x + k)$ for constant k?

Implicit 2D Plotting

The 2D plots above are all either graphs of functions $y = f(x)$, or parametric curves. In addition to these, we will also want to be able to plot the affine variety

$$\mathbf{V}(g(x, y)) = \{(x, y) \in \mathbf{R}^2 : g(x, y) = 0\}$$

for a general polynomial $g(x, y)$ in two variables. The technical name for this in Mathematica is *implicit curve plotting*. To plot these curves, we need a new command called `ImplicitPlot`. This command is not part of the basic Mathematica system; instead it is a part of a separate *package* of commands for various plotting operations called the `Graphics` package. Before you can use this command, you will need to "load" the plots package with the command

```
<< Graphics`ImplicitPlot`
```

(those are "back-quotes"). The format for the `ImplicitPlot` command is

```
ImplicitPlot[equation,x-range,y-range]
```

The expression should be the equation $g(x, y) == 0$. This command generates a plot of the part of the variety $\mathbf{V}(g(x, y))$ in the rectangular box in the plane defined by the x- and y-ranges.

Example 4

For instance, try the following command:

```
ImplicitPlot[x^3 - 3 x y^2 - 3 == 0,{x,-3,3},{y,-3,3}]
```

to plot the variety $\mathbf{V}(x^3 - 3xy^2 - 3)$. (Some questions to think about as you look at this: If you set $x = c$ (a constant), how many points (c, y) are there on the curve? How many points (x, d) are there for $y = d$? Do your answers depend on c, d? How?)

3D Curve and Surface Plotting

Now we move up a dimension and consider plotting curves and surfaces in $\mathbf{R}^3$. The `ParametricPlot3D` command can be used to draw parametric curves and parametric surfaces in $\mathbf{R}^3$. To use it to draw a parametric curve, you would enter a command of the form

```
ParametricPlot3D[{x-comp,y-comp,z-comp},{t,low,high}]
```

where `x-comp` $= x(t)$, `y-comp` $= y(t)$, `z-comp` $= z(t)$ are the parametric equations of the curve and the range of parameter values to be plotted is $a \leq t \leq b$.

Example 5

For instance, try entering

```
ParametricPlot3D[{t,t^2,t^3},{t,-2,2}]
```

to plot a portion of the *twisted cubic* curve from class. The first plot you see here might be rather uninformative. Fortunately, Mathematica also lets you look at a 3D plot from different viewpoints, by adding an option

```
ViewPoint -> {x0,y0,z0}
```

for suitable points $(x_0, y_0, z_0) \in \mathbf{R}^3$. The curve plot will then be redrawn as though your eye was located at the given point. If you know the coordinates of the point you want, you can just enter them in the option. If you don't and you want to experiment to find a good viewpoint, here's another way:

1) First position the cursor after the range of t-values in the input `ImplicitPlot` command above. Then, from the `Input` pull-down menu, choose `3D Viewpoint Selector`, and use the input boxes or sliders to position the "viewing box" in the orientation you want.
2) Pressing the `Paste` button at this point will create an option like the one above (for a particular viewpoint), based on the settings in the Viewpoint Selector.
3) If you now press Shift-ENTER again on the plotting command, you will redraw the plot from the new viewpoint.

Practice a few times repositioning the viewpoint (rotating the viewing box) and redrawing the graph.

Parametric Surface Plotting

The `ParametricPlot3D` command is also used for plotting parametric surfaces in $\mathbf{R}^3$. The differences between this use of the command and that above is that the component functions will depend on *two parameters*, say u, v, and you will need to specify plotting ranges for each one.

Plotting a graph $z = f(x, y)$.

The command for plotting graphs of functions of two variables is called `Plot3D` (naturally enough!) Its format is similar to, but not exactly the same as, the format of `Plot`. To draw a graph with `plot3d`, you use a command of the format:

```
Plot3D[function,xrange,yrange,options]
```

The `function` is the function $f(x, y)$, entered in the usual Mathematica syntax for expressions. The `xrange` and `yrange` specify a rectangular box in the plane that the plot will be constructed over; the `options` can be used to specify how the plot is drawn. Look at the online help for `Plot3D` if you want to see what things are possible. You can change viewpoint, using the pulldown menus. The method is the same as that for the `ParametricPlot3D` command described above.

Implicit Surface Plots

To plot an surface defined as a variety $\mathbf{V}(g(x, y, z))$, you use the `ContourPlot3D` command. Look up the on-line help listing for this command.

Assignment on Plotting

Prepare and hand in a printout of a Mathematica notebook showing the plots asked for in the following questions. Answer any questions posed here with text annotations.

A) Generate a plot of the variety $\mathbf{V}(x^3 - 3x + 2xy^2 - y^4 - 1) \subset \mathbf{R}^2$. Add a text region answering the following questions: How many intersections of this variety are there with vertical lines ($x = a$) and horizontal lines ($y = b$) in $\mathbf{R}^2$? Does the answer depend on the values of a, b? Is it possible to "see" these numbers from the form of the equation $g(x, y)$? Explain.

B) What happens if you plot the twisted cubic curve from Example 5 above, from the view points

$$(20, 0, 0), (0, 20, 0), (0, 0, 20)$$

Show each of these plots and explain their shapes. (Hint: These view points are quite far away from the origin compared to the points that are plotted on the twisted cubic. If we look at the curve from a point far out on the x-axis, what are we seeing?)

C) Generate a plot of the following variety in $\mathbf{R}^3$:

$$S = \mathbf{V}(z^2 - (16 - x^2 - y^2)((x + 2)^2 + y^2 - 1)((x - 2)^2 + y^2 - 1)/50)$$

and explain the shape you see. You will want to "walk around" this one a lot by rotating and looking at it from different viewpoints (For example, for which $(x, y) \in \mathbf{R}^2$ are there points (x, y, z) on S and why (that is, what is the projection of S into the (x, y) plane? You will also need to think about how you choose the x-, y-, and z-ranges. Be sure you take the x- and y- ranges big enough to see all the points on the variety.) Suggestion: Use the options `Mesh->None` and `PlotPoints -> {6,6}` to see more detail. Be aware though that this larger than normal number of grid points will mean the surface takes longer to compute and "render" or draw and your worksheet will require more space to store if you save the graphics.

Choose either one of the following two problems, or both if you are really up for a challenge!

D) The line segment from (a, b, c) to (d, e, f) can be parametrized as follows

$$\alpha(t) = (a + t(d - a), b + t(e - b), c + t(f - c))$$

for $0 \leq t \leq 1$. The points $P = (0, 0, 0)$, $Q = (2, 0, 0)$, $R = (0, \sqrt{3}, 1)$, $S = (0, -1, \sqrt{3})$ are four corners of a cube in $\mathbf{R}^3$ with edges PQ, PR, and PS, because the vectors $\vec{u} = Q - P, \vec{v} = R - P, \vec{w} = S - P$ all have magnitude 2, and $\vec{u} \cdot \vec{v} = \vec{u} \cdot \vec{w} = \vec{v} \cdot \vec{w} = 0$. Your assignment, should you decide to accept it (just kidding!), is to create a picture of this cube by drawing the 12 edges together on the same set of axes in $\mathbf{R}^3$ (begin by drawing the line segments from P to Q, P to R, and P to S). To plot several parametric curves together on the same set of axes, consult the online documentation for the `Show` command. Also explain how you found the other 4 corners of the cube. This can be done in a text region.

E) One very interesting and useful class of parametric curves are the *Bezier* curves we saw in discussion earlier this week. A single Bezier cubic is a plane parametric curve

$$\beta(t) = t^3 P_0 + 3t^2(1 - t)P_1 + 3t(1 - t)^2 P_2 + (1 - t)^3 P_3$$

where the P_i are called the *control points* – they control the location and the shape of the curve.

Mathematica also lets you define *piecewise* Bezier curves linked together in a very particular way. You can list any number of points in a list like the list `points` above, and change `Bezier` to `CompositeBezier`. The curves that are generated use the control points in a similar fashion to the basic Bezier cubics, but as you will see, the curve passes through every other point, and the intermediate points control the tangent directions.

In Mathematica, a composite Bezier curve, together with its control points, can be plotted via commands like this:

```
points = {{0,0},{1,0},{0,2},{1,1},{1,2}}
Show[Graphics[Spline[points,CompositeBezier, SplineDots -> Automatic]],
        PlotRange -> {{-1,4},{-1,4}}]
```

Try varying the locations of the control points and redrawing the curve to see how this works – how the endpoints of the curve and the tangent directions at the end points are determined by the points $P_0, P_1, P_2, P_3, \ldots, P_n$.

Curves like this are a major tool in *computer-aided geometric design*. They can be used to specify curved shapes appearing in many objects. In this problem your goal will be to construct a composite Bezier curve (that is determine a suitable collection of control points) that has the shape of the outline of the letter "c" in the typeface used for this document. Try match the overall shape, proportions, and details as closely as you can. In particular, note

- the small "knob" at the upper end,
- the gradual increase in thickness (distance between left and right boundaries) in the vertical portion, and
- the different curvature on the bottom end.

(Designing typefaces is one "real-world" application of this mathematical idea.)

Symbolic Computation in Mathematica

Much of our work starting this week will consist of *symbolic* computations in Mathematica – mainly various manipulations of polynomials in several variables. Today, we will look at some first examples of this sort of computation, with polynomials in one variable.

The assignment operator =

A Mathematica command of the form

```
name = expression;
```

assigns the right hand side to the name on the left. Usually, the right hand side will involve a known expression, or performing some operation on some information we already know. We want to take the result of the operation and give it an abbreviated "name" for later use. For instance,

```
cube = (x^2 - 16)^3;
```

Then later on, we could "reuse" the value of `cube` in other expressions, just by putting in the name `cube` at the appropriate place (e.g. `Plot[cube, {x,5,10}]`).

Some additional symbolic commands on polynomials

Mathematica has built-in commands:

- `Expand` to multiply out (and also simplify) a factored polynomial. Format: `Expand[poly]` This works on polynomials in any number of variables.
- `Factor` to factor a polynomial. This gives the factorization with *rational number coefficients*. The factor command does not know about radicals, complex numbers, etc. Format: `Factor[poly]` This works on polynomials in any number of variables.
- `PolynomialQuotient` to compute the quotient on division of one polynomial $f(x)$ by another $g(x)$. Format: `PolynomialQuotient[f,g,x]` Note: the variable must be included. If f or g contain other variables besides x, they will be treated as constants in the coefficients.

- `PolynomialRemainder` to compute the quotient on division of one polynomial $f(x)$ by another $g(x)$. Format: `PolynomialRemainder[f,g,x]` Note: the variable must be included again.
- `PolynomialGCD` to compute the greatest common divisor of two polynomials $f(x)$ and $g(x)$. Format: `PolynomialGCD[f,g];` Note: the variable is *not* included in this one. The greatest common divisor of a set of several polynomials can be computed by nested gcd's (see p. 43 of "IVA"; for example $gcd(f, g, h)$ can be computed by `PolynomialGCD[f,PolynomialGCD[g,h]]`.
- `D` to compute derivative of a polynomial (or any other function) with respect to a given variable. Format `D[f,x]`

Here is a sample sequence of Mathematica input lines illustrating some of these commands. Try entering and executing them in sequence. Look carefully at the output and make sure you understand what happened in each case:

```
cube = (x^2 - 16)^3
D[cube,x]
cubee = Expand[cube]
Factor[cubee]
s = x^2-x+1
q = PolynomialQuotient[cubee,s,x]
r = PolynomialRemainder[cubee,s,x]
Factor[Expand[q s + r]]
```

Assignment, part 2

Using Mathematica and the commands above, do problems 8, 9, 15 b from Chapter 1, §5 of "IVA". For 15 b, use the formula in 15 a.

An "Extra" – A First Taste of Mathematica Programming

In addition to a large collection of built-in commands for performing various mathematical computations, Mathematica also contains a *programming language* that you can use to code and implement your own new commands or procedures. Here is a first example of a Mathematica procedure which takes as input two polynomials in one variable, $f(x)$ and $g(x)$, and computes their gcd $h(x)$, together with polynomials $A(x)$ and $B(x)$ satisfying

$$h(x) = A(x)f(x) + B(x)g(x)$$

(recall this is question 10 from Chapter 1, §5 of "IVA" from today's discussion). Here is Mathematica code for one way to do this:

```
ExtEuc[f_,g_,v_] := Module[
    {A=1,B=0,C=0,Csave,D=1,Dsave,q,r,s=g,h=f},
    While[!(s===0),
        q = PolynomialQuotient[h,s,x];
```

```
        r = PolynomialRemainder[h,s,x];
        h = s;
        s = r;
        Csave = C;
        Dsave = D;
        C = Expand[A - q C];
        D = Expand[B - q D];
        A = Csave;
        B= Dsave]
    {h,A,B}];
```

Note that the syntax here is very similar to our pseudocode for algorithms. To use the procedure to compute the gcd and the polynomials A, B, you would enter a command like this:

```
ExtEuc[fpoly,gpoly,var]
```

where `fpoly` and `gpoly` are the two polynomials, and `var` is the variable. The output from the procedure consists of the last value of h (the gcd, up to a constant multiple), then the polynomials A and B.