

Advanced Numerical Techniques

March 9, 2019

1 LAB 6

2 PDE

2.1 Question 6

2.1.1 2nd order PDE of linear principle part.

Parabolic PDE, $B^2 - AC = 0$. Heat Equation.

$$u_t = u_{xx}$$

Initial Condition:

$$u(x, 0) = \sin(\pi x)$$

Boundary Condition:

$$u(0, t) = 0$$

$$u(1, t) = 0$$

$$\delta t = \frac{1}{64}, \quad \text{Experiment with different } \delta x.$$

```
In [1]: import numpy as np
import pandas as pd
```

```
In [2]: x1 = 0
x2 = 1
```

```
In [3]: def thomas_(a,b,c,d):
    c_ = np.zeros(c.size)
    d_ = np.zeros(d.size)

    c_[0] = c[0]/b[0]
    d_[0] = d[0]/b[0]

    for i in range(1, c.shape[0]-1):
```

```

        c_[i] = c[i]/(b[i] - a[i]*c_[i-1])

    for i in range(1, d.shape[0]):
        d_[i] = (d[i] - a[i]*d_[i-1])/(b[i] - a[i]*c_[i-1])

    return [c_, d_]

def main_(h=0.25):
    err = 0.00001

    n = int((x2-x1)/h)

    x_f = np.zeros(n+1)
    u_j_n = np.zeros(n+1)

    for i in range(n+1):
        x_f[i] = i*h
        u_j_n[i] = np.sin(np.pi*x_f[i])
    u_j_n[0]=u_j_n[-1]=0
    r = 1/(64*h*h)

    flag=1
    while flag==1:

        a = np.zeros(n-1)
        b = np.zeros(n-1)
        c = np.zeros(n-1)
        d = np.zeros(n-1)

        for i in range(n-1):
            a[i] = r
            b[i] = -1 * (1+2*r)
            c[i] = r
            d[i] = -1 * u_j_n[i+1]

        a[0] = 0
        c[-1] = 0

        c_, d_ = thomas_(a,b,c,d)
        res1 = np.zeros(n-1)

        res1[-1] = d_[-1]
        for i in range(n-2):
            res1[n-3-i] = d_[n-3-i] - res1[n-2-i]*c_[n-3-i]

        res = np.zeros(n+1)
        for i in range(n-1):

```

```

        res[i+1] = res1[i]

    flag=0
    for i in range(n+1):
        if np.absolute(res[i]-u_j_n[i]) > err:
            flag = 1

    u_j_n = res

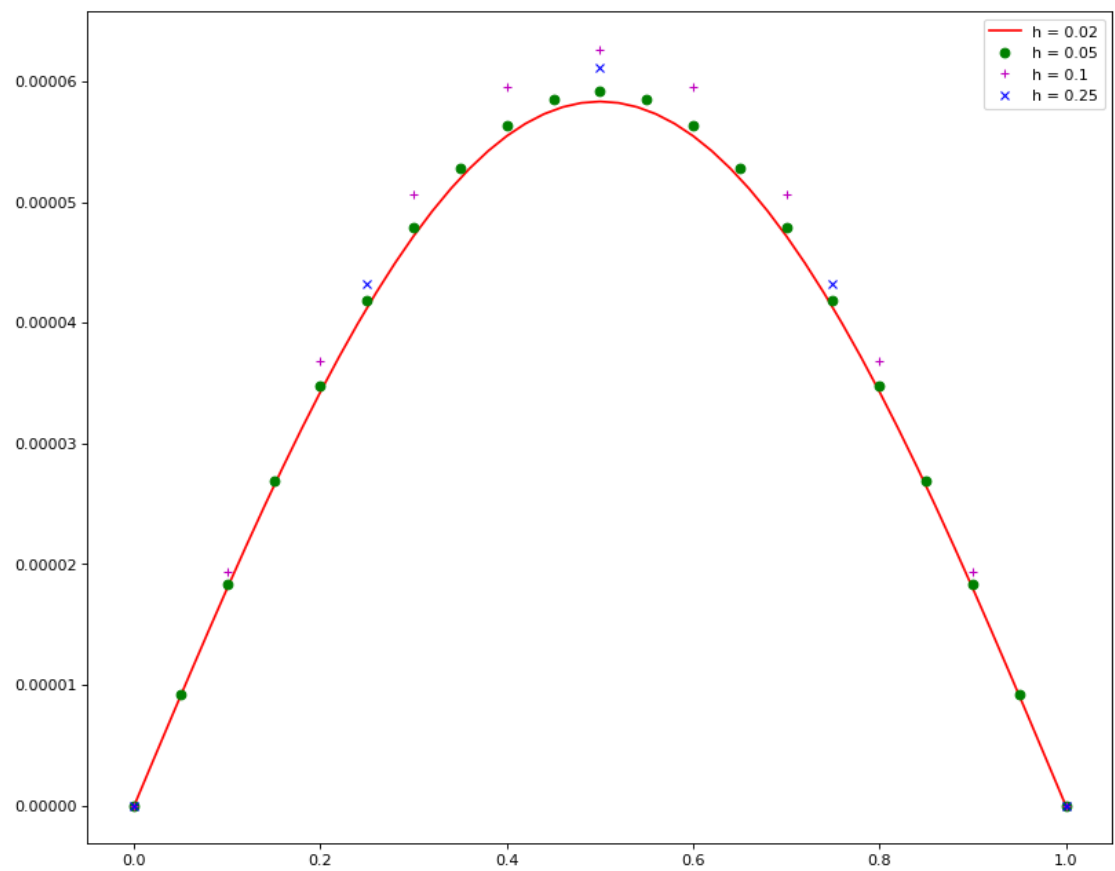
    return [u_j_n, x_f]

In [4]: a_1, x_1 = main_(0.02)
        a_2, x_2 = main_(0.05)
        a_3, x_3 = main_(0.1)
        a_4, x_4 = main_(0.25)

In [6]: import matplotlib.pyplot as plt
        from matplotlib.pyplot import figure
        figure(num=None, figsize=(12, 10), dpi=80, facecolor='w', edgecolor='k')

        plt.plot(x_1, a_1, 'r-', label = 'h = 0.02')
        plt.plot(x_2, (a_2), 'go', label = 'h = 0.05')
        plt.plot(x_3, (a_3), 'm+', label = 'h = 0.1')
        plt.plot(x_4, (a_4), 'bx', label = 'h = 0.25')
        plt.legend(loc='best')
        plt.show()

```



In []: