Math 371 — Computing Project #2
Sept. 27, 2018, Due Oct. 9, 2018

The object here is to write a code to implement naive Gaussian elimination (no row interchanges) adapted to banded matrices. Let $q \geq 1$ be an integer. Suppose the matrix A is banded in the sense that $a_{ij} = 0$ if $|i - j| \geq q$. For example, for $q = 1$, A is diagonal; for $q = 2$, A is *tridiagonal*; for $q = 3$, A is *pentadiagonal*. The bandwidth of A is the band symmetrically located along the main diagonal and contains $2q - 1$ diagonals.

While we can still think of A as a *regular* square matrix, it should nevertheless be stored in a $2q - 1 \times n$ rectangular array B. We give an example below for $q = 3$.

$$B = \begin{bmatrix} 0 & 0 & a_{13} & \dots & & & a_{n-2,n} \\ 0 & a_{12} & a_{23} & \dots & & & a_{n-1,n} \\ a_{11} & a_{22} & a_{33} & \dots & & & a_{nn} \\ a_{21} & a_{32} & a_{43} & \dots & & a_{n,n-1} & 0 \\ a_{31} & a_{42} & a_{53} & \dots & a_{n,n-2} & 0 & 0 \end{bmatrix}.$$

The mapping between A and B is given by

$$a_{ij} = b_{i-j+q,j}, \quad 1 \leq i, j \leq n, \quad |i - j| \leq q - 1.$$

1. Write a code for LU factorization of A. For this, you can adapt the naive Gaussian elimination code by suitably restricting the ranges of the *for* loops. You can store U in the upper part of B and L in the (strictly) lower part. If you wish to keep A intact, then create another $2q - 1 \times n$ array to store U and L.

2. Write a code for the forward substitution again adapted to the banded storage form. Also write a code for back substitution.

   Remark The codes above should work for any q and n, provided as input.

3. Let $q = 2$ and $n = 20$, $a_{ii} = 4$, $a_{i,i-1} = 2$, $a_{i,i+1} = -1$, $b_i = i + 1$. Solve for $\mathbf{x}$. Print all the components of $\mathbf{x}$.

4. Let $q = 3$ and $n = 20$, $a_{ii} = 6$, $a_{i,i-1} = a_{i,i+1} = -2$, $a_{i,i-2} = a_{i,i+2} = -1$, $b_i = i + 1$. Solve for $\mathbf{x}$. Print all the components of $\mathbf{x}$.

Question Besides being banded, what special properties do the two matrices above have? In particular, is naive Gaussian elimination guaranteed not to fail here?