CS556 Iterative Methods Fall 2024 Quiz 1.

Due Thursday, Aug. 29, 5 PM.
(Turn in via the Relate Page.)

Answer the following questions. (Use Saad, *Iterative Methods*, 2nd Ed., to look up any unfamiliar terms.)

1. Suppose $A \in \mathbb{R}^{n\times n}$ is diagonally dominant matrix with $a_{ii} > 0$, $i = 1, \ldots, n$ and let D:=diag(A) be the diagonal matrix with entries $d_{ii} = a_{ii}$.

Using Gershgorin's theorem (Theorem (4.6) in Saad, 2nd. Ed.), describe the region in the complex λ-plane that contains the eigenvalues λ_k of $D^{-1}A$.

How does this region change if we only require $|a_{ii}| > 0$ (i.e., the diagonal entries of A may be of arbitrary sign)?

2. Use `matlab` or `octave` or `python` to answer the following questions.

Suppose A is the $n \times n$ tridiagonal matrix, $A = \frac{1}{h^2}$tridiag(-1,2,-1), with $h = 1/(n+1)$ and (L, U) represent the lower- and upper-triangular factors such that $A = LU$. These matrices are generated in `matlab` by

```
e=ones(n,1);
A=spdiags([-e 2*e -e],-1:1,n,n);
[L,U]=lu(A);
nnzL = nnz(L)
```

The last statement reports the number of nonzeros in L. Asymptotically, we have $nnz \sim Cn^{\beta}$ as $n \longrightarrow \infty$. THe following asks you to find the constants C and β for several matrices.

- How does nnz(L) scale with n?
- How does nnz(L^{-1}) scale with n?
- How does nnz(A) scale with n?
- How does nnz(A^{-1}) scale with n?

Note that you can visualize the content of sparse matrices via `spy(L)`, etc. Just do not do so for large n.

3. Consider the following timing loop:

```
n=1;
for k=1:100 n=ceil( 1.35*(n+1) );
   if n>1e6; break end;
   e=ones(n,1);
   A=spdiags([-e 2*e -e],-1:1,n,n);
   [L,U]=lu(A);  % Warm start to cache L,U
   nloop = max(ceil(1000/n),1);
   tstart=tic;
   for loop=1:nloop;
      [L,U]=lu(A);
      A(1,1) = A(1,1) + 1e-9; % Prevent loop analysis
   end;
   telapse=toc(tstart)/nloop;
   nk(k)=n; tk(k)=telapse;
end;
model=1.e-6*nk;
loglog(nk,model,'k-',nk,tk,'ro','linewidth',2);
xlabel('Matrix Order, n','fontsize',14);
ylabel('Factor Time (sec)','fontsize',14);
title('Tridiagonal Matrix Factor Times','fontsize',14);
print -dpng lutime.png
```

Run this code in octave (the free version of matlab) or matlab. It will generate a plot in lutime.png.

Modify the code to plot the time to instead solve $A\underline{x} = \underline{b}$ where `b=rand(n,1);` is the matlab/octave definition of a random n-vector. You should replace `[L,U]=lu(A);` with `x=A\b;`. Plot the time for this case.

How do these times compare? Do you have a hypothesis regarding why the times differ as they do?

NOTE: In matlab/octave, a ";" at end-of-line suppresses printing of the command output.