Math 131/135/194, Fall 2004

Applied Calculus

Profs. Kaplan & Flath

Macalester College

Instructions for Matrix Software

To make the manipulation of vectors and matrices somewhat easier in R, we have written some programs that are contained in the file `matrix.r` which is available both on the web site and the course folder (for Math 194 — the same place the other lab files are located).

Before you can use any of the programs, you will have to "source" the `matrix.r` file into R.

Two of the new functions provide a simple way to construct vectors and matrices.

The built-in `matrix` command described in the handout is a bit hard to use. We have provided you with two new functions, `vec` and `mat`, that can be made available by sourcing the file `matrix.r`. The `vec` function collects scalar arguments into a vector. The `mat` function collects vectors into a matrix.

Here is an example, using problem 4 in the lab instructions:

```
> source('matrix.r') % do this once in each session
> u = vec(3,1)
> v = vec(-1,3)
> w = vec(-4,1)
> b = vec(5,9)
```

The statements above defined 4 vectors to be used in the problem. The `vec` program will work with vectors of any length — just increase the number of arguments to give the dimension you want. Keep in mind that vectors will print out as a column:

```
> b
     [,1]
[1,]    5
[2,]    9
```

To package up a set of vectors into a matrix, use the `mat` command:

```
> A = mat(u,v,w)
```

The printed form of the matrix should be familiar

```
> A
     [,1] [,2] [,3]
[1,]    3   -1   -4
[2,]    1    3    1
```

The new program `project` replaces both `solve` and `lsfit`. The syntax is the same as the old programs, but `project` handles correctly (I hope!) all the situations we need.

```
> x = project(A,b)
> x
           [,1]
[1,]  1.9518519
[2,]  2.4851852
[3,] -0.4074074
```

Remember, they syntax is `project(A,b)`: project $\vec{b}$ onto $\mathbf{A}$. If you get it backwards, there will be an error message:

```
> project(b,A)
Error in project(b, A) : b must be a vector
```

This output suggests that $1.95\vec{u} + 2.48\vec{w} - 0.407\vec{w}$ will be as close as possible that you can get to $\vec{b}$ using the vectors $\vec{u}, \vec{v}$ and $\vec{w}$. Let's check:

```
> 1.9518*u + 2.4852*v - .4074*w
      [,1]
[1,]    5
[2,]    9
```

It seems that we were able to get there exactly!

Using the notation of matrix multiplication, we can compute the above sum with a more concise command:

```
> A%*%x
     [,1]
[1,]    5
[2,]    9
```

The residual is just the difference between the projection of $\vec{b}$ and $\vec{b}$ itself.

```
> resid = A%*%x - b
> resid
              [,1]
[1,] -2.664535e-15
[2,]  1.776357e-15
```

If you look carefully, you will see that these are very small numbers, essentially zero.