Michael Hanke
CSC NA
November 2, 2010

DN2222
Numerical Algebra

KTH Computer Science and Communication

Programming assignment 1

Study the relevant topics before preparing solutions to the assignments. Reading hints: D means Demmel book *Applied Numerical Linear Algebra*, L means the lecture notes *Topics in Numerical Linear Algebra.*

You are encouraged to work in groups of two, and you may work at any time. The course assistant, Jennifer Grünig, will answer questions concerning these assignments at the scheduled lab sessions.

Hand in a report by November 12, at the Student Expedition! The report may be hand written, but would preferably include plots and matrices from MATLAB.

Do not hand in your entire MATLAB code, only a few lines where the interesting computations are done. *I will not like to see long* MATLAB *programs or* `diary` *files!*

A 1. **Properties of floating point arithmetic** (D 1.4-1.6)

There are several ways to compute the values of a polynomial numerically. Three of these are:

Sum: $p(x) = \sum_{k=0}^{n} c_k x^k$ for c a vector of coefficients.

Product: $p(x) = \prod_{i=1}^{n} (x - r_i)$ for r_i a set of roots.

Eigenvalue problem: $p(x) = \det(A - xI)$ for A an $n \times n$ matrix.

The interesting thing is now that these representations behave differently numerically in floating point arithmetic. The product can be computed with a small relative forward error, while the two others only can be computed with a bounded backward error in the coefficients or matrix elements. You will feel the difference in the neighbourhood of a zero of the polynomial and specially if the zero is multiple.

As an example consider the polynomial given by the product and sum

$$p_{65} = (x-6)^5 = x^5 - 30x^4 + 360x^3 - 2160x^2 + 6480x - 7776$$

Do the following:

(a) Compare the values of the polynomial computed these two ways. Do this by plotting the values for an interval around the multiple root at $x = 6$ say $[5.992, 6.008]$ Take 200 points distributed evenly or randomly over the interval. Use the MATLAB routines `poly`, to find the coefficients c_k when the roots r_i are given, and `polyval`, to evaluate a polynomial with given coefficients c_k. Does the plot give a good clue to where the polynomial has its zeros?[1]

(b) Compute the roots of the polynomial! You may get any values in a rather large interval around $x = 6$. The MATLAB routine `roots` will give you a set of complex values of r_i. Plot them as points in the complex plane. Use the command `axis('equal');` to make sure that real and imaginary parts will be scaled in the same way! You will get a five star of size like $(\|c\|\epsilon)^{(1/5)} = (7776 * 2.22e-16)^{(1/5)} = .0044$. For different root finding algorithms, you may get any values in a rather large interval around $x = 6$. (A much more elaborate excercise of this is given in D 1 Question 1.20 3 on page 30)

Remark: The routine `roots` uses the third representation. It finds a representation of the polynomial as an eigenvalue problem for a specially chosen matrix A and computes its eigenvalues. In this case this representation is as ill conditioned as the sum representation. There are several other matrices that have the same eigenvalues as the polynomial roots. Many of these are better conditioned. We will discuss eigenvalue computation later in the course.

A 2. Operations on vectors and matrices in MATLAB, Gauss elimination (L 1.1, D 2.3)

Your task is to write a MATLAB routine for Gaussian elimination without pivoting.

Formulate your routine in terms of operations on vectors and matrices!

Hint: At elimination step i, the computation of the multipliers, $l_{j,i}$, can be done as an operation on a column vector, taken from the ith column of the matrix A. Moreover the innermost loop can be replaced by a rank one modification of the lower southeast part of the matrix, subtracting the outer product of the column vector from the ith column of L that was just computed, and the row vector taken from the ith row of the matrix A.

Show by a few simple examples that your routine is correct.

Remark: The built in routine, called by the \ (backslash) operator or when invoking the `lu` command, does row pivoting.

A 3. The need for pivoting (D2.4.1)

Try the routine[2] you just written to compute a solution x of the simple 2×2 system with

$$A = \begin{bmatrix} \epsilon & 1 \\ 2 & -3 \end{bmatrix} \text{ and } b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Choose a sequence of smaller and smaller ϵ like 10^{-k}, $k = 1 : 17$. Compare the computed solution $\tilde{x}$ to what you got with the built in

[1] If you are in doubt on how to use the different MATLAB commands, `help <command>` is your friend!

[2] If you have not succeed in solving **A** 2, you can use a self-written routine using the standard nested for loops as shown in Section 1.1 of the Lecture Notes.

routine, it is assumed to be close to the exact solution $\hat{x}$! Plot the norm of the error in the solution $e = \tilde{x} - \hat{x}$ and residual $r = A * \tilde{x} - b$, as a function of the pivot ϵ! You will get the best result using logarithmic scaling, command `loglog` in MATLAB.

A 4. **Pivoting, singular submatrices** (L1.3)

The following matrix,

```
B =
       1     2     3     4     5
       6     7     8     9    10
       1     2     3     4     2
      21    22    20    21    20
      17    17    18    20    20
```

is specially devised to give interesting results when doing Gaussian elimination without pivoting. Try it yourself by using your self-written routine! You will get `Inf`, which means “infinity” and occurs when a nonzero number is divided by zero, and `NaN`, which means “not a number” and occurs when zero is divided by zero or two infinities are subtracted.

These things occur because the leading 3×3 submatrix of B is singular, giving a zero pivot at step $k = 3$. Run the standard routine `lu` with pivoting, and no exceptional number will show up, the matrix is nonsingular. For testing this, use MATLAB to compute the condition number of B.

A 5. Modify the leading element b_{11} to $1 + \epsilon$ for a small number $\epsilon \approx 10^{-8}$. Now the infinities will be large numbers. How large will they be compared to ϵ? What happens with the “not a number”?