
```

%{
    ENG 180 Project 4
    File:          ENG180_PJ3_main_Hieu_Bui.m
    Author:        Hieu Bui
    Date:          10/19/2022
    Description:    Finding answers through iterative approximations

%}

clc, close all;
clear;

k = 10;
jacob = 'Jacobi';
gauss = 'GSeidel';
sor = 'SOR';
eqn = @(x,n) cos(pi.*x./(n+1));

%Plotting problem 1
[clk1,iterations1,res1,cor1] = iterationEfficiency(10,1.2,jacob,eqn);
[clk2,iterations2,res2,cor2] = iterationEfficiency(10,1.2,gauss,eqn);
[clk3,iterations3,res3,cor3] = iterationEfficiency(10,1.2,sor,eqn);
figure('Name','Problem 1 ')
subplot(2,1,1)
loglog(iterations1,res1)
hold on
loglog(iterations2,res2)
hold on
loglog(iterations3,res3)
hold on
title('Residuals vs Iterations')
legend1=sprintf('Jacobi      %fs ',clk1);
legend2=sprintf('Gauss S.   %fs',clk2);
legend3=sprintf('SOR        %fs',clk3);
xlabel('Iterations')
ylabel('Residuals')
legend(legend1,legend2,legend3,'Location','southwest')
subplot(2,1,2)
loglog(iterations1,cor1)
hold on
loglog(iterations2,cor2)
hold on
loglog(iterations3,cor3)
hold on
title('Corrections vs Iterations')
legend1=sprintf('Jacobi      %fs ',clk1);
legend2=sprintf('Gauss S.   %fs',clk2);
legend3=sprintf('SOR        %fs',clk3);
xlabel('Iterations')
ylabel('Corrections')
legend(legend1,legend2,legend3,'Location','southwest')

```

```

%Plotting problem 2
figure('Name','Fixed Point Iterations','NumberTitle','off')
subplot(3,1,1)
[r11,r21]=p2f('a',k,.5,3,.7);
title(sprintf('Propellant Burnt when n = .5, r1 = %.2f, r2 = %.2f', r11, r21))
xlabel('x')
ylabel('y')
hold on;
subplot(3,1,2)
[r12,r22]=p2f('a',k,2.1,1,.6);
title(sprintf('Propellant Burnt when n = 2.1, r1 = %.2f, r2 = %.2f', r12,r22))
xlabel('x')
ylabel('y')
hold on;
subplot(3,1,3)
[r13,r23]=p2f('b',k,1,2,.8);
title(sprintf('Semenov Equation, r1 = %.2f, r2 = %.2f', r13, r23))
xlabel('x')
ylabel('y')
hold on;

figure('Name','Newton Method')
subplot(2,1,1)
[newtonR1]=p2n('a',k,2.1,3,.49);
[newtonR2]=p2n('a',k,2.1,3,.2);
title(sprintf('Propellant Burnt when n=2.1 using Newton; root 1 = %.2f; root 2
= %.2f', newtonR1,newtonR2))
xlabel('x')
ylabel('y')
hold on;
subplot(2,1,2)
[newtonR3]=p2n('b',k,1,3,.29);
title(sprintf('Semenov using Newton; root = %.2f', newtonR2))
xlabel('x')
ylabel('y')
hold on;

%Function for problem 1
function [clk,iter,resValues, corValues] =
    iterationEfficiency(n,guess,method,eqn)
    %create tridiagonal matrix
    a = ones(n-1,1);
    b = -2.*ones(n,1);
    c = ones(n-1,1);
    matrixA = diag(a,-1)+diag(b)+diag(c,1);
    iterations = 0;
    resid = 1;
    tolerance = 1e-15;
    f = eqn(1:n,n);
    xNew = guess.*ones(n,1);

    switch method
        case 'Jacobi'

```

```

        start = tic;
        while resid > tolerance
            xOld = xNew;
            for i=1:n
                xNew(i) = xOld(i)+1/matrixA(i,i)*(f(i)-
matrixA(i,:)*xOld(:));
            end
            resid = max(abs(matrixA*xNew-f'));
            cor = max(abs(xNew-xOld));
            iterations = iterations + 1;
            resValues(iterations) = resid;
            corValues(iterations) = cor;
        end
        clk = toc(start);
        iter = linspace(1,iterations,iterations);
    case 'GSeidel'
        start = tic;
        while resid > tolerance
            xOld = xNew;
            for i = 1:n
                xNew(i) = xOld(i)+(f(i)-matrixA(i,1:i-1)*xNew(1:i-1)-
matrixA(i,i:n)*xOld(i:n))/matrixA(i,i);
            end
            resid = max(abs(matrixA*xNew-f'));
            cor = max(abs(xNew-xOld));
            iterations = iterations + 1;
            resValues(iterations) = resid;
            corValues(iterations) = cor;
        end
        clk = toc(start);
        iter = linspace(1,iterations,iterations);
    case 'SOR'
        omega = 1.7;
        start = tic;
        while resid > tolerance
            xOld = xNew;
            for i = 1:n
                xGS = xOld(i)+(f(i)-matrixA(i,1:i-1)*xNew(1:i-1)-
matrixA(i,i:n)*xOld(i:n))/matrixA(i,i);
                xNew(i) = omega*(xGS-xOld(i))+xOld(i);
            end
            resid = max(abs(matrixA*xNew-f'));
            cor = max(abs(xNew-xOld));
            iterations = iterations + 1;
            resValues(iterations) = resid;
            corValues(iterations) = cor;
        end
        clk = toc(start);
        iter = linspace(1,iterations,iterations);
    end
end

% Fixed point iterations problem 2A
% e = which equation ['a' = 2a; 'b' = 2b]

```

```

% k is number of iterations to do
% n is the power for problem 2a
% s is number of points for exact graph
% g is initial guess
% r is the approximated root
function [r1,r2] = p2f(e,k,n,s,g)
    c1 = 1.6;
    c2 = 1.1;
    a = -.13;
    b = .44;
    c = .25;
    alpha = 2;
    r = zeros(k,1);
    x = ones(k,1);
    x(1) = g;
    if strcmp(e,'a')
        %=====root 1=====
        fun1 = @(x)(c1/c2).*x.^n;
        fun2 = @(x)(c2/c1).*x.^(1/n);
        for i = 1:k-1
            r(i) = fun1(x(i));%propellant burning root equation 1
            x(i+1) = r(i);
        end
        r(k) = fun1(x(k));
        y = linspace(0,s,100);
        rexact = fun1(y);

        plot(y,y,y,rexact,'--')
        hold on;
        xline(g);
        plot(x(k),r(k),'^','MarkerFaceColor','red')
        hold on;
        r1 = r(k);
        %=====root 2=====
        for i = 1:k-1
            r(i) = fun2(x(i)); %propellant burning root equation 2
            x(i+1) = r(i);
        end
        r(k) = fun2(x(k));
        y2 = linspace(0,s,100);
        rexact2 = fun2(y);

        plot(y2,rexact2,'--')
        hold on;
        plot(x(k),r(k),'^','MarkerFaceColor','red')
        hold on;
        r2 = r(k);
    elseif strcmp(e,'b')
        %semenov root equation 1
        semenov1 = @(x) (c.*exp(1).^(-alpha.*x)-a)./b;
        for i = 1:k
            r(i) = semenov1(x(i));
            x(i+1) = r(i);
        end
    end
end

```

```

    r(k) = semenov1(x(k));
    y = linspace(0,s,20);
    rexact = semenov1(y);

    plot(y,y,y,rexact,'--')
    xline(g);
    hold on;
    plot(x(k),r(k),'^','MarkerFaceColor','red')
    hold on;
    r1 = r(k);

    %semenov root equation 2
    semenov2 = @(x) alpha./(log((alpha+b.*x)./c));
    for i = 1:k
        r(i) = semenov2(x(i));
        x(i+1) = r(i);
    end
    r(k) = semenov2(x(k));
    y = linspace(0,s,20);
    rexact2 = semenov2(y);

    plot(y,y,y,rexact2,'--')
    xline(g);
    hold on;
    plot(x(k),r(k),'^','MarkerFaceColor','red')
    hold on;
    r2 = r(k);
end
end

%Approximate with newton method
function [root] = p2n(e,k,n,s,g)

c1 = 1.6;
c2 = 1.1;
a = -.13;
b = .44;
c = .25;
alpha = 2;

%root function 1 for propellant burnt
if strcmp(e,'a') %if paramater e is 'a' then solve for propellant burn
equation
    x = ones(k,1);
    r = ones(k,1);
    x(1) = g;
    func = @(x) c1.*x.^n - c2.*x;
    funcdot = @(x) c1*n.*x.^(n-1)-c2;
    for i=1:k-1
        r(i) = func(x(i));
        rdot = funcdot(x(i));
        x(i+1) = x(i) - r(i)/rdot;
    end
    r(k) = func(x(k));

```

```

y = linspace(-2,s,100);
rexact = func(y);

plot(y,rexact,'--')
xline(x(1))
hold on; grid on;
plot(x(k),r(k),'bo')
hold on;
root = x(k);
elseif strcmp(e,'b') %if paramater e is 'b' then solve for semenov equation
x = ones(k,1);
r = ones(k,1);
x(1) = g;
func = @(x) c.*exp(1).^(-alpha./x)-a-b.*x;
funcdot = @(x) (c*alpha./x^2).*exp(1).^-alpha./x - b;
%===root equation 1 for semenov===%
for i = 1:k-1
    r(i) = func(x(i));
    rdot = funcdot(x(i));
    x(i+1) = x(i) - r(i)/rdot;
end
r(k) = func(x(k));

y = linspace(-2,s,100);
rexact = func(y);

plot(y,rexact,'--')
xline(g)
hold on; grid on;
plot(x(k),r(k),'bo')
hold on;
root = x(k);
end
end

function [root1,root2] = p2secant(eqn,x1,x2,x11,x22,k,n)
c1 = 1.6;
c2 = 1.1;
a = -.13;
b = .44;
c = .25;
alpha = 2;

if strcmp(eqn, 'a')
    rootFunc1 = @(x) (c1/c2)*x.^n;
    rootFunc2 = @(x) ((c2/c1)*x).^(1/n);
    x(1) = x1;
    x(2) = x2;

    root = ones(k,1);
    root(1) = rootFunc1(x(1));

    for i = 2:k-1

```

```

        root(i) = rootFunc1(x(i));
        rdot = root(i)-root(i-1)/(x(i)-x(i-1));
        x(i+1) = x(i) -root(i)/rdot;
    end
    root(k)=rootFunc1(x(k));
    root1 = root(k);
    y = linspace(0,5,20);
    xline(x(1))
    hold on; grid on;
    plot(y,rootFunc1(y),x(k),root(k), 'bo')

    x(1) = x11;
    x(2) = x22;
    root(1) = rootFunc2(x(1));
    for i = 2:k-1
        root(i) = rootFunc2(x(i));
        rdot = root(i)-root(i-1)/(x(i)-x(i-1));
        x(i+1) = x(i) -root(i)/rdot;
    end
    root(k) = rootFunc2(x(k));
    root2 = root(k);
    y = linspace(0,5,20);
    xline(x(1))
    hold on; grid on;
    plot(y,rootFunc2(y),x(k),root(k), 'bo')

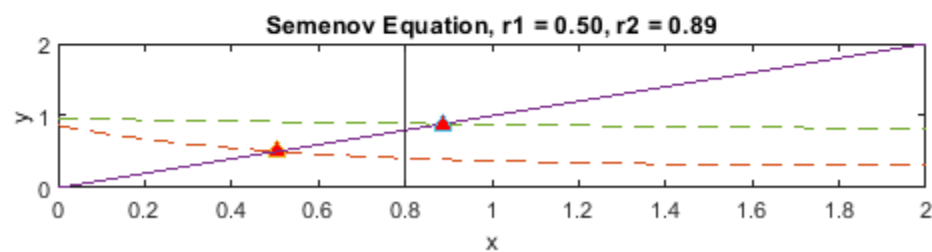
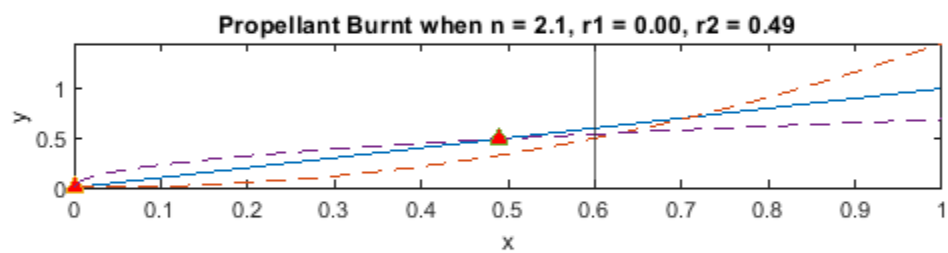
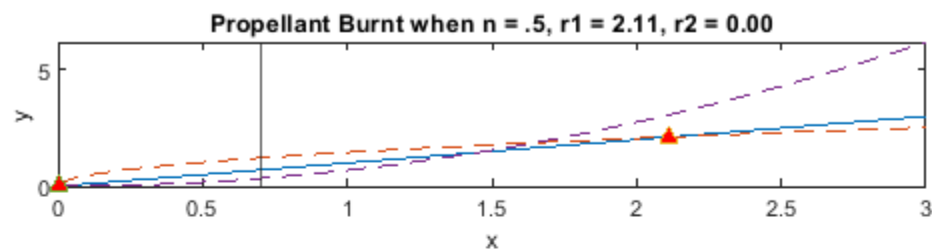
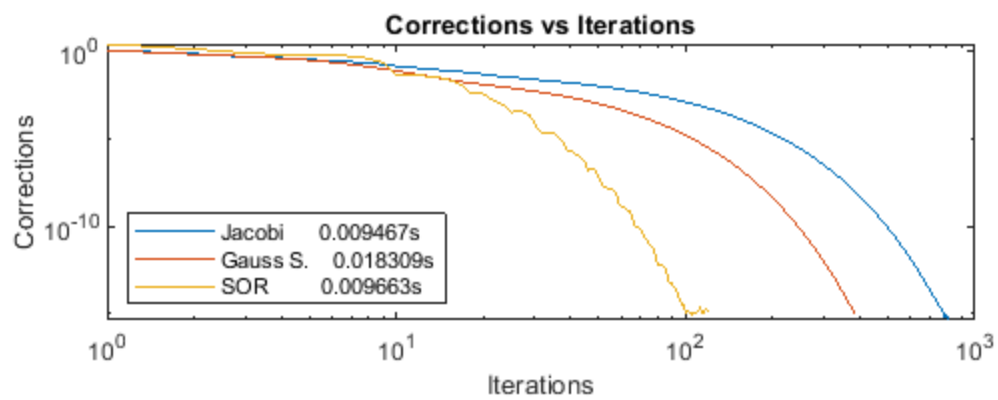
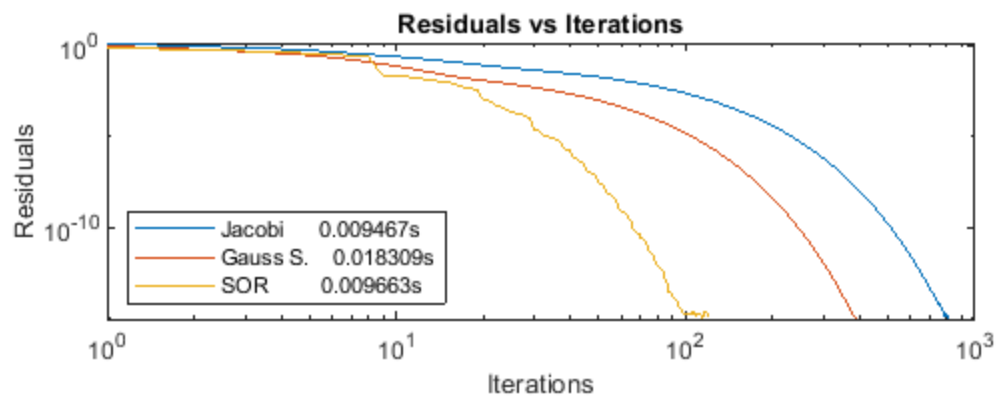
elseif strcmp(eqn, 'b')
end

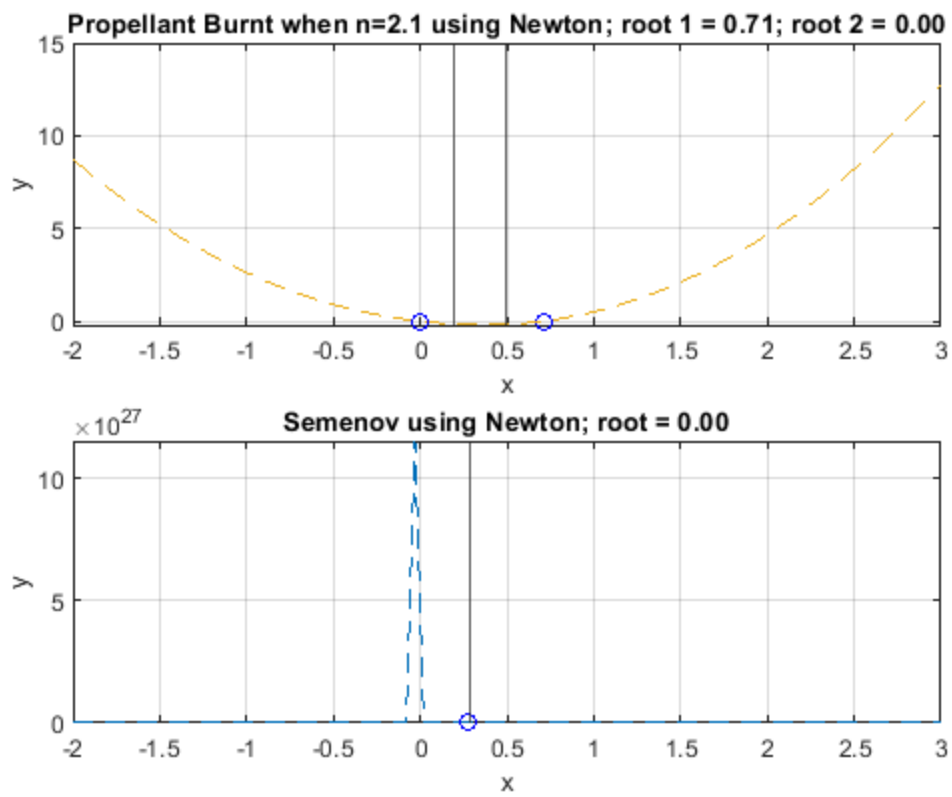
end

```

Warning: Imaginary parts of complex X and/or Y arguments ignored.

Warning: Imaginary parts of complex X and/or Y arguments ignored.





Published with MATLAB® R2022b