NATIONAL INSTITUE OF TECHNOLOGY KARANATAKA, SURATHKAL

ASSIGNMENT 9

APPLIED COMPUTATIONAL METHODS IN MECHANICAL SCIENCES

RAJAT A CHANDAVAR – 16ME156

22-Oct-19

ASSIGNMENT ON NON-UNIFORM GRID

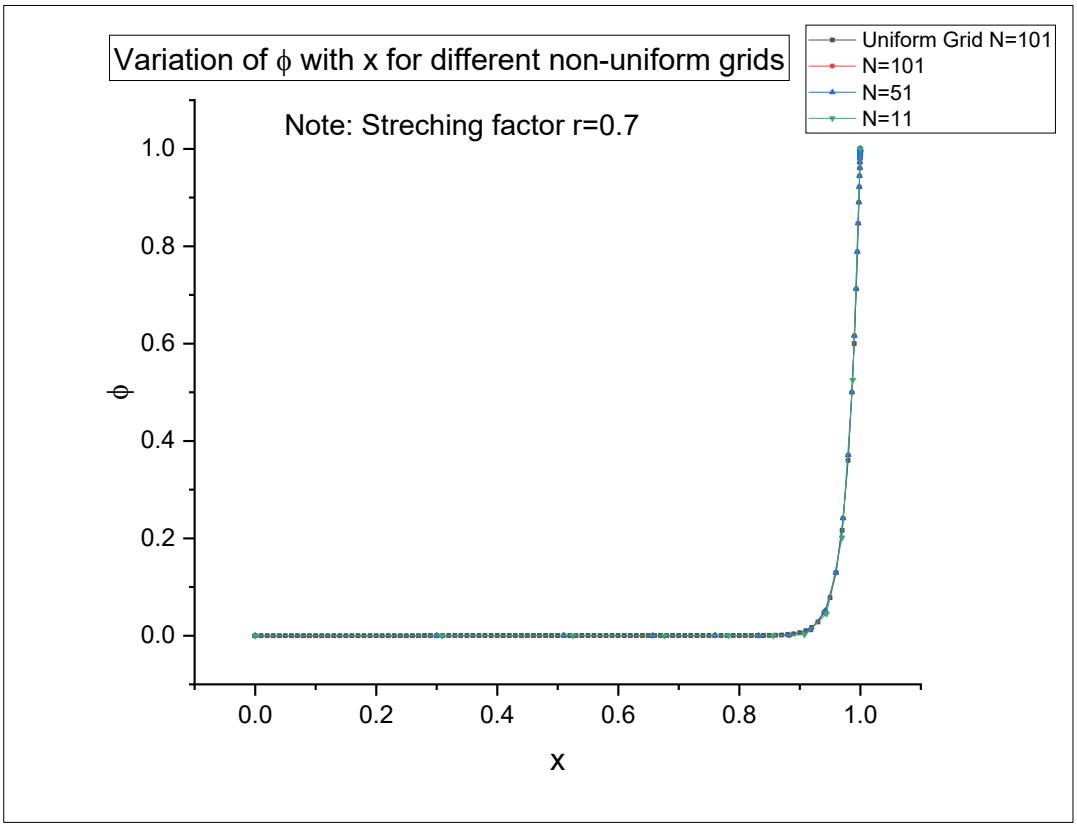
Answer

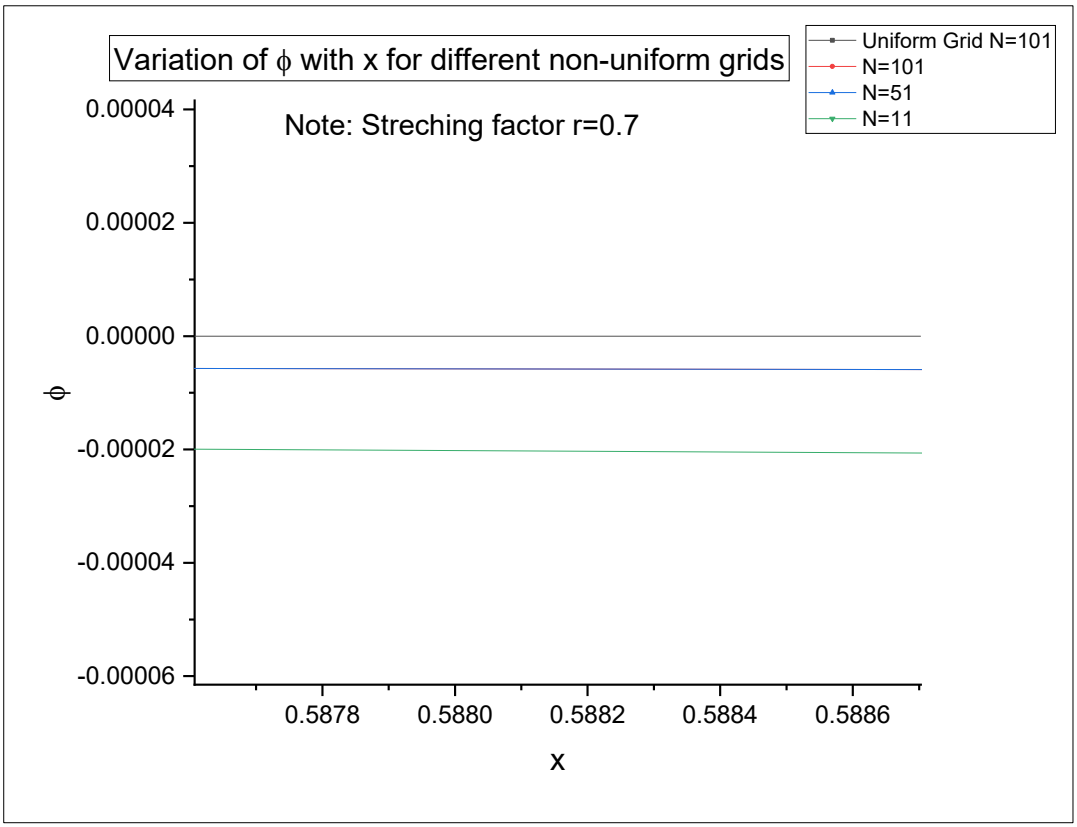
Validation was done with a uniform grid of 100 elements.

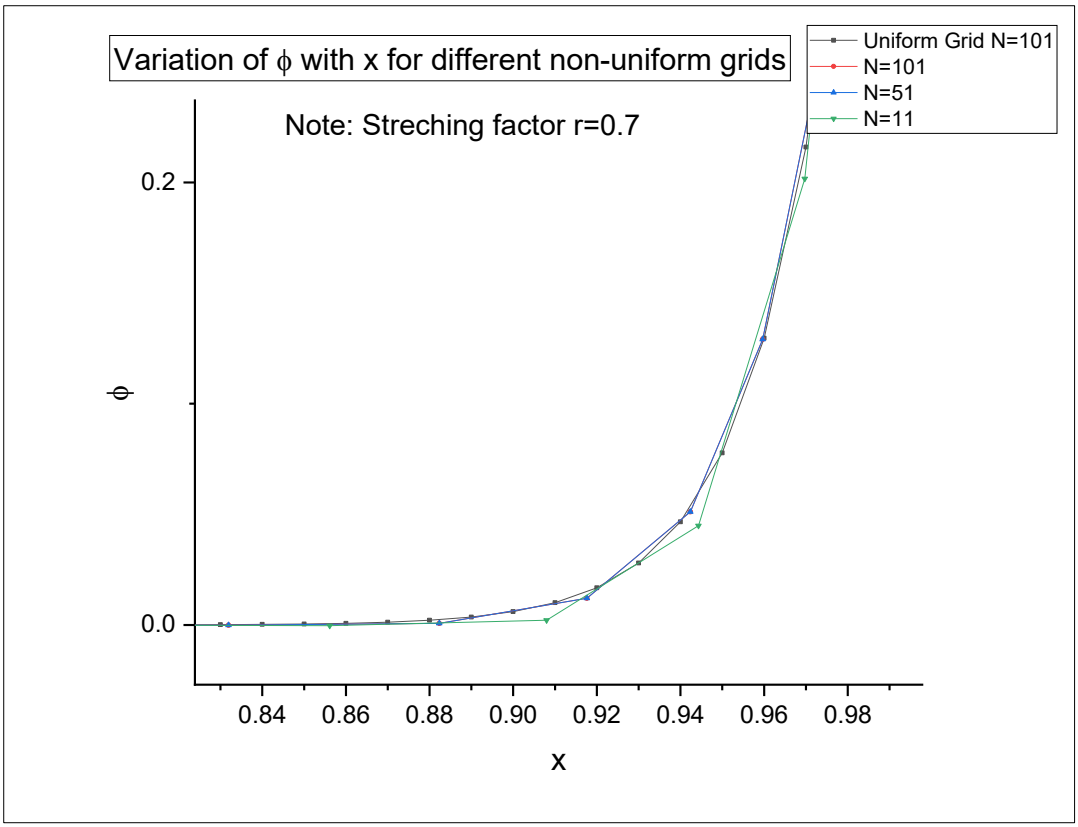
Code(C++)

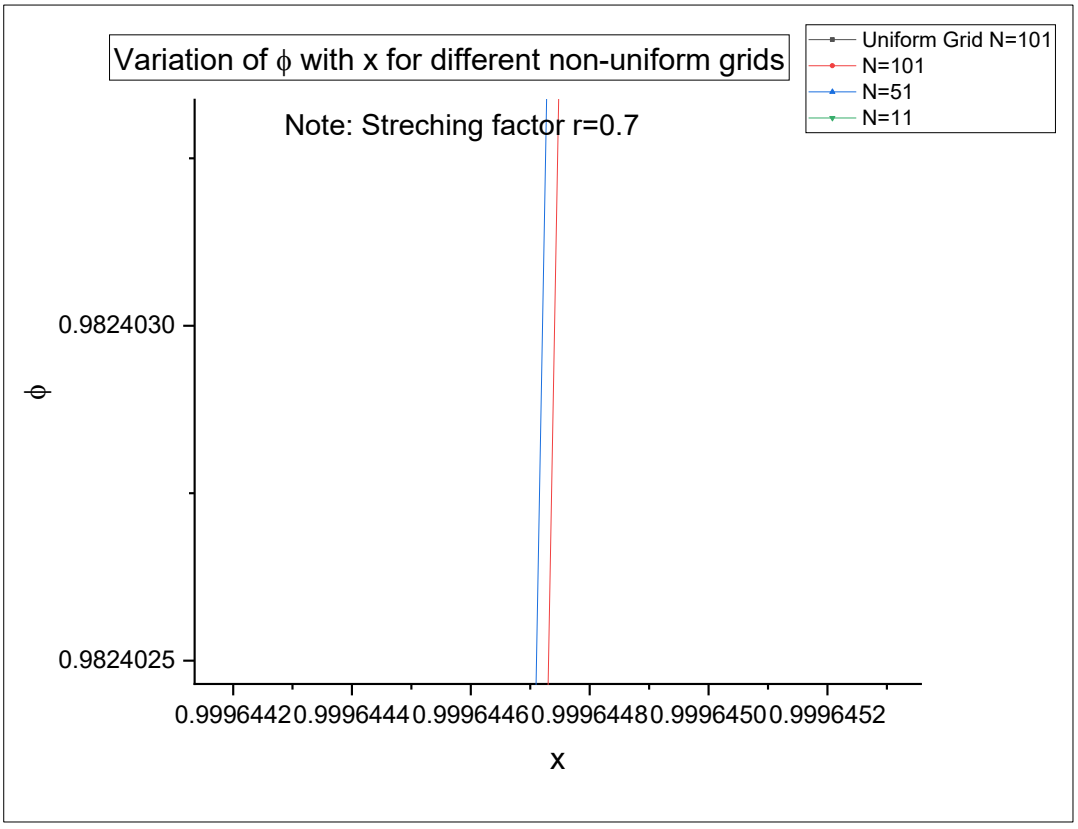
```
#include<iostream>
 #include<cmath>
 #include<fstream>
 #include<time.h>
 using namespace std;
 main()
 {
     clock_t start=clock();
     fstream f;
     f.open("DATA.txt",ios::out);
     int n=51,i;
     double rho=1,u=1,T=0.02,l=1,r=1,dx0,dxi,dxi_1,Ap[n-2],Aw[n-2],Ae[n-2],B[n-2]={ },phi[n],factor,dist=0;
     phi[0]=0;
     phi[n-1]=1;
     dx0=0.02;
     //dx0=l*(1-r)/(1-pow(r,n-1));
     for(i=1;i<n-1;++i)
     {
         dxi=pow(r,i)*dx0;
         dxi_1=pow(r,i-1)*dx0;
         Aw[i-1]=(1/(dxi+dxi_1))*(-rho*u-(2*T/dxi_1));
         Ae[i-1]=(1/(dxi+dxi_1))*(rho*u-(2*T/dxi));
         Ap[i-1]=2*T/(dxi+dxi_1)*(1/dxi+1/dxi_1);
         if(i==1)
             B[0]=-phi[0]*Aw[0];
         if(i==n-2)
             B[n-3]=-phi[n-1]*Ae[n-3];
     }
     for(i=1;i<n-2;++i)
     {
         factor=Aw[i]/Ap[i-1];
         Ap[i]=Ap[i]-factor*Ae[i-1];
         B[i]=B[i]-factor*B[i-1];
     }
     phi[n-2]=B[n-3]/Ap[n-3];
     for(i=n-3;i>=1;--i)
         phi[i]=(B[i-1]-Ae[i-1]*phi[i+1])/Ap[i-1];
     f<<'0'<<" "<<phi[0];
     for(i=1;i<n-1;++i)
     {
         dist+=pow(r,i-1)*dx0;
         f<<"\n"<<dist<<" "<<phi[i];
     }
     f<<"\n"<<l<<" "<<phi[n-1];
     f.close();
     clock_t stop=clock();
     double timespent = (double)(stop-start)/(double)CLOCKS_PER_SEC;
     cout<<"\nCPU Time:"<<timespent<<" seconds";
 }
```

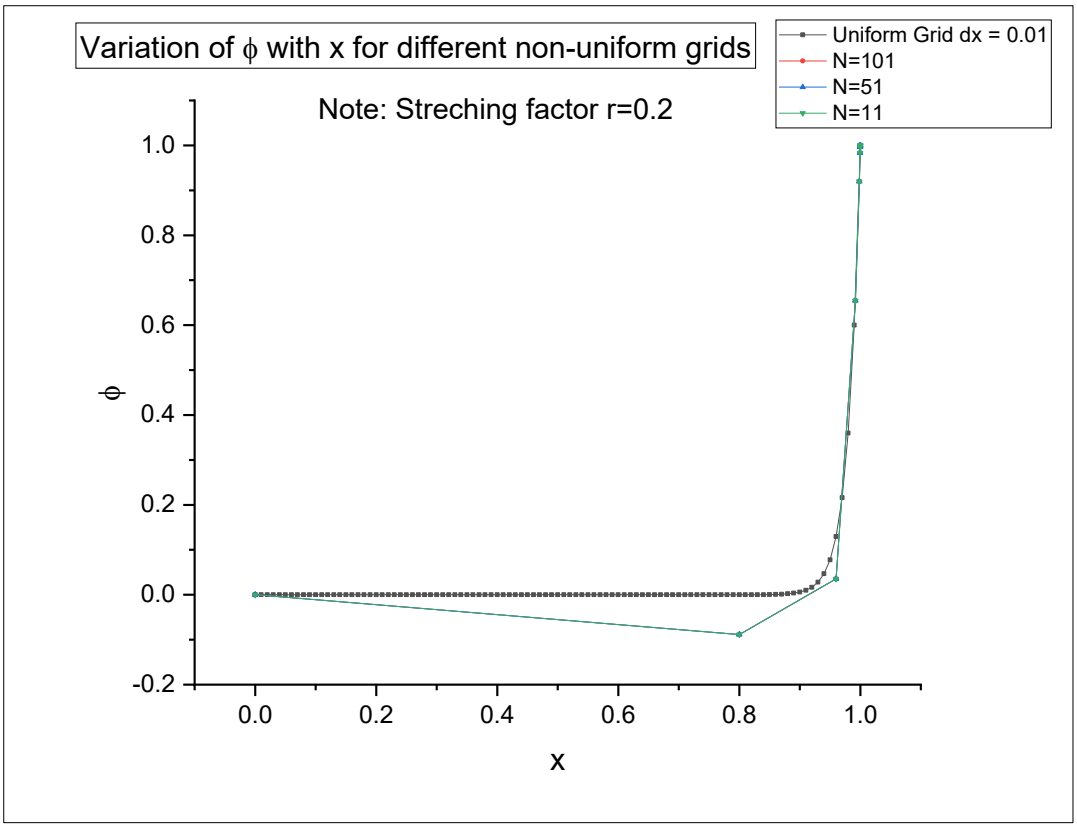
Output

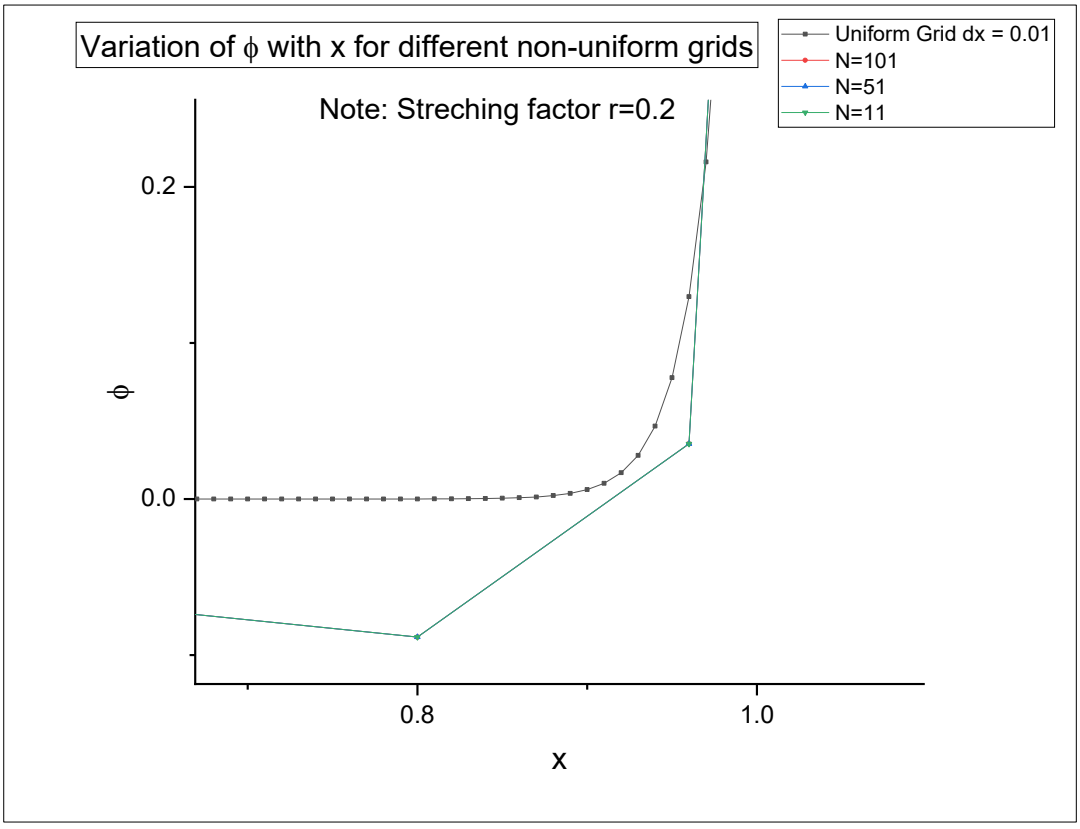

Variation of ϕ with x for different non-uniform grids
Note: Streching factor r=0.7
Uniform Grid N=101
N=101
N=51
N=11
ϕ
x
0.0
0.2
0.4
0.6
0.8
1.0

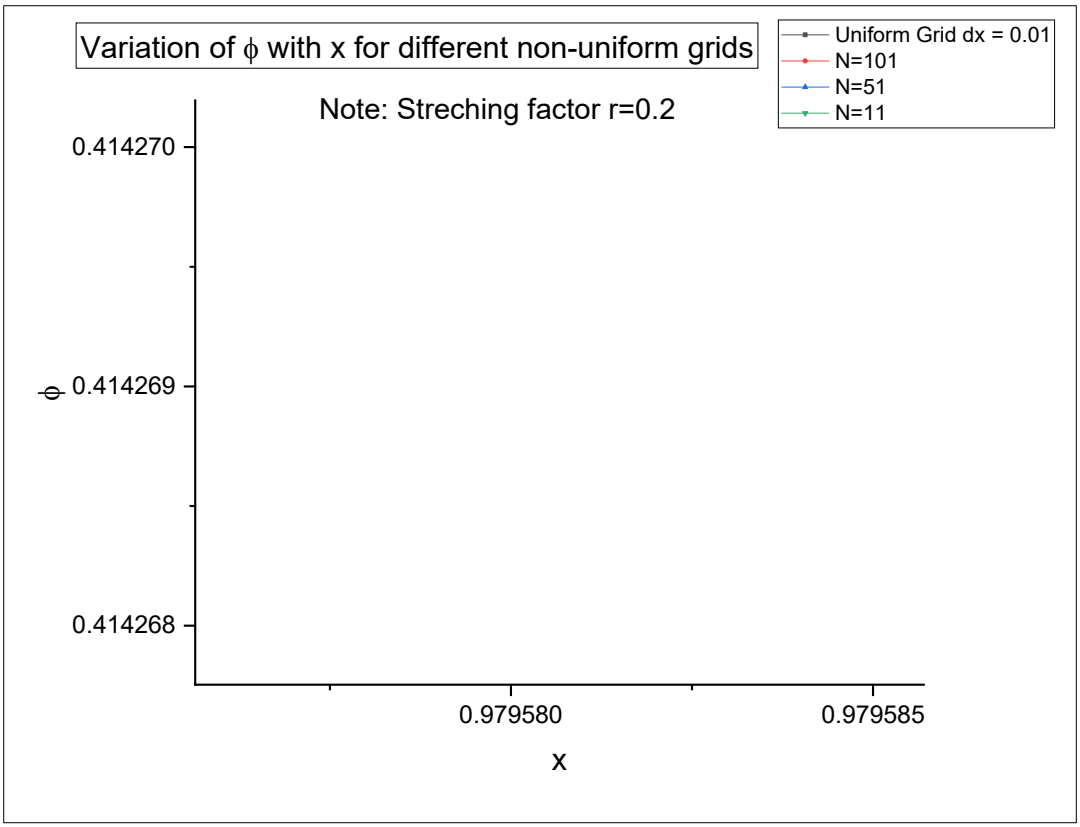

Variation of ϕ with x for different non-uniform grids
Note: Streching factor r=0.7
Uniform Grid N=101
N=101
N=51
N=11
ϕ
x
0.00004
0.00002
0.00000
-0.00002
-0.00004
-0.00006
0.5878
0.5880
0.5882
0.5884
0.5886


Variation of ϕ with x for different non-uniform grids
Note: Streching factor r=0.7
Uniform Grid N=101
N=101
N=51
N=11
0.2
0.0
ϕ
0.84
0.86
0.88
0.90
0.92
0.94
0.96
0.98
x


Variation of ϕ with x for different non-uniform grids
Note: Streching factor r=0.7
Uniform Grid N=101
N=101
N=51
N=11
0.9824030
0.9824025
ϕ
0.9996442
0.9996444
0.9996446
0.9996448
0.9996450
0.9996452
x


Variation of ϕ with x for different non-uniform grids
Note: Streching factor r=0.2
Uniform Grid dx = 0.01
N=101
N=51
N=11
ϕ
x
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0.0
0.2
0.4
0.6
0.8
1.0


Variation of ϕ with x for different non-uniform grids
Note: Streching factor r=0.2
Uniform Grid dx = 0.01
N=101
N=51
N=11
ϕ
x
0.2
0.0
0.8
1.0



Computational Times (s)

	N=11	N=51	N=101
Uniform Grid	0.003	0.005	0.006
Non-uniform grid	0.003	0.004	0.004

For r=0.7, reasonably good accuracy is found with the uniform grid. For r=0.2, significant deviation was found. This is because the first Δx is very huge (0.8 units) leading to incorrect results whereas in the former case first Δx was just 0.30872 units. Results seemed to almost coincide for different grid sizes in both cases.

As expected, the uniform grid took more computational time for a more number of elements. For lower no. of elements, much differnece wasn't found.