FINITE DIFFERENCE METHOD
Lab Report for Assignment No. 11

PAWANPREET KAUR
(2020PHY1092)

KABIR SETHI
(2020PHY1097)

S.G.T.B. Khalsa College, University of Delhi, Delhi-110007, India.

May 1, 2022

Submitted to Dr. Mamta
"32221401 - MATHEMATICAL PHYSICS III"

Contents

1 Theory

Theory

Date

Finite difference method

→ To Solve 2-Point linear BVP?

Given

$$y''(x) = P(x)\,y'(x) + Q(x)\,y(x) + R(x) \quad -①$$

$$x \in [a,b]$$

Let's start from the Drichlet Conditions.

$$y(a)=\alpha \;,\; y(b)=\beta$$

We divide $[a,b]$ in N subintervals. corresponding to x_i & the exact value of function $= y_i$

Now we Replace each derivative with the finite difference approximation. formula.

$$y_i''(x) = \frac{\cancel{y_{i+1}}\; y_{i+1} - 2y_i + y_{i-1}}{h^2} + O(h^2)$$

$$y_i'(x) = \frac{y_{i+1}-y_{i-1}}{2h} + O(h^2)$$

So Substituting this in eq ①, we get

$$\frac{y_{i+1}-2y_i+y_{i-1}}{h^2} + O(h^2) = P_i\left(\frac{y_{i+1}-y_{i-1}}{2h}\right) + Q y_i + R + O(h^2)$$

OR.

$$\frac{y_{i+1}-2y_i+y_{i-1}}{h^2} = P\frac{(y_{i+1}-y_{i-1})}{2h} + Q y_i + R \quad -②$$

To solve these we need two more equations from initial & boundary condition i.e.

$$y(a)=y(x_0)=\alpha \;,\; y(b)=y(x_N)=\beta$$

OR we can write the eq② as

$$\left(-1-\frac{h}{2}P_i\right)y_{i-1} + (2+h^2Q_i)y_i + \left(-1+\frac{h}{2}P_i\right)y_{i+1} = -h^2R_i \quad -③$$

As we know $y_0=\alpha$

Using it we can write.

Putting $i=1$

$$\left(-1-\frac{h}{2}P_i\right)y_0 + (2+h^2Q_i)y_1 + \left(-1+\frac{h}{2}P_1\right)y_2 = -h^2R_i$$

OR $\left(-1-\frac{h}{2}P_i\right)\alpha + (2+h^2Q_i)y_1 + \left(-1+\frac{h}{2}P_i\right)y_2 = -h^2R_i$

Similarly we can write the last eq as

$$\left(-1-\frac{h}{2}P_{i-2}\right)y_{i-2} + (2+h^2Q_{i-1})y_{i-1} = -h^2R_i + \left(1-\frac{h}{2}P_0\right)\beta$$

we can write it in matrix form

$$\begin{bmatrix} d_i & u_1 & & & & & \\ l_2 & d_2 & u_2 & & & & \\ & l_3 & d_3 & u_3 & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & l_{n-3} & d_{N-3} & u_{N-3} & \\ & & & & l_{N-2} & d_{N-2} & u_{N-2} \\ & & & & & l_{N-1} & d_{N-1} \end{bmatrix}$$

where $d_i = 2+h^2Q_i$

$u_i = -1+\frac{h}{2}P_i$

$l_i = -1 \cancel{+} - \frac{h}{2}P_i$

Date

With ~~ye~~ vectors $\vec{Y}$ & $\vec{B}$ as

$$\vec{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-2} \\ y_{N-1} \end{bmatrix} \qquad \vec{B} = \begin{bmatrix} -h^2 r_1 + \left(1 + \frac{h}{2} P_1\right)\alpha \\ -h^2 r_2 \\ \vdots \\ -h^2 \cdot r_{N-2} \\ -h^2 r_{N-1} + \left(1 - \frac{h}{2} P_{N-1}\right)\beta \end{bmatrix}$$

And thus

$$\vec{Y} = A^{-1} \cdot \vec{B}$$

If we use Newmann OR Robin Conditions We would have.

$$A = \begin{bmatrix} a_{11} & a_{12} & & & \\ l_1 & d_1 & u_1 & & \\ & l_2 & d_2 & u_2 & \\ & & \ddots & \ddots & \ddots \\ & & l_{N-1} & d_{N-1} & u_{N-1} \\ & & & d_{N+1N} & a_{N+1,N+1} \end{bmatrix}$$

$$\vec{B} = \begin{bmatrix} b_1 \\ -h^2 r_1 \\ -h^2 r_2 \\ \vdots \\ -h^2 r_{N-1} \\ b_{N+1} \end{bmatrix}$$

where

$d_i = 2 + h^2 Q_i$ $\quad u_i = -1 + \frac{h}{2} P_i$, $\quad l_i = -1 - \frac{h}{2} P_i$

&

$$a_{11} = \begin{cases} 1 & \text{Dirichlit B.C} \\ d_0 & \text{Newmam} \\ d_0 + 2h l_0 \frac{\alpha_1}{\alpha_2} & \text{Robin} \end{cases}$$

$$a_{12} = \begin{cases} 0 & \text{Dirichlet} \\ -2 & \text{otherwise} \end{cases}$$

$$a_{N+1,N+1} = \begin{cases} 1 & \text{Dirichlet} \\ d_N & \text{Newmen} \\ d_N - 2h u_N \frac{\beta_1}{\beta_2} & \text{Robin} \end{cases}$$

$$b_1 = \begin{cases} \alpha & \text{Dirichlet} \\ -h^2 \gamma_0 + 2h l_0 \alpha & \text{Newmenn} \\ -h^2 \gamma_0 + 2h l_0 \frac{\alpha_3}{\alpha_2} & \text{Robin} \end{cases}$$

$$b_{N+1} = \begin{cases} \beta & \text{Dirichlet} \\ -h^2 \gamma_N - 2h u_N \beta & \text{Newmenn} \\ -h^2 \gamma_N - 2h u_N \frac{\beta_3}{\beta_2} & \text{Robin} \end{cases}$$

(b) Eq. $-y'' + \pi^2 y = 2\pi^2 \sin(\pi x)$, $x \in [0,1]$

$p(x) = 0 = p_i$

$q(x) = \pi^2 = q_i$

$r(x) = -2\pi^2 \sin(\pi x)$

$$h = \frac{1-0}{N}$$

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 2+(\pi/4)^2 & -1 & 0 & 0 \\ 0 & -1 & 2+(\pi/4)^2 & -1 & 0 \\ 0 & 0 & -1 & 2+(\pi/4)^2 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \sqrt{2}\,(\pi/4)^2 \\ 2\,(\pi/4)^2 \\ \sqrt{2}\,(\pi/4)^2 \\ 0 \end{bmatrix} \quad \& \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

Solution : $Ax = B$

$x = A^{-1} B$

$$x = \begin{bmatrix} 0 \\ 0.725371 \\ 1.0259 \\ 0.725470 \\ 0 \end{bmatrix}$$

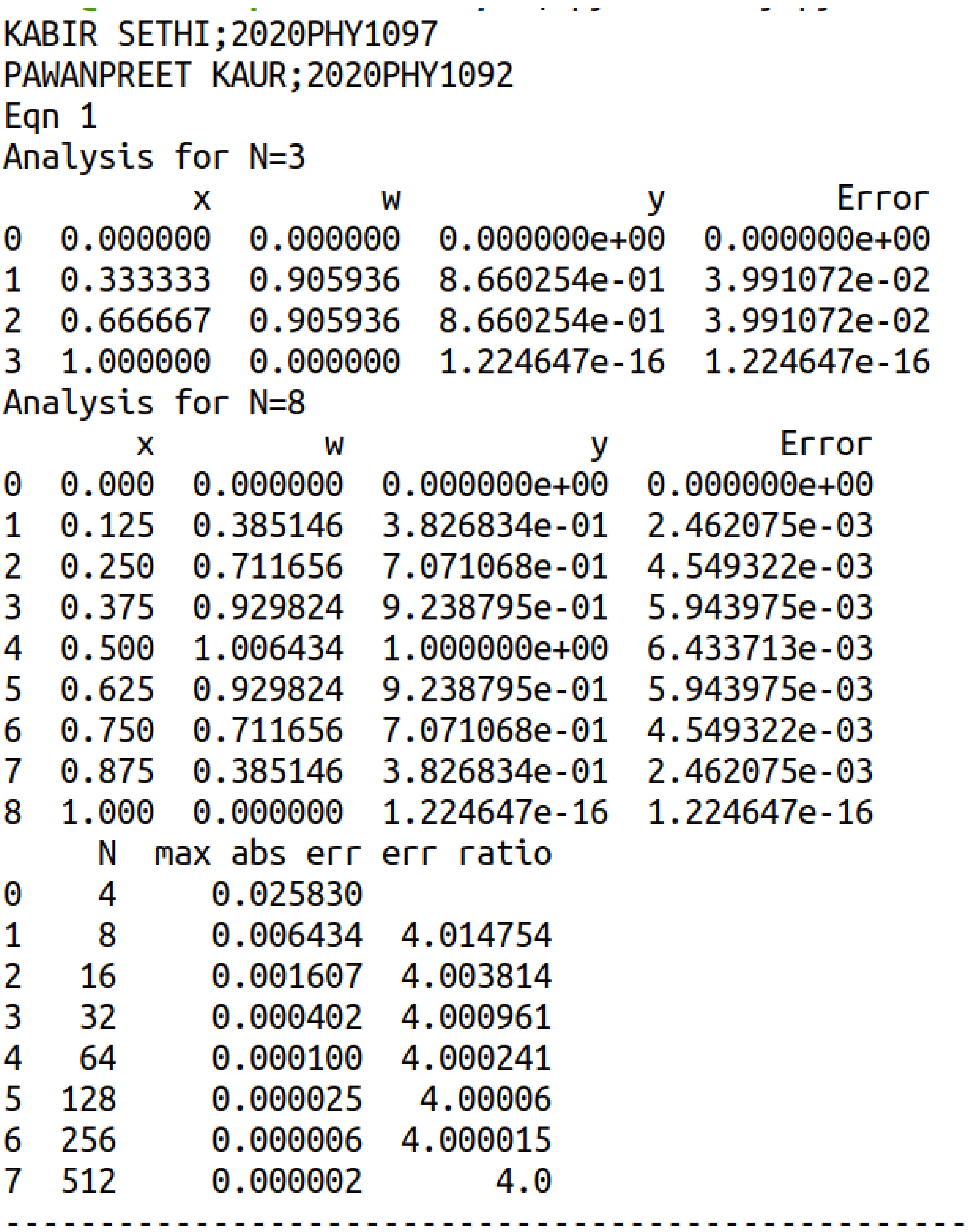
2 Programming

```
#KABIR SETHI;2020PHY1097
#PAWANPREET KAUR;2020PHY1092
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

print("KABIR SETHI;2020PHY1097")
print("PAWANPREET KAUR;2020PHY1092")

def solve(l,d,u,B):
    #l = Lower Diagnol, d = Main Diagnol, u = Upper Diagnol, B = solution vector
    n = len(B)
    w = np.zeros(n-1)
    g = np.zeros(n)
    x = np.zeros(n)
    w[0] = u[0]/d[0]
    g[0] = B[0]/d[0]
    for i in range(1,n-1):
        w[i] = u[i]/(d[i] - l[i-1]*w[i-1])
    for i in range(1,n):
        g[i] = (B[i] - l[i-1]*g[i-1])/(d[i] - l[i-1]*w[i-1])
    x[n-1] = g[n-1]
    for i in range(n-1,0,-1):
        x[i-1] = g[i-1] - w[i-1]*x[i]
    return x

def FiniteDifference(a,b, , ,N,p,q,r,y,key,c1,c2):
  h = (b-a)/N #size of each subinterval
  x = []
  W = []
  Y = []
  l=[]
  err = []
  for i in range(0,N+1):
    x.append(a + i*h)
  if c1 == 'd':
    B = [ [0]]
    d = [1]
    u = [0]
  elif c1 == 'n':
    B = [-h*h*r(x[0]) + 2*h*(-1-h*p(x[0])/2)*( [0])]
    d = [2+h*h*q(x[0])]
    u = [-2]
  elif c1 == 'r':
    B = [-h*h*r(x[0]) + 2*h*(-1-h*p(x[0])/2)*( [2]/ [1])]
    d = [2+h*h*q(x[0]) + 2*h*(-1-h*p(x[0])/2)*( [0]/ [1])]
    u = [-2]
  for i in range(1,N):
    d.append(2+h*h*q(x[i]))
    u.append(-1+h*p(x[i])/2)
    l.append(-1-h*p(x[i])/2)
    B.append(-h*h*r(x[i]))
  if c2 == 'd':
    B.append( [0])
    d.append(1)
    l.append(0)
  elif c2 == 'n':
    B.append(-h*h*r(x[N]) - 2*h*(-1+h*p(x[N])/2)*( [0]))
    d.append(2+h*h*q(x[N]))
    l.append(-2)
  elif c2 == 'r':
    B.append(-h*h*r(x[N]) - 2*h*(-1+h*p(x[N])/2)*(  [2]/  [1]))
    d.append(2+h*h*q(x[N]) - 2*h*(-1+h*p(x[N])/2)*(  [0]/  [1]))
    l.append(-2)
  W = solve(l,d,u,B)
  Y = []
  err = []
  for i in range(0, N+1):
    Y.append(y(x[i]))
    err.append(abs(Y[i]-W[i]))
  if key==1:
    print(f"Analysis for N={N}")
    data = {"x" : x, "w" : W, "y" : Y, "Error" : err}
    table = pd.DataFrame(data)
    print(table)
  return W, max(err), x

def Analysis(Ns, name):
  max_err_for_N = []
  err_ratio = [" "]
  for i in range(0,8):
    W, err, x = FiniteDifference(a,b,  ,  ,Ns[i],p,q,r,y,0,c1,c2)
    max_err_for_N.append(err)
    plt.plot(x,W,label=f"N = {Ns[i]}")
    if i!=0:
      err_ratio.append(max_err_for_N[i-1]/max_err_for_N[i])
  x = np.linspace(a,b,1000)
  Y = []
  for i in range(0,len(x)):
    Y.append(y(x[i]))
  plt.plot(x,Y,label="Analytic",color="black")
  plt.xlabel("x")
  plt.ylabel("y")
  plt.title("Variation of solution with N")
  plt.legend()
  plt.grid()
  plt.savefig(f"Solution_{name}.pdf")
  plt.show()
  data1 = {"N" : Ns, "max abs err" : max_err_for_N, "err ratio" : err_ratio}
  table1 = pd.DataFrame(data1)
  print(table1)
  plt.scatter(Ns,max_err_for_N)
  plt.xlabel("No. of subintervals")
  plt.ylabel("Max. absolute error")
  plt.title("N vs Max abs error")
  plt.xscale("log")
  plt.yscale("log")
  plt.grid()
  plt.savefig(f"Analysis_{name}.pdf")
  plt.show()

#Q1
print("Eqn 1")
a = 0
b = 1
   = [0]
   = [0]
c1 = 'd'
c2 = 'd'
name = "Q1"
p = lambda x : 0
q = lambda x : np.pi*np.pi
```

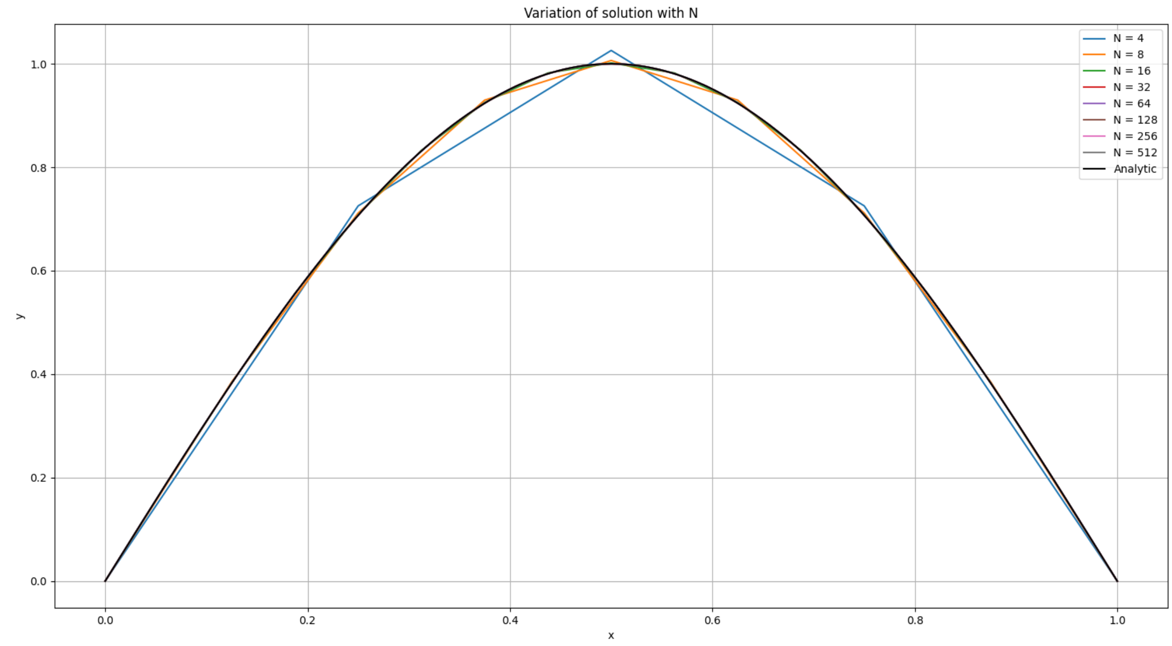
```
r = lambda x : -2*np.pi*np.pi*np.sin(np.pi*x)
y = lambda x : np.sin(np.pi*x)

W, err, x = FiniteDifference(a,b,  ,  ,3,p,q,r,y,1,c1,c2)
W, err, x = FiniteDifference(a,b,  ,  ,8,p,q,r,y,1,c1,c2)

Ns = []
for i in range(2,10):
  Ns.append(pow(2,i))

Analysis(Ns, name)
print("---------------------------------------------------")
#Q2
print("Eqn 2")
a = 0
b = np.pi/2
   = [1,1,-1]
   = [1]
c1 = 'r'
c2 = 'n'
name = "Q2"
p = lambda x : 0
q = lambda x : -1
r = lambda x : np.sin(3*x)
y = lambda x : -1*np.cos(x) + (3/8)*np.sin(x) - (1/8)*np.sin(3*x)

W, ErrArr, x = FiniteDifference(a,b,  ,  ,3,p,q,r,y,1,c1,c2)
W, ErrArr, x = FiniteDifference(a,b,  ,  ,8,p,q,r,y,1,c1,c2)

Ns = []
for i in range(2,10):
  Ns.append(pow(2,i))

Analysis(Ns, name)
```

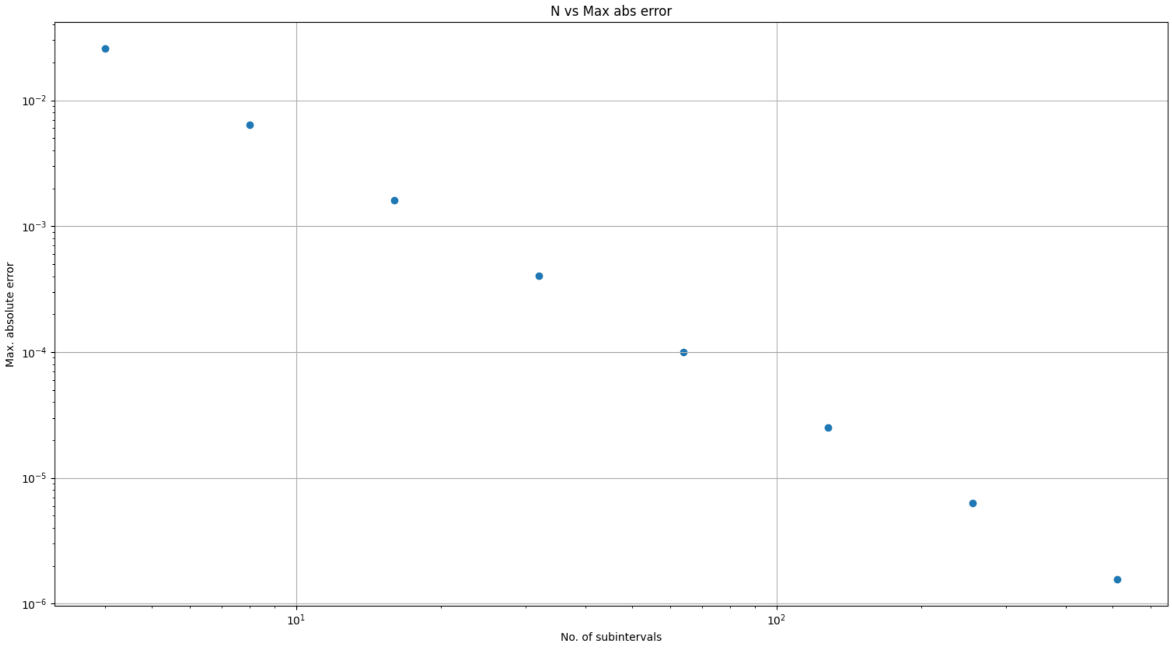
3 Results

Equation 1

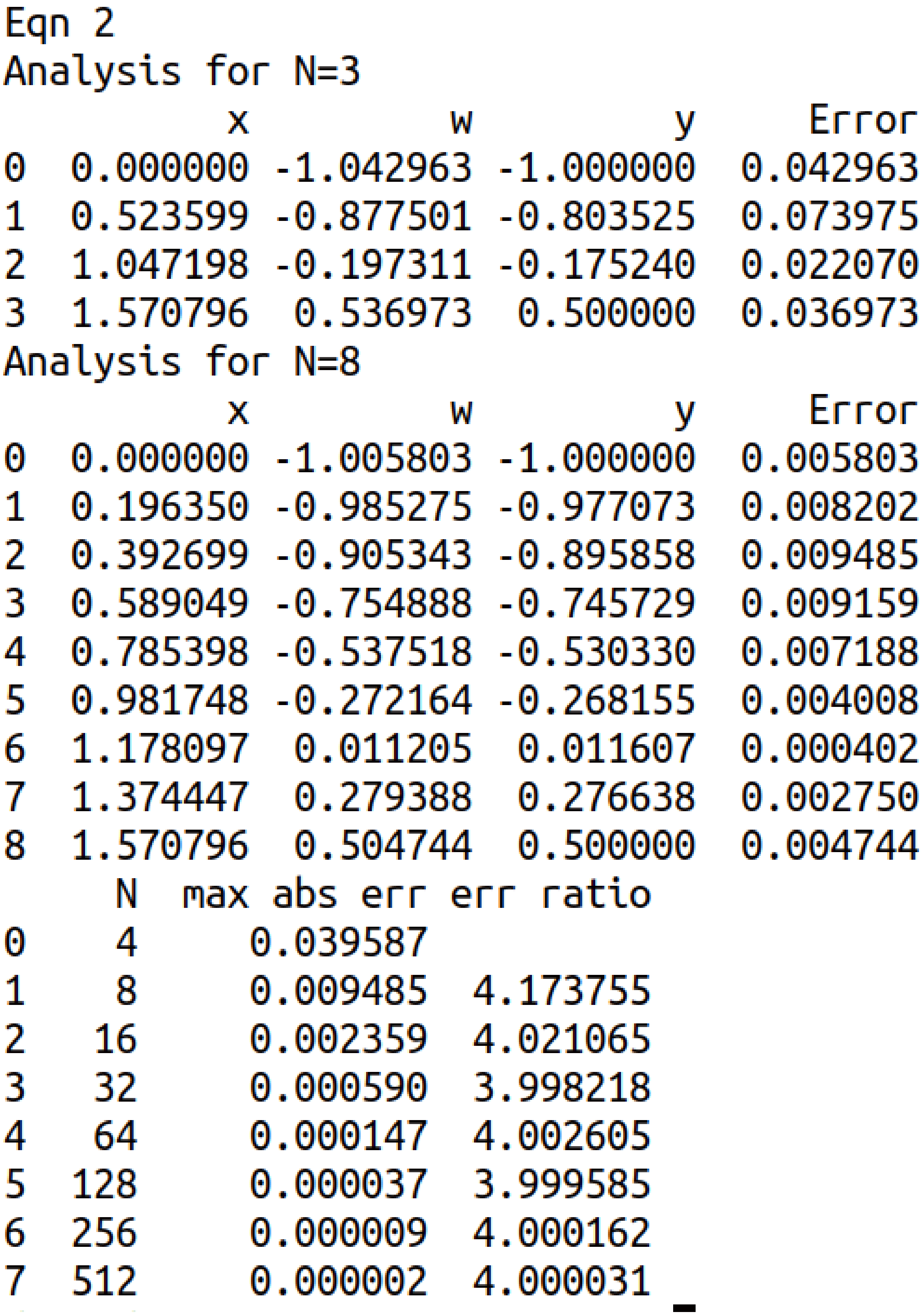


```
KABIR SETHI;2020PHY1097
PAWANPREET KAUR;2020PHY1092
Eqn 1
Analysis for N=3
          x          w             y         Error
0  0.000000  0.000000  0.000000e+00  0.000000e+00
1  0.333333  0.905936  8.660254e-01  3.991072e-02
2  0.666667  0.905936  8.660254e-01  3.991072e-02
3  1.000000  0.000000  1.224647e-16  1.224647e-16
Analysis for N=8
       x          w             y         Error
0  0.000  0.000000  0.000000e+00  0.000000e+00
1  0.125  0.385146  3.826834e-01  2.462075e-03
2  0.250  0.711656  7.071068e-01  4.549322e-03
3  0.375  0.929824  9.238795e-01  5.943975e-03
4  0.500  1.006434  1.000000e+00  6.433713e-03
5  0.625  0.929824  9.238795e-01  5.943975e-03
6  0.750  0.711656  7.071068e-01  4.549322e-03
7  0.875  0.385146  3.826834e-01  2.462075e-03
8  1.000  0.000000  1.224647e-16  1.224647e-16
     N  max abs err err ratio
0    4     0.025830
1    8     0.006434  4.014754
2   16     0.001607  4.003814
3   32     0.000402  4.000961
4   64     0.000100  4.000241
5  128     0.000025   4.00006
6  256     0.000006  4.000015
7  512     0.000002       4.0
---------------------------------------------------
```

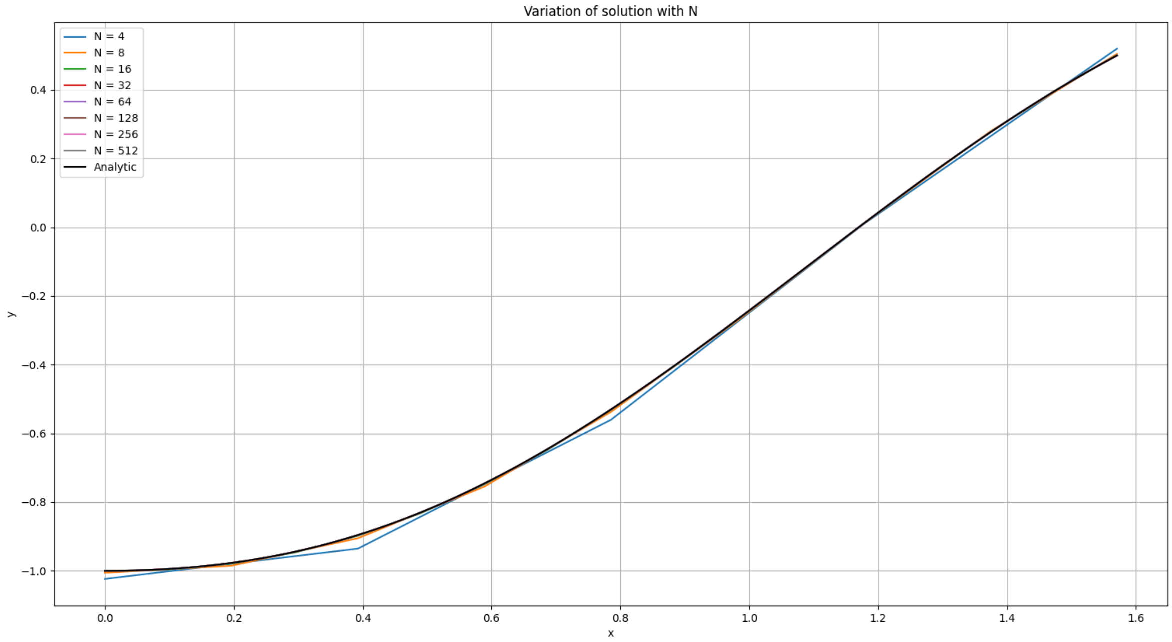

Variation of solution with N
N = 4
N = 8
N = 16
N = 32
N = 64
N = 128
N = 256
N = 512
Analytic
y
1.0
0.8
0.6
0.4
0.2
0.0
x
0.0
0.2
0.4
0.6
0.8
1.0

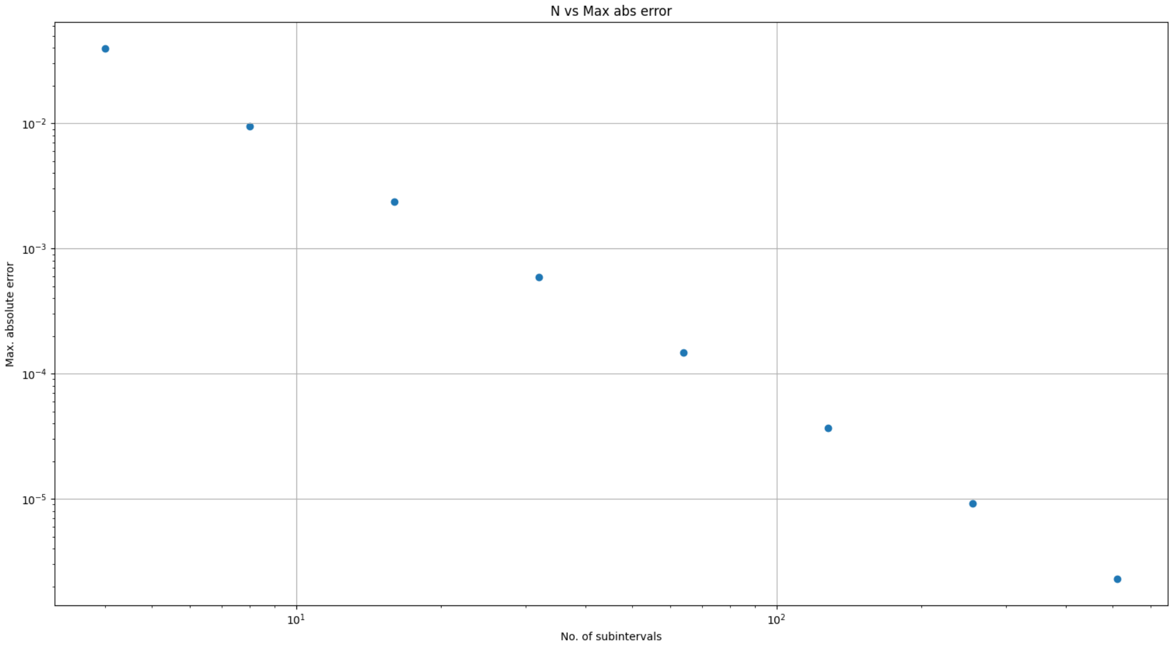

N vs Max abs error
Max. absolute error
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
No. of subintervals
10^{1}
10^{2}

Equation 2



```
Eqn 2
Analysis for N=3
          x          w          y     Error
0  0.000000 -1.042963 -1.000000  0.042963
1  0.523599 -0.877501 -0.803525  0.073975
2  1.047198 -0.197311 -0.175240  0.022070
3  1.570796  0.536973  0.500000  0.036973
Analysis for N=8
          x          w          y     Error
0  0.000000 -1.005803 -1.000000  0.005803
1  0.196350 -0.985275 -0.977073  0.008202
2  0.392699 -0.905343 -0.895858  0.009485
3  0.589049 -0.754888 -0.745729  0.009159
4  0.785398 -0.537518 -0.530330  0.007188
5  0.981748 -0.272164 -0.268155  0.004008
6  1.178097  0.011205  0.011607  0.000402
7  1.374447  0.279388  0.276638  0.002750
8  1.570796  0.504744  0.500000  0.004744
     N  max abs err err ratio
0    4     0.039587
1    8     0.009485  4.173755
2   16     0.002359  4.021065
3   32     0.000590  3.998218
4   64     0.000147  4.002605
5  128     0.000037  3.999585
6  256     0.000009  4.000162
7  512     0.000002  4.000031 _
```


Variation of solution with N
N = 4
N = 8
N = 16
N = 32
N = 64
N = 128
N = 256
N = 512
Analytic
y
x
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6


N vs Max abs error
Max. absolute error
No. of subintervals
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{1}
10^{2}