HW 1

You may (and should) talk about problems with each other and with me, providing attribution for any good ideas you might get. Your final write-up should be your own.

1: A problem of performance MATLAB supports both sparse and dense matrix data structures, and they have different performance characteristics. For a variety of square matrices of size n and sparsity level s (where s is the fraction of entries that are nonzero), compare the speed of dense and sparse matrix-vector multiply. You may use `As = sparse(A)` to make a sparse version of a dense matrix A. What do you observe about the relative performance of these options?

Note: Your performance will vary depending on your machine and MATLAB version. Feel free to write these experiments using Octave, Python, or Julia if you prefer.

2: Toeplitz and transforms Consider a circulant matrix C whose first column is c, i.e.

$$C = \begin{bmatrix} c_1 & c_n & c_{n-1} & \dots & c_2 \\ c_2 & c_1 & c_n & \dots & c_3 \\ \vdots & & & & \vdots \\ c_n & c_{n-1} & c_{n-2} & \dots & c_1 \end{bmatrix}.$$

The following MATLAB code computes a matrix-vector product with C in $O(n \log n)$ time:

```
y = ifft(fft(c) .* fft(x)); % Equivalent to y = C*x
```

Using the circulant multiply as a building block and following the approach sketched in the notes from the first week, write an $O(n \log n)$ multiply for the Toeplitz matrix T with first row r and first column c, i.e. a fast version of

```
y = toeplitz(c, r)*x;
```

3: Low-rank limbo Suppose $A = xy^T$ where $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$. Show that $\|A\|_2 = \|A\|_F = \|x\|_2\|y\|_2$, $\|A\|_1 = \|x\|_1\|y\|_\infty$, and $\|A\|_\infty = \|x\|_\infty\|y\|_1$.

4: Noodling with Neumann Suppose that if $A \in \mathbb{R}^{n \times n}$ satisfies $|A| < M$ (interpreted elementwise) for some $M \in \mathbb{R}^{n \times n}$. Show that if $\|M\| < 1$ in some consistent norm, then both $(I - A)^{-1}$ and $(I - M)^{-1}$ exist and $|(I - A)^{-1}| \leq (I - M)^{-1}$.

5: Simple substitutions Consider the code for forward substitution to solve the linear system $Lx = b$ where L is lower triangular:

```
for i = 1:n
  x(i) = ( b(i)-L(i,1:i-1)*x(1:i-1) )/L(i,i);
end
```

Show that the vector $\hat{x}$ computed in floating point arithmetic satisfies

$$(L + \delta L)\hat{x} = b$$

where $|\delta l_{ij}| \leq n\epsilon_{\text{mach}}|l_{ij}| + O(\epsilon_{\text{mach}}^2)$.