1. Write a matlab script to generate the mesh shown on the figure to the right. The external and internal radii are 1 and 0.4 respectively. There are 20 wedges. The three circular bands have the same width (which is 0.20). You will need to use `delaunay` along the script `assmbl1` shown in class, and `gplot`. You must enter ‘axis equal’ after calling `gplot` in order to get something that looks like a real circle. You need to draw the two axes [but not the text 0.4, 0.6,... shown on the x-axis] Along with your plot of the mesh, show also the sparsity pattern of the assembled matrix you get.

2. Write a matlab script to implement a post-order DFS traversal of an acyclic graph. In fact you will write this for a lower triangular matrix $A$. The function is recursive. It should take for form:

```matlab
function [Lst, Mark] = dfs1(u, A, Lst, Mark)
```

On input, `Mark` is used as a marker array for visited nodes, $u$ is the starting node in the traversal, and `Lst` is an empty array (`Lst = []`) on input and the post-order traversal on return. Both `Mark` and `Lst` must be passed from one recursive call to the next in the procedure. Test your script for the lower triangular matrix $A = \text{tril}(B)$ in which $B = \text{lap2d}(4,4)$. Show the array `Lst` returned when $u = 1$. After this is done, implement a topological sort algorithm in a script like:

```matlab
function [Lst, Mark] = TopSort(A, b)
```

On input $A$ is a sparse lower triangular matrix, and $b$ is a sparse right-hand side. On output, `Lst` is the topological sort order, and `Mark` is an optional output which marks all visited nodes. Apply your script to the same matrix $A$ as above and to $b = e_4 + e_8$ (zeros everywhere except in locations 4 and 8).

3. Give an interpretation of the adjacency graph for powers of matrices (recall that edges $i \rightarrow i$ are not represented). Consider the (strictly lower triangular) matrix on the right. Show the (directed) adjacency graph of $L$. [It will be convenient to draw the vertices 1, 2, 6, 4, 8 3, 5, clockwise on a rough circle and 7 at the centre of the circle]. What are the patterns of the powers of $L$? (use graphs only to show this). Now consider the product $B = L^2$, where each $*$ in the matrix is replaced by one. You will find that $B(8, 1) = 2$. What does this imply on the paths in the original graph of $L$? What can you infer on the pattern of the inverse of $M = I + L$?
4. The goal of this exercise is to write a script for partitioning a graph in 2 subgraphs. Consider a graph Laplacean $L = D - W$, associated with a graph $G = (V, E)$. For now the only requirement is that $w_{ij} \geq 0$ with $w_{ii} = 0$ for all $i$, and $W$ is symmetric. Show that for any vector $x$ we have

$$(Lx, x) = \frac{1}{2} \sum_{i,j} w_{ij} |x_i - x_j|^2.$$ 

The weight matrix $W$ is defined as $w_{ij} = 1$ for $(i, j) \in E$ and zero elsewhere (recall that $w_{ii} = 0$). Given a vector $x \in \{-1, 1\}^n$, i.e., a vector with values $\pm 1$, show that $(Lx, x)$ is a multiple of the number of edge-cuts between the subgraph consisting of vertices having values 1 and those having values -1. What is the min of $(Lx, x)$ over all vectors $x \perp e$? Use this to find a good partitioning of the graph - see class notes. As an example, consider the finite element mesh of Question 1. Find the spectral graph partitioning of this graph by computing the Fielder vector $v$ and assigning partitions based on the sign of $v - \text{median}(v)$. Show the graph with the partitions colored differently for each partition [a modification for gplot will be provided for this].

5. Consider the problem of solving a sparse triangular system $Ax = b$ in the situation where the right-hand side is sparse.

a. Consider first the case when $A$ is the triangular matrix $A = I + L$ where $L$ is shown in (1) of Question 3 and $b$ is the vector $e_3$ (all zeros except $b_3 = 1$). Show the progress of the structure of $x$ in the Column forward solve algorithm in this case. Show how to exploit topological sorting to determine the nonzero pattern.

b. Use the topological sort script you developed in Question 2, to write a script

$$[x] = \text{DepthSolv}(L, b, \text{list})$$

to solve a sparse triangular system with sparse right-hand side. [note the diagonal entries of $A$ are ones and are not needed. Only $L$ is passed]

c. Apply the scripts of the previous question for the following case:

```matlab
%%----------------- sparse right-hand side.
B = lap2D(20, 10);
L = tril(B, -1)/4.0;
%%----------------- rhs = L*e + e
n = size(B, 1);
rhs = zeros(n, 1);
rhs(100) = 1;
%%----------------- get top. sort
[list, depths] = TopSort(L, rhs);
%%----------------- sparse solve
[x] = DepthSolv(L, rhs, list);
```

How many nonzero entries does $x$ have? compare your result with the exact one, i.e., one obtained with the 'backslash' operator.