Applied Computational Methods in Mechanical Sciences

(ME466)

Assignment 7

Himanshu Kumar

16ME234

October 14, 2019

Problem Statement:

Obtain the variation of ϕ vs x , given the governing differential equation as:

$$\Gamma \frac{d^2\phi}{dx^2} + \frac{d\phi}{dx} - \phi = 0$$

1. Use central difference scheme with $\Gamma = 0.1$, where $\Delta x = 0.1$ and $\Delta x = 0.01$.
2. Use forward difference scheme with $\Gamma = 0.1$, where $\Delta x = 0.01$.
3. Use backward difference scheme with $\Gamma = 0.1$, where $\Delta x = 0.01$.

The boundary conditions are:

1. At $x = 0$; $\phi = 1.0$
2. At $x = 1.0$; $\phi = 0$

Python Code:

```
import time

import matplotlib.pyplot as plt


# formulate tridiagonal matrix

def solve_plot(delx,scheme):

    def matrix_form(delx,scheme):

        rng = 1.0

        n = int((rng/delx) -1)

        #print (n)
```

```
a = [[0 for i in range(n)] for i in range(n)]
t = 0.1

xm = [0 for i in range(n+1)]
for i in range(n):
    xm[i+1] = xm[i]+delx
#print (x)
if(scheme is "cds"):
    alpha = 1+ (delx/2*t)
    beta = -2-(delx*delx/t)
    gamma = 1- (delx/2*t)
elif(scheme is "bds"):
    alpha = 1
    beta = -2-(delx*delx/t) +(delx/t)
    gamma = 1- (delx/t)
elif(scheme is "fds"):
    alpha = 1+ (delx/t)
    beta = -2-(delx*delx/t) -(delx/t)
    gamma = 1

for i in range(1,n-1):
    a[i][i-1] = alpha
    a[i][i] = beta
    a[i][i+1] = gamma

#row 0
a[0][0]=beta
a[0][1] = gamma

#row n-1
a[n-1][n-1]=beta
a[n-1][n-1]=alpha
```

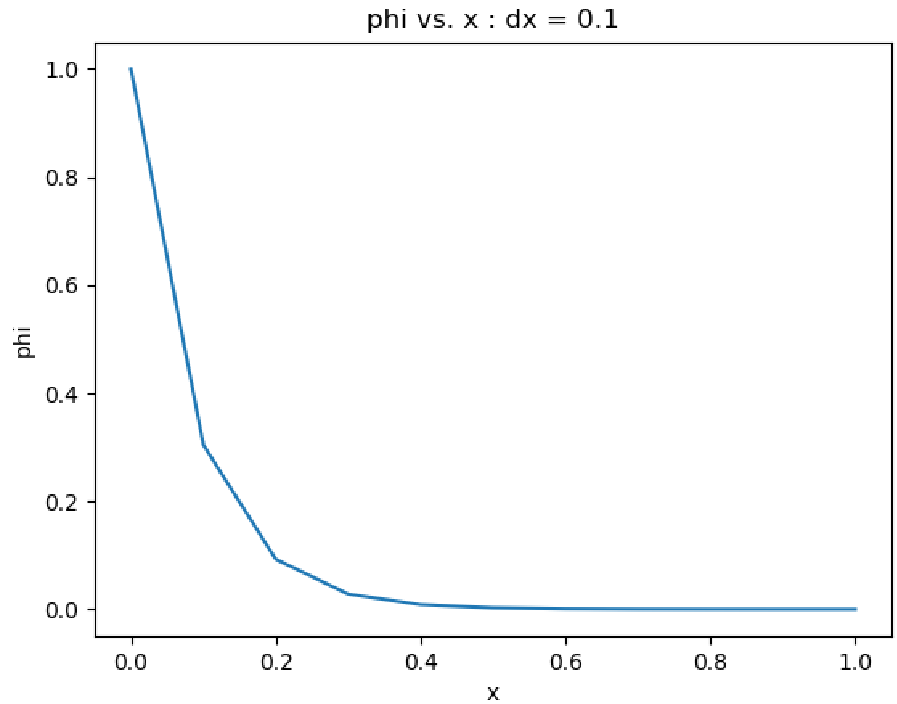
```
    #rhs vector

    b = [0 for i in range(n)]

    b[0] = -gamma

    b[n-1] =  0

    return(a,b,xm,n)


def thomas(a,b):

    n= len(b)

    for i in range(1,n):

        a[i][i-1]=a[i][i-1]/a[i-1][i-1]

        a[i][i] = a[i][i]-a[i][i-1]*a[i-1][i];

        b[i]= b[i]-a[i][i-1]*b[i-1]

        a[i][i-1]=0


    #backward substitution

    x=[0 for i in range(n)]

    x[n-1]= b[n-1]/a[n-1][n-1]

    for k in range(n-1):

        i=n-2-k

        x[i] = ( b[i] + x[i+1] )/ a[i][i]


    #print(x)

    return(x)


#display matrices and check

a,b,xm,n = matrix_form(delx,scheme)

# for i in range(n):

#     print(a[i])

# for i in range(n):

#     print(b[i])
```

```
    phi=thomas(a,b)

    mod_phi = [0 for i in range(n+1)]
    for i in range(len(phi)):
        mod_phi[i+1] = phi[i]
    mod_phi[0] = 1
    mod_phi[n]= 0
    print(mod_phi)
    print(xm)
    print ("\n CPU time: ", time.process_time(),'s')

    #plotting
    plt.plot(mod_phi, xm)
    plt.xlabel('x')
    plt.ylabel('phi')
    meta = str(delx)
    meta = "phi vs. x : dx = " + meta
    plt.title(meta)
    plt.show()

solve_plot(0.1,"cds")
# solve_plot(0.01,"cds")

# solve_plot(0.01,"fds")

#solve_plot(0.01,"bds")
```

Results:

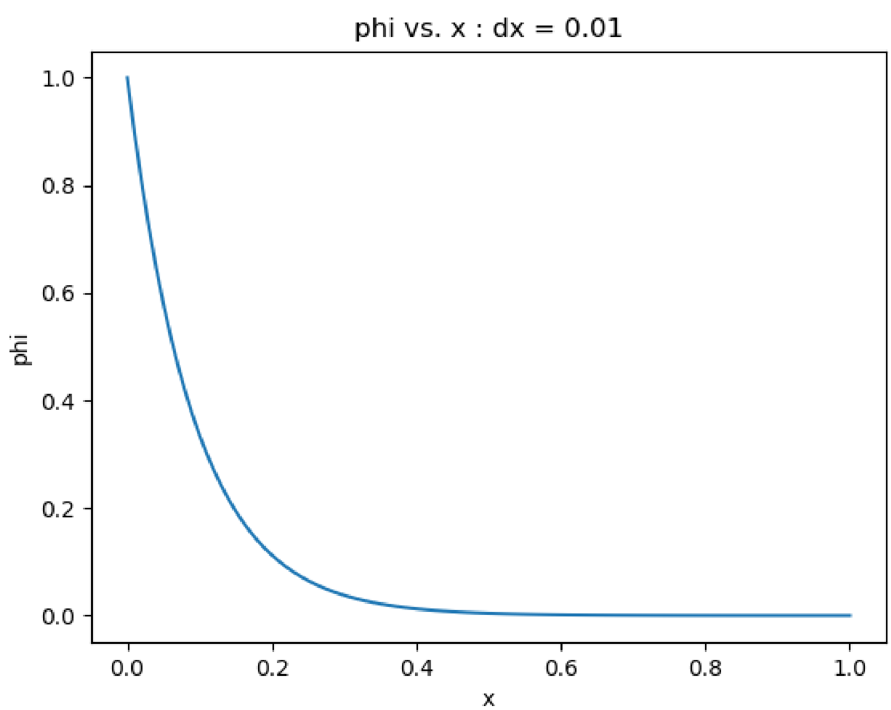
1. **With CDS and $\boldsymbol{\Delta x = 0.1}$:**

```
2.  CPU time:  0.390625 s
```



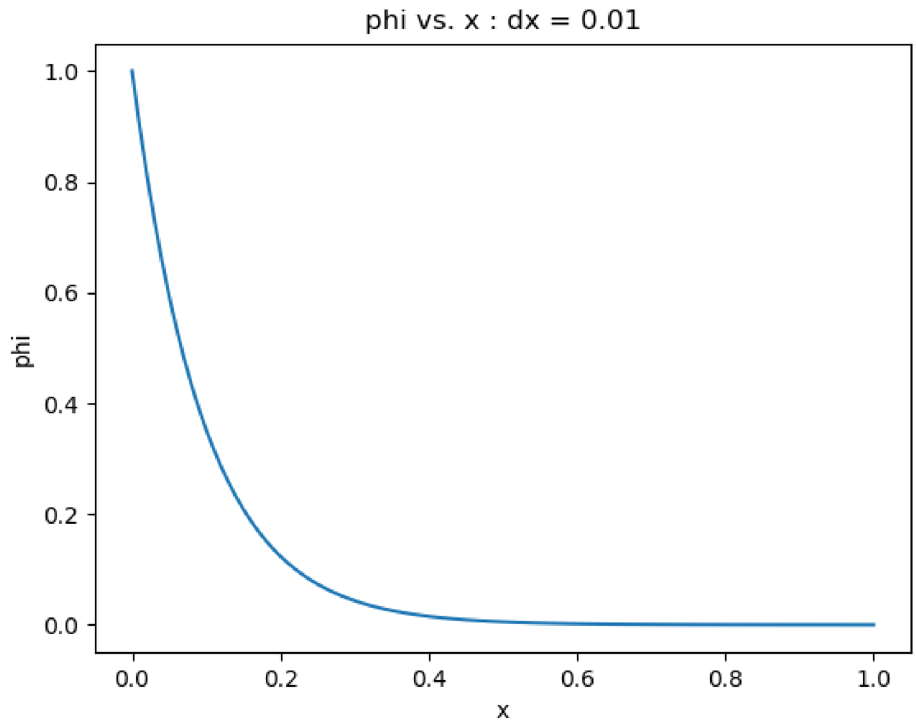
2. **With CDS and $\boldsymbol{\Delta x = 0.01}$:**

```
3.  CPU time:  0.734375 s
```



3. With FDS and $\Delta x = 0.01$:

```
3.  CPU time:  0.4375 s
```



4. With BDS and $\Delta x = 0.01$:

```
5.  CPU time:  0.828125 s
```