3.3 Hermite Interpolation

1.a

The Hermite approximating polynomial is the third-degree polynomial defined by the diagonal of the divided difference table, interpolated at x=8.4

> x = c(8.3, 8.6)

> y = c(17.56492, 18.50515)

> z = c(1.116256, 1.151762)

> A = cbind(x, y, z)

> hermite (A, 8.4)

$table

[,1] [,2] [,3] [,4]

[1,] 17.56492 NA NA NA

[2,] 17.56492 1.116256 NA NA

[3,] 18.50515 3.134100 6.726147 NA

[4,] 18.50515 1.151762 -6.607793 -44.44647

$coef

[1] 17.564920 1.116256 6.726147 -44.446467

$interp

[1] 17.8327

Thus, the polynomial is

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2.a

> f = function(x){x\*log(x)}

> ploy = hermite(A, 8.4)

> ploy$interp

[1] 17.8327

> acctualError = abs(f(8.4) - ploy$interp)

> acctualError

[1] 0.04444633

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3.a Using the error formula to find a bound for the error.

Since it’s degree three polynomial, we need to find the **maximum** 4th derivative of f(x)

Since is always negative for [8.3, 8.6], then it follows that is always decreasing.

Because is always positive on [8.3, 8.6], then the maximum is

Therefore the bound error for approximation f(8.4) is

It follows that its value is

> x = A[,1]

> x

[1] 8.3 8.6

> M = 2/(8.3)^3

> M

[1] 0.003497806

> error = (M /factorial(4))\*prod(8.4-x)^2

> error

[1] 5.829677e-08

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4.a

Taking the derivative of and creating a table at the values of 1 and 1.05:

> f = function(x){3\*x\*exp(x) - exp(2\*x)}

> fp = function(x){ 3\*exp(x) + 3\*x\*exp(x) - 2\*exp(2\*x) }

> x = c(1, 1.05)

> A = cbind(x, f(x), fp(x))

> A

x

[1,] 1.00 0.7657894 1.531579

[2,] 1.05 0.8354311 1.242215

Applying the Hermite program to this matrix yields the interpolation

> poly = hermite(A, 1.03)

> poly

$table

[,1] [,2] [,3] [,4]

[1,] 0.7657894 NA NA NA

[2,] 0.7657894 1.531579 NA NA

[3,] 0.8354311 1.392834 -2.774886 NA

[4,] 0.8354311 1.242215 -3.012398 -4.750237

$coef

[1] 0.7657894 1.5315788 -2.7748863 -4.7502366

$interp

[1] 0.8093249

The **actual error** is:

> actual = abs( f(1.03) - poly$interp)

> actual

[1] 1.237317e-06

The bound error follows the same formula as for Newton’s Interpolated Divided Difference formula.

In this case, we need to find the maximum of the fourth derivative of f(x).

Using a mesh to find an approximate max:

> xx = seq(1, 1.05, len=1000)

> fp4 = function(x){exp(x) \* (3\*x + 12 - 16\*exp(x))}

> fn = max( abs( fp4(xx)))

> fn

[1] 87.3653

So the **error bound** is:

> error = (fn / factorial(4)) \* prod(1.03 - x**)^2**

> error

[1] 1.31048e-06

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.a Use Hermite polynomial to predict the position of the car and its speed when t=10s.

> x = c(0, 3, 5, 8, 13)

> y=c(0,225,383,623,993)

> z=c(75,77,80,74,72)

> A=cbind(x,y,z)

> poly = hermite(A, 10)

> poly$interp

[1] 742.5028

So the position of the car is 742.5208, when t =10s.

To approximate the speed, instead of taking the derivative of the polynomial, we can use the forward divided difference to approximate it when h is very small.

This is the computer way to calculate derivative.

> x = c(0, 3, 5, 8, 13)

> y=c(0,225,383,623,993)

> z=c(75,77,80,74,72)

> A=cbind(x,y,z)

> h = 1e-8

> speed = ( hermite(A, 10+h)$interp - hermite(A, 10)$interp ) / h

> speed

[1] 48.38174

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.b

55 mph = 242/3 = 80.6666666 ft/sec =

f = function(x){

h = 1e-8

herm=rep(0,length(x))

for (i in 1:length(x)) {

herm[i]=(hermite(A,x[i]+h)$interp-hermite(A,x[i])$interp)/h-242/3

}

return(herm)

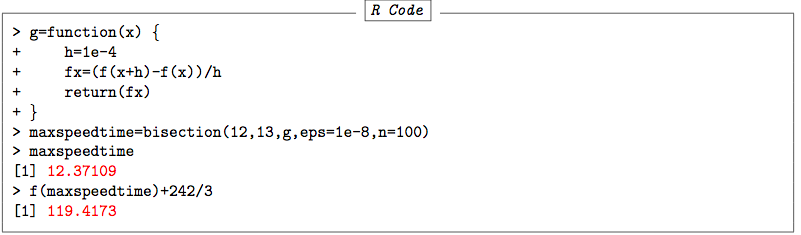
}

Firsttime = bisection(5, 6, f, eps=1e-12, n=100)

> firsttime

[1] 5.648806

The maximum speed that the car attains can be found by taking the derivative of the function above using a divided difference again. We use bisection method to find the answer between 12 and 13 (which can be seen on a graph).



Therefore, it attains the speed of 119.4173 ft/sec (81.42 mph) at 12.37109 seconds.

\*