

Numerical Analysis Take-Home Final, Fall 2019

Maxwell Greene

Problem #1

```
% Using Newton's Method for multivariate equations:
X = [1;1];Tol=.001;df=20;ak = 1.0;
funcs = @(x,y) [3*x^2-x*y^2+x^2*y-13; x*y-2*x^2+y^2-4];
while df > Tol
    Xn = X-ak*InvJac(X)*funcs(X(1),X(2));
    df =(dot((Xn-X),(Xn-X)))^0.5;
    X = Xn;
end
X
```

```
X =
    2.3502
    2.8780
```

Problem #2

```
%Creating data
x = [0.90,0.95,0.77,0.79,0.25,0.55,0.09,0.85,0.24,0.06];
x2 = [0.81,0.90,0.59,0.62,0.06,0.30,0.01,0.72,0.06,0.00];
b = [4.71,4.87,4.42,4.46,3.88,4.06,3.93,4.60,3.88,3.95];
```

Problem #11

```
%Load Data from Problem
tx = [0:.1111:1];
yx = [-0.0148;0.0714;0.0999;0.1099;0.1342;0.1399;0.1448;0.1417;0.1664;0.1679];

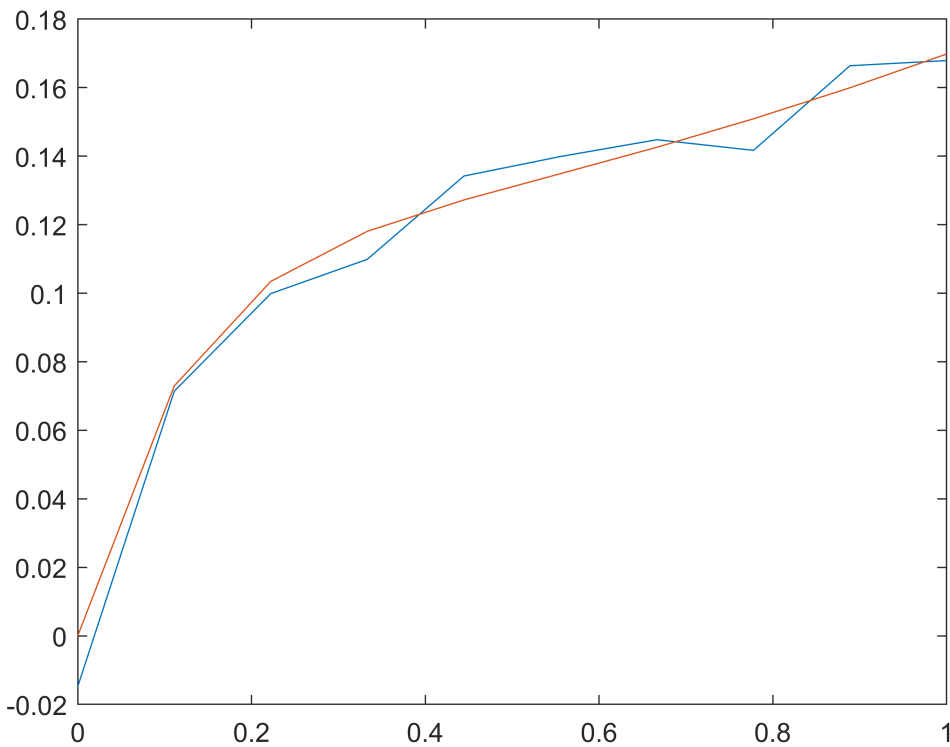
% Solving for optimum values of a and b in equation
% y''+ay'-y = t*exp(-bt), 0<=t<=1

% Optimization routine for values of a and b
% least squares and odefunc.
optFunc = @(input) errorFunction(input(1),input(2),tx,yx);
optVals = optimize(optFunc,[0,0])
```

```
optVals =
    8.2989    0.4248
```

```
a = optVals(1); b = optVals(2);
```

```
[t,y] = ode23s(@(t,y) odefunc(t,y,a,b),tx,[0,1]);
plot(tx,yx,t,y(:,1));
```



Functions used in questions:

Problem #1

```
function InvJ = InvJac(X)
    x = X(1); y = X(2);
    J = [6*x-y^2+2*x*y, -2*x*y+x^2; y-4*x, x+2*y];
    InvJ = inv(J);
end
```

Problem #11

```
function dydt = odefunc(t,z,c,d)
    dydt = [z(2); -c*z(2)+z(1)+t*exp(-d*t)];
end

function [a,b] = optimize(optfunc,params)
    [a,b] = fminsearch(optfunc,[params(1),params(2)]);
```

```
end
```

```
function error = errorFunction(a1,a2,tx,yx)
    [~,y] = ode23s(@(t,y) odefunc(t,y,a1,a2),tx,[0,1]);
    error = sum((y(:,1)-yx).^2);
end
```