Yunus Güngör No:150150701 Date:17.04.2016

BLG202E CRN:21843 Homework 3

Q1)a)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Regular Prices | | | Crash Prices | | |
| Day | 0 | 7 | 14 | 21 | 28 | 35 |
| Stock Price | 100 | 98 | 101 | 50 | 51 | 50 |

,

Adding (0,100) data point:

, ,

By using backward substitution:

Adding (21,50) data point:

, ,

The most accurate interpolation is because it does not involve extreme data points (stock prices after crash) and it includes all the other data points.

b)

Matlab Code:

>> w=[0 7 14 21 28 35];

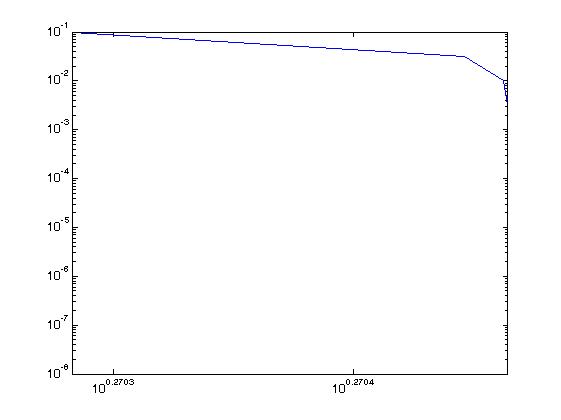
>> p=[100 98 101 50 51 50];

>> p2=100-0.642857142857143\*w+0.0510204081632653\*power(w,2);

>> p3=41+12\*w-0.5510\*power(w,2);

>> figure

>> plot(p,w,p1,w,p2,w)



and is giving the right values before and after crash, in that order. But there is no function we can use for all data.

Q2)

a)

, ,

Using polynomial interpolation:

, ,

, ,

By using backward substitution:

We need to find the maximum point of , by using derivation the maximum point is:

b)

c)

Matlab code:

>> syms x;

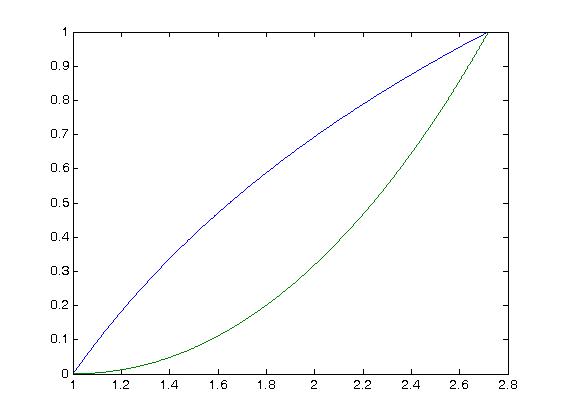
>> p(x)=1-0.12339674565853\*x+1.84167857411758\*x^2;

>> x=0:0.001:1;

>> p=1-0.12339674565853\*x+1.84167857411758\*sqrt(x);

>> g=exp(x);

>> plot(g,x,p,x)



d)

Matlab code:

>> syms x;

>> p(x)=1-0.12339674565853\*x+1.84167857411758\*x^2;

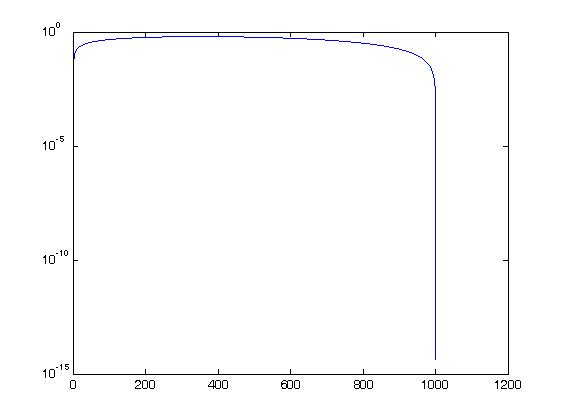
>> x=0:0.001:1;

>> p=1-0.12339674565853\*x+1.84167857411758\*sqrt(x);

>> g=exp(x);

>> d=abs(g-p);

>> semilogy(d)



Q3)a)

By using taylor series:

Transaction error: and error’s order:

b)

Matlab Code:

>> syms x;

>> f(x)=exp(x);

>> syms h;

>> d(x,h)=3\*(((f(x+h)-f(x-h))/h^3)-(2\*f(x)/h^2));% f'(x)=f(x) for f(x)=e^x

>> for k=1:9

l=10^-k;

vpa(d(0,l))

l

end

ans =

1.0005001190641549423763011314301

l =

0.1

ans =

1.0000050000119047784391684704281

l =

0.01

ans =

1.0000000500000011904762070105818

l =

0.001

ans =

1.0000000005000000001190476193302

l =

0.0001

ans =

1.0000000000050000000000122870454

l =

1e-05

ans =

1.0000000000000499999999696164495

l =

1e-06

ans =

1.000000000000000498732999343332

l =

1e-07

ans =

1.0

l =

1e-08

ans =

1.0

l =

1e-09

For , ,

Which is smaller than , this means that created formula is indeed second order accurate. For and formula gives the most accurate result, which is .

c) For very small round-off error eliminates truncation error. Truncation error becomes so small and gets ignored because of the round-off error.

d) To get a fourth order formula for , can be used with Taylor series.

This method will require , , and , which means 4 points required.

Q4)

Matlab code:  
>> syms x;

>> syms h;

>> d(x,h)=(sin(x+h)-2\*sin(x)+sin(x-h))/h^2;

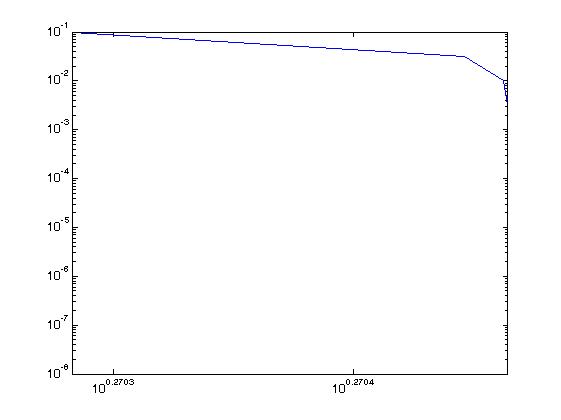
>> k=[1:0.5:8];

>> figure

>> h=power(10,-k);

>> g=sin(1.2)-d(1.2,h);

>> loglog(g,h)Th



Sudden change in the plot caused by round-off error. The most optimal is.