Homework 7

Yash Ganatra

April 18, 2016

Problem 0: Homework checklist

Please identify anyone, whether or not they are in the class, with whom you discussed your homework. This problem is worth 1 point, but on a multiplicative scale.

- Zhehuan Xu, Nate Majid Kesto, Emily Zimovan,Woi Sok Oh
- Coding Language: Matlab

Problem 1: Gautschi Exercise 4.19

Consider the equation

$$f(x) = \tan x + \lambda x = 0, 0 < \lambda < 1..$$

Figure 1 shows the plot of $tan\, x, -\lambda\, x,\ f\prime(x)$ for $\lambda = 0, 0.5, 1$. It can be seen that there is only one root for the limiting cases ($\lambda \to 0, \lambda \to 1$). Hence this function has only 1 root in the given interval. Newtons method is given by Equation 1

$$x_{k+1} = x_k - \frac{f(x_k)}{f\prime(x_k)} \tag{1}$$

For an initial value, $x_0 = \pi$, it can be seen from Figure 1 that $f\prime(\pi) = 1 + \lambda \neq 0$. Figure 2 shows the plot of function $f(x)$ for $\lambda = 0, 0.5, 1$, it can be seen that Newtons method will converge.

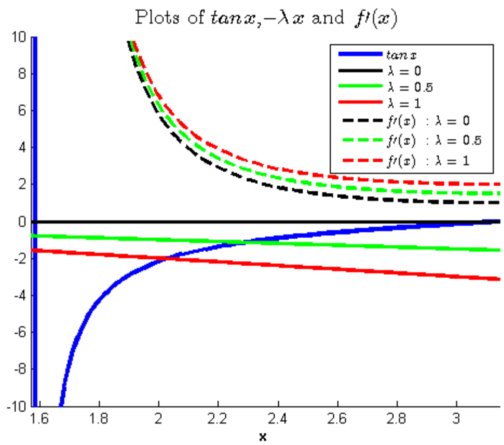

Figure 1: Plot of $tan\, x, -\lambda\, x,\ f\prime(x)$ for $\lambda = 0, 0.5, 1$

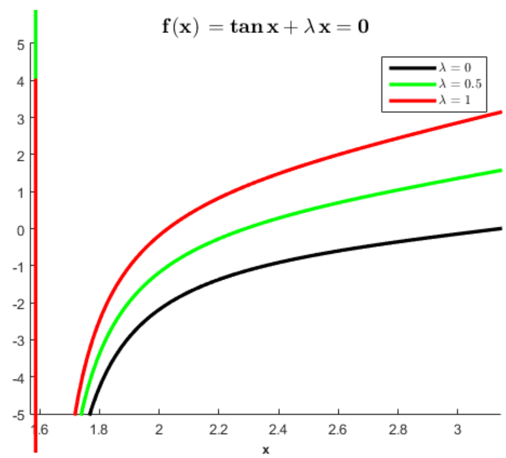

Figure 2: Plot of $\tan x + \lambda x$ for $\lambda = 0, 0.5, 1$

Problem 2: Gautschi Exercise 4.25

Consider Newton's method

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_n}\right), \qquad a > 0,$$

for computing $\alpha = \sqrt{a}$. Let $d_k = x_{k+1} - x_k$. Rearranging given equation to form a quadratic equation in x_k and using $d_k = x_{k+1} - x_k$ to eliminate x_{k+1}, as shown in Equation 2

$$
\begin{aligned}
& x_k^2 - 2\,x_{k+1}\,x_k + a = 0 \\
& -x_k^2 - 2\,d_k\,x_k + a = 0 \\
& x_k = -d_k \pm \sqrt{d_k^2 + a} \\
& \text{Take the - sign since } x_k, d_k > 0 \\
& \text{Multiply and divide by the conjugate} \\
& x_k = \frac{a}{d_k + \sqrt{d_k^2 + a}}
\end{aligned}
\tag{2}
$$

To find $d_k = f(d_{k-1})$, use $d_{k-1} = x_k - xk - 1$, as shown in Equation **??**

$$
\begin{aligned}
& x_k = -d_k + \sqrt{d_k^2 + a} \\
& \text{Substitute this in } d_{k-1} = x_k - xk - 1 \text{ to get} \\
& \sqrt{d_{k-1}^2 + a} = -dk + \sqrt{d_k^2 + a} \\
& \text{Squaring both sides} \\
& d_{k-1}^2 = d_k^2 + 2\,d_k\,\sqrt{d_k^2 + a} \\
& \text{Substitute } x_k + d_k = \sqrt{d_k^2 + a} \\
& 2d_k^2 - 2\,(x_k + d_k)d_k - d_{k-1}^2 = 0 \\
& |d_k| = \frac{d_{k-1}^2}{2\,x_k}
\end{aligned}
\tag{3}
$$

Comparing the denominator of Equation 3, we need to prove that $x_k = \sqrt{d_{k-1}^2 + a}$.

$$
\begin{aligned}
& \text{To prove that} \\
& -d_k + \sqrt{d_k^2 + a} = \sqrt{d_{k-1}^2 + a} \\
& 2d_k^2 - 2\,(x_k + d_k)d_k - d_{k-1}^2 = 0 \\
& \textbf{which is already proved in Equation 3}
\end{aligned}
\tag{4}
$$

Comments

- Since Newtons method is a fixed point method, for convergence to be achieved d_k should decrease with iterations according to contraction mapping theorem.
- For $a = 0$, $d_k = \frac{d_{k-1}}{2}$ which is the bisection method. Hence, the convergence is linear.

Problem 3: Nonlinear systems

1. $x_1^2 + x_1 x_2^3 = 9$, $3x_1^2 x_2 - x_2^3 = 4$

2. $x_1 + x_2 - 2x_1x_2 = 0$, $x_1^2 + x_2^2 - 2x_1 + 2x_2 = -1$

For multivariate functions, Newtons method can be written as (in vector notation) shown in Equation 5

$$\mathbf{x_{n+1}} = \mathbf{x_n} - J^{-1}\mathbf{f}(\mathbf{x_n}) \tag{5}$$

The Jacobi Matrix is defined as

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix}$$

For the first system of equations, Joacobi matrix is

$$\begin{pmatrix} 2x_1 + x_2^3 & 3x_1\,x_2^2 \\ 6x_1\,x_2 & 3x_1^2 - 3x_2^2 \end{pmatrix}$$

$\implies J_1(0,0) = [\mathbf{0}]$. The matrix is singular and hence **Newtons method wont converge**

For the second system of equations, Joacobi matrix is

$$\begin{pmatrix} 1 - 2\,x_2 & 1 - 2\,x_1 \\ 2\,x_1 - 2 & 2\,x_2 + 2 \end{pmatrix}$$

$\implies J_2(0,0) =$

$$\begin{pmatrix} 1 & 1 \\ -2 & 2 \end{pmatrix}$$

Newtons method converges to give $x_1 = 0.2158, x_2 = -0.3795$

Problem 4: Fixed point methods

Roots of given systems of equations calculated in Wolfram Alpha are shown in Table 1

Table 1: Roots of given system of equations caculated in *Wolfram Alpha*

$F(x)$	Roots $[x, y]$			
1	-3.00162,0.148108	-0.901266,-2.08659	1.33636,1.75424	2.99837,0.148431
2	0.215761.-0.379541	0.39098,-1.79315		

For first modification, $\phi = x - \alpha F(x)$, the fixed point iteration ($x_{n+1} = \phi(x_n)$) is done. To validate the modification, contraction mapping principle is used as shown in Equation 6

$$\gamma = \frac{||\phi(x) - \phi(x^\star)||}{||x - x^\star||}, 0 \le \gamma \le 1 \tag{6}$$

For $\alpha = 1e-2$, the solutions for both systems converges

1. $x_1 = 2.98, x_2 = 0.15$
2. $x_1 = 0.216, x_2 = -0.38$

The corresponding plots of $\phi\, and\, \gamma$ are shown in Figures 3-6

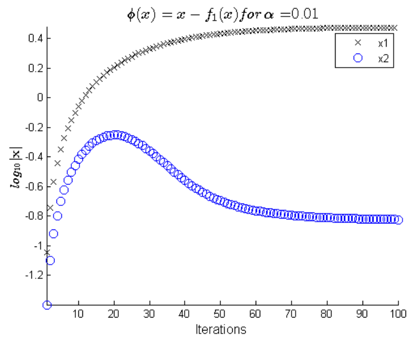

Figure 3: $\phi_1(x)$

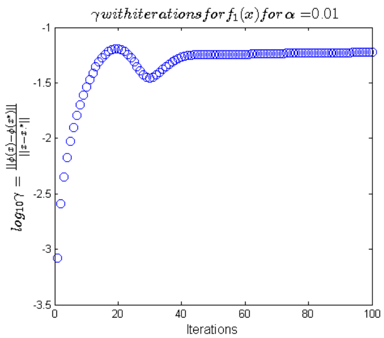

Figure 4: γ_1

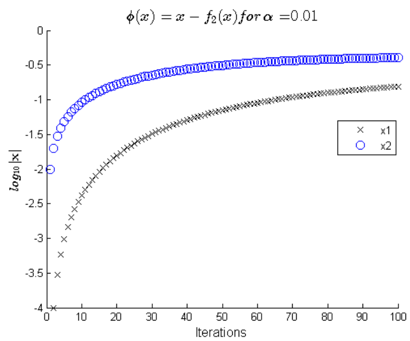

Figure 5: $\phi_2(x)$

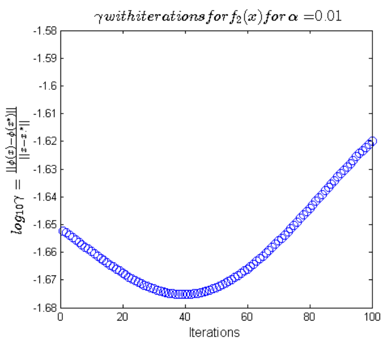

Figure 6: γ_2

Appendix

Listing 1: Rates of convergence for given series

```
clc;close all;clear all;
a = pi/2;
b = pi;
n = 100;
x = linspace(a,b,n);
my_lambda = [0 0.5 1];
%%
C = {'k','g','r','b','y'};
c1 =0;c2 =0;c3=0;
%%
figure(1)
hold on
plot(x,tan(x),'b-','LineWidth',3)
for lambda = my_lambda
    c1 =c1+1;
    plot(x,-lambda.*x,'color',C{c1},'LineWidth',2.5)
    axis([-inf pi -10 10])
    legendInfo1{c1} = ['$\lambda$ = ' num2str(lambda)];
end
% Derivative
for lambda = my_lambda
    c2 =c2+1;
    plot(x,(1./(cos(x).^2))+lambda,'color',C{c2},'
        LineStyle','--',...
         'LineWidth',2.5)
    axis([-inf pi -10 10])
    legendInfo2{c2} = ['$f\prime(x)\,\,:\lambda$ = '
        num2str(lambda)];
end
l = ['$tan\,x$' legendInfo1 legendInfo2];
legend(l,'interpreter','latex')
title('Plots of $tan\,x$,$-\lambda\,x$ and $f\prime(x)$'
    ,...
    'FontSize',14,'interpreter','latex')
hold off
xlabel('x','FontSize',9,'FontWeight','bold')
%%
figure(2)
hold on
for lambda = my_lambda
    c3 =c3+1;
    plot(x,tan(x)+lambda.*x,'color',C{c3},'LineWidth'
```

```
        ,2.5)
    axis([-inf pi -5 5])
    legendInfo3{c3} = ['$\lambda$ = ' num2str(lambda)];
%          hline = refline([0 0]);
% set(hline,'LineStyle','--')
end
xlabel('x','FontSize',9,'FontWeight','bold')
legend(legendInfo3,'interpreter','latex')
title('$ \mathbf{f(x)\,= tan\,x+\lambda\,x = 0}$',...
    'FontSize',14,'interpreter','latex')
%%
dat = tan(x)+(1e-5).*x;
der_end = dat(end)-dat(end-1)./(x(end)-x(end-1));
%%%% P3
clc;close all;clear all;
syms x1;
syms x2;
g1_ana = x1 + x2 - 2.*x1.*x2;
g2_ana = x1.^3 +x2.^2 -2.*x1 +2.*x2+1;
% f1 = @(x1,x2) x1 + x2 - 2.*x1.*x2;
% f2 = @(x1,x2) x1.^2 +x2.^2 -2.*x1 +2.*x2+1;
g_der_11_ana = diff(g1_ana,x1);g_der_12_ana = diff(g1_ana
    ,x2);
g_der_21_ana = diff(g2_ana,x1);g_der_22_ana = diff(g2_ana
    ,x2);
%% Derivs depend on functions (some cases only are funcs
    of 1 variable) -- check manually
g_der_11 = matlabFunction(g_der_11_ana);
g_der_12 = matlabFunction(g_der_12_ana);
g_der_21 = matlabFunction(g_der_21_ana);
g_der_22 = matlabFunction(g_der_22_ana);
g1 = matlabFunction(g1_ana);
g2 = matlabFunction(g2_ana);
%%
x1_guess =0;
x2_guess = 0;
maxiter = 6;
error = NaN.*ones([1,maxiter]);
count =0;
error_check = 1;
tol = 1e-8;
x = [x1_guess;x2_guess];
while(error_check > tol)
count = count+ 1;
J = [g_der_11(x(2)),g_der_12(x(1));g_der_21(x(1)),
    g_der_22(x(2))];
```

```
x_new = [x(1);x(2)] - inv(J)*[g1(x(1),x(2));g2(x(1),x(2))
    ];
error(count) = max(abs(x_new-x));
x = x_new;
error_check = vpa(error(count));
if(count>maxiter)
    disp('Max Iter Reached');
    break;
end
end
count
%% Plug-in values
g1_check = g1(x(1),x(2))
g2_check = g2(x(1),x(2))
%%
figure(1)
plot(1:count,log10(error(1:count)),'kx')
xlabel('Iterations','FontSize',12)%'FontWeight','bold'
ylabel('$ -log_{10}|\mbox{Error}|$',...
    'FontSize',12,'interpreter','latex')
%% P4
clc;close all;clear all;
syms x1;
syms x2;
g1_ana = x1.^2 + x1.*x2.^3 - 9;
g2_ana = 3.*x1.^2.*x2 -x2.^3 -4;
% g1_ana = x1 + x2 - 2.*x1.*x2;
% g2_ana = x1.^2 +x2.^2 -2.*x1 +2.*x2+1;
g1 = matlabFunction(g1_ana);
g2 = matlabFunction(g2_ana);
%%
x1_guess =0;
x2_guess = 0;
maxiter = 5;
error = NaN.*ones([1,maxiter]);
count =0;
error_check = 1;
tol = 1e-10;
x = [x1_guess;x2_guess];
%%
alpha = 1;
gamma_val = 0.5;
%%
while(error_check > tol)
    count = count+ 1;
    if(count>maxiter)
```

```
        disp('Max Iter Reached');
        count = count-1;
        fprintf('%d\n',x)
        break;
    end
    x_old = [x(1);x(2)];
    func_mat = [g1(x(1),x(2));g2(x(1),x(2))];
    disp('func')
    func_mat'
    %x_new = x_old - alpha*func_mat; %Fixed Point
        iteration
    x_new = (x_old - gamma_val*x_old)./(1-gamma_val);
    func_diff = [g1(x_new(1),x_new(2));g2(x_new(1),x_new
        (2))] - func_mat;
    gamma(count) = norm(func_diff,2)./norm(func_mat,2);
    disp('x')
    x_new'
    error(count) = max(abs(x_new-x));
    x= x_new;
    error_check = vpa(error(count));
    x1_fp(count) = x(1);
    x2_fp(count) = x(2);
end
%%
% figure(1)
% hold on
% plot(1:(count),log10(x1_fp),'kx',1:(count),log10(x2_fp)
    ,...
%     'bo','MarkerSize',8)
% legend('x1','x2')
% xlabel('Iterations','FontSize',12)%'FontWeight','bold'
% ylabel('$ log_{10}|\mbox{\textbf{x}}|$',...
%     'FontSize',12,'interpreter','latex')
% title_name1 = (['$ \phi(x) = x - f_{1}(x) for\,\alpha =
   $' num2str(gamma_val)]);
% title(title_name1,'FontSize',14,'interpreter','latex')
% axis tight
% %%
% figure(2)
% plot(1:(count),log10(gamma),'bo','MarkerSize',8)
% xlabel('Iterations','FontSize',12)%'FontWeight','bold'
% ylabel('$ log_{10}\gamma = \frac{||\phi(x)-\phi(x^\star
   )||}{||x - x.^\star||}$',...
%     'FontSize',14,'interpreter','latex')
% title_name2 = (['$\gamma with iterations for f_1(x) for
   \,\alpha =$' num2str(gamma_val)]);
```

```
% title(title_name2,...
%     'FontSize',14,'interpreter','latex')
```