This assignment involves some analysis and some programming. Our goal will be to understand the Jacobi method applied to a matrix-vector problem with matrix $A_3$. Here $A_3$ is the tri-diagonal $n \times n$ matrix given in reading assignment 2.

**Analysis:** Unfortunately, the matrix $A_3$ is not diagonally dominant so we cannot directly apply the theorem given for the Jacobi method. The first problem shows why the Jacobi method still works in this case.

**Problem 1.** A symmetric matrix is always diagonalizable (this is a spectral theorem).

(a) Show that the vectors $\phi_j \in \mathbb{R}^n$ given by

$$\phi_j = (\sin(\pi j/(n+1)), \sin(2\pi j/(n+1)), \ldots, \sin(n\pi j/(n+1)))^t,$$

for $j = 1, 2, \ldots, n$, are the $n$ eigenvectors of $A_3$.

(b) Compute the corresponding eigenvalues.

(c) Let $G$ be the matrix corresponding to the Jacobi method applied to the problem $A_3x = b$ with $x, b \in \mathbb{R}^n$. Show that $\rho(G)$ is less than one. (Hint: Write $G$ as a linear combination of the matrix $A_3$ and the identity matrix so that the eigenvalues of $G$ can be trivially calculated from those of $A_3$.)

The matrix $G$ is symmetric so (as we shall see in a later class), $\rho(G) = \|G\|_2$. This implies that $\|G\|_2$ is less than one so that the Jacobi method applied to $A_3$ converges for any right hand side and starting iterate. Actually,

$$\|e_i\|_2 \leq \rho(G)^i\|e_0\|_2.$$

**Problem 2.** Do Exercises 1 and 2 of Class Notes 2.

**Problem 3.** Decide on a CRS structure in matlab. This involves $n$ and the arrays rind, val and cind. Write a matlab-m function which given $n$, produces the CRS representation of $A_3$.

**Problem 4.** Write a matlab m-file function which does one step of the Jacobi method. This function has the arguments:

(a) whatever structures you use for storing the CRS matrix,

(b) a vector $x_i$ (the $i$’th iterate),

(c) A vector $b$ (the right hand side, of $Ax = b$),

and returns $x_ip$ where

$$x_ip = x_i + D^{-1}(b - A \ast x_i).$$
To implement the above line, use the equivalent form illustrated below:

function \[x_{ip}\]=jacobi(\ldots,xi,b) 

for \(i=1:n\) 
\(s=b(i)\) 
for \(j \text{ in } (1\ldots n) \text{ such that } A_{ij} \neq 0 \text{ and } j \neq i\) 
\(s=s-A_{ij} \ast xi(j)\); 
end 
\(x_{ip}(i)=s/A_{ii}\); 
end

Of course, you need to use the CRS structure to determine which indices are included in the second for loop as well as the values of \(A_{ij}\). This routine must work for general \(A\) in modified CSR storage as it will be used for other matrices in later assignments.

Since the Jacobi method is linear, the convergence is determined by the error behavior which is given by the formula \(e_{i+1} = Ge_i\). Thus, to numerically study the error behavior of the method, we can solve a problem with right hand side 0 and initial iterate \(x_0\). In this case, \(e_i = -x_i\) and so have the same norm.

**Problem 5.** Write a driver routine for doing the Jacobi iteration on the problem \(Ax = 0\) with \(A\) given in CRS form and initial iterate \(x_0 = (1,1,1,\ldots,1)^t\). After each step in your iteration, compute \(\|e_i\|_2\) and continue iteration until the relative error is less than \(\epsilon\), i.e.

\[\|e_i\|_2/\|e_0\|_2 < \epsilon.\]

Using the above driver, run the Jacobi method applied to \(A_3x = 0\) for \(n = 4, 8, 16, 32, 64\) with initial iterate given by \(x_0 = (1,1,1,\ldots,1)^t\). The driver code should look like

```
\(.\)
\(.\)
\(.\)
\i=0; \% iteration counter
repeat
  x=jacobi(\ldots,x,b);
  i=i+1;
until rel_norm(x)<epsilon
```
end

print(i);  % the number of iterations required.

For each \( n \) report
\[
 n \quad i_n \quad (1 - \rho(G_n))^{-1}.
\]
Here \( i_n \) is the number of iterations required to reduce the relative error in
\( \| \cdot \|_2 \) by a factor of .001 and \( G_n \) is the Jacobi reduction matrix associated
with the \( n \times n \) matrix \( A_3 \).

**Problem 6. (Extra Credit.)** You should observe that \( i \) and
\( (1 - \rho(G_n))^{-1} \) grow more or less proportionally in the above problem.
Find a mathematical argument which explains this behavior.