README.md

HW3 :: Joey Bingham :: Math 373

1.

```
CODE DONE IN R
a <- c(1.0, 1.1, 1.3, 1.5, 1.9, 2.1)
ones <- array(data = 1, dim = 6)
a_s <- a ^ 2
A <- matrix(data = c(ones, a, a_s), nrow = 6, ncol = 3, byrow = FALSE)
At <- t(A)
At_A <- At %*% A
b <-c(1.84, 1.96, 2.21, 2.45, 2.94, 3.18)
At_b = At %*% b
ans = solve(At_A, At_b)
ans
f <- function(x){ans[1] + ans[2]*x + ans[3]*x*x}
plot.function(f, from = .5, to = 2.5)
err_val = (b - f(a))
err_val_s = err_val ^ 2
err = sum(err_val_s)
err

which yields
            [,1]
[1,]  0.59658071
[2,]  1.25329314
[3,] -0.01085343
So the polynomial would be  0.59658071 +  1.25329314x - 0.01085343x^2.
Plot below
This would have error
[1] 1.801484e-05
```

2.

a.

```
X = array(dim = 10)
ones = array(dim = 11, data = 1)
X[1] = -1
for(i in 2:11){ X[i] = X[i-1] + 1.0/5.0}
X_s = X ^ 2
f = exp(X)
A <- matrix(data = c(ones, X, X_s), nrow = 11, ncol = 3, byrow = FALSE)
At <- t(A)
At_A <- At %*% A
At_f = At %*% f
ans = solve(At_A, At_f)
ans
g <- function(x){ans[1] + ans[2]*x + ans[3]*x*x}
plot.function(g, from = -1.5, to = 2.5)
err_val = (f - g(X))
err_val_s = err_val ^ 2
err = sum(err_val_s)
err
Which yields
          [,1]
```

```
[1,] 0.9949934
[2,] 1.1237483
[3,] 0.5430256
So the polynomial would be 0.9949934 +  1.1237483x + 0.5430256x^2
Plot below
This would have error
[1] 0.01266033
```

b.

```
X = array(dim = 10)
ones = array(dim = 11, data = 1)
X[1] = -1
for(i in 2:11){ X[i] = X[i-1] + 1.0/5.0}
X_new = (X ^ 2)
for(i in 1:10){ X_new[i] = 1.5 * X_new[i] - .5}
f = exp(X)
A <- matrix(data = c(ones, X, X_new), nrow = 11, ncol = 3, byrow = FALSE)
At <- t(A)
At_A <- At %*% A
At_f = At %*% f
ans = solve(At_A, At_f)
ans
g <- function(x){ans[1] + ans[2]*x + ans[3]*x*x}
plot.function(g, from = -1.5, to = 2.5)
err_val = (f - g(X))
err_val_s = err_val ^ 2
err = sum(err_val_s)
err
which yields
              [,1]
[1,] 1.1760019
[2,] 1.1237483
[3,] 0.3620171
So the polynomial would be 1.1760019 +  1.1237483x +  0.3620171x^2
Plot below
This would have error
[1] 0.1873847
```

3.

a.

```
X = c(-0.3826, 0.3826, -0.92388, 0.92388)
ones = c(1, 1, 1, 1)
X_new = (X ^ 2)
f = exp(X)
A <- matrix(data = c(ones, X, X_new), nrow = 4, ncol = 3, byrow = FALSE)
At <- t(A)
At_A <- At %*% A
At_f = At %*% f
ans = solve(At_A, At_f)
ans
g <- function(x){ans[1] + ans[2]*x + ans[3]*x*x}
plot.function(g, from = -1.5, to = 2.5)
Which yields
          [,1]
[1,] 0.9946177
[2,] 1.1303203
[3,] 0.5428980
Which are the coefficents to the degree 3 polynomial, like the previous problems.
The error, E, since we are using chebyshev nodes, is bounded
E < max(-1 <= l, x <= 1; |f''''(l)/4!||(x - x1)(x - x2)(x - x3)(x - x4)|
< max(-1 <= l <= 1; |f''''(l)/4!||1/2^3|) = e/2^3
```

b.

```
#need to translate [0,2] to [-1,1] using t = (2x-(a+b))/(b-a) -> (2x-2)/2 -> x-1, whose inverse is x+1
tr <- function(x){x + 1}
X <- tr(c(-0.3826, 0.3826, -0.92388, 0.92388))
ones <- c(1, 1, 1, 1)
X_new <- (X ^ 2)
f <- exp(X)
A <- matrix(data = c(ones, X, X_new), nrow = 4, ncol = 3, byrow = FALSE)
At <- t(A)
At_A <- At %*% A
At_f <- At %*% f
ans <- solve(At_A, At_f)
ans
g <- function(x){ans[1] + ans[2]*x + ans[3]*x*x}
plot.function(g, from <- -.25, to = 2.5)
Which yields
          [,1]
[1,] 1.1068717
[2,] 0.1210297
[3,] 1.4757498
Which are the coefficents to the degree 3 polynomial, like the previous problems.
The error, E, since we are using chebyshev nodes, is bounded
E < max(0 <= l, x <= 2; |f''''(l)/4!||(x - x1)(x - x2)(x - x3)(x - x4)|
< max(0 <= l <= 2; |f''''(l)/4!||1/2^3|) = (e^2)/(2^3)
```

4.

```
nexp <- function(x){exp(-x)}
ss <- chebfun(deg = 2), nexp, from = -1, to = 1)
plot.function(ss, from = -1, to = 1)
plot below
```

5.

a.

```
n <- 3
f <- function(x){exp(x)}
pi <- acos(-1)

x <- array(dim = 17, data = 0)
for(i in 0:16){x[i+1] <- -pi + i/n*pi}
y <- f(x)
a <- array(dim = n, data = 0)
b <- array(dim = n, data = 0)
for(i in 1:n){
temp_x <- i*x
temp_cos <- cos(temp_x)
temp_sin <- sin(temp_x)
temp_cos_y <- y %*% temp_cos
temp_sin_y <- y %*% temp_sin
overm_cos <- temp_cos_y/n
overm_sin <- temp_sin_y/n
a[i] <- overm_cos
b[i] <- overm_sin
}
a0 <- sum(y)/n
x <- seq(-pi, pi, length = n)
k <- 1:(n)
s1 <- cos(n*x)
s2 <- a[n]/2*s1
s3 <- a0/2 + s2 #split at sum
```

```
s4 <- x %*% k
s5 <- cos(s4)
s6 <- s5 %*% a[1:n] #split at sum
s7 <- x %*% k
s8 <- sin(s7)
s9 <- s8 %*% b
s <- s3 + s6 + s9
c <- fft(y)*16
cc_s <- array(dim = 9, data = 0)
for(i in 0:n){cc_s[i+1] = ((-1)^i)*(1/n)}
cc <- cc_s %*% c[1:9]
aa <- Re(cc)
bb <- Im(cc)
aa0 <- aa[1]
aa <- aa[2:length(aa)]
bb <- bb[2:length(bb)]
ss = aa0/2*(aa[n]/2)%*%cos(n%*%x) + cos(x%*%k[1:n])%*%aa[1:n] + sin(x%*%k[1:n])%*%bb[1:n]
plot.function(ss, from = -pi, to = pi)
Plot of polynomial below
```

b.

```
n <- 3
m <- 6
f <- function(x){exp(x)}
pi <- acos(-1)

x <- array(dim = 17, data = 0)
for(i in 0:16){x[i+1] <- -pi + i/n*pi}
y <- f(x)
a <- array(dim = n, data = 0)
b <- array(dim = n, data = 0)
for(i in 1:n){
temp_x <- i*x
temp_cos <- cos(temp_x)
temp_sin <- sin(temp_x)
temp_cos_y <- y %*% temp_cos
temp_sin_y <- y %*% temp_sin
overm_cos <- temp_cos_y/n
overm_sin <- temp_sin_y/n
a[i] <- overm_cos
b[i] <- overm_sin
}
a0 <- sum(y)/n
x <- seq(-pi, pi, length = n)
k <- 1:(n)
s1 <- cos(n*x)
s2 <- a[n]/2*s1
s3 <- a0/2 + s2 #split at sum
s4 <- x %*% k
s5 <- cos(s4)
s6 <- s5 %*% a[1:n] #split at sum
s7 <- x %*% k
s8 <- sin(s7)
s9 <- s8 %*% b
s <- s3 + s6 + s9
c <- fft(y)*16
cc_s <- array(dim = 9, data = 0)
for(i in 0:n){cc_s[i+1] = ((-1)^i)*(1/n)}
cc <- cc_s %*% c[1:9]
aa <- Re(cc)
bb <- Im(cc)
aa0 <- aa[1]
aa <- aa[2:length(aa)]
bb <- bb[2:length(bb)]
ss1 = aa0/2 #splitting at sum
ss2 = n%*%x
ss2_2 = cos(ss2)
```

```
ss2_3 = (aa[n]/2)%*%ss2_2
ss3 = cos(x%*%k[1:n])%*%aa[1:n]
ss4 = sin(x%*%k[1:n])%*%bb[1:n]
ss = ss1 + ss2 + ss3 + ss4
plot.function(ss, from = -pi, to = pi)
Plot of polynomial below
```

c.

```
n <- 8
f <- function(x){exp(x)}
pi <- acos(-1)

x <- array(dim = 17, data = 0)
for(i in 0:16){x[i+1] <- -pi + i/n*pi}
y <- f(x)
a <- array(dim = n, data = 0)
b <- array(dim = n, data = 0)
for(i in 1:n){
temp_x <- i*x
temp_cos <- cos(temp_x)
temp_sin <- sin(temp_x)
temp_cos_y <- y %*% temp_cos
temp_sin_y <- y %*% temp_sin
overm_cos <- temp_cos_y/n
overm_sin <- temp_sin_y/n
a[i] <- overm_cos
b[i] <- overm_sin
}
a0 <- sum(y)/n
x <- seq(-pi, pi, length = n)
k <- 1:(n)
s1 <- cos(n*x)
s2 <- a[n]/2*s1
s3 <- a0/2 + s2 #split at sum
s4 <- x %*% k
s5 <- cos(s4)
s6 <- s5 %*% a[1:n] #split at sum
s7 <- x %*% k
s8 <- sin(s7)
s9 <- s8 %*% b
s <- s3 + s6 + s9
c <- fft(y)*16
cc_s <- array(dim = 9, data = 0)
for(i in 0:n){cc_s[i+1] = ((-1)^i)*(1/n)}
cc <- cc_s %*% c[1:9]
aa <- Re(cc)
bb <- Im(cc)
aa0 <- aa[1]
aa <- aa[2:length(aa)]
bb <- bb[2:length(bb)]
ss1 = aa0/2 #splitting at sum
ss2 = n%*%x
ss2_2 = cos(ss2)
ss2_3 = (aa[n]/2)%*%ss2_2
ss3 = cos(x%*%k[1:n])%*%aa[1:n]
ss4 = sin(x%*%k[1:n])%*%bb[1:n]
ss = ss1 + ss2 + ss3 + ss4
plot.function(ss, from = -pi, to = pi)
Plot of polynomial below
```