11.2 (a) How closely, as measured in L_2 norm on the interval $[1,2]$, can the function $f(x) = x^{-1}$ be fitted by a linear combination of e^x, $\sin x$ and $\Gamma(x)$? Write a program that determines the answer to at least two digits of relative accuracy using a discretization of $[1,2]$ and a discrete least squares problem. Write down an estimate of the answer and also of the coefficients of the optimal linear combination, and produce a plot of the optimal approximation.

(Optional) (b) Repeat with $[1,2]$ replaced by $[0,1]$. You may find the following helpful:

$$\text{if } g(x) = \frac{1}{\Gamma(x)} \Rightarrow g'(0) = 1.$$

We can use least squares:

$$\text{let } b_i = \frac{1}{1+\Delta x_i} \equiv b(x_i) \qquad i = 0, 1, 2 \ldots, N$$

$$\Delta x = \frac{1}{N}$$

$$\begin{bmatrix} \vdots & \vdots & \vdots \\ e^{x_i} & \sin(x_i) & T'(x_i) \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{pmatrix} c_a \\ c_b \\ c_c \end{pmatrix} = \begin{bmatrix} \vdots \\ b(x_i) \\ \vdots \end{bmatrix}$$

$A \in \mathbb{R}^{N+1 \times 3}$ $\qquad c \in \mathbb{R}^{3 \times 1}$ $\qquad b \in \mathbb{R}^{N+1}$

~~$A = \mathbb{R}^{N \times 3}$~~

$A \in \mathbb{R}^{N+1 \times 3}$

$$Ac = b \qquad \Rightarrow c = A^{+} b$$

$$\begin{cases} c_a \sim -1.0777e-1 \\ c_b \sim 9.2282e-3 \\ c_c \sim 1.2871e+0 \end{cases}$$

$$\frac{\|b - Ac\|}{\|b\|} \sim 7.5864e-4$$

see eleventwo.m

11.3 Take $m=50$, $n=12$. Using matlab linspace define t to be the m-vector corresponding to linearly spaced grid points from 0 to 1. Using matlab vander and fliplr, define A to be mxn matrix associated with least squares fitting on this grid by a polynomial of degree n-1. Take b to be the function $\cos(4t)$ evaluated on the grid. Now, calculate and print to 16 digit precision the least squares coefficient vector x by 6 methods:

(a) via normal equations, using matlab's \
(b) QR factorization via mgs
(c) QR via Householder Triangulation
(d) QR via matlab QR
(e) $x = A\backslash b$ in matlab
(f) SVD using matlab

Comment on differences in outcomes

Round off errors, from smallest to largest:
(1) A\b, (2) SVD, (3) QR, (4) Householder, (5) Modified GS, (6) Normal equations

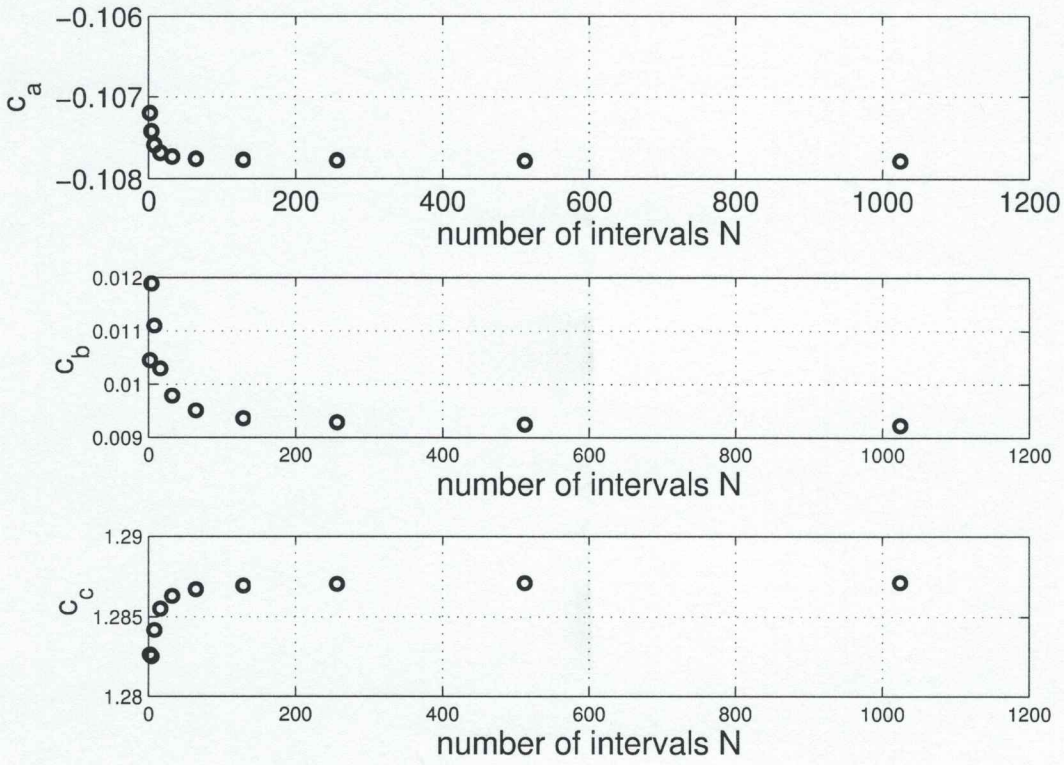


Figure 1: Convergence behavior of coefficients for least square approximation of $f(x)$

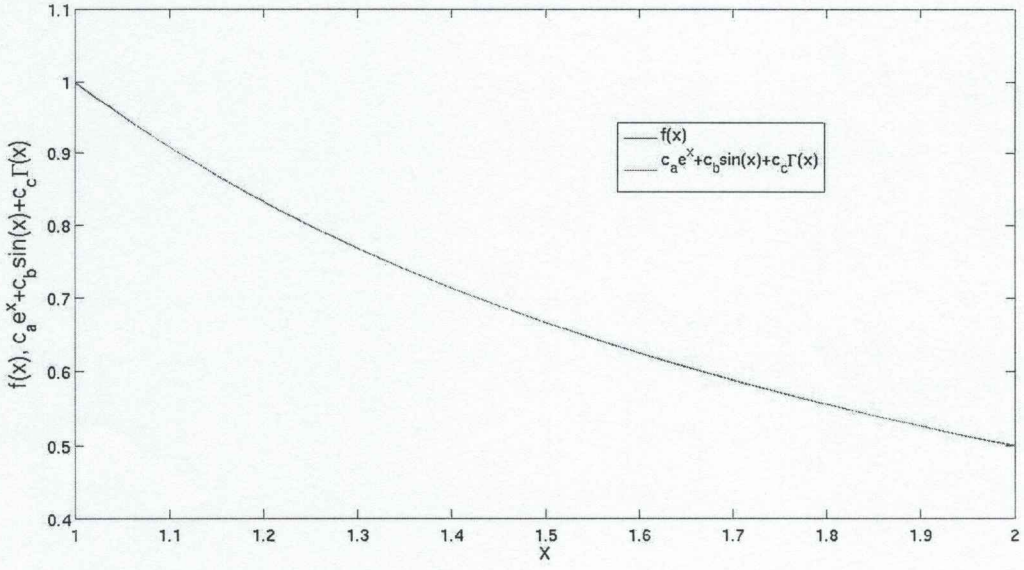


Figure 2: Least square approximation of $f(x)$ in the interval $[1, 2]$

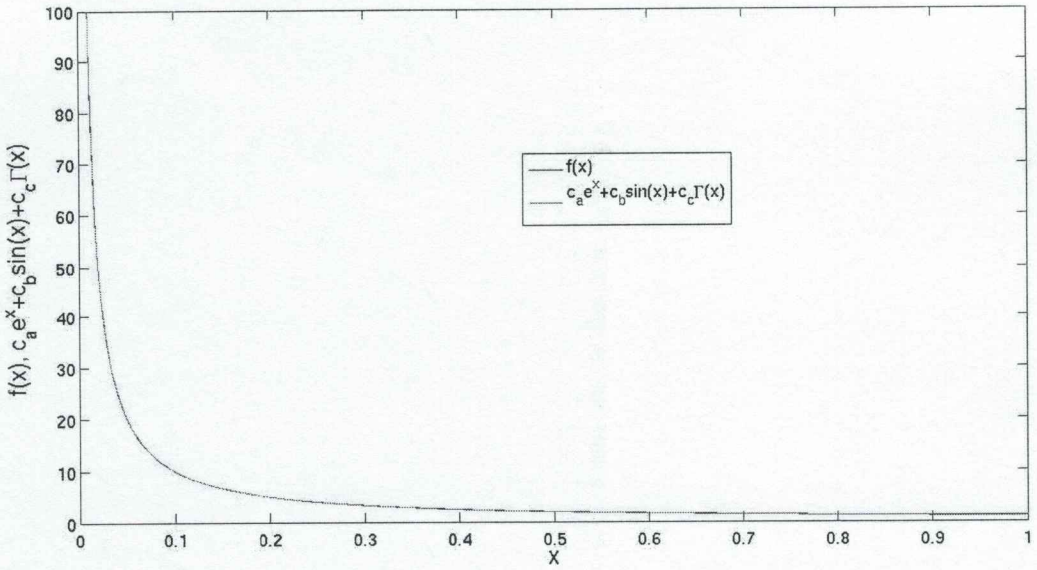


Figure 3: Least square approximation of $f(x)$ in the interval $[0, 1]$

Figure 3 shows that the function $f(x) = x^{-1}$ can be approximated very well with a least square procedure also in the interval $[0, 1]$. However, I circumvented the problem that arises at x=0 by considering a slighly different problem.

Problem 11.3)

The normal equations exhibit instability, i.e. the matrix A^+ is ill-conditioned. From the results shown in table 1 it can be seen that the solution with the normal equations is even worse than the *mgs*-procedure which is known to be unstable. The other 4 methods are equally good for solving the problem.

	1	2	3
norm. eqn.	1.000000007972887	-0.000002500274451	-7.999902739915186
mgs	0.999999998591600	0.000000270667532	-8.000007041537414
house	1.000000000996603	-0.000000422742961	-7.999981235685353
qr	1.000000000996608	-0.000000422742973	-7.999981235687727
$A\backslash b$	1.000000000996608	-0.000000422743228	-7.999981235679865
svd	1.000000000996608	-0.000000422742972	-7.999981235687831
	4	5	6
norm. eqn.	-0.001481595176351	10.678376546894064	-0.054370245025213
mgs	0.000058859566744	10.666556140132517	-0.000909059738683
house	-0.000318763248631	10.669430796050715	-0.013820288685859
qr	-0.000318763206579	10.669430795727829	-0.013820287294560
$A\backslash b$	-0.000318763301264	10.669430796332966	-0.013820289609699
svd	-0.000318763204773	10.669430795712323	-0.013820287217179
	7	8	9
norm. eqn.	-5.531642469895595	-0.287488484503984	1.945133753544229
mgs	-5.683533226645578	-0.008785565866526	1.615241918057822
house	-5.647075625462318	-0.075316027551260	1.693606967008906
qr	-5.647075629157884	-0.075316021267386	1.693606960149606
$A\backslash b$	-5.647075623539531	-0.075316030120297	1.693606969166215
svd	-5.647075629395676	-0.075316020803934	1.693606959575385
	10	11	12
norm. eqn.	-0.179680484459608	-0.296574794164493	0.073989380164090
mgs	0.063558858313696	-0.398174788732086	0.092350019063863
house	0.006032106375708	-0.374241702539007	0.088040575921693
qr	0.006032111024733	-0.374241704319368	0.088040576215936
$A\backslash b$	0.006032105309351	-0.374241702274479	0.088040575901462
svd	0.006032111462623	-0.374241704506582	0.088040576250244

Table 1: Least square coefficients

12.1 Suppose A is 202×202 matrix with $\|A\|_2 = 100$ and $\|A\|_F = 101$

Give the sharpest possible lower bound on the 2-norm condition number $k(A)$

Using Theorem 5.3 we know that $100 = \|A\| = \sigma_1$

and $\|A\|_F = 101 = \sqrt{\sum \sigma_j^2}$

the $k(A) = \frac{\sigma_1}{\sigma_{202}} = \frac{100}{\sigma_{202}}$,

A lower bound on $k(A)$ means an upper bound for σ_{202}.

$$(101)^2 = \|A\|_F^2 = 100^2 + \sum_{j=2}^{202} \sigma_j^2$$

so $\sum_{j=2}^{202} \sigma_j^2 = 201$, We know that σ_j are decreasing. Hence $201 \geq 201 \sigma_{202}^2$ so

$$\sigma_{202}^2 \leq 1$$

$$\therefore k(A) = \frac{\sigma_1}{\sigma_{202}} \geq 100$$

12.2 Polynomial interpolation on equi-spaced points is ill-conditioned. To illustrate phenomenon, let $x_1, x_2 \ldots, x_n$ and $y_1, y_2 \ldots y_m$ be n & m equispaced points from $[-1, 1]$

(a) Derive formula for $m \times n$ matrix A that maps n-vector of data at $\{x_j\}$ to m-vector of sampled values $\{p(y_i)\}$, where p is the degree $n-1$ polynomial interpolant of the data (see Example 1.1)

So A is a linear transformation matrix. It's the composition of the map B that takes n values $\{d_i\}_{i=1}^{n}$ at given points x_i to coefficients c_j of the polynomial

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

that interpolates those values and the map C that takes the coefficients c_j and sends them to the values $p(y_k)$ $1 \le k \le m$.

The matrix of the first map is the inverse of the $n \times n$ Vandermonde matrix V sampled at

x_i. Hence the first map

$$\begin{pmatrix} 1 & x_1^1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2^1 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & & & & \\ 1 & x_n^1 & x_n^2 & \cdots & x_n^{n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix}$$

The second map corresponds to an $m \times n$ Vandermonde matrix

$$V = \begin{pmatrix} 1 & y_1^1 & y_1^2 & \cdots & y_1^{n-1} \\ 1 & y_2^1 & y_2^2 & \cdots & y_2^{n-1} \\ \vdots & & & & \\ 1 & y_m^1 & y_m^2 & \cdots & y_m^{n-1} \end{pmatrix}$$

sampled at y_j. Thus the matrix

$$A = WV^{-1}$$

(b) Write program to calculate A and plot $\|A\|_\infty$ on a semilog scale for $n = 1, 2, \ldots, 30$, $m = 2n-1$

In the continuous limit $m \to \infty$ the numbers $\|A\|_\infty$ are known as the Lebesgue.

for equispaced interpolation, which are asymptotic to

$$\frac{2^n}{(e(n-1)\log n)} \quad \text{as } n \to \infty$$

(see twelvetwo.m)

(c) For $n = 1, 2 \ldots, 30$ & $m = 2n-1$ what is the ∞-norm condition number k of the problem of interpolating the constant function 1? (Use 12.6)

(d) How close is your result for $n = 11$ to the bound in Fig 11.1?

The condition number is given by (12.9)

$$\frac{\|A\| \, \|x\|}{\|Ax\|}$$

In this case we use the ∞-norm, and $\|x\|_\infty = 1$. The code twelvetwo computes the norm. We note that the polynomial that interpolates 1 is 1 and thus the interpolated values are also 1.

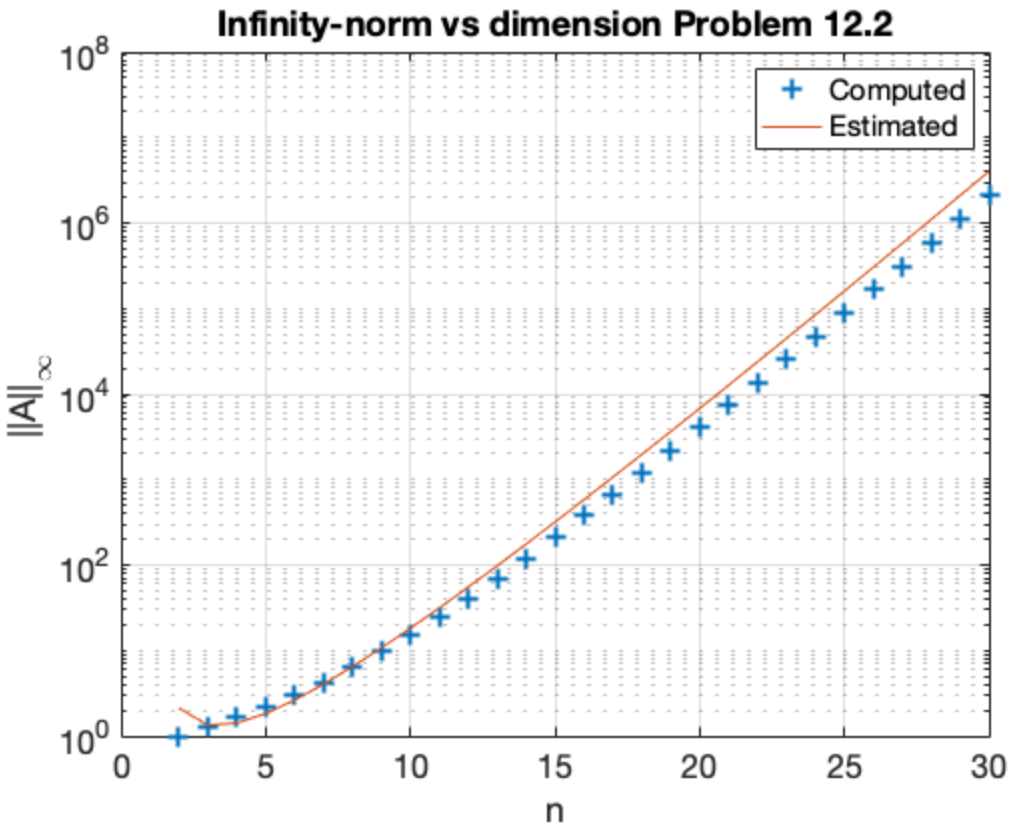
$\therefore \|Ax\|_\infty = 1$ and the condition number

of the problem is the the condition number of the matrix shown in (b).

(d) Let the input data be $d = 1_n + \varepsilon u$ where $u \in \mathbb{C}^n$ is arbitrary and $1_n \in \mathbb{C}^n$ is the vector with all entries 1. Then $\Delta d = \Delta 1_n + \varepsilon \Delta u$

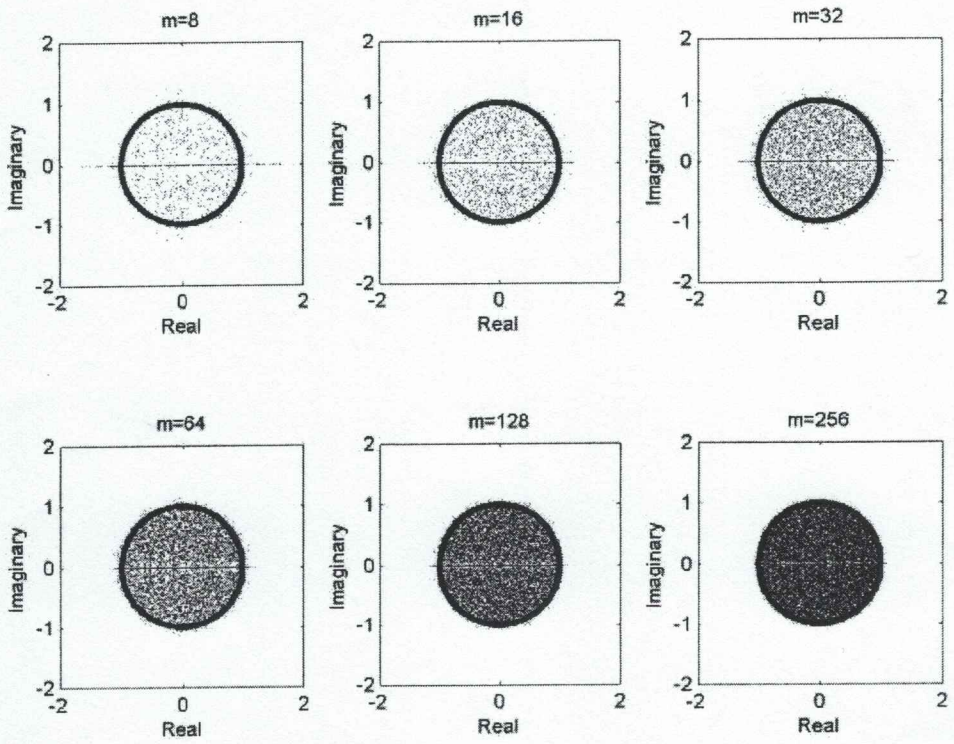
$= 1_m + \varepsilon \Delta u$. The ratio of the size of the perturbation of the output to that of the input is $\|\Delta u\| / \|u\|$. Of course the maximum of this ratio is the condition number computed in part (c). Suppose we take $u = (0,0,0,1,1,1,0,0,0,0)^*$ (this could lead to $\|A\|_{\infty} \sim 4$. But the computed value (also see Fig 11.1) is $\|A\|_{\infty} \sim 24.661$

```
%twelvetwo.m
clear, close all
for nn = 2:30
    mm = 2*nn-1;
    xx=linspace(-1,1,nn)';
    yy=linspace(-1,1,mm)';
    V=fliplr(vander(xx));
    W=fliplr(vander(yy));
    W=W(:,1:nn);
    A=W/V;
    nrm(nn)=norm(A,inf);
end
N=[2:30];
estimate=(2.^N)./(exp(1)*log(N).*(N-1));
semilogy(N,nrm(2:30),'+',N,estimate,'-');
grid
set(gca,'Fontsize',16);
xlabel('n')
ylabel('||A||_\infty')
title('Infinity-norm vs dimension Problem 12.2')
legend('Computed','Estimated')
```



Published with MATLAB® R2018b

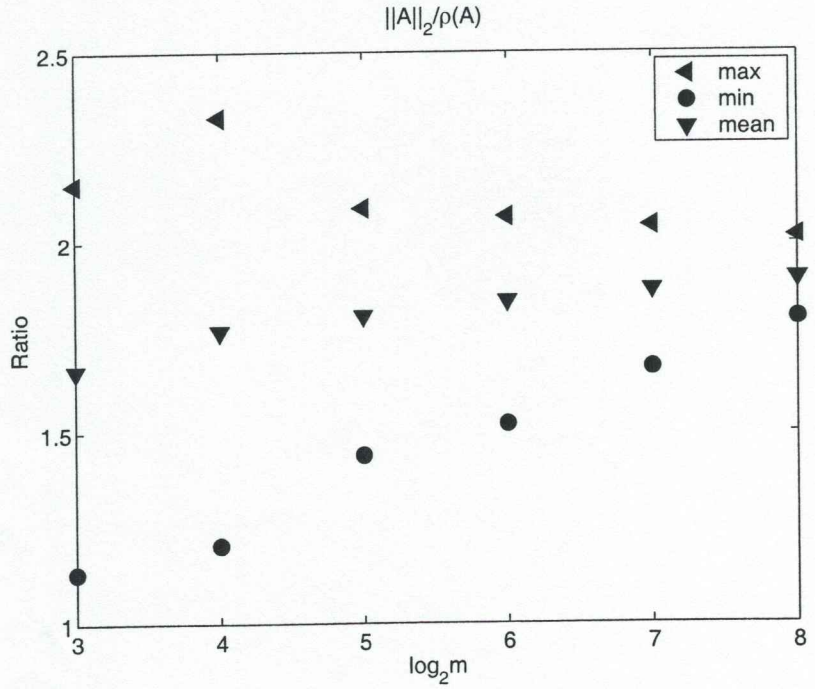
12.3a)



The figure above shows the scatter plots of the imaginary vs. the real parts of the eigenvalues of random matrices normalized as suggested by the exercise. It seems that the eigenvalues mostly lie within the unit circle (shown in the figures). The conjecture that one is led to is that in the limit of large m, the eigenvalues form a distribution that fills the unit circle. It is harder to be sure what the nature of the distribution is, but, it seems to be roughly uniform. The figure was produced using the following matlab code

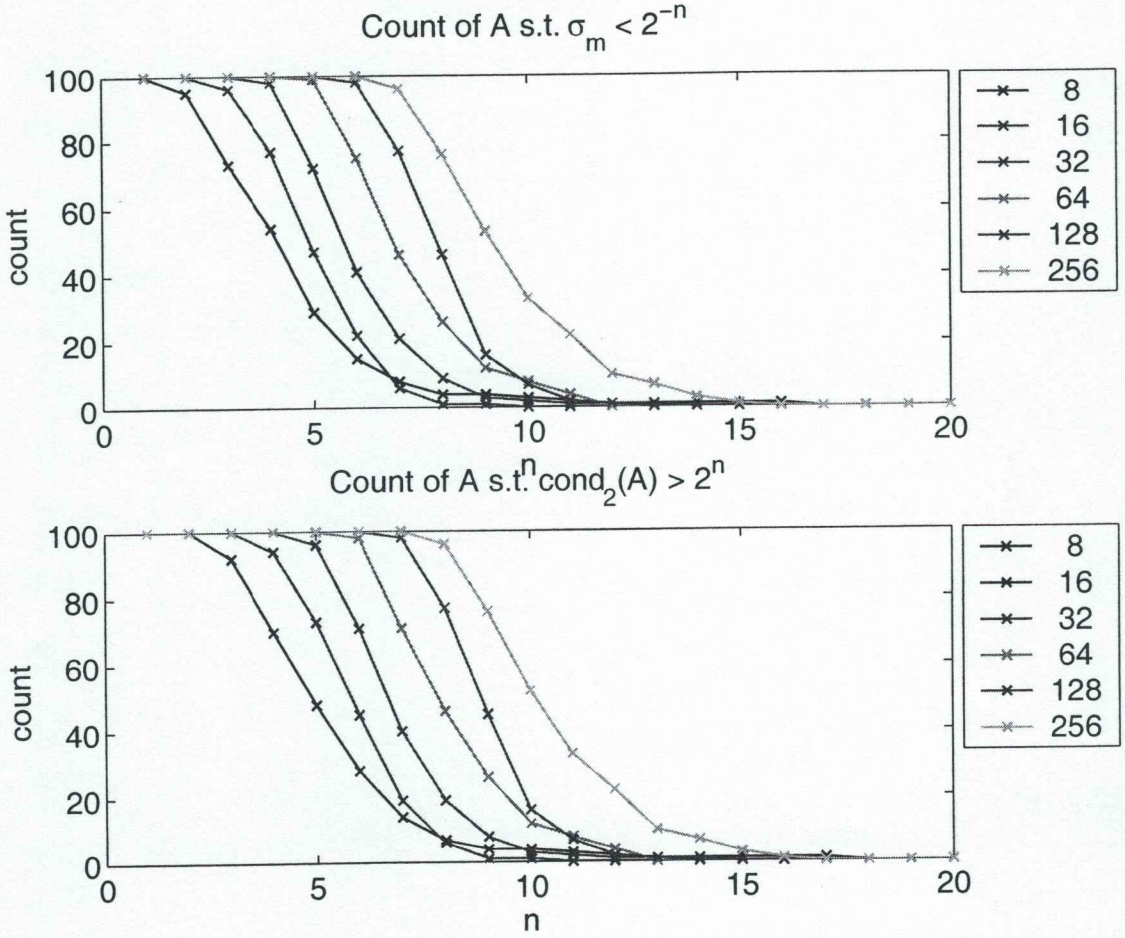
```
ratio=zeros(100,9);
th=linspace(-pi,pi,200);
hold on;
for k=3:8
    m=2^k;
    subplot(2,3,k-2);% this puts multiple plots on one page (2 rows, 3 columns)
    for j=1:100
            if j == 1
                hold off;
            else
                hold on;
            end
```

```
        A=randn(m,m)/sqrt(m); %generate the scaled random matrix
        E=eig(A); % calculate eigenvalues
        plot(E,'.'); %scatter plot.  Note that complex data is automatically graphed
        rho=max(abs(E));
        n2=norm(A,2);
        ratio(j,k)=n2/rho; % find the ratio of 2-norm and spectral radius
    end
    plot(cos(th),sin(th),'k','LineWidth',3); % lay down a comparison unit circle
    xlabel('Real');
    ylabel('Imaginary');
    title(['m=' num2str(m)]);
    axis square % this makes the aspect ratio square (circles don't look like ellipse
                % note this command should be after all title and labels
                % otherwise it may get over-ridden
    hold off;
end
print('-deps2','HW12_3'); % this prints the figure to a file named HW12_3.eps (color
figure(2) hold off;
plot(ks,max(ratio(:,ks)),'<',ks,min(ratio(:,ks)),'o',ks,mean(ratio(:,ks)),'v','Marker
set(gca,'FontSize',16); legend('max','min','mean');
title('||A||_2/\rho(A)'); xlabel('log_2m'); ylabel('Ratio');
print('-deps2','HW12_3_2');
```



The second figure produced by the code above shows how the ratio of induced 2-norm and spectral radius behaves as the dimension grows. The plot suggests that the values of that ratio are heading towards (converging to?) a number a bit less than 2. The sample sizes are small, so the meaning of the minimum and maximum is not obvious. The best I would want to predict is that the typical value of the values seem to be tightening their range.

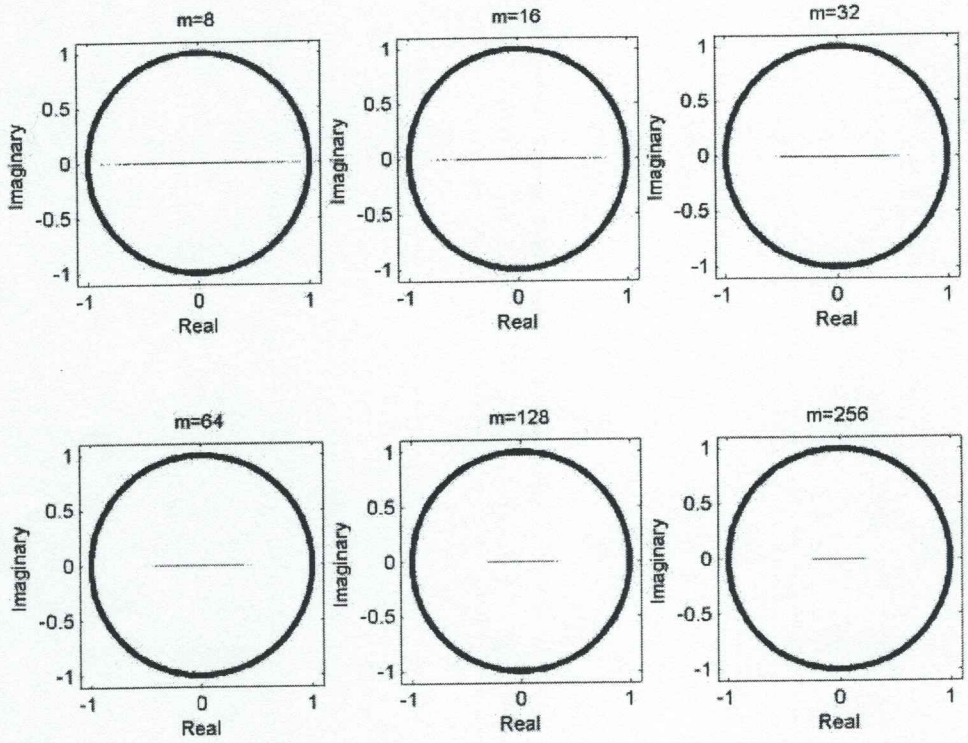
The following figures show how the distributions of the smallest singular value and of the condition number vary with the size of the matrices.

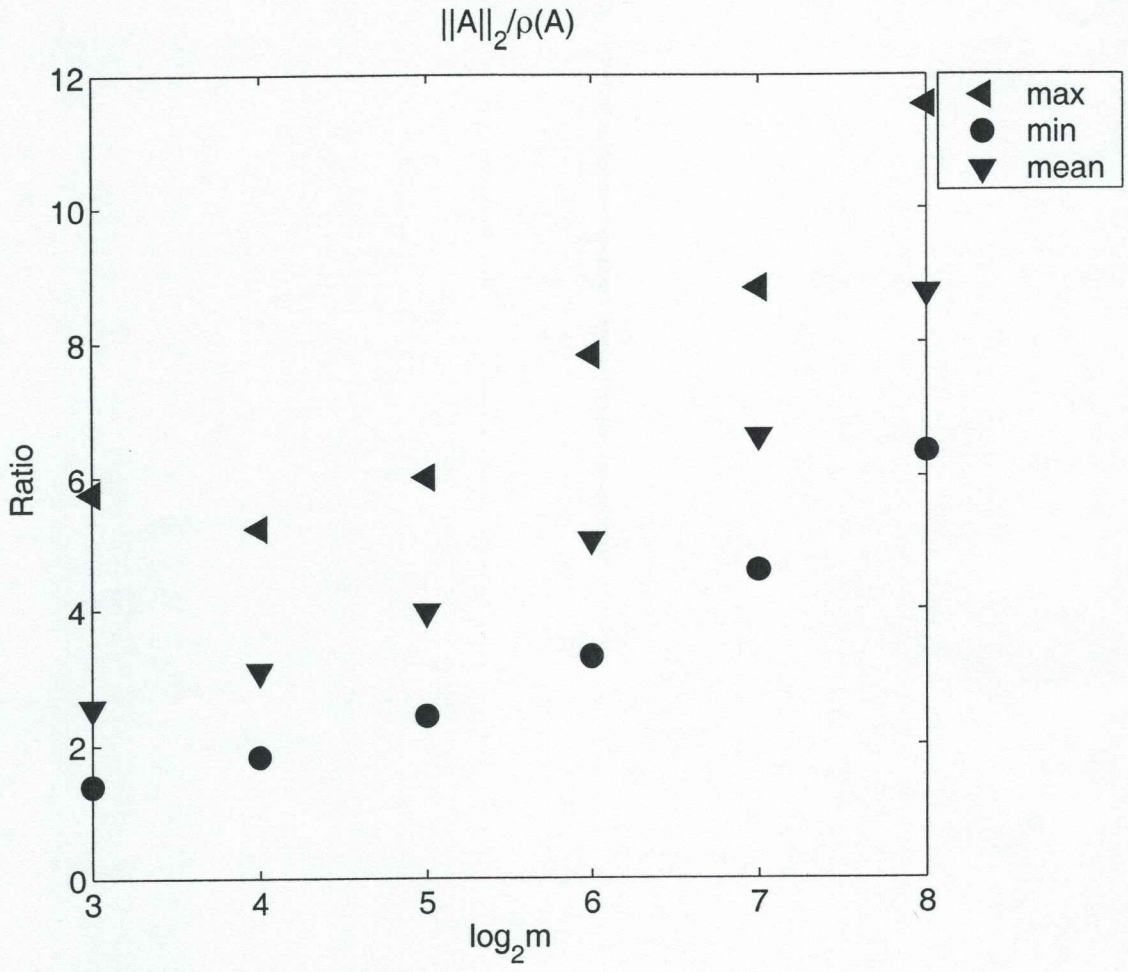


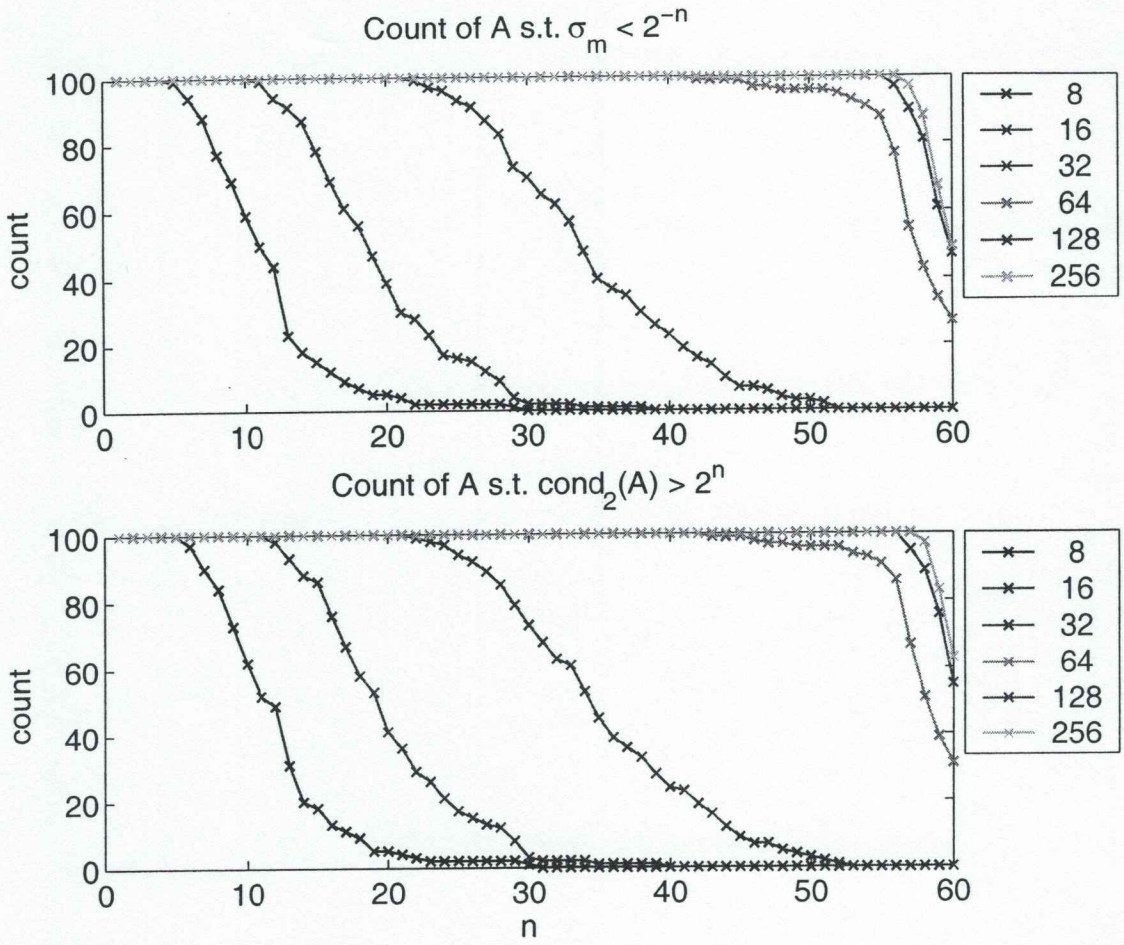
The distribution of both smallest singular value and of the condition number seem to have a consistent shape with increasing matrix size, but be shifted to the right. That is there are more poorly conditioned matrices as the dimension increases. One might speculate that there is a range where the shape of the cumulative distribution function (with exponential scale on the variable) is almost linear.

The same graphs for triangular matrices are quite different, In this case, the eigs routine does a rather poor job. You can get better accuracy by just using "E=diag(A)". This extracts the diagonal elements. They are the eigenvalues. Otherwise, running essentially the

same code you get the following plots:




$||A||_2/\rho(A)$
Ratio
$\log_2 m$
max
min
mean
0
2
4
6
8
10
12
3
4
5
6
7
8



As the first plot reveals, the eigenvalues are all real. They are distributed with a normal distribution with 0 mean and $1/\sqrt{m}$ variance.

The ratio of the norm and the spectral radius appears to be growing with the size of the matrix. It looks faster than linear in $\log(m)$.

The smallest singular values and the condition number are behaving much more poorly. The condition number for larger matrices is always of order $1/\epsilon_m$. Typical triangular matrices are very poorly conditioned.

13.1 $2^{29} - 1$

13.2 a) $\beta^t + 1$

b) $2^{53} + 1$ for double and $2^{24} + 1$ for single

c) There are lots of ways to pursue this. One way is to try to calculate $2^53 + 1 + k$ for $k = -3 : 3$ and compare the results by checking equality. Another is to write out the results as binary ieee doubles, and then read them back in as an array of integers or bits, and compare the actual bit pattern. The code fragment below does the latter.