

Lab Session 2

MA-423 : Matrix Computations Lab

2018

R. Alam

1. The machine epsilon `eps` of a floating point system is the distance from 1 to the next floating number bigger than 1 and `u = eps/2` is the unit roundoff (default). You can compute `eps` and `u` in MATLAB by writing a small script. What is it that the following MATLAB script computes?

```
x = 2;
while x > 1
x = x/2
end
```

On the other hand, if the condition $x > 1$ in the above script is replaced with $1 + x > 1$ then what will be the output?

2. This exercise illustrates the difficulty in handling polynomials in finite precision computation. Consider the polynomial

$$p(x) = (x - 2)^9 = x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512$$

Write a MATLAB script to evaluate p at 151 equidistant points (use `linspace` command) in the interval $[1.95, 2.05]$ using two methods:

- (a) Apply Horner's method, or call MATLAB function `polyval` (Type `help polyval` for more info).
- (b) Calculate $p(x) = (x - 2)^9$ directly.

Plot these results in two separate figures. For example, if x is a row vector of points in the given interval then the commands

```
>> y = p(x); plot(x, y)
```

will do the job. Do the plots differ from one another? If yes, can you think of possible reasons?

3. A *magic square* is an n -by- n matrix in which each integer $1, 2, \dots, n^2$ appears once and for which all the row, column, and diagonal sums are identical. MATLAB has a command `magic` that returns magic squares. Check its output for a few values of n and use MATLAB to verify the summation property. (The antidiagonal sum will be the trickiest. Look for help on how to “flip” a matrix.)

The MATLAB command `eig` computes eigenvalues and eigenvectors of a square matrix and the command `schur` computes Schur decomposition of a square matrix. Type `help eig` and `schur` for more information. What is the largest eigenvalue of `magic(n)` for $n = 4, 5, 6$ and why? Compute Schur decomposition of `magic(5)`.

4. The function `eigshow` is available in the MATLAB demos directory. The input to `eigshow` is a real, 2-by-2 matrix A , or you can choose an A from a pull-down list in the title. Initially, `eigshow` plots the unit vector $x = [1, 0]^T$, as well as the vector Ax , which starts out as the first column of A . You can then use your mouse to move x , shown in green, around the unit circle. As you move x , the resulting Ax , shown in blue, also moves. What is the shape of the resulting orbit of Ax ? SVD tells us that the blue curve is an ellipse. `eigshow` provides a proof by GUI.

5. The MATLAB file `imagesvd.m` helps you investigate the use of SVD in digital image processing. If you have them available, use your own photographs (jpg format):

```
>> imagesvd('photo.jpg' )
```

The M-file `imagesvd`, with no arguments, provides popup menu access to several images from the demos directories.

How does the choice of approximating rank affect the visual qualities of the images? There are no precise answers here. Your results will depend upon the images you choose and the judgments you make.

6. Consider the matrix given by the MATLAB command `A = gallery(5)`. Compute $B := A^5$. What are the eigenvalues of B ? Now compute the eigenvalues of A using MATLAB command `eig`. What are the eigenvalues of A ? Now plot the eigenvalues with the following commands:

```
A = gallery(5)
e = eig(A)
plot(real(e),imag(e),'r*',0,0,'ko')
axis(.1*[-1 1 -1 1])
axis square
```

What do you observe? Now repeat the experiment with a matrix where each element is perturbed by a single roundoff error. The elements of `gallery(5)` vary over four orders of magnitude, so the correct scaling of the perturbation is obtained with

```
e = eig(A + eps*randn(5,5).*A)
```

Put this statement, along with the plot and axis commands, on a single line and use the up arrow to repeat the computation several times. You will see that the pentagon flips orientation and that its radius varies between 0.03 and 0.07, but that the computed eigenvalues of the perturbed problems behave pretty much like the computed eigenvalues of the original matrix.

This experiment provides evidence for the fact that the computed eigenvalues are the exact eigenvalues of a matrix $A + E$ where the elements of E are on the order of roundoff error compared to the elements of A . This is the best we can expect to achieve with floating-point computation.

*****End*****