Scientific Computing Homework 7

Michael Nelson

Problem 1

Exercise 1. Solve the following nonlinear system using Newton's method in Matlab (use newton.m):

$$0 = x^2 - y - \sin z + 1$$
$$0 = x + 1 + \sin(10y) - y$$
$$0 = (1 - x)z - 2.$$

Print the norm of the residual to check that your solution is in fact a root. Hint: you need to find a suitable value for x, y and z so that the method converges. Submit hw07q1.m and make sure the definitions of your functions f and ∇f are also included (if you create them in separate files).

Solution 1. We create f in a file called hw7fun.m. Its code is given below

```
function y = hw7fun(x)
y(1,1) = x(1)^2 - x(2) - sin(x(3)) + 1;
y(2,1) = x(1) + 1 + sin(10*x(2)) - x(2);
y(3,1) = z - x*z - 2;
end
```

Next we create ∇f in a file called hw7gradfun.m. Its code is given below:

```
function y = hw7gradfun(x)
y(1,1) = 2*x(1);
y(1,2) = -1;
y(1,3) = -cos(x(3));
y(2,1) = 1;
y(2,2) = 10*cos(x(2))-1;
y(2,3) = 0;
y(3,1) = -x(3);
y(3,2) = 0;
y(3,3) = 1 - x(1);
end
```

Now we apply newton.m with an initial guess of $(0, 0, 1)$:

```
[root,numits] = newton(@hw7fun,@hw7gradfun,[0;0;1],1e-8)


root =

        1.20653487050551              2.19978940708985              -9.68359480946174

numits =

               61

norm(hw7fun(root)) %checking solution is a root by computing norm of residual

ans =

        1.88008173473976e-08
```

Problem 2

Exercise 2. Compute the interpolating polynomial in a) monomial, b) Lagrange, and c) Newton basis form for the points $(-1,2)$, $(0,0)$, $(2,1)$. Also show that they produce the same polynomial. Note: This question is done on paper.

Solution 2. First we compute the interpolating polynomial in monomial form, which has the form

$$p(t) = c_1 + c_2 t + c_3 t^2$$

To do this, we solve the linear equation

$$\begin{pmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 2 & 4 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}.$$

In other words, we have the system of equations

$$\begin{aligned} c_1 - c_2 + c_3 &= 2 \\ c_1 &= 0 \\ c_1 + 2c_2 + 4c_3 &= 1 \end{aligned}$$

From $c_1 = 0$, we see that we can reduce this system of equations to

$$\begin{aligned} -c_2 + c_3 &= 2 \\ 2c_2 + 4c_3 &= 1 \end{aligned}$$

Setting $c_3 = 2 + c_2$ into the second equation, we are reduced to the single equation

$$6c_2 + 8 = 1$$

It follows that $c_2 = -7/6$, and hence $c_3 = 5/6$. Thus the interpolating polynomial in monomial form is

$$p(t) = -(7/6)t + (5/6)t^2.$$

Next we compute the Lagrange form, it is given by

$$\begin{aligned} p(t) &= 2\ell_1(t) + 0\ell_2(t) + 1\ell_3(t) \\ &= 2\ell_1(t) + \ell_3(t) \\ &= 2\frac{t(t-2)}{(-1-0)(-1-2)} + \frac{(t+1)t}{(2+1)(2-0)} \\ &= 2\left(\frac{t^2-2t}{3}\right) + \left(\frac{(t^2+t)}{6}\right) \\ &= (2/3)t^2 - (4/3)t + (1/6)t^2 + (1/6)t \\ &= -(7/6)t + (5/6)t^2, \end{aligned}$$

where the last part shows that it has the same form as the monomial form. Finally we compute the Newton basis which has the form

$$p(t) = a_1 + a_2(t+1) + a_3(t+1)t.$$

form. To do this, we solve the linear equation

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 3 & 6 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}.$$

In other words, we have the system of equations

$$\begin{aligned} a_1 &= 2 \\ a_1 + a_2 &= 0 \\ a_1 + 3a_2 + 6a_3 &= 1 \end{aligned}$$

From the first equation, we have $a_1 = 2$. Plugging this into the second equation, we obtain $a_2 = -2$. Then plugging all of this into the third equation, we obtain $a_3 = 5/6$. Thus the Newton basis form is

$$\begin{aligned} p(t) &= 2 - 2(t+1) + (5/6)(t+1)t \\ &= 2 - 2t - 2 + (5/6)t^2 + (5/6)t \\ &- (7/6)t + (5/6)t^2 \end{aligned}$$

where the last part shows that it has the same form as the monomial form.

Problem 3

Exercise 3. Interpolate the function $f(x) = 1/(1+25x^2)$ with a polynomial of degree $n-1$ for $n = 3, 5, 7, 9, 11$ equally spaced points x_i between -1 and 1 (so, for $n = 3$ the points are $x_1 = -1$, $x_2 = 0$, and $x_3 = 1$) using monomial representation.

1. For each n, compute the error $\|f - p_n\|_\infty = \max_x |f(x) - p_n(x)|$ (you can approximate the maximum by evaluating it for a large number of x values, for example using linspace(-1,1).
2. Notice that the error is increasing with n. Confirm this visually by generating a plot of f and p_n for $n = 5$ and $n = 9$.
3. Reconcile your result with the error estimate for polynomial interpolation as discussed in class.

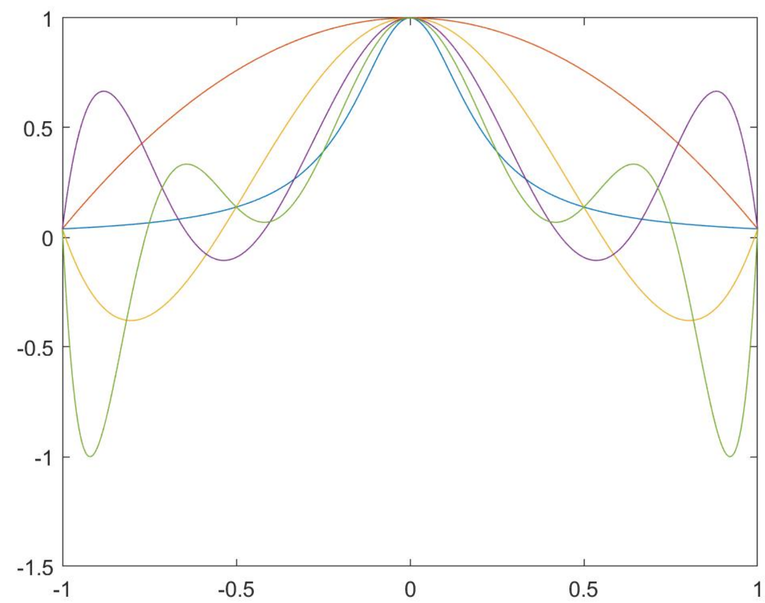
Solution 3. We do this in the following code below:

```
format longg
f = @(x)  1/(1+25*x^2)
v = @(n)  -1:(2/(n-1)):1
apply_func_2_cols = @(f,M) cell2mat(cellfun(f,num2cell(M,1), 'UniformOutput',0))  % need to apply f e
w = @(n) apply_func_2_cols(f,v(n)) % this is f applied to v element-wise
A = @(n) fliplr(vander(v(n)))
c = @(n) A(n)\(w(n)')
p = @(m,n) polyval(flip(c(m)'),v(n));
error = @(m,n) norm((p(m,n)-w(n)),Inf)
for m = 3:2:11
        disp([m error(m,1000)]);
end
```

3	0.646228542340267
5	0.438349795129046
7	0.616925958093177
9	1.04517051816102
11	1.91563314750099

```
plot(v(1000),w(1000),v(1000),p(3,1000),v(1000),p(5,1000),v(1000),p(7,1000),v(1000),p(9,1000))
```

The plot is given by


1
0.5
0
-0.5
-1
-1.5
-1
-0.5
0
0.5
1