Numerical Linear Algebra

Due at 2:00pm on Friday October 8, 2021

What you need to get

- `a2q4q5_YOU.ipynb`: a jupyter notebook for **both Q4 and Q5**

What to do

1. **[7 marks]** Consider the system of equations:

$$\begin{aligned} -24x_0 + 12x_1 + 24x_2 - 12x_3 &= -36 \\ -6x_0 + 27x_1 - 42x_2 - 15x_3 &= -69 \\ -12x_0 + 14x_1 + 2x_2 + 20x_3 &= 4 \\ -6x_0 - 9x_1 + 18x_2 + 15x_3 &= 9 \end{aligned}$$

   (a) **[1 mark]** Write the coefficient (system) matrix A and the right-hand-side vector b so that $Ax = b$.

   (b) **[4 marks]** By manual calculation (showing your work), compute the LU factorization (with row pivoting) of the system matrix in part (a). That is, find a permutation matrix P, a unit-diagonal, lower-triangular matrix L, and an upper-triangular matrix U such that $PA = LU$.

   (c) **[3 marks]** Solve the system manually using the LU factorization above. Show your work.

2. **[6 marks]** Suppose a square $N \times N$ nonsingular matrix A has already been factored as

$$A = LU,$$

   where L is unit-diagonal and lower triangular, and U is upper triangular. Consider the system

$$A^2 x = b,$$

   where b is a given vector and x is an unknown vector.

   (a) **[3 marks]** Show how to use the L and U factors of A to solve the linear system *efficiently*.

   (b) **[2 marks]** How many flops does your algorithm use to compute the solution, x? (Include the flops of the initial LU factorization of A.)

   (c) **[1 mark]** Show that the number of multiplications to form A^2 is higher than the flop count for computing x.

3. **[8 marks]** Consider the upper triangular system:

$$\begin{bmatrix} 1 & -1 & \cdots & \cdots & -1 \\ & 1 & -1 & \cdots & -1 \\ & & \ddots & & \vdots \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix}_{N\times N} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

   (a) **[5 marks]** Suppose there is a small perturbation, ϵ, in the last entry of the right hand side, i.e., $b = [1, 0, \ldots, 0, \epsilon]^T$. Assume $0 < \epsilon < 1$. Let x^ϵ be the solution of the perturbed problem. What is the relative error,

$$\frac{\|x - x^\epsilon\|_\infty}{\|x\|_\infty},$$

   measured in the ∞-norm? Compute the error directly without using any estimate. Is this a well-conditioned or ill-conditioned problem for large N? Why?

(b) **[3 marks]** The inverse of the coefficient matrix of part (a) is given by

$$\begin{bmatrix} 1 & 1 & 2 & 4 & \cdots & 2^{N-2} \\ & 1 & 1 & 2 & \cdots & 2^{N-3} \\ & & \ddots & \ddots & \ddots & \vdots \\ & & & 1 & 1 & 2 \\ & & & & 1 & 1 \\ & & & & & 1 \end{bmatrix}.$$

Estimate the relative error in part (a) without using the knowledge of x or x^{ϵ}. Show your calculations.

4. **[7 marks] Google Page Rank:** The objective of this task is to develop Python code that computes the Page Rank of a set of web pages based on the network adjacency graph. The adjacency graph is represented by a matrix G, where

$$G_{ij} = \begin{cases} 1 & \text{if } \exists \text{ a link from } j \text{ to } i \\ 0 & \text{otherwise} \end{cases}$$

(a) **[2 marks]** Complete the Python function

```
y = SparseMatMult(G, x)
```

which multiplies the sparse matrix `G` by the vector `x`. The matrix is stored using the *dictionary-of-keys* method. You can get the key-value pairs using the built-in call `G.nonzero()`. See the function's documentation for full details.

(b) **[5 marks]** Complete the Python function

```
p, iters = PageRank(G, alpha)
```

that finds the steady-state solution (eigenvector for $\lambda = 1$) of the Page Rank problem using the iterative method discussed in class. The output `p` is an $R \times 1$ vector (where R is the number of nodes) containing the node scores, and `iters` is the number of iterations the method took to converge. Your method should use your implementation of `SparseMatMult` from question 4a, and must **not** form a full $R \times R$ matrix at any time (even as an intermediate while evaluating an expression). To input the initial adjacency matrix, you can use `dok_matrix` from the `scipy.sparse` module (see class demonstration `Randy_demo`). Your iterations should start with a uniform distribution in `p`, and terminate once the solution is found to within a tolerance of 10^{-8} (ie. none of the elements in `p` changes by more than the tolerance).

5. **[5 marks] Illegal Trading Network** The employee of First National Bank that wrote the code for `CalculateNet` is named Dennis Reader. After interrogating Dennis Reader, evidence of an underground crime ring was uncovered. The police plea-bargained with Dennis, and he agreed to inform them about the illegal trading of contraband occurring in a black market. The police want to arrest the most influential criminal in the trading network and replace him or her with an undercover police officer. This officer will then use the criminal's accounts to infiltrate the trading network, and find out more about the identities of the other criminals.

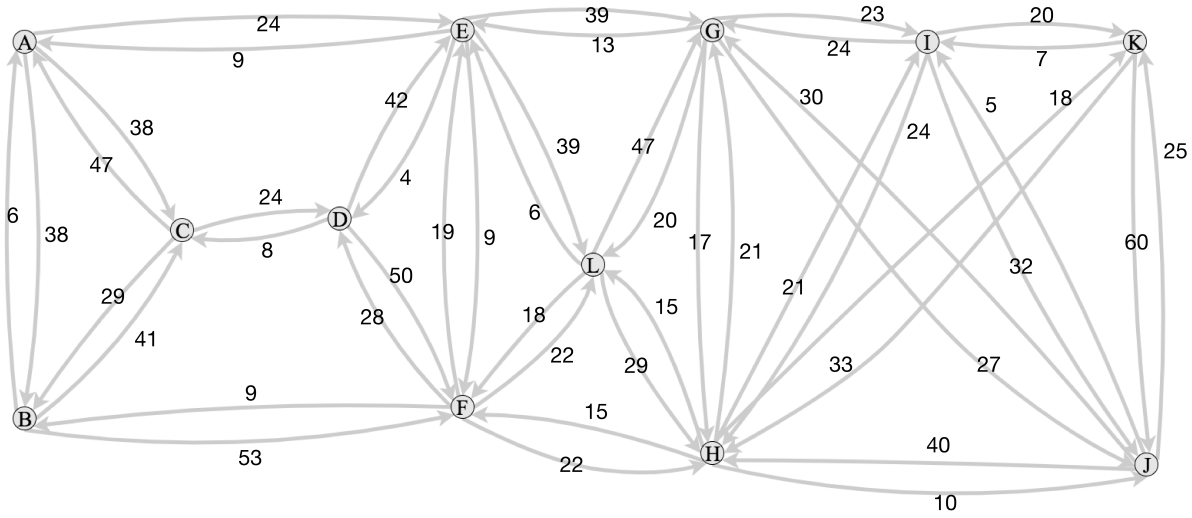
The information Dennis has provided is shown in the diagram below, where each member of the network is represented by a letter. For each member, the arrows pointing away from the node show the percentage of their sales that go along that channel. To find the most influential criminal in the ring, do the following:

(a) **[2 marks]** Add lines of Python code to the notebook to create a sparse matrix representing the network. You should use the `scipy.sparse` module.

(b) **[2 marks]** Add lines to the notebook that run your `PageRank` function (from question 4b) on the network with $\alpha = 1$, creating a bar plot (or stem plot) of the node scores to determine the most influential 'node'. As with any plot, make sure it is labelled appropriately.

(c) **[1 mark]** Write a brief note to the police indicating which node in the trading network is the most influential.

Note: You may assume that the connection matrix has only one eigenvalue of 1. The Markov matrix is not a *positive* matrix, but it is *aperiodic*, which is a different but related property that gives us the same uniqueness guarantee on the largest eigenvalue.



What to submit

For handwritten or typeset questions (such as Q1, Q2 and Q3 in this assignment), you can:

- typeset your solutions using a word-processing application, such as Microsoft Word, LaTeX, Google docs, etc., or
- write your solutions using a tablet computer, or
- write your solutions on paper and take photographs.

In either case, it is **your responsibility** to make sure that your solutions are sufficiently legible. Crowdmark will accept PDFs or image files (JPEG or PNG). You may submit multiple files for each question, if needed.

For programming questions, you must submit your code in two places: on Crowdmark, and on D2L.

Crowdmark: We suggest you rename your jupyter notebooks, replacing "`YOU`" with your WatIAM ID. For example, Prof. Orchard would change the file to `a2q4q5_jorchard.ipynb`. Export each jupyter notebook as a PDF, and submit the PDF to Crowdmark.

D2L: Submit any `.ipynb` files to Desire2Learn in the designated dropbox. You do not need to zip these files.