Question 3. Try different diffusion coefficients in Example 1 of Note 4-2. Measure the diffusion times and compare them with the theoretical model.

In [1]:

```
# 1D diffusion solver by FTCS method
# Suggested parameters: xmax=1, dx=0.01, dt=0.001, D=0.05, imax=1000

import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
# Get the parameters
xmax=float(2) # Domain size
dx=float(0.1)   # Mesh size
dt=float(0.01)   # Time step
D=float(0.5)   # Diffusion coefficient
imax=int(500)  # max step to run
```

In [3]:

```
def diffusion_eq(xmax,dx,dt,D,imax) :
    # Parameters
    a=D*dt/dx**2
    Nx=int(xmax/dx)

    # Variables
    x=np.arange(0,Nx+1); x = x*dx
    u=0.0*x

    # initial shape of u
    u=np.sin(np.pi*x/xmax)+0.3*np.sin(2.0*np.pi*x/xmax)+0.2*np.sin(3.0*np.pi*x/xmax) + 0.1*np.s

    # Show the initial shape of u
    #plt.ylim(0.0,1.3); plt.yticks(np.arange(0,1.4,0.2)); plt.plot(x,u); plt.show()

    # Boundary condition (zero)
    u[0]=u[Nx]=0.0

    # Main loop
    i=0
    sum0 = np.sum(u)
    #print("sum0 : ",sum0)
    while i<imax:
        u[1:-1] += a*(u[2:]-2*u[1:-1]+u[0:-2])
        """
        if i%10==0:
            plt.ylim(0.0,1.3)
            plt.yticks(np.arange(0.0,1.4,0.2))
            plt.plot(x,u); plt.draw(); plt.pause(0.01); plt.clf()
        """

    # find  diffusion time
        if np.sum(u)/sum0 < 1/np.e :
            diffusion_t = dt*i
            break
        i+=1

    #plt.show()
    return diffusion_t
```

In [4]:

```
#example
print(diffusion_eq(xmax,dx,dt,0.5,imax))
print(diffusion_eq(xmax,dx,dt,0.25,imax))
```

```
0.75
1.51
```

characteristic time of diffusion is inversely proportional to D. Compare the values with them

In [40]:

```
#measure
df = []
for i in np.arange(0.1,0.51,0.025) :
    df.append(diffusion_eq(xmax,dx,dt,i,imax))
```

In [42]:

```
plt.scatter(np.arange(0.1,0.51,0.025),df, c='r', label = 'measured')
thm=0.375/np.arange(0.1,0.51,0.01) #thm values
plt.plot(np.arange(0.1,0.51,0.01),thm,label = 'thm')
plt.legend()
plt.show()
```