MATH 609 Numerical Analysis
Programming assignment #1
Direct methods for linear system

1 Problem Formulation

Write a program for solving the system $Ax = b$ by Gaussian elimination.

(1) Write your own routine for the Gaussian elimination with pivoting in a column, and (2) use the corresponding simple driver routine DGESV from LAPACK (since the systems below have banded matrices you may use the corresponding banded variant DGBSV).

Note, that DGESV and DGESVX are the double precision variants of SGESV and SGESVX subroutines described in the 20 pages of the LAPACK User's Guide from the Internet (see the home page of the labs on: http://www.math.tamu.edu/ xjchen/math609/601.html).

2 Specifications

1. The program should use double precision and should be well documented.
2. Report the computed solution and compare the results obtained by different methods. For the report use the provided template on http://www.math.tamu.edu/ xjchen/math609/601.html.
3. First test your program on the following simple example of a diagonally dominant matrix: $a_{ii} = 20, \ \ a_{ij} = 1, \ i \neq j, \ b_i = 19 + n, \ \ i, j = 1, ..., n, \ \ n = 10, \ 20.$ (This problem is just to test your program, the solution is $x_1 = x_2 = ... = x_n = 1$; do not report the results).

3 Computational examples

Solve the following linear systems:

1. $a_{ij} = 1/(i + j - 1), \ b_i = 10, \ \ i, j = 1, ..., n, \ \ n = 10, \ 20, \ 40.$
   Remark. This is so-called Hilbert matrix which is known to be very ill-conditioned. Do not be surprised if you get different answers with different solvers.
2. Consider the following linear system:

   $$x_0 = 0, \quad k_{i-1}(x_i - x_{i-1}) + k_i(x_i - x_{i+1}) + c_i x_i = b_i, \quad i = 1, ..., n, \quad x_{n+1} = 1,$$

   which after the elimination of x_0 and x_{n+1} is written in the form $Ax = b$ with $x \in R^n$. Solve for $n = \ 20, \ 40.$

   Introduce the following notations: $h = 1/(n + 1)$, $k_i = k((i + 0.5)h)$, where $k(x)$ is given below in two cases:

   (1) (constant k(x)): $k(x) = 1$, $c_i = 0$, $b_i = 0$, $i = 1, \ldots, n$.

   (2) (jump in k(x)): $K = 2, \ 10, \ 100$ and

   $$k(x) = 1, \ for \ 0 < x < 0.5, \ and \ k(x) = K, \ for \ 0.5 \leq x < 1, \ c_i = 0, \ b_i = 0.$$

Remark. Since the solution of this system represents an approximation of the solution of a boundary value problem on the interval $(0, 1)$, plot x_i.

3. Now the unknowns are given as a two dimensional array $x_{ij}, i, j = 0, \ldots, n+1$ that satisfy the system

$$(4 + h^2)x_{i,j} - x_{i-1,j} - x_{i+1,j} - x_{i,j-1} - x_{i,j+1} = h^2, x_{0,j} = x_{n+1,j} = x_{i,0} = x_{i,n+1} = 0.$$

   Here $h = 1/(n+1)$ so that the system represents a finite difference approximation of the boundary value problem $-\Delta u + u = 1$ in $\Omega = (0, 1) \times (0, 1)$ and $u = 0$ on the boundary of Ω. Solve for $n = 20,\ 40$. You need to represent this in the standard from $Ax = b$, where A is an $R^{n^2 \times n^2}$ matrix.

   Remark. Again, since the solution $x_{i,j}$ of this system represents an approximation of the solution $u(ih, jh)$ of a boundary value problem in Ω it makes sense to plot $x_{i,j}$.

4 Penalties

There will be penalties for delaying the programming assignment - for each day 5 points (out of 100).