project i path

February 12, 2016

In [19]:
```
import numpy as np
import matplotlib as plt
import scipy.linalg as la
import time

mat_loop = (100,100)
for m in mat_loop:
    n=m
    a_1 = 2
    a_1D = a_1 * np.ones((1, n-1)) #1D array of 2s
    a=np.diag(a_1D) #2D array with main diagonal filled w/ a
    b_1 = -1
    b_1D = b_1 * np.ones((1, n-2)) #1D array of -1
    b=np.diag(b_1D, k=-1)#2D array with adj down diagonal filled w/ b
    c=np.diag(b, k=1)#2D array w/ adj up diagonal filled w/ c
    h=1.0/((n+1.0)-1.0)
    x=n*h
    source = (h ** 2.0) * 100.0 * np.exp(-10.0 * x)
    A_0 = np.zeros((1, n+1))

    #starts system clock to measure computation time
    begin=time.clock()

    for i in range(1,n-1):   #begin Gaussian Elimination
        a[i] = a[i] - (b[i+1] * b[i+1])/ a[i]
        source[i] = source[i] - b[i + 1]/a[i] * source[i]

    #backward substitution
    A_0[0, n-2] = source[n-2]/a[0,n-2]

    ####
    for k in range(n-3, -1,-1):
        A_0[0,i]=(source[i]-b[0,i]*source[0, i+1])/d[0,i]

    end=time.clock()
    print('Gaussian elimination comp time ', finish - start, 's')

    exact=1-(1-np.exp(-10.0))*x-np.exp(-10.0*x)
    err=np.log10(abs(u-exact)/exact)
    err_max=max(error)
    print('Maximum error using Gaussian elimination is:', err_max)

    print('Time for Gaussian Elimination is: ', begin-end, 'seconds')
```

```
    #plot

    #LU Decomposition

for j in A:
    A=np.diag(2*np.ones(j-1),0)+np.diag(-1*np.ones(j-2),1)+np.diag(-1*np.ones(m-2),-1)
    h=1.0/((n+1.0)-1.0)
    x=n*h
    source = (h ** 2.0) * 100.0 * np.exp(-10.0 * x)
    begin = time.clock()
    LU_decomp = la.lu_factor(A)
    V = la.lu_solve(LU_decomp,source)
    end = time.clock()
    V = V[range(0,m-1)]
    exact = 1-(1-np.exp(-10.0))*x-np.exp(-10.0*x)
    err = np.log10(np.abs(V-exact)/exact)
    err_max = max(err)

    print('Time for LU Decomposition is: ', begin-end, 'seconds')
    print('Maximum error using LU Decomposition is:', err_max)
```

```
---------------------------------------------------------------------------

IndexError                                Traceback (most recent call last)

<ipython-input-19-d173b35e70ea> in <module>()
 23
 24     for i in range(1,n-1):   #begin Gaussian Elimination
---> 25         a[i] = a[i] - (b[i+1] * b[i+1])/ a[i]
 26         source[i] = source[i] - b[i + 1]/a[i] * source[i]
 27


IndexError: index 1 is out of bounds for axis 0 with size 1
```

```
In [23]: import scipy.linalg as la
for j in A:
    A=np.diag(2*np.ones(j-1),0)+np.diag(-1*np.ones(j-2),1)+np.diag(-1*np.ones(m-2),-1)
    h=1.0/((n+1.0)-1.0)
    x=n*h
    source = (h ** 2.0) * 100.0 * np.exp(-10.0 * x)
    begin = time.clock()
    LU_decomp = la.lu_factor(A)
    V = la.lu_solve(LU_decomp,source)
    end = time.clock()
    V = V[range(0,m-1)]
```

```
        exact = 1-(1-np.exp(-10.0))*x-np.exp(-10.0*x)
        err = np.log10(np.abs(V-exact)/exact)
        err_max = max(err)

        print('Time for LU Decomposition is: ', begin-end, 'seconds')
        print('Maximum error using LU Decomposition is:', err_max)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)

<ipython-input-23-47e945051da1> in <module>()
  1 import scipy.linalg as la
  2 for j in A:
----> 3     A=np.diag(2*np.ones(j-1),0)+np.diag(-1*np.ones(j-2),1)+np.diag(-1*np.ones(m-2),-1)
  4     h=1.0/((n+1.0)-1.0)
  5     x=n*h


/Users/jmmilem/anaconda/lib/python3.5/site-packages/numpy/core/numeric.py in ones(shape, dtype,
181
182     """
--> 183     a = empty(shape, dtype, order)
184     multiarray.copyto(a, 1, casting='unsafe')
185     return a


ValueError: sequence too large; cannot be greater than 32
```

In [ ]: