ASSIGNMENT 2 SOLUTIONS

Nick Trefethen, 12 November 2019

1. Condition numbers by direct method

We set up anonymous functions to construct the matrix A as indicated:

```
b = @(n) ones(n,1);    A = @(n) .1*sprandsym(n,10/n) + speye(n);
```

Here is a table of condition numbers. With `eig`, we can't get up to a very large dimension.

```
k = 0; t = 0;
while t < 2
  n = 100*2^k; k = k+1;
  rng(1), tic, e = eig(A(n)); t = toc; kappa = max(e)/min(e);
  fprintf('n = %5d  cond(A) = %4.1f  time = %6.3f\n',n,kappa,t)
end

n =   100  cond(A) =  5.6  time =  0.001
n =   200  cond(A) =  6.3  time =  0.005
n =   400  cond(A) =  6.6  time =  0.038
n =   800  cond(A) =  6.5  time =  0.324
n =  1600  cond(A) =  7.8  time =  4.220
```

2. Condition numbers by iterative method

With `eigs`, we can get much farther. The condition numbers jump around but remain $O(10)$.

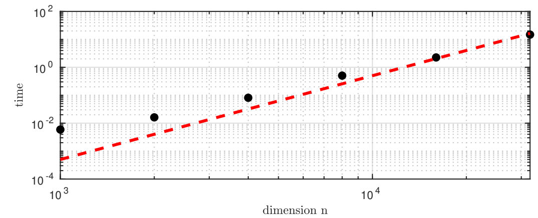
```
k = 4; t = 0;
while t < 4
  n = 100*2^k; k = k+1; tic
  rng(1), emax = eigs(A(n),1,'largestreal');
  rng(1), emin = eigs(A(n),1,'smallestreal');
  t = toc; kappa = emax/emin;
  fprintf('n = %6d  cond(A) = %4.1f  time = %6.3f\n',n,kappa,t)
end
n =   1600  cond(A) =  7.8  time =  0.022
n =   3200  cond(A) =  6.5  time =  0.052
n =   6400  cond(A) =  6.9  time =  0.217
n =  12800  cond(A) =  7.6  time =  0.234
n =  25600  cond(A) =  7.6  time =  0.585
n =  51200  cond(A) =  8.1  time =  1.503
n = 102400  cond(A) =  9.9  time =  2.079
n = 204800  cond(A) = 11.6  time =  4.147
```

3. System of equations by direct method

The time on my machine looks like about $5 \times 10^{-13} n^3$ seconds, which would be 5e5 seconds for $n = 10^6$, i.e., about a week.

```
k = 0; t = 0;
while t < 10
  n = 1000*2^k; k = k+1; rng(1), tic
  x = A(n)\b(n); t = toc;
  fprintf('n = %6d  time = %6.3f\n',n,t)
  loglog(n,t,'.k','markersize',16), hold on
```

```
end
loglog([1000 n],5e-13*[1000 n].^3,'--r','linewidth',2), grid on
xlabel('dimension n'), ylabel('time'), hold off
n =   1000  time =  0.006
n =   2000  time =  0.016
n =   4000  time =  0.081
n =   8000  time =  0.502
n =  16000  time =  2.252
n =  32000  time = 14.774
```

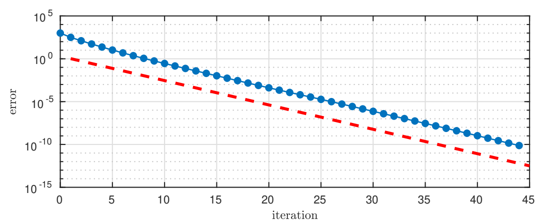


4. System of equations by iterative method

An iterative solution with conjugate gradients, on the other hand, converges in less than a second. Here I use the code `prcg` from the lectures without a preconditioner. The curve corresponds to the classic estimate $2((\sqrt{\kappa}-1)/(\sqrt{\kappa}+1))^n$ with $\kappa = 10$.

```
n = 1e6; rng(1)
tic, x = prcg(A(n),b(n),speye(n)); toc, hold on
kap = 10; r = (sqrt(kap)-1)/(sqrt(kap)+1);
semilogy([1 45],2*r.^[1 45],'--r','linewidth',2)
xlabel('iteration'), ylabel('error'), hold off


Elapsed time is 6.462129 seconds.
```



Here is entry x_{1000} of the solution:

```
format long, x(1000)


ans = 3.105943759802163
```

5. A modified system of equations

Now we make the problem harder, much more ill-conditioned:

```
A = @(n) .1*sprandsym(n,10/n) + diag(sparse(1:n));
```

Unpreconditioned CG is now very slow, but a diagonal preconditioner works beautifully:

```
n = 1e6; rng(1)
tic, x = prcg(A(n),b(n),diag(sparse(1:n))); toc, x(1000)
kap = 10; r = (sqrt(kap)-1)/(sqrt(kap)+1)
xlabel('iteration'), ylabel('error')

Elapsed time is 3.743674 seconds.
ans = 0.001000001373131
r = 0.519493853295916
```