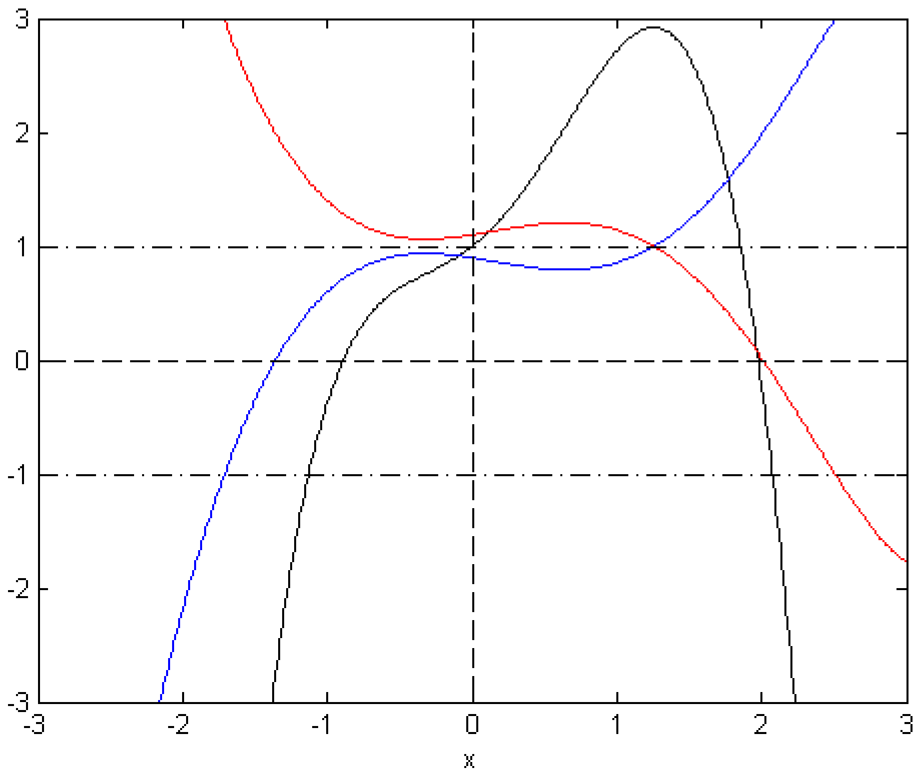
Prob 1 Solution, Example 1 (prepared by HPH)

This example uses $g_1(x) = x + 0.1(x e^x - x^4 + 0.1)$ and $g_2(x) = x - 0.1(x e^x - x^4 + 0.1)$ for the positive and negative solutions, respectively. The factor of 0.1 helps us adjust the magnitude of g'(x) so as to bring it to within |g'(x)| < 1 in the neighborhood of the desirable solution. The original function f(x) and the first derivative of g1(x) and g2(x) are shown in the following figure in black, red, and blue. The good interval for the initial guess for each solution can be readily inferred from the plot. We use x = 2 and x = 0 for the positive and negative solutions. from the plot. The matlab code is in next page. (The actual segment of the code for finding the solution is just a few lines.)



Positive solution is 1.976937

Negative solution is –0.892466

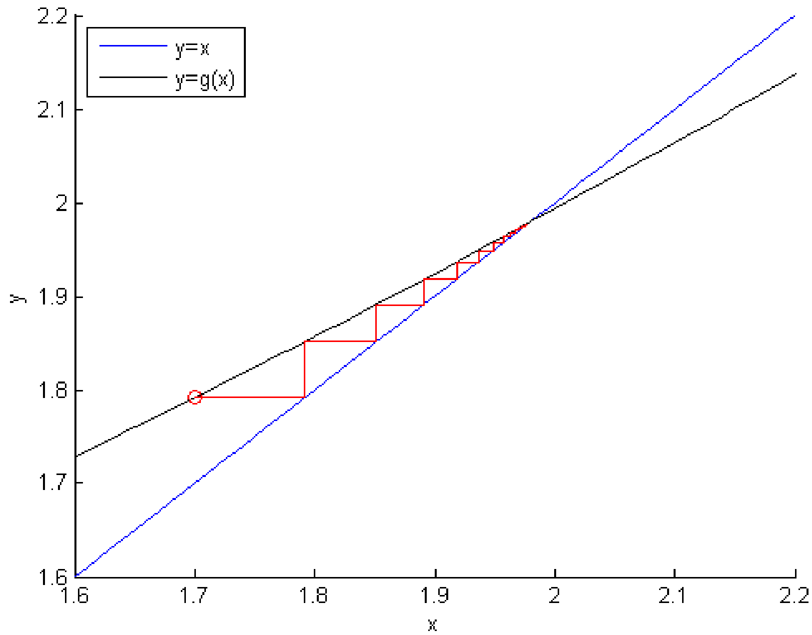
(Matlab code for Prob 1 Example 1)

```
x = [-3:0.01:3];
f = inline('x*exp(x)-x^4+1','x');
g1 = inline('x+0.1*x*exp(x)-0.1*(x^4)+0.1','x');
g2 = inline('x-0.1*x*exp(x)+0.1*(x^4)-0.1','x');
g1p = inline('1+0.1*exp(x)+0.1*x*exp(x)-0.4*(x^3)','x');
g2p = inline('1-0.1*exp(x)-0.1*x*exp(x)+0.4*(x^3)','x');
for k = 1:601
    p1(k) = f(x(k));
    p2(k) = g1p(x(k));
    p3(k) = g2p(x(k));
    z(k) = 0;
    zp1(k) = 1;
    zn1(k) = -1;
end
plot(x,p1,'-k',x,p2,'-r',x,p3,'-b',...
    x,z,'--k',x,zp1,'-.k',x,zn1,'-.k')
xlabel('x')
hold on; plot([0 0],[-3 3],'--k')
axis([-3 3 -3 3])
%
% positive sol (using g1)
%
xs = 2;
for iter = 1:50
    xs = g1(xs);
end
fprintf('Positive solution is %10.6f \r',xs)
%
% Negative sol (using g2)
%
xs = 0;
for iter = 1:50
    xs = g2(xs);
end
fprintf('Negative solution is %10.6f \r',xs)
```

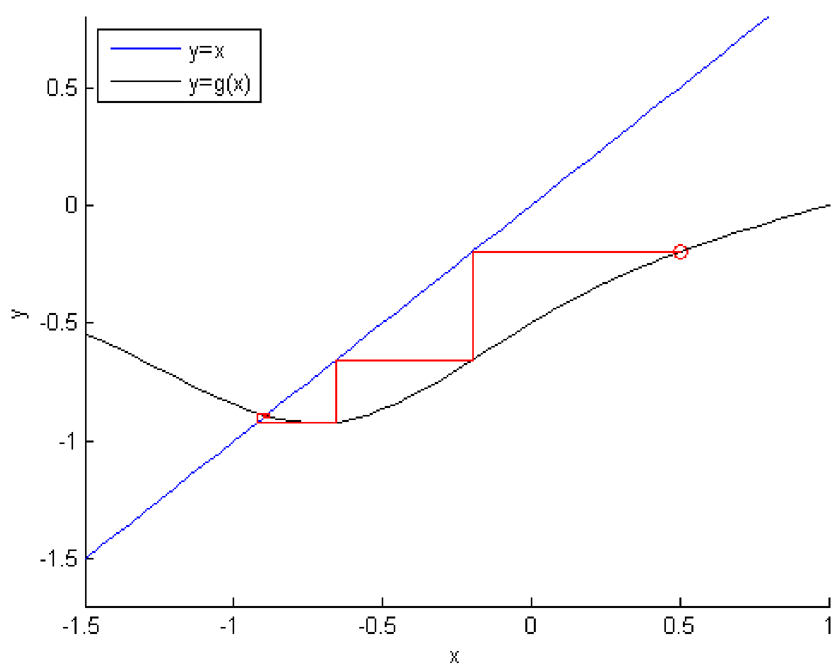
Prob 1 Solution, Example 2

(This version is inspired by the work of Eduardo Perez and Lindsay Fleming. Thanks to both.)

This example demonstrates another way to visualize the process of convergence. Here, $g_1(x)=(x e^x+1)^{1/4}$ (initial guess = 1.7) and $g_2(x)=(x-1)/(e^x-x^3+1)$ (initial guess = 0.5) are chosen for the positive and negative solutions. The following figures (top for the positive solution, bottom for the negative solution) show the two curves of y = x and y = g(x) whose intersection is the solution. The red segments indicate the progress of the iterative process (starting from the initial guess as the red circle), in the same fashion as Fig. 3-18 in the textbook. The matlab code for the negative solution (bottom) is in next page.



Solution is 1.9768 after 19 iterations



Solution is −0.8924 after 9 iterations

(Matlab code for Prob 1 Example 2)

```
x = [-1.5:0.05:1];
g = inline('(x-1)/(exp(x)-x^3+1)','x');
for k = 1:51
    gplot(k) = g(x(k));
end
hold on
plot(x,x,'b-',x,gplot,'k-','LineWidth',1.3)
xlabel('x');ylabel('y');
legend('y=x','y=g(x)','Location','NorthWest')
axis([-1.5 1 -1.7 0.8]);
x0 = 0.5;
x1 = g(x0);
plot([x0 x1],[x1 x1],'r-','LineWidth',1.3)
plot(x0,x1,'o','MarkerEdgeColor','r')
ktot = 0
while (abs(x1-x0) > 0.0001)
    ktot = ktot+1
    x0 = x1;
    x1 = g(x0);
    plot([x0 x0],[x0 x1],'r-','LineWidth',1.3)
    plot([x0 x1],[x1 x1],'r-','LineWidth',1.3)
end
hold off
fprintf('Solution is %9.4f after %4u iterations \r',x1,ktot)
```

Prob 1 Further notes

Most people got the positive solution right. What's causing trouble is the negative solution. The following table summarizes the different forms of g(x)'s you have chosen that work for the *negative* solution. The number in the second column indicates the number of students who chose the particular form of g(x). There would be some deduction if a correct pick of g(x) is accompanied by a wrong initial guess, of if an initial guess is not specified. An incorrect (improbable) pick of g(x) that "leads to the right solution" would receive no credit.

g(x)	# of students	Remarks
$-(x e^x+1)^{1/4}$	27	Note that $(x e^x+1)^{1/4}$ and $-(x e^x+1)^{1/4}$ are two distinctive choices of g(x). The former works for the positive solution while the latter works for the negative solution. Some students claimed that the former works for both (by arbitrarily inserting a negative sign in the iterative process). This would lead to a deduction.
$x+A(e^x-x^3+1/x)$ or $x+A(x e^x-x^4+1)$, where *A* is a small negative number	16	See instructor's solution which uses $A = -0.1$.
$\dfrac{x-1}{e^x-x^3+1}$	7	An interesting choice. See if you can figure out how to obtain this form of g(x).
$x\cos(x e^x-x^4+1)$	4	It works, but convergence is painfully slow in this case. For example, using x = −1 as the initial guess, it takes over 2000 iterations to reach within 0.0001 of the true solution.
$x\left(\dfrac{x e^x+1}{x^4}\right)^{1/100}$	2	Convergence is somewhat slow in this case; Needs about 200 iterations to reach a reasonably accurate result starting from the initial guess of x = −1.
$(e^x+1/x)^{1/3}$	2	While this is a valid pick, it could be somewhat tricky to implement the calculation in matlab (or using a regular calculator) due to the 1/3 power and the fact that the number in the parentheses is negative. You might need to use a special function ("nthroot" in matlab) to take the root or matlab would return a complex number (which is a valid root but not what we want).
$(x^2 e^x+x)^{1/5}$	2	See above remark.
$\left(\dfrac{-1}{x^{-2}e^x-x}\right)^{1/3}$	1	See above remark.
$x^5(x e^x-x^4+1+1/x^4)$	1	
Other choices that did not lead to convergence to the negative solution	31	

Prob 2 Solution (Thanks to Nicholas Ramseyer)

(a) The result of Gauss elimination is (x1,x2,x3) = (1 , 2 , 0.5)

Matlab code for (b)-(d):

```
%                    JACOBI ITERATION METHOD
K = 0;
X1 = 0;
X2 = 0;
X3 = 0;

fprintf('Below are the solutions for Jacobi Iteration Mehtod')
disp('  ')
disp('  ')
for K = 1:15
    N = K;
    Y1 = X1;
    Y2 = X2;
    Y3 = X3;
    X1 = (9 - 1*Y2 - 2*Y3)/6;
    X2 = (13 - 2*Y1 - 2*Y3)/5;
    X3 = (6 - 1*Y1 - 1*Y2)/6;

    R1 = (6*(X1)+ 1*(X2)+ 2*(X3)) - 9;
    R2 = (2*(X1)+ 5*(X2)+ 2*(X3)) - 13;
    R3 = (1*(X1)+ 1*(X2)+ 6*(X3)) - 6;

    E = ((R1^2 + R2^2 + R3^2)^(1/2));
    E1(K) = E;
    fprintf('%g E = %f  X1 = %5f , X2 = %5f , X3 = %5f \r',N+1,E,X1,X2,X3)
end
```

Jacobi

```
disp('  ')
disp('  ')
fprintf('Below are the solutions for Gauss Seidel Method')
disp('  ')
disp('  ')
%                    GAUESS SEIDEL ITERATION
k = 0;
x1 = 0;
x2 = 0;
x3 = 0;

for k = 1:15
    I = k;
    x1 = (9 - 1*x2 - 2*x3)/6;
    x2 = (13 - 2*x1 - 2*x3)/5;
    x3 = (6 - 1*x1 - 1*x2)/6;
```

Gauss Seidel (

(Matlab code continues to next page)

(Prob 2, Matlab code continued)

```
    R1 = (6*(x1)+ 1*(x2)+ 2*(x3)) - 9;
    R2 = (2*(X1)+ 5*(X2)+ 2*(x3)) - 13;
    R3 = (1*(X1)+ 1*(x2)+ 6*(x3)) - 6;

    e = ((R1^2 + R2^2 + R3^2)^(1/2));
    E2(k) = e;
    fprintf('%g E = %f  x1 = %5f , x2 = %5f , x3 = %5f \r',I+1,e,x1,x2,x3)
end

n = 1:15;

plot(n,E1,'Color',[1 0 0])
hold on
plot(n,E2)
xlabel('Iterations','FontSize',12)
ylabel('Values of percent error','FontSize',12)
title('Jacobi vs. Gauss Seidel Plot','FontSize',14)
legend('Jacobi','Gauss Seidel')
```

(Prob 2 continued)

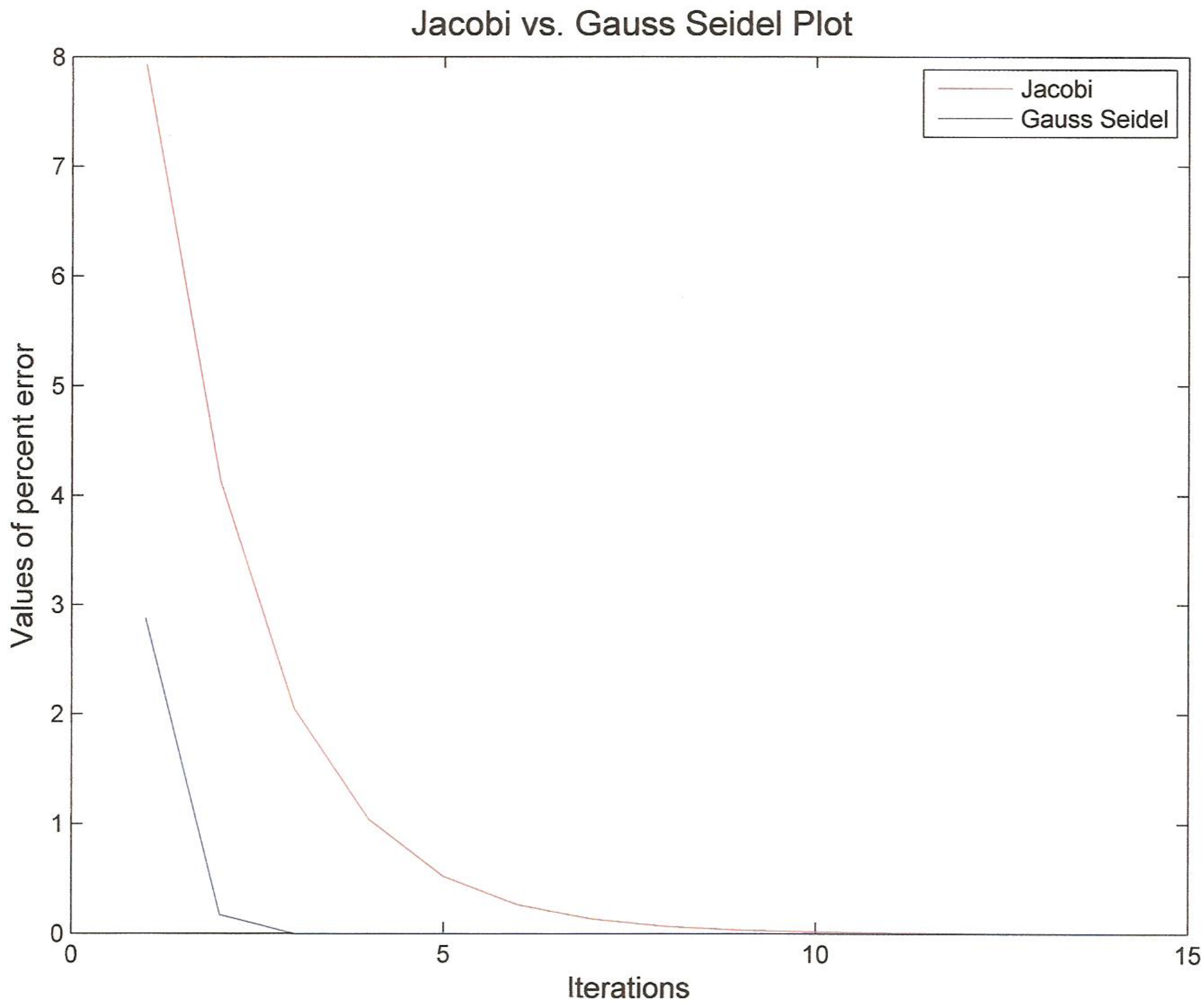
```
Below are the solutions for Jacobi Iteration Mehtod

2 E = 7.935364  X1 = 1.500000 , X2 = 2.600000 , X3 = 1.000000
3 E = 4.139109  X1 = 0.733333 , X2 = 1.600000 , X3 = 0.316667
4 E = 2.052831  X1 = 1.127778 , X2 = 2.180000 , X3 = 0.611111
5 E = 1.045047  X1 = 0.932963 , X2 = 1.904444 , X3 = 0.448704
6 E = 0.525566  X1 = 1.033025 , X2 = 2.047333 , X3 = 0.527099
7 E = 0.265806  X1 = 0.983078 , X2 = 1.975951 , X3 = 0.486607
8 E = 0.134106  X1 = 1.008473 , X2 = 2.012126 , X3 = 0.506829
9 E = 0.067726  X1 = 0.995703 , X2 = 1.993880 , X3 = 0.496567
10 E = 0.034191  X1 = 1.002164 , X2 = 2.003092 , X3 = 0.501736
11 E = 0.017263  X1 = 0.998906 , X2 = 1.998440 , X3 = 0.499124
12 E = 0.008716  X1 = 1.000552 , X2 = 2.000788 , X3 = 0.500442
13 E = 0.004400  X1 = 0.999721 , X2 = 1.999602 , X3 = 0.499777
14 E = 0.002222  X1 = 1.000141 , X2 = 2.000201 , X3 = 0.500113
15 E = 0.001122  X1 = 0.999929 , X2 = 1.999899 , X3 = 0.499943
16 E = 0.000566  X1 = 1.000036 , X2 = 2.000051 , X3 = 0.500029
17 E = 0.000286  X1 = 0.999982 , X2 = 1.999974 , X3 = 0.499985
18 E = 0.000144  X1 = 1.000009 , X2 = 2.000013 , X3 = 0.500007
19 E = 0.000073  X1 = 0.999995 , X2 = 1.999993 , X3 = 0.499996
20 E = 0.000037  X1 = 1.000002 , X2 = 2.000003 , X3 = 0.500002
21 E = 0.000019  X1 = 0.999999 , X2 = 1.999998 , X3 = 0.499999
22 E = 0.000009  X1 = 1.000001 , X2 = 2.000001 , X3 = 0.500000
23 E = 0.000005  X1 = 1.000000 , X2 = 2.000000 , X3 = 0.500000
24 E = 0.000002  X1 = 1.000000 , X2 = 2.000000 , X3 = 0.500000
25 E = 0.000001  X1 = 1.000000 , X2 = 2.000000 , X3 = 0.500000
26 E = 0.000001  X1 = 1.000000 , X2 = 2.000000 , X3 = 0.500000
27 E = 0.000000  X1 = 1.000000 , X2 = 2.000000 , X3 = 0.500000
28 E = 0.000000  X1 = 1.000000 , X2 = 2.000000 , X3 = 0.500000
```

```
Below are the solutions for Gauss Seidel Method

2 E = 2.881936  x1 = 1.500000 , x2 = 2.000000 , x3 = 0.416667
3 E = 0.175242  x1 = 1.027778 , x2 = 2.022222 , x3 = 0.491667
4 E = 0.003071  x1 = 0.999074 , x2 = 2.003704 , x3 = 0.499537
5 E = 0.002421  x1 = 0.999537 , x2 = 2.000370 , x3 = 0.500015
6 E = 0.000372  x1 = 0.999933 , x2 = 2.000021 , x3 = 0.500008
7 E = 0.000035  x1 = 0.999994 , x2 = 1.999999 , x3 = 0.500001
8 E = 0.000002  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
9 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
10 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
11 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
12 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
13 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
14 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
15 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
16 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
17 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
18 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
19 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
20 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
21 E = 0.000000  x1 = 1.000000 , x2 = 2.000000 , x3 = 0.500000
>>
```

(Prob 2 continued)



(Prob 2 continued - Part (e))

```
clear all
close all
clc

A = [6,1,2;2,5,2;1,1,6];

% A(Row,Column)

B =  ((A(1,1)^2) + (A(1,2)^2) + (A(1,3)^2)...
    + (A(2,1)^2) + (A(2,2)^2) + (A(2,3)^2)...
    + (A(3,1)^2) + (A(3,2)^2) + (A(3,3)^2))^(1/2);

fprintf('Value of Norm for A is %g',B)
disp('  ')
disp('  ')
C = inv(A);

D =  ((C(1,1)^2) + (C(1,2)^2) + (C(1,3)^2)...
    + (C(2,1)^2) + (C(2,2)^2) + (C(2,3)^2)...
    + (C(3,1)^2) + (C(3,2)^2) + (C(3,3)^2))^(1/2);

fprintf('norm of inverse = %g',D)
disp('  ')
disp('  ')
E = B*D;

fprintf('Condition number is equal to %g',E)
disp('  ')
```

The condition number is 3.8147. The system is not ill-conditioned.