1 The LAPACK Library

The LAPACK library is a collection of routines for more advanced problems in linear algebra such as solving linear systems of equations and computing eigenvalues of matrices. It works in conjunction with the BLAS library. The LAPACK library is quite extensive, consisting of over 3000 subroutines and functions, though most of these are not used directly by the user. In this assignment you will use the LAPACK routines:

`DGETRF` - this routine computes the LU factorization of a matrix with partial pivoting. *The original matrix A is overwritten with the L and U factors.*

`DGETRS` - this routine solves a linear system given the factored matrix and the pivoting vector from `DGETRF` and the right hand side vector b. (*i.e.*, it applies the pivoting vector to b, solves $Lz = b$ for z, then solves $Ux = z$ for x). *This routine overwrites the vector b with the solution x.*

`DLAMCH` - this routine is a function that will retrieve various numerical properties of the computer you are running the program on.

`DLANGE` - this routine is a function that will compute the 1 and infinity norms of matrices and vectors.

In addition, there is a subroutine `GETCOND` that will compute an estimate of the condition number of A once it has beeen factored using `DGETRF`. This can be used to estimate the number of correct digits in the computed solution using the rule of thumb

$$\text{digits} = \text{INT}(-\log_{10}\left(10\,\epsilon_{\text{mach}} \cdot \text{cond}(A)\right)).$$

To compile your programs, use the syntax

```
pgf95 (list of .f90 files) -llapack -lblas -o prog.exe
```

Note that the order of the libraries is important.

You should make use of BLAS routines where appropriate.

1) **Plate Temperature**

   The data files `prob1_A.dat` and `prob1_b.dat` define the coefficient matrix and right hand side vector respectively of a linear system of equations, $Ax = b$. The solution to this set of equations describes the temperature in a particular flat plate.

   A is a special type of matrix called a *sparse* matrix. This means that most of the elements of A are zero. Storing the entire matrix in a data file would waste a large amount of disk space, so instead, only the non-zero elements of A are stored in the file. The elements of A are stored in *coordinate* format. Each line in the file `prob1_A.dat` looks like

   ```
   row  column   value
   ```

   The first two entries, `row` and `column` give the row and column indicies of the non-zero `value`, i.e., `A(row,column) = value`.

   There is also a file (`prob1_points.dat`) that contains 2 columns of numbers. These numbers represent the x and y coordinates of the points in the plate. These are ordered to be consistent with the rows of A (i.e., the first row of coordinates in `prob1_points.dat` correspond to the first computed temperature).

   You should write a Fortran 90 problem that answers the following questions:

   a) What is the maximum temperature in the plate?

   b) At what x and y coordinates does this maximum temperature occur?

2) **Random Matrix Solution**

Write an F90 program that solves $Ax = b$ when A is a random matrix of size 10000×10000. Use the intrinsic `RANDOM_NUMBER` subroutine to generate A. Use the trick from HW 5 to compute a right hand side vector b that corresponds to an exact solution of all ones.

Solve the system and compute the (true) relative error. Compare the actual number of correct digits in your solution to the estimate obtained from the rule of thumb.

3) **Structured Matrix Solution**

Write a subroutine to generate the matrix

$$A_{i,j} = \begin{cases} 0 & \text{if } j < i-1 \\ 1 & \text{if } j > i-1 \\ -2^{j-1} & \text{if } j = i-1 \end{cases} .$$

For example, if $n = 4$,

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ 0 & -2 & 1 & 1 \\ 0 & 0 & -4 & 1 \end{pmatrix} .$$

Repeat Problem 4 using this matrix. You should run this case using $n = 20$ and $n = 50$.

You should include copies of your programs and output in addition to answering any questions posed. Make sure your programs are well commented.