Assignment 1 for ME4233

Lecturer: Mengqi Zhang

Due date: Nov. 5th, 2018

Consider the following Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -8\pi^2 \sin[2\pi(x+y)] \tag{1}$$

defined on $\Omega = [0,1]^2$ with boundary conditions $\partial\Omega = 0$.

- (a) Discretize the equation (in the form of $Au = b$) with the classical 5-point scheme of 2nd order accuracy on a grid resolution of $N_x \times N_y = 61 \times 61$;
- (b) Use LU and QR decompositions to solve the discretized equation. You can think of the solution of LU or QR method as the "true" solution u_{true}. Plot the solution.
- (c) Use Jacobi, Gauss-Seidel and successive over-relaxation (SOR, $\omega = 1.5$) methods to solve iteratively the discretized equation with a convergence criterion $|r| = |u - u_{true}| < 10^{-7}$. Use a random vector to kick off the iteration. Plot the residual $|r|$ as a function of iteration number for the three methods and comment on the results. Try different ω for SOR and comment on the results.
- (d) Try different initial guesses (at least 4) in the iterative methods and compare the decaying rate of the residue with respect to the iteration number. One of them could be $u = \sin[(2\pi(x+y))]$ (which satisfies the equation but not the boundary conditions). Comment on what you get.
- (e) Now temporally change the boundary conditions to
  at $x = 0, 1, u = \sin(2\pi y)$ and
  at $y = 0, 1, u = \sin(2\pi x)$.
  The analytical solution is $u = \sin[(2\pi(x+y))]$ as in (d).
  Solve the new equation numerically using LU or QR decomposition with $N_x = N_y = 11, 21, 31, 41, 51, 61$ and verify the method is of 2nd order accuracy (you are supposed to plot $R_{L^2} = \sqrt{\frac{\sum_{i,j}(u-u_{analytical})^2}{N_x N_y}}$ as a function of h, where $h = dx = dy$).

Suggestions:

- Please use double-precision arithmetic for the computation.

- If you program in Matlab, consider to define sparse matrices as there are many zeros in the matrices you will construct.
- If your code takes too much time due to the operations in the loop, you might want to take advantage of matrix multiplication. For example, in Matlab, for multiplicating two vectors A and B, the following loop

  $u = 0$;
  for i=1:1000
  $u = u + A(i) * B(i)$;
  end

  is worse than $u = A' * B$;

  As another example, it degrades the general efficiency of the code if you do multiplication of constants in a loop. Define a constant as the multiplication of the constants before you go into the loop.
- Please do not use built-in routines to do LU or QR decomposition. If you use the built-in routines, you have to explain what the algorithm and tricks used in those routines.