Getting Acquainted with Octave

1 Starting Octave

Open an SSH connection to `octave.math.gordon.edu`. This can be done from a Macintosh computer by opening a terminal and typing the command `ssh octave@octave.math.gordon.edu`. From a Windows computer you can use PuTTY[1]; connect to `octave.math.gordon.edu` and login as `octave` using the password ____________. (Ask the instructor if you would like to connect to Octave from off-campus.) You'll then get a message like

```
Last login: Mon Jan 12 10:46:24 2009 from 172.26.8.92
Have a lot of fun...
octave:1>
```

at which point the computer is ready to accept octave commands.

2 Entering data into Octave

Let's start by creating some variables in octave. Type

```
a=1
b=2;
```

where each line is followed by <ENTER> . Notice the difference; the line followed by the semicolon was not echoed. The use of semicolons in this way is optional. Spacing around the variable names and the equal sign is ignored. Now type `a` followed by <ENTER> – you are shown the contents of the variable.

We can overwrite `a` with any other value, this time we'll create a matrix:

```
a = [1 2 ; 3 4]
```

Notice that entries are listed row–by–row, and that the rows are separated by semicolons. Alternatively, each row can be terminated by <ENTER> .

[1] This is already installed in Gordon's lab; you can download it for free from *http://www.chiark.greenend.org.uk/~sgtatham/putty/*. The direct download link for Windows XP, Vista, etc. is *http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe*

3 Solving Linear Systems with Octave

Let's try one of the text exercises. Many of the exercises have already been entered to the computer for us. The naming scheme is "cNsM" where "N" is the chapter number and "M" is the section number. For Section 1.1 you need to enter

```
c1s1
```

and press <ENTER> . You'll then see a prompt for the particular exercise number. Let's try exercise 14. This creates the matrix `M` for us:

```
M =

   1  -3   0   5
  -1   1   5   2
   0   1   1   0
```

which is the augmented matrix corresponding to the linear system

$$\begin{array}{rcrcrcl} x_1 & - & 3x_2 & + & & = & 5 \\ -x_1 & + & x_2 & + & 5x_3 & = & 2 \\ & & x_2 & + & x_3 & = & 0 \end{array}.$$

Notice that the matrix name is an uppercase `M` – variable names are case-sensitive. To solve the linear system, we can perform elementary row operations on `M`. For this purpose we have three *m-files* provided along with the exercises:

SCALE	Scales row `r` of matrix `A` by a nonzero scalar `c`.
Format:	`Y = scale(A,r,c)`
SWAP	Interchanges rows `r` and `s` of matrix `A`.
Format:	`Y = swap(A,r,s)`
REPLACE	Replaces row `r` of matrix `A` by its sum with `m` times row `p`.
Format:	`Y = replace(A,r,m,p)`

Following our algorithm for row-reduction to reduced echelon form, we identify column one as the pivot column and the 1 at the top of it as the pivot element. We now want to create zeros in all entries below the pivot in the first column.

We will use the `replace` command, which requires that we always **add** a scalar multiple of some row to another row. Thus, we need to multiply the first row by 1 and add it to the second row, replacing the second row. The command

```
replace(M,2,1,1)
```

will do this. **Note:** You are asked to verify if this what you really wanted to do; press <ENTER> to accept the result or type anything other than 'y' or 'Y' to undo the change.

You should notice that the output of `replace` operation is named `ans` and if you examine `M` you'll see that it is unchanged. The default variable name for any operation with output is `ans`. What we really wanted was to replace `M` by the modified matrix we created. To do this we can modify our last command so as to assign the result to the matrix `M` (hint: you can use the arrow keys to recall and edit previous commands).

```
M = replace(M,2,1,1)
```

The first column has now been reduced to all zeros except for the pivot so we can turn our attention to the second column. We could scale the second row by (-1/2) to convert the pivot entry to a 1, or we could just swap the second and third rows; let's try that. While this is not strictly necessary, it will make things a little easier — remember that a matrix needs 1's in the pivot positions (leading entries in each row) in order to be in reduce echelon form.

```
M = swap(M,2,3)
```

This time the work is done without a verification prompt. Now `M` looks like:

```
M =

   1  -3   0   5
   0   1   1   0
   0  -2   5   7
```

We can eliminate the -2 in the $(3,2)$ position with the command

```
M = replace(M,3,2,2)
```

Our matrix is now in echelon form. The following commands will complete the reduction to reduced echelon form:

```
M = scale(M,3,1/7)
M = replace(M,2,-1,3)
M = replace(M,1,3,2)
```

We are left with

```
M =

   1   0   0   2
   0   1   0  -1
   0   0   1   1
```

from which we conclude that the solution is $x_1 = 2$, $x_2 = -1$, and $x_3 = 1$. You should verify that this is indeed the solution.

Now that you've row-reduced this matrix step-by-step, you'll probably be happy to know that there is an easier way. Reload the original matrix `M` by using the `c1s1` command and choosing exercise 14. Now type

```
M = rref(M)
```

which replaces `M` with its reduced row echelon form.

4 Finishing Up

When you are ready to quit using octave, just type `quit` or `exit`. Be sure to copy any information you want before doing this, however, as often the window containing your octave session will disappear.

5 More about Octave

Octave is a powerful tool and can do much more than row-reduce matrices. Later in the semester we will be performing more complicated operations, a tool like Octave can be a big help. Time spent using Octave (or a similar tool) now will help pay dividends later in the semester.

If you would like to use Octave on your own computer or just learn more about what it can do, check out the Octave homepage at *http://www.octave.org*.