18.085, PROBLEM SET 3 SOLUTIONS

Question 1. (40 pts.) This exercise will show you how to find the Singular Value Decomposition (SVD) of a matrix. Let

$$A = \begin{pmatrix} 1 & 4 \\ 2 & 8 \end{pmatrix}.$$

a) We did this in class! $A = U\Sigma V^T$ so $A^T = V\Sigma^T U^T = V\Sigma U^T$, since Σ is a diagonal matrix. And so $A^T A = V\Sigma U^T U \Sigma V^T = V\Sigma^2 V^T$ since $U^T = U^{-1}$, that is, U is orthonormal. (And this is the eigenvalue decomposition of the matrix $M = A^T A$! Eigenvalues of $M = A^T A$ are the squares of the singular values of A.)

b) $M = A^T A = \begin{pmatrix} 5 & 20 \\ 20 & 80 \end{pmatrix}$. $\det(M - \lambda I) = (80-\lambda)(5-\lambda) - 400 = \lambda(\lambda - 85) = 0$ so eigenvalues are $\lambda_1 = 85$, $\lambda_2 = 0$ (we expect a zero eigenvalue since the matrix has rank 1). Eigenvectors solve $(M - \lambda_1 I)v_1 = 0$ and $(M - \lambda_2 I)v_2 = 0$, and they can be chosen orthonormal since M is symmetric: $v_1 = (1,4)/\sqrt{17}$ and $v_2 = (4,-1)/\sqrt{17}$ ($-v_1$ and $-v_2$ would also work, your answer for u_1 would then be $-u_1$).

c) So $\sigma_1 = \sqrt{\lambda_1} = \sqrt{85}$ and $\sigma_2 = \sqrt{\lambda_2} = 0$.

d) $u_1 = Av_1/\sigma_1 = (1,2)/\sqrt{5}$. But $\sigma_1 = 0$, so we cannot divide by 0 to use this formula.

e) But U has to be orthonormal, hence its second column has to be orthonormal to the first. By inspection, we find that $u_2 = (2,-1)/\sqrt{5}$ would work (or, again, its negative).

f) We do:

```
>> A=[1 4; 2 8];
>> [U S V]=svd(A);
>> U
U =
    -4.472135954999577e-01    -8.944271909999159e-01
    -8.944271909999157e-01     4.472135954999580e-01
>> S
S =
     9.219544457292887e+00                         0
                         0     2.154149081657523e-16
>> V
V =
    -2.425356250363330e-01    -9.701425001453319e-01
    -9.701425001453319e-01     2.425356250363330e-01
```

And we notice that u_1 and v_1 have opposite signs to what we got - this is fine, because we have the choice of signs when we pick eigenvectors - as long as u_1 and v_1 are compatible through $u_1 = Av_1/\sigma_1$ with positive σ_1, it's ok. And signs in u_2, v_2 don't matter as long as they are orthonormal to u_1, v_1 respectively (they don't need to "match" through a formula like u_1 and v_1 do).

g) The reduced SVD of A is $A = \tilde{U}\tilde{\Sigma}\tilde{V}$ for $\tilde{U} = u_1 = \begin{pmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{pmatrix}$, $\tilde{\Sigma} = \begin{pmatrix} \sqrt{85} \end{pmatrix}$, $\tilde{V}^T = v_1 = \begin{pmatrix} 1/\sqrt{17} & 4/\sqrt{17} \end{pmatrix}$. Because σ_2 is 0, it cancels out u_2 (which is orthonormal to the column space of A, so cannot be useful to us because Ax has to be in the column space of A) and v_2 (which is in the nullspace of A and orthonormal to v_1, hence Ax would be 0 anyways for x in the null space of A). This is why we can get rid of u_2, v_2, and the second row and column of Σ.

h) We do:

```
>> Ut=[1/sqrt(5);2/sqrt(5)]
Ut =
     4.472135954999579e-01
     8.944271909999159e-01
>> St=sqrt(85)
St =
     9.219544457292887e+00
>> Vt=[1/sqrt(17);4/sqrt(17)]
Vt =
     2.425356250363330e-01
     9.701425001453319e-01
>> Ut*St*Vt'
ans =
     1     4
     2     8
```

which is indeed equal to A.

Question 2. (40 pts.) We want

$$p(x_j) = y_j \qquad \text{for } j = 1, \ldots, n,$$

with

$$p(x) = \sum_{j=0}^{n-1} c_j x^j,$$

written as a system

$$Ac = y,$$

where c is the vector of coefficients c_j, y is the vector of target values of p at the given points, that is, y contains the y_j's, and we want to solve $Ac = y$ exactly. Finally, we will evaluate this polynomial at m other points.

a) Entries of A are $A_{ij} = x_i^{(j-1)}$, that is,

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{(n-1)} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{(n-1)} \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^{(n-1)} \end{pmatrix}.$$

This is called a Vandermonde matrix, and is famously ill-conditioned!

b) Entries of B are $B_{ij} = t_i^{(j-1)}$, that is,

$$A = \begin{pmatrix} 1 & t_1 & t_1^2 & \ldots & t_1^{(n-1)} \\ 1 & t_2 & t_2^2 & \ldots & t_2^{(n-1)} \\ \vdots & & & & \vdots \\ 1 & t_m & t_m^2 & \ldots & t_m^{(n-1)} \end{pmatrix}.$$

c) Let $n = 4$. This is what you should run and obtain:

```
>> format short e
>> n=4;h=1/(n-1);x=0:h:1;t=h/2:h:1-h/2;
>> A=zeros(n,n); for i=1:n, A(:,i)=x.^(i-1);end
>> B=zeros(n-1,n); for i=1:n, B(:,i)=t.^(i-1);end
>> A
A =
   1.0000e+00            0            0            0
   1.0000e+00   3.3333e-01   1.1111e-01   3.7037e-02
   1.0000e+00   6.6667e-01   4.4444e-01   2.9630e-01
   1.0000e+00   1.0000e+00   1.0000e+00   1.0000e+00
>> B
B =
   1.0000e+00   1.6667e-01   2.7778e-02   4.6296e-03
   1.0000e+00   5.0000e-01   2.5000e-01   1.2500e-01
   1.0000e+00   8.3333e-01   6.9444e-01   5.7870e-01
```

d) What are the condition numbers of A, B, M?

```
>> [cond(A) cond(B) cond(M)]
ans =
   9.8868e+01   2.0618e+01   3.7002e+00
```

e) Let now $n = 10$, and construct A, B and M again. What are their condition numbers?

```
>> n=10;h=1/(n-1);x=0:h:1;t=h/2:h:1-h/2;
>> A=zeros(n,n); for i=1:n, A(:,i)=x.^(i-1);end
>> B=zeros(n-1,n); for i=1:n, B(:,i)=t.^(i-1);end
>> M=B*inv(A);
>> [cond(A) cond(B) cond(M)]
ans =
   1.5193e+07   1.7999e+06   5.4263e+02
```

A and B are badly conditioned (high condition number) but M is not so bad, because Matlab used carefully modified versions of A and B before constructing M.

f) What is the norm of the error in M? `norm(M*A-B) =  3.9960e-10` The error is not as small as we would expect, because of the conditioning of A and B.

g) Let $n = 20$ now. This is what you should run, and what happens:

```
>> n=20;h=1/(n-1);x=0:h:1;t=h/2:h:1-h/2;
>> A=zeros(n,n); for i=1:n, A(:,i)=x.^(i-1);end
>> B=zeros(n-1,n); for i=1:n, B(:,i)=t.^(i-1);end
>> M=B*inv(A);
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 2.235816e-17.
```

h) Report the condition numbers of A and B and the norm of the error in M:

```
>> [cond(A) cond(B) cond(M)]
ans =
   1.1471e+16   9.2524e+14   6.6207e+07
>> norm(M*A-B)
ans =
   2.6316e-01
```

So A and B are very ill-conditioned, which is why there is so much error in M (at this point, A and B are probably not quite accurate either).

I hope you are now convinced of two things: first, worrying about the conditioning of your matrix can save you from making inaccurate calculations (i.e., lots of error). Second, interpolating data at equispaced points is a terrible idea. If you ever have to do this, try cubic splines; or use Chebyshev points (we will not discuss those techniques though, they are beyond the scope of this class); or do not *interpolate*, but *approximate* such as least squares does.

Question 3. (30 pts.)

$$\begin{pmatrix} -\beta & \alpha \\ \alpha & -\beta \end{pmatrix} v = \lambda v. \tag{0.1}$$

a) $\lambda_1 = -\beta - \alpha$ and $v_1 = (1, -1)$, $\lambda_2 = -\beta + \alpha$ and $v_2 = (1, 1)$. Since the matrix is symmetric, we have 2 real eigenvalues and 2 independent eigenvectors (could be made orthonormal).

For $k_1 = k_2 = m = 1$, $\alpha = \frac{k_2}{m} = 1$, $\beta = \frac{k_1+k_2}{m} = 2$. So the matrix is $\begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$, which looks like a small second difference matrix!

b) $-\omega_1^2 = \lambda_1 = -\beta - \alpha = -3$ so $\omega_1 = \sqrt{3}$ and $-\omega_2^2 = \lambda_2 = -\beta + \alpha = -1$ so $\omega_2 = 1$.

c) For $k_1 = .1$, $k_2 = 1000$, $m = 1$, $\alpha = \frac{k_2}{m} = 1000$, $\beta = \frac{k_1+k_2}{m} = 1000.1$. Then $-\omega_1^2 = \lambda_1 = -\beta - \alpha = -2000.1$ so $\omega_1 = \sqrt{2000.1}$ and $-\omega_2^2 = \lambda_2 = -\beta + \alpha = -.1$ so $\omega_2 = \sqrt{.1}$.

d) Since the matrix is symmetric negative definite (eigenvalues are negative), the condition number is the largest eigenvalue in absolute value (= largest singular value) divided by the smallest eigenvalue in absolute value (= smallest singular value). Hence the matrix in (b) has condition number $\frac{|-3|}{|-1|} = 3$, which is not bad at all. But the matrix in (c) has condition number $\frac{|-2000.1|}{|-.1|} = 20001$, which is quite high - we can say this matrix is badly conditioned. Notice that the condition number is the square of the ratio of the frequencies: cond $= \frac{|\lambda_1|}{|\lambda_2|} = \left(\frac{\omega_1}{\omega_2}\right)^2$. So the ratio ω_1/ω_2 of the frequencies of oscillations is the square root of the condition number. This means that, when we have a high condition number, that ratio is moderately large. Or, if we expect to have a moderately large ratio between frequencies in our system, we can expect the system to be badly conditioned, hence we can expect initial errors to be amplified a lot, sadly.