Part A

1. Suppose you had a computer called the Marc-5 that used floating point arithmetic in binary with 5 bits for the fraction. So, for example, the decimal number ~~0.3~~ 0.4 would be represented in the Marc-5 by

$$1.10011 \cdot 2^{-2}$$

Explain what would happen if you entered the following commands in MATLAB on the Marc-5.

```
t = 0.1
n = 1:10
e = n/10 - n*t
```

Solution. $t = 0.1$ is represented in the Marc-5 in floating point as $t = 1.10011 \cdot 2^{-4}$. The following table lists the elements in the last line.

n	n/10	n*t	e
1	$1.10011 \cdot 2^{-4}$	$1.10011 \cdot 2^{-4}$	0
2	$1.10011 \cdot 2^{-3}$	$1.10011 \cdot 2^{-3}$	0
3	$1.00110 \cdot 2^{-2}$	$1.00110 \cdot 2^{-2}$	0
4	$1.10011 \cdot 2^{-2}$	$1.10011 \cdot 2^{-2}$	0
5	$1.00000 \cdot 2^{-1}$	$1.11111 \cdot 2^{-2}$	$1 \cdot 2^{-6}$
6	$1.00110 \cdot 2^{-1}$	$1.00110 \cdot 2^{-1}$	0
7	$1.01100 \cdot 2^{-1}$	$1.01100 \cdot 2^{-1}$	0
8	$1.10011 \cdot 2^{-1}$	$1.10011 \cdot 2^{-1}$	0
9	$1.11010 \cdot 2^{-1}$	$1.11001 \cdot 2^{-1}$	$1 \cdot 2^{-6}$
10	$1.0000 \cdot 2^{0}$	$1.11111 \cdot 2^{-1}$	$1 \cdot 2^{-6}$

□

2. Show that every rank r matrix A can be written as a linear combination of r rank 1 matrices. Is the set of all rank r matrices in the space of $m \times n$ matrices a vector space?

Solution. The first part is basically an application of the singular value decomposition. Given a matrix A, we can find unitary U and V and diagonal Σ with nonnegative diagonal entries $\sigma_1, \ldots, \sigma_r$ such that

$$A = U\Sigma V^*$$

Let x_i and y_i be the columns of U and V, respectively. Then

$$\sum_{j=1}^{r} \sigma_j x_j y_j^* y_i = \sigma_i x_i$$

Similarly, since $V^* y_i = e_i$

$$Ay_i = U\Sigma e_i = \sigma_i x_i$$

Since $\{y_1, \ldots, y_n\}$ is a basis, we have

$$A = \sum_{j=1}^{r} \sigma_j x_j y_j^*$$

i.e. a linear combination of rank one matrices.

For the second part, the answer is no since the set of rank r matrices does not contain the zero vector. (One can also show that it is not closed under addition.) □

3. Find the *plane* that gives the best fit to the 4 values $z = (0, 1, 3, 4)$ at the corners $(1, 0)$ and $(0, 1)$ and $(-1, 0)$ and $(0, -1)$ of a square. The equations $z = C + Dx + Ey$ at those 4 points are $Ax = z$ with three unknowns (C, D, E). What is A? At the center of the square, show that $C + Dx + Ey =$ average of the z's.

Solution. If the points fit exactly through the plane we would have

$$\begin{aligned} C + D &= 0 \\ C + E &= 1 \\ C - D &= 3 \\ C - E &= 4 \end{aligned}$$

We write in matrix form as

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} C \\ D \\ E \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 4 \end{pmatrix}$$

We thus formulate this as $Av = z$, where

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix}, \quad v = \begin{pmatrix} C \\ D \\ E \end{pmatrix}, \quad z = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 4 \end{pmatrix}$$

The system has no solution so we form the normal equations

$$A^T A v = A^T z, \qquad \begin{pmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} C \\ D \\ E \end{pmatrix} = \begin{pmatrix} 8 \\ -3 \\ -3 \end{pmatrix}$$

which has the solution $C = 2,\ D = -3/2,\ E = -3/2$. Thus the least squares plane is

$$\boxed{z = 2 - \frac{3}{2}x - \frac{3}{2}y}$$

At the center of the square ($x = y = 0$), we have $z = 2$, which is the average of $0, 1, 3, 4$.

□

4. Let A be the $n \times n$ upper triangular matrix with elements

$$a_{ij} = \begin{cases} -1, & i < j \\ 1, & i = j \\ 0, & i > j \end{cases}$$

Show that $\det(A) = \det(A^{-1}) = 1$, but

$$\kappa_1(A) = n2^{n-1}$$

Thus A is badly conditioned for largish n, even though A and A^{-1} are far from singular. For which n does $\kappa_1(A)$ exceed `realmax`? Show by example that there are vectors x such that $\|Ax\|$ is much smaller than $\|x\|$. Since A is already upper triangular, solving $Ax = b$ by Gaussian elimination is done straight away by back-substitution. How much would you trust a solution to $Ax = b$?

Solution. That $\det(A) = 1$ follows from the fact that A is upper triangular. A^{-1} is the Toeplitz matrix with (i, j) entry t_{j-i} where $t_k = 2^{k-1}$ if $k > 1$, 1 if $k = 0$ and 0 if $k < 0$. So, for example, for $n = 5$:

$$A^{-1} = \begin{bmatrix} 1 & 1 & 2 & 4 & 8 \\ 0 & 1 & 1 & 2 & 4 \\ 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A^{-1} is therefore upper triangular with ones on the main diagonal, so $\det(A^{-1}) = 1$.

The $\|\cdot\|_1$ norm is the max column sum. For A and A^{-1} this is the last column:

$$\|A\|_1 = n, \quad \|A^{-1}\|_1 = 1 + \sum_{j=0}^{n-2} 2^j = 1 + \frac{1 - 2^{n-1}}{1 - 2} = 2^{n-1}$$

Thus

$$\kappa_1(A) = \|A\|_1 \|A^{-1}\|_1 = n2^{n-1}$$

$\kappa_1(A) = n2^{n-1}$ exceeds `realmax` when $\boxed{n = 1016}$.

To compare Ax with x, notice that

$$(Ax)_i = x_i - \sum_{j=i+1}^{n} x_j$$

So for any ϵ let $x_n = \epsilon$ and define recursively $x_i = \sum_{j=i+1}^{n} x_j$. Then $(Ax)_i = 0$ for $i = 1, \ldots, n-1$ and $(Ax)_n = \epsilon$. So $\|Ax\| = \epsilon$ but $x_j = 2^{n-j-1}\epsilon$, hence

$$\|x\| = \epsilon 2^{n-1}$$

So for any $\epsilon > 0$ and any M we can find n and x such that $\|Ax\|_1 < \epsilon$ and $\|x\|_1 > M$.

For small n (say, about 10 or less), back substitution should produce an accurate result. But, since A is badly conditioned for large n, any result for large n is suspect. □

Part B

1. The goal of this exercise is to see what happens to the eigenvalues of a matrix if the matrix is perturbed "a little." Suppose $A = SDS^{-1}$, where D is diagonal. The Bauer-Fike Theorem, which can be derived from the Gershgorin circle theorem, says that if ν is an eigenvalue of the perturbed matrix $A+E$, then A has an eigenvalue λ_i such that

$$|\nu - \lambda_i| \leq \kappa_\infty(S)\|E\|_\infty$$

(In other words, the perturbation of the eigenvalues is bounded by the condition number of the eigenvector matrix.)

Define a 6×6 matrix A by

$$A = \begin{bmatrix} 6 & 9 & & & & \\ & 5 & 9 & & & \\ & & 4 & 9 & & \\ & & & 3 & 9 & \\ & & & & 2 & 9 \\ & & & & & 1 \end{bmatrix}$$

(a) Use MATLAB's `eig` function to compute a nonsingular S and diagonal D such that $A = SDS^{-1}$. The columns of S are the eigenvectors of A.

Solution.

```
>> A = diag(6:-1:1) + 9*diag(ones(1,5),1);
>> [S,D] = eig(A);
```

□

(b) Use MATLAB's `cond` command to compute $\kappa_\infty(S) = \|S\|_\infty\|S^{-1}\|_\infty$.

Solution.

```
>> kappa = cond(S,inf)

ans =

   4.0498e+04
```

□

(c) Let A_ϵ denote the matrix obtained from A by changing the $(6,1)$ entry to ϵ. What is $\|E\|_\infty$? Show that $\kappa_\infty(S)\|E\|$ overestimates the perturbations of the eigenvalues. (Try $\epsilon = 10^{-10}, 10^{-6}, 10^{-1}$.)

Solution. $\|E\|$ is the max row sum of E, which is ϵ. Thus $\kappa_\infty(S)\|E\| = \epsilon\kappa_\infty(S)$.

E can be generated by `diag(epsilon,-5)`. A_ϵ can be generated by `A+diag(epsilon,-5)`. Thus, we can calculate the maximal amount an eigenvalue is moved by the command

```
>> max(abs(sort(eig(A+diag(epsilon,-5)))-(1:6)'))
```

The following table show the maximal movement of an eigenvalue and $\kappa_\infty(S)\|E\|$:

ϵ	max movement of eigenvalue	$\kappa_\infty(S)\|E\|$
0.1	3.8295	4.05×10^{3}
10^{-6}	4.9×10^{-3}	4.05×10^{-2}
10^{-10}	4.9207×10^{-7}	4.05×10^{-6}

We see that for small ϵ, $\kappa_\infty(S)\|E\|$ overestimates the movement of the eigenvalues by about an order of magnitude. □

(d) Compute the left eigenvectors of A, that is, those vectors w such that $w^TA = \lambda w^T$, in other words, the eigenvectors of A^T. The condition number of a simple eigenvalue λ is $\kappa(\lambda) = \|w\|\|v\|$, where w is the left eigenvector and v is the right eigenvector, normalized so that $w^Tv = 1$. Show that $\kappa(\lambda)\epsilon$ gives a good estimate for how far the eigenvalue was moved by the perturbation ϵ.

Solution. Perhaps the easiest way to incorporate the normalization is to write the condition number as

$$\kappa(\lambda) = \frac{\|w\|\|v\|}{w^T v}$$

We find the left eigenvectors by `[St,Dt]=eig(A')`. Then the eigenvalue condition numbers can be found by

```
>> kappa=0;
>> for i=1:6
kappa(i)=norm(St(:,7-i))*norm(S(:,i))/(St(:,7-i)'*S(:,i));
end
>> kappa

kappa =

   1.0e+03 *

    0.5774    2.7349    5.3283    5.3283    2.7349    0.5774
```

We see that the condition numbers are around 5×10^3. From the table in part (c), we see that this corresponds pretty closely to how much the eigenvalues are moved. □

2. The general n-body problem involves n mutually attracting masses $m_1, \ldots, m_n$ at position vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, satisfying the $3n$-dimensional second order differential equation

$$\ddot{\mathbf{x}}_i = \sum_{j \neq i} \frac{g m_j (\mathbf{x}_j - \mathbf{x}_i)}{\|\mathbf{x}_i - \mathbf{x}_j\|^3}, \quad i = 1, 2, \ldots, n \qquad (\star)$$

In the *restricted three body problem*, it is assumed that two larger bodies affect but are not affected by a third, smaller body. This could describe, for example, the motion of an Earth-Moon satellite. To simplify, the masses of the larger bodies are scaled to $1-\mu$ and μ and their positions relative to the center of mass are $(\mu, 0, 0)$ and $(\mu - 1, 0, 0)$. Write y_1, y_2, y_3 as the coordinates of the smaller body and y_4, y_5, y_6 as the corresponding velocities. Under these assumptions the equations of motion are

$$\begin{aligned}
\ddot{y}_1 &= 2y_5 + y_1 - \frac{\mu(y_1 + \mu - 1)}{(y_2^2 + y_3^2 + (y_1 + \mu - 1)^2)^{3/2}} - \frac{(1-\mu)(y_1 + \mu)}{(y_2^2 + y_3^2 + (y_1 + \mu)^2)^{3/2}} \\
\ddot{y}_2 &= -2y_4 + y_2 - \frac{\mu y_2}{(y_2^2 + y_3^2 + (y_1 + \mu - 1)^2)^{3/2}} - \frac{(1-\mu) y_2}{(y_2^2 + y_3^2 + (y_1 + \mu)^2)^{3/2}} \\
\ddot{y}_3 &= -\frac{\mu y_3}{(y_2^2 + y_3^2 + (y_1 + \mu - 1)^2)^{3/2}} - \frac{(1-\mu) y_3}{(y_2^2 + y_3^2 + (y_1 + \mu)^2)^{3/2}}
\end{aligned}$$

(a) Write the equations as a system of six first order equations.

(b) Periodic planar orbits exist. For $\mu = 1/81.45$, corresponding to the Earth-Moon system, a planar periodic orbit with $y_3 = 0$ has initial conditions
$(y_1, y_2, y_3, \dot{y}_1, \dot{y}_2, \dot{y}_3) = (0.994, 0, 0, 0, -2.0015851063790825224, 0)$. The period of this orbit is 17.06521656. Write a code to compute this orbit and plot the trajectory in the y_1y_2-plane. Use 'events' to find the period. (Hint: Use as a stopping criterion $\|y(t) - y(0)\|^2 < \delta$ and $t > 1$.)

(c) Another periodic planar orbit has initial conditions $(y_1, y_2, y_3, \dot{y}_1, \dot{y}_2, \dot{y}_3) = (0.87978, 0, 0, 0, -0.3797, 0)$. Compute and plot this orbit and find its period.

(d) (BONUS) Consider the 3 body problem in which the 3 bodies have equal masses. Use the equations $(\star)$ to find a *figure-of-eight orbit*, i.e. a solution in which the three bodies trace out a figure 8.

Solution. Below is the function to integrate the equations. Note that you need a pretty small tolerance in the ode solver to find the periodic solutions.

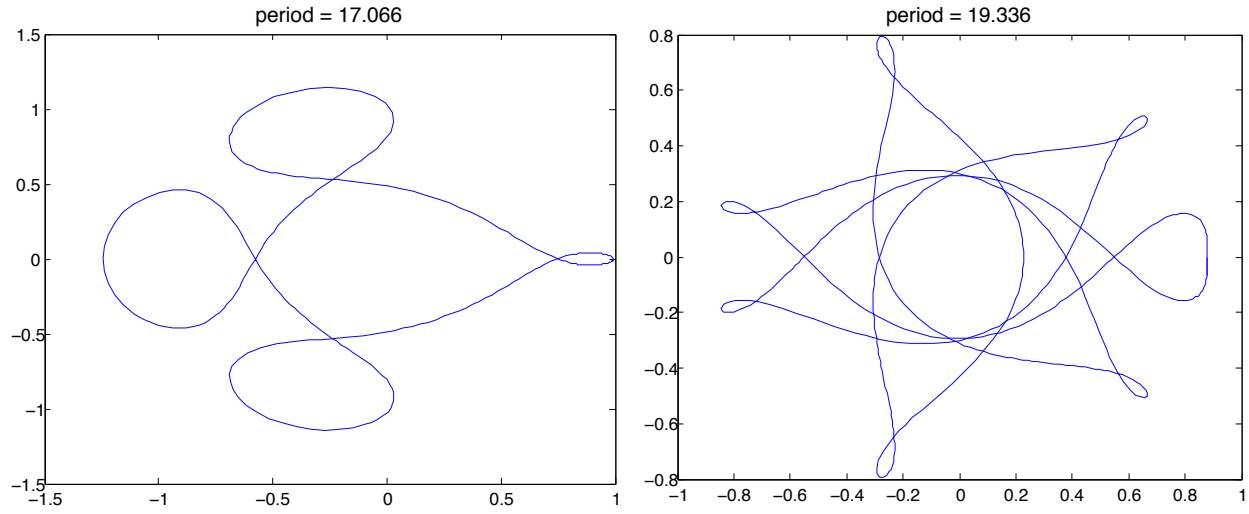
```
function threebody
  global mu
  mu = 1/81.45;

  ic=[.994 0 0 0 -2.0015851063790825224 0]';
  ic=[.87978 0 0 0 -.3797,0]';
  opts=odeset('events',@TBstop,'reltol',1e-10);
  [T, Y] = ode45(@rhs,[0 200],ic,opts,ic);
  figure(1)
  plot(Y(:,1),Y(:,2))
  title(sprintf('period = %.3f',T(end)),'fontsize',12)

function y = rhs(t,x,ic)
  global mu
  y = [x(4);
       x(5);
       x(6);
       2*x(5)+x(1)-mu*(x(1)+mu-1)/(x(2)^2+x(3)^2+(x(1)+mu-1)^2)^(3/2)-...
       (1-mu)*(x(1)+mu)/(x(2)^2+x(3)^2+(x(1)+mu)^2)^(3/2);
       -2*x(4)+x(2)-mu*x(2)/(x(2)^2+x(3)^2+(x(1)+mu-1)^2)^(3/2)-...
       (1-mu)*x(2)/(x(2)^2+x(3)^2+(x(1)+mu)^2)^(3/2);
       -mu*x(3)/(x(2)^2+x(3)^2+(x(1)+mu-1)^2)^(3/2)-...
       (1-mu)*x(3)/(x(2)^2+x(3)^2+(x(1)+mu)^2)^(3/2)];

function [gstop,isterminal,direction]=TBstop(t,y,ic)
  d=[y(1)-ic(1) y(2)-ic(2) y(4)-ic(4) y(5)-ic(5)];
  gstop=d*d'-.01 + (t<1);
  isterminal=1;
  direction=1;
```

The figures produced by this code using the two different initial conditions are shown below.



□

3. The goal of this exercise is to compare the number of prime numbers less than x to $x/\ln x$ and $\mathrm{Li}(x)$, where $\mathrm{Li}(x) = \displaystyle\int_2^x \frac{dt}{\ln t}$ is the offset logarithmic integral. The prime counting function $\pi(x)$ is defined to be the number of prime numbers less than or equal to x. For instance, $\pi(10) = 4$, since there are four prime numbers (2,3,5 and 7) less than or equal to 10. The Prime Number Theorem (Hadamard and Vallée-Poussin) states that

$$\lim_{x\to\infty} \frac{\pi(x)}{x/\ln x} = 1$$

or $\pi(x) \sim \dfrac{x}{\ln x}$. Dirichlet (1838) conjectured that a better approximation to $\pi(x)$ is $\mathrm{Li}(x)$.

   (a) Write a program to compute $\pi(x)$. (Hints: To test if N is prime, one need only test if $p = 2, 3, 5, \ldots, \sqrt{N}$ divide N. When computing $\pi(x+1)$, you may use $\pi(x)$ if you have already computed it.)

   (b) Write a program to compute $\mathrm{Li}(x)$. Write your program so that it can return values fairly quickly for large values of x. (You will have to decide which kind of quadrature algorithm to use, e.g. trapz or quad, and what kind of steps to use. Should you use equal step-sizes?)

   (c) Plot $\pi(x)$, $x/\ln x$ and $\mathrm{Li}(x)$ for x up to 100,000. Can you go to 1 million? 1 billion? Verify that $\pi(x)/(x/\ln x)$ is approaching 1. Which gives a better approximation, $x/\ln x$ or $\mathrm{Li}(x)$? (Note: It is known that $\pi(x) > \mathrm{Li}(x)$ for some x, but the smallest x for which this has been proven is 1.39822×10^{316}.)

   (d) (BONUS) Show that $\pi(x) = \mathrm{Li}(x) - \frac{1}{2}\mathrm{Li}(\sqrt{x}) - \sum_\rho \mathrm{Li}(x^\rho) +$ smaller terms, where the sum is over zeros ρ of the Riemann zeta function. (Note: To prove this you will first have to verify the Riemann hypothesis.)

Solution. The most computationally expensive part of this problem is to compute $\pi(x)$. Probably the most efficient way to do it is to use the Sieve of Eratosthenes. The following code uses the sieve to make a list of all the prime numbers up to a certain number. It takes about 21 seconds on my Dell desktop to find the primes up to 1 million.

```
function primes = EratosthenesSieve(N)

% Returns all prime numbers less than N
% using the "Sieve of Eratosthenes" algorithm.

primes=[2 3:2:N];

j=1;
while primes(j)<=sqrt(N)
    primes(nonzeros((rem(primes(j+1:end),primes(j))==0).*(j+1:length(primes))))=[];
    j=j+1;
end
```

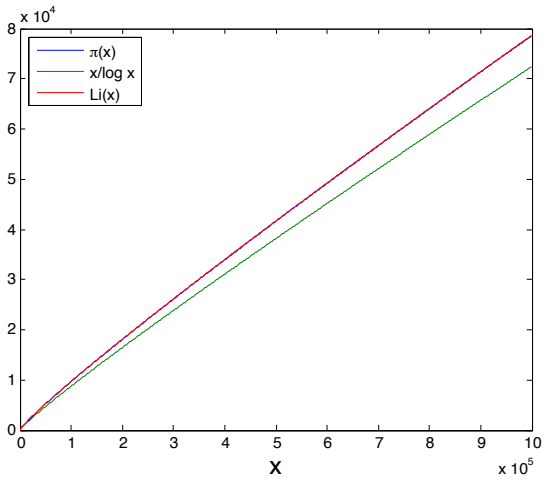
To compute $\mathrm{Li}(x)$ it is best to use `trapz` or `cumtrapz`. `quad` will take forever. We can use `cumtrapz` with the x values being the prime numbers (this is more than enough). The following code computes $\pi(x)$ and compares with $\mathrm{Li}(x)$ and $x/\log x$. The graph follows.

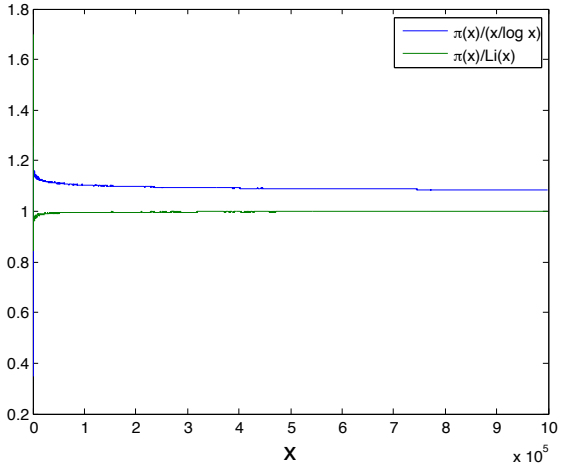
```
N=10^6;
primes=EratosthenesSieve(N);

Li=cumtrapz(primes,1./log(primes));

figure(2)
plot(primes,1:length(primes),primes,primes./log(primes),...
    primes,Li)
legend('\pi(x)','x/log x','Li(x)','location','northwest')
xlabel('x','fontsize',16)

figure(3)
plot(primes,(1:length(primes))./(primes./log(primes)),...
       primes,(1:length(primes))./Li)
xlabel('x','fontsize',16)
legend('\pi(x)/(x/log x)','\pi(x)/Li(x)')
```



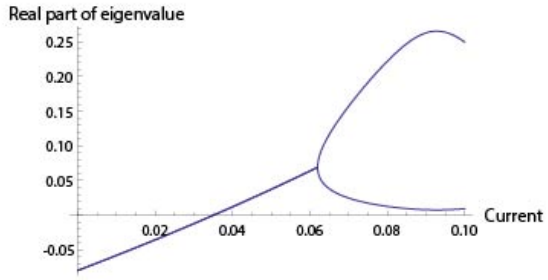


From the figures we see that $\mathrm{Li}(x)$ gives a much better approximation. The ratio of $\pi(x)$ to $x/\log x$ does approach 1, but very slowly. □

4. The Fitzhugh-Nagumo equations model the response of a neuron to an input current I:

$$\begin{aligned} \frac{dv}{dt} &= I - v(v-a)(v-1) - w \\ \frac{dw}{dt} &= \epsilon(v - \gamma w) \end{aligned} \qquad (\star)$$

Here v represents the voltage and w is a "helper" variable. Time t is in ms. Parameters used are usually $\epsilon = 0.008$, $a = 0.139$, $\gamma = 2.54$. There is one fixed point. A neuron "fires" when its voltage spikes. Below is a figure showing the real parts of the eigenvalues of the fixed point. The goal of this exercise is to recreate this graph and interpret what it means through simulation.



(a) Write a function that returns the eigenvalues of the fixed point for any input I. (This function will first have to find the fixed point, then calculate the Jacobian at the fixed point, and finally calculate the eigenvalues of the Jacobian.)

Solution. The following function will return the eigenvalues:

```
function lambda = FNeig(I)

epsilon=0.008;
a=0.139;
gamma=2.54;

vs = fzero(@(x)(I-x.*(x-a).*(x-1)-x/gamma),0);
ws = vs/gamma;

A = [-(vs-a)*(vs-1)-vs*(vs-1)-vs*(vs-a) -1;
    epsilon -epsilon*gamma];

lambda = eig(A);
```

□

(b) Plot the real parts of the eigenvalues as a function of I for $0 \leq I \leq 0.1$.

Solution. This will give a graph like the one above:

```
Ivals = linspace(0,.1,100);
index=1; lambda=[0 0];
for I=Ivals
    lambda(index,:)=FNeig(I)
    index=index+1;
end

plot(Ivals,real(lambda))
```

□

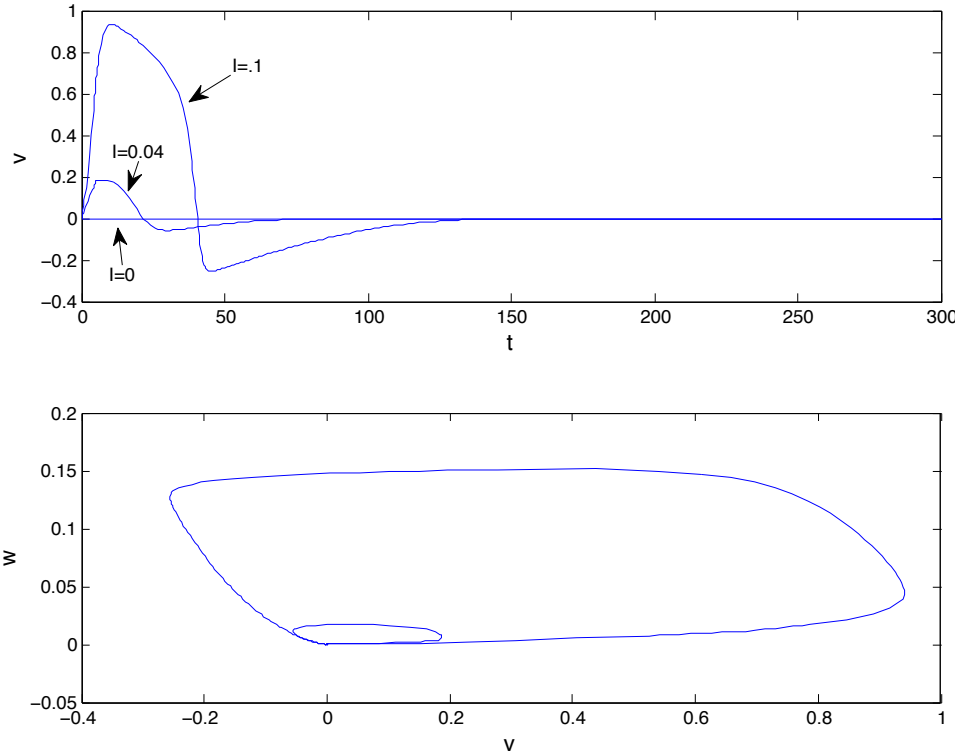
(c) Solve the system ($\star$) when I is turned on for 5 ms at values of $I = 0$, $I = 0.04$ and $I = 0.1$. (So, e.g. in the latter case $I = 0.1$ if $0 \leq t \leq 5$ and 0 for $t > 5$.) Use $(v, w) = (0, 0)$ for your initial condition, and run the simulation to 300 ms. Plot the phase portraits and $v(t)$ as a function of time. Interpret the results in terms of what you found in part (a).

Solution. The following code will produce a time plot of $v(t)$ and an orbit in the $vw-$plane. To get the other ones change I to 0.04 and 0.1.

```
I=0;
epsilon=0.008;
a=0.139;
gamma=2.54;

FNrhs=@(t,x)[I*(t<5)-x(1)*(x(1)-a)*(x(1)-1)-x(2);
    epsilon*(x(1)-gamma*x(2))];
[T,Y] = ode45(FNrhs,[0 300],[0 0]);
figure(1)
subplot(2,1,1)
hold on
plot(T,Y(:,1))
xlabel('t','fontsize',12)
ylabel('v','fontsize',12)
subplot(2,1,2)
hold on
plot(Y(:,1),Y(:,2))
xlabel('v','fontsize',12)
ylabel('w','fontsize',12)
```

With the three values of I, we get the following plot:



When I is large enough so that the eigenvalues have a positive real part (around .038), the fixed point becomes unstable. The trajectory gets kicked away from the fixed point and then comes back. This corresponds to an action potential in the neuron.

□