Graficas taller 1

November 13, 2018

1 Métodos

```
In [2]: import numpy as np
        import matplotlib.pyplot as plt
        import scipy as sp
        %matplotlib inline
```

2 Numeral 1 del punto 4 del taller, graficas

2.1 Datos de los 4 tiempos característicos

```
In [3]: D = np.loadtxt('FEesfera.dat', unpack = True)
        Dt1 = np.loadtxt('FEesferat1.dat', unpack = True)
        Dt2 = np.loadtxt('FEesferat2.dat', unpack = True)
        Dt3 = np.loadtxt('FEesferat3.dat', unpack = True)
        Dt4 = np.loadtxt('FEesferat4.dat', unpack = True)

        np.shape(D)

Out[3]: (21, 10000)
```

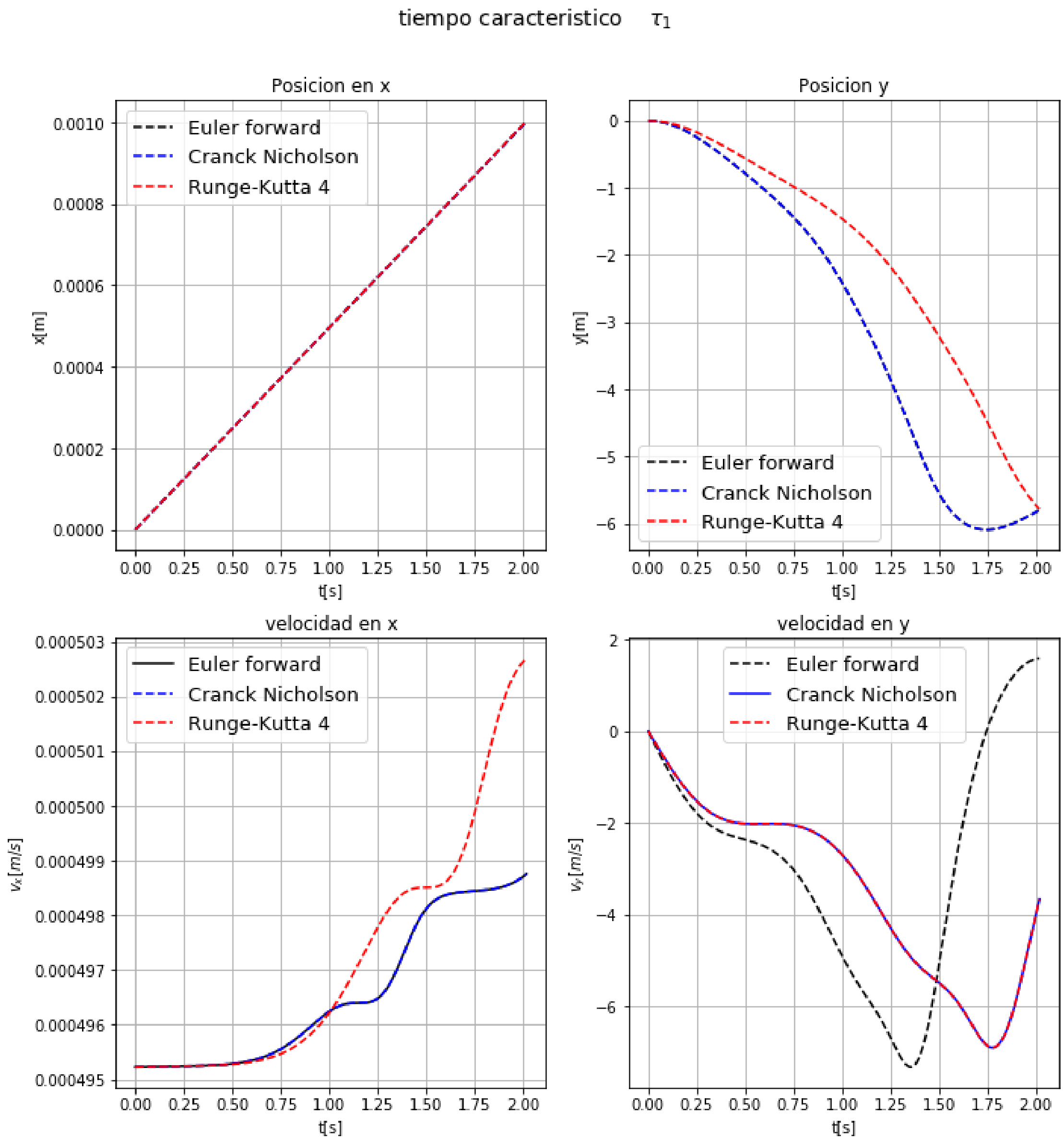
3 Funciones graficadoras

```
In [4]: def plot4(D,i):
            fig, axes = plt.subplots(nrows=2, ncols=2,figsize=(10,10))
            ax0, ax1, ax2, ax3 = axes.flatten()

            ax0.plot(D[0],D[1],"k--", label = "Euler forward")
            ax0.plot(D[0],D[3],"b--", label = "Cranck Nicholson")
            ax0.plot(D[0],D[5],"r--", label = "Runge-Kutta 4")
            ax0.legend(prop={'size': 13})
            ax0.grid(True)
            ax0.set_title('Posicion en x')
            ax0.set_xlabel("t[s]")
            ax0.set_ylabel("x[m]")

            ax1.plot(D[0],D[2],"k--", label = "Euler forward")
```

```
ax1.plot(D[0],D[4],"b--", label = "Cranck Nicholson")
ax1.plot(D[0],D[6],"r--", label = "Runge-Kutta 4")
ax1.legend(prop={'size': 13})
ax1.grid(True)
ax1.set_title('Posicion y')
ax1.set_xlabel("t[s]")
ax1.set_ylabel("y[m]")

ax2.plot(D[0],D[7],"k-", label = "Euler forward")
ax2.plot(D[0],D[9],"b--", label = "Cranck Nicholson")
ax2.plot(D[0],D[11],"r--", label = "Runge-Kutta 4")
ax2.legend(prop={'size': 13})
ax2.grid(True)
ax2.set_title('velocidad en x')
ax2.set_xlabel("t[s]")
ax2.set_ylabel("$v_{x}[m/s]$")

# Make a multiple-histogram of data-sets with different length.
ax3.plot(D[0],D[8],"k--", label = "Euler forward")
ax3.plot(D[0],D[10],"b-", label = "Cranck Nicholson")
ax3.plot(D[0],D[12],"r--", label = "Runge-Kutta 4")
ax3.legend(prop={'size': 13})
ax3.grid(True)
ax3.set_title('velocidad en y')
ax3.set_xlabel("t[s]")
ax3.set_ylabel("$v_{y}[m/s]$")


fig.tight_layout()
fig.suptitle("tiempo caracteristico    $\\tau_{"+str(i)+"}$",size=14, y=1.05)

plt.savefig("punto4_grafica"+str(i)+".eps",format = "eps", dpi =400)
plt.show()
```

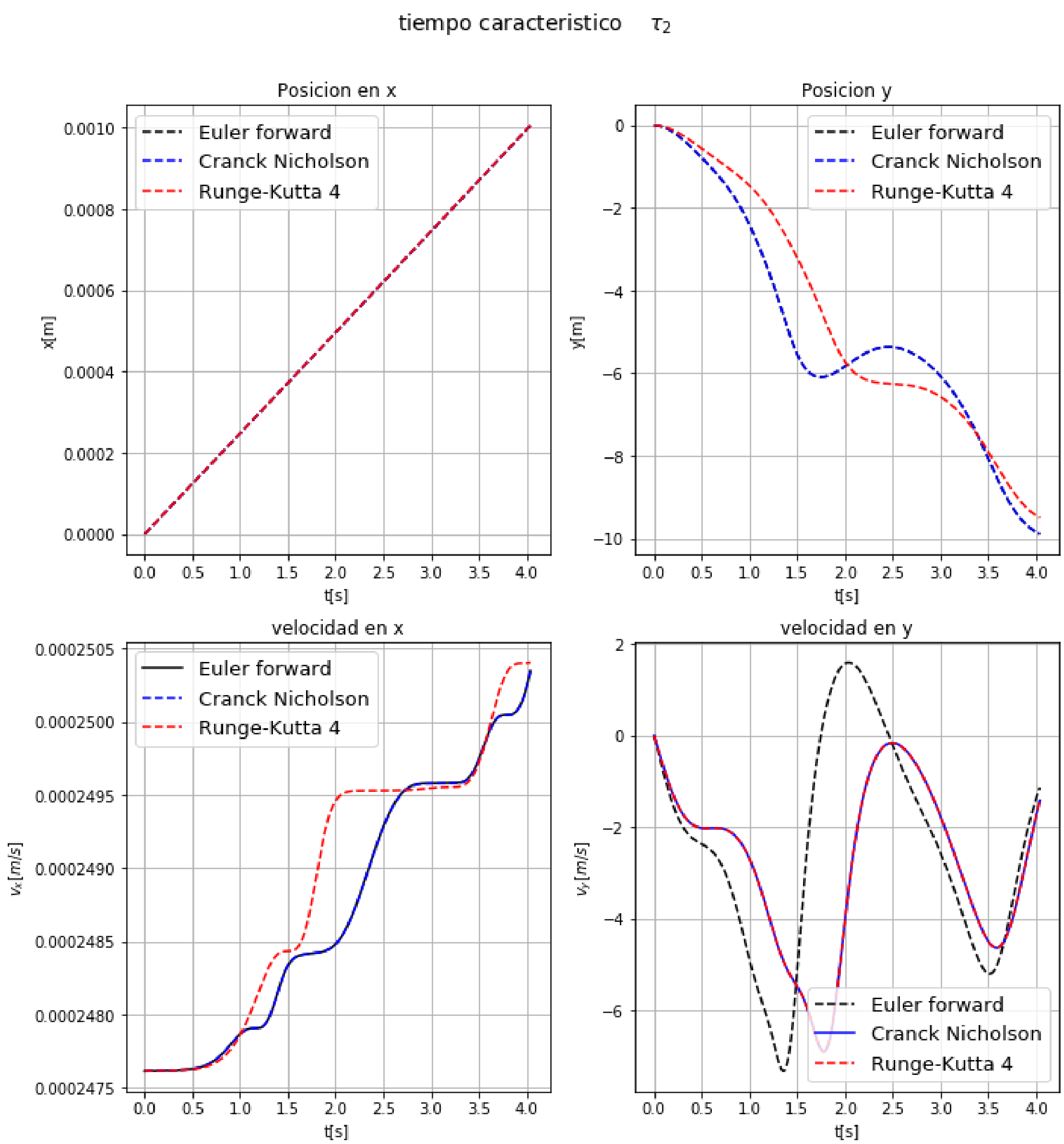
3.1 τ_1

```
In [5]: plot4(Dt1,1)
```



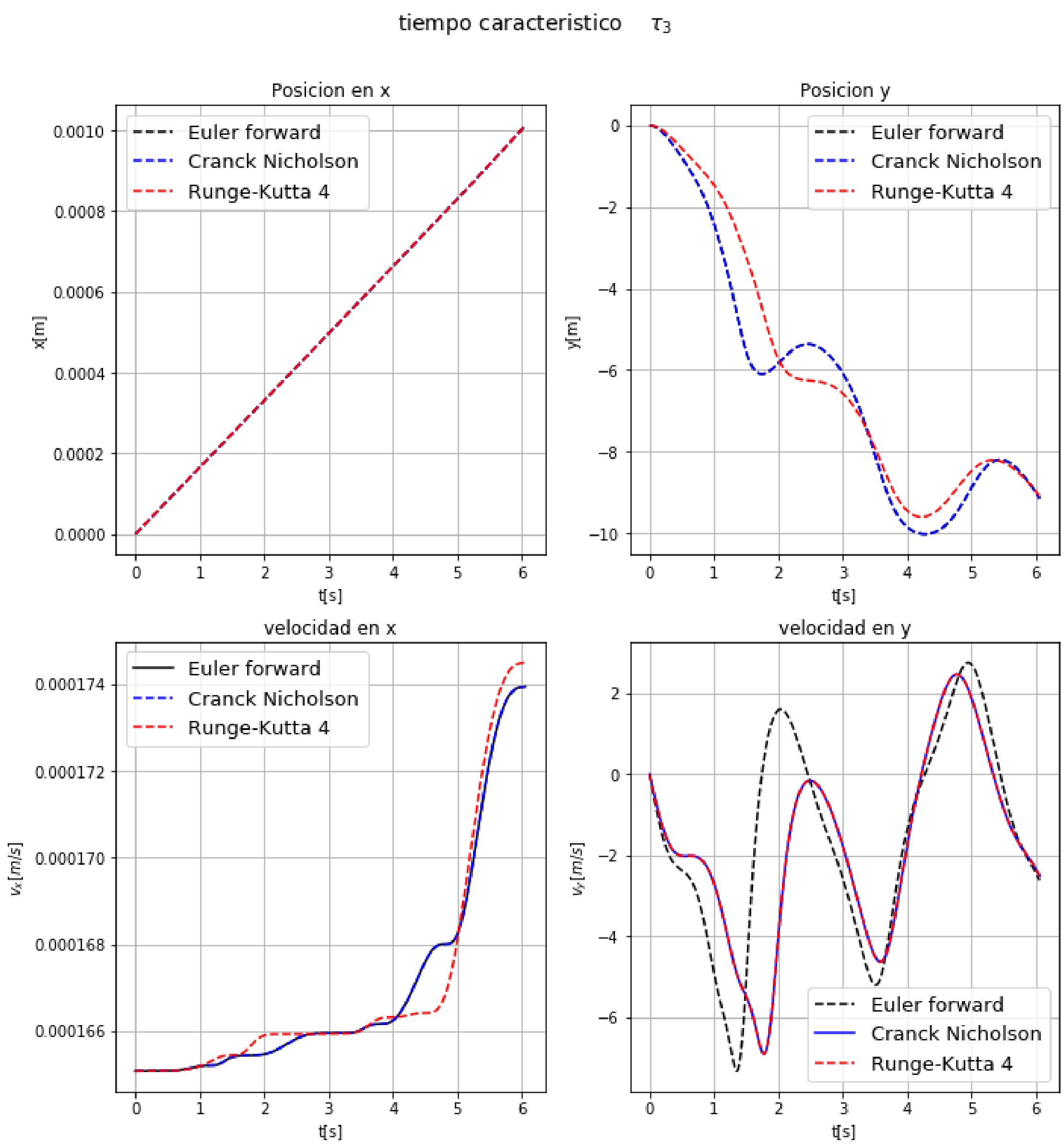
3.2 τ_2

```
In [6]: plot4(Dt2,2)
```



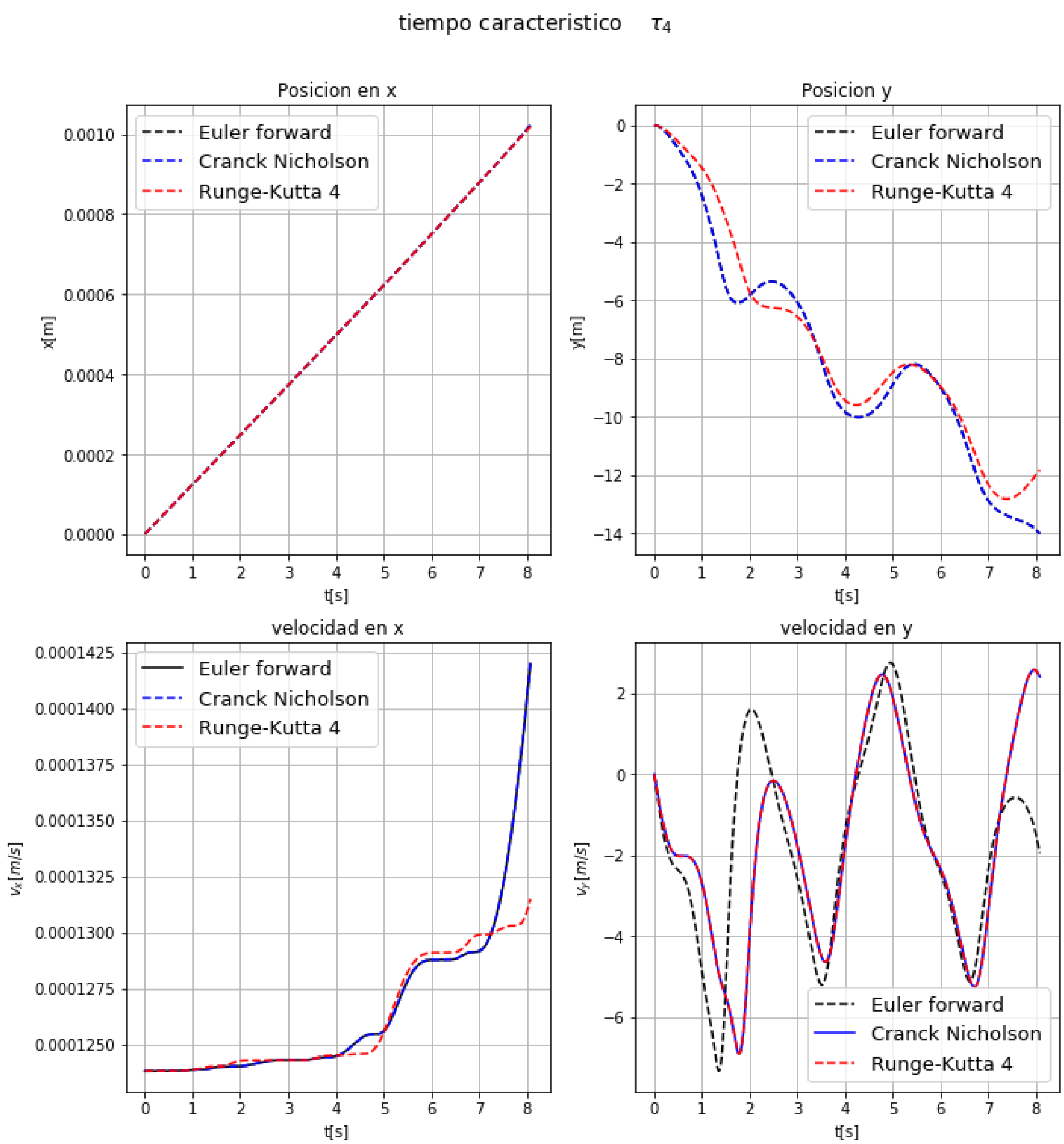
3.3 τ_3

```
In [7]: plot4(Dt3,3)
```



3.4 τ_4

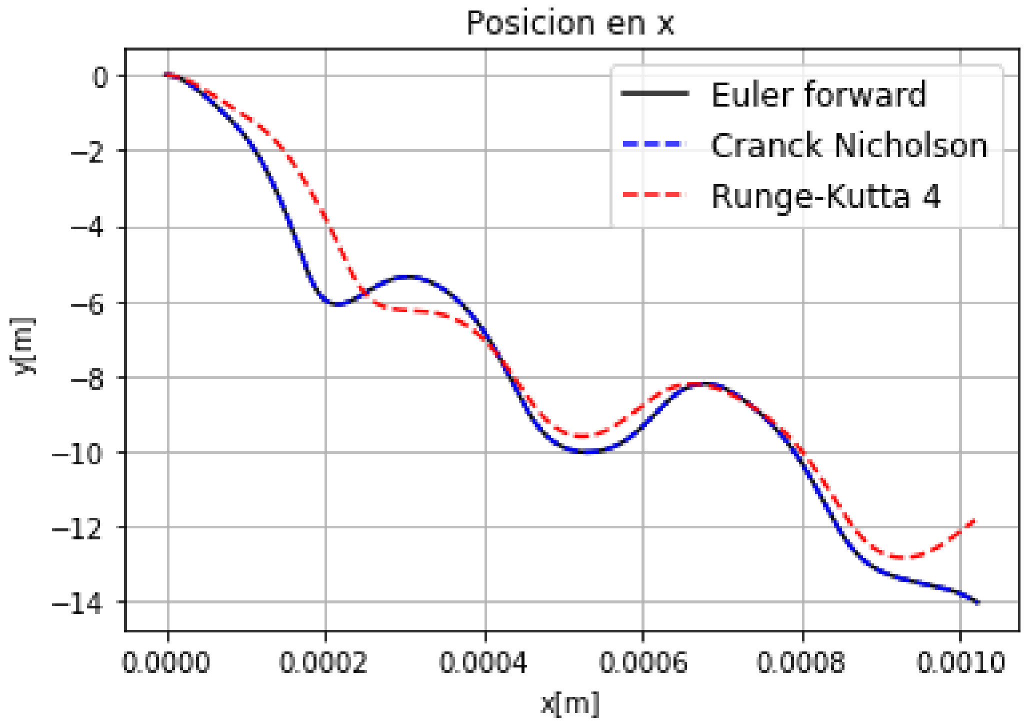
```
In [8]: plot4(Dt4,4)
```



4 Numeral 2 del punto 4 del taller, graficas

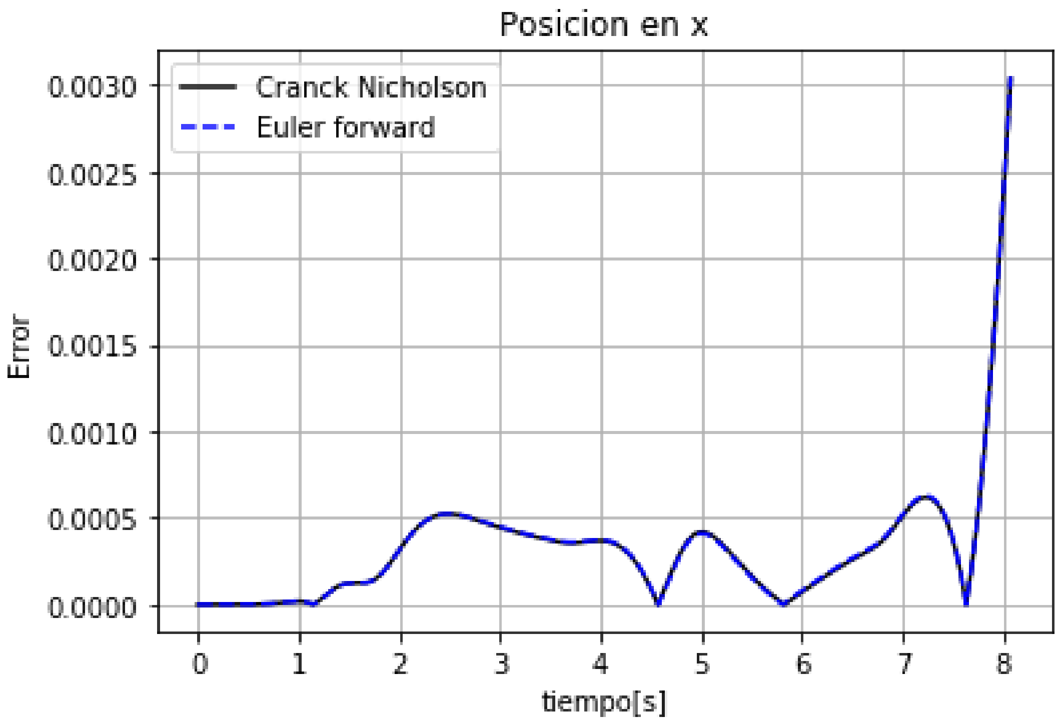
In [9]:
```
plt.plot(D[1],D[2],"k-", label = "Euler forward")
plt.plot(D[3],D[4],"b--", label = "Cranck Nicholson")
plt.plot(D[5],D[6],"r--", label = "Runge-Kutta 4")
plt.legend(prop={'size': 12})
plt.xlabel("x[m]")
plt.ylabel("y[m]")
plt.grid(True)
plt.title('Posicion en x')
```

```
plt.savefig("punto4_trayectorias.eps",format = "eps", dpi =400)
```



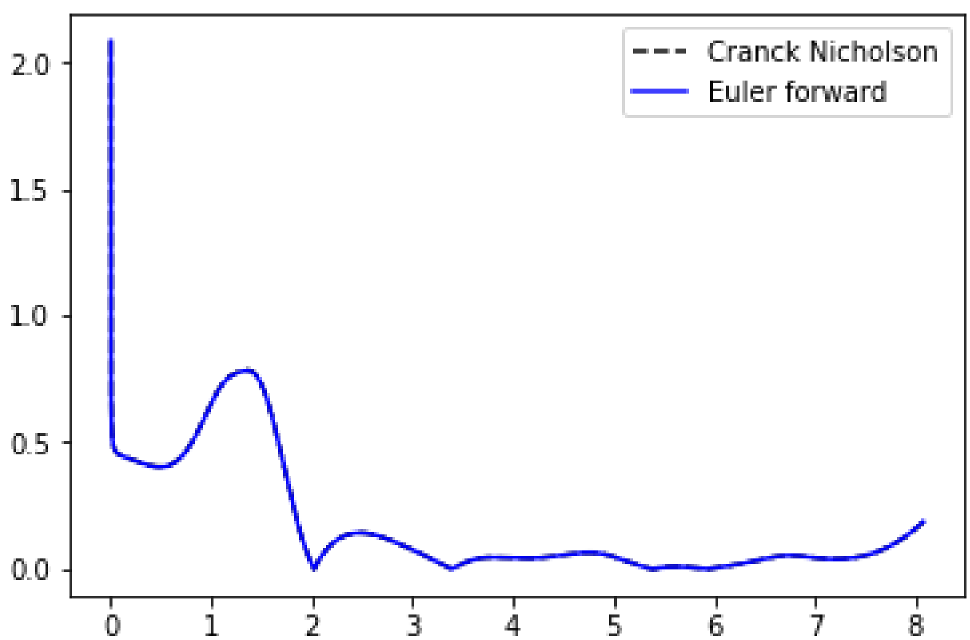
5 graficas de error

```
In [10]: plt.plot(D[0],D[13],"k-", label = "Cranck Nicholson")
         plt.plot(D[0],D[14],"b--", label = "Euler forward")
         plt.grid(True)
         plt.xlabel("tiempo[s]")
         plt.ylabel("Error")
         plt.title("Posicion en x")
         #plt.xlim(4.0,5.0)
         plt.legend()

Out[10]: <matplotlib.legend.Legend at 0x11d2a0e50>
```



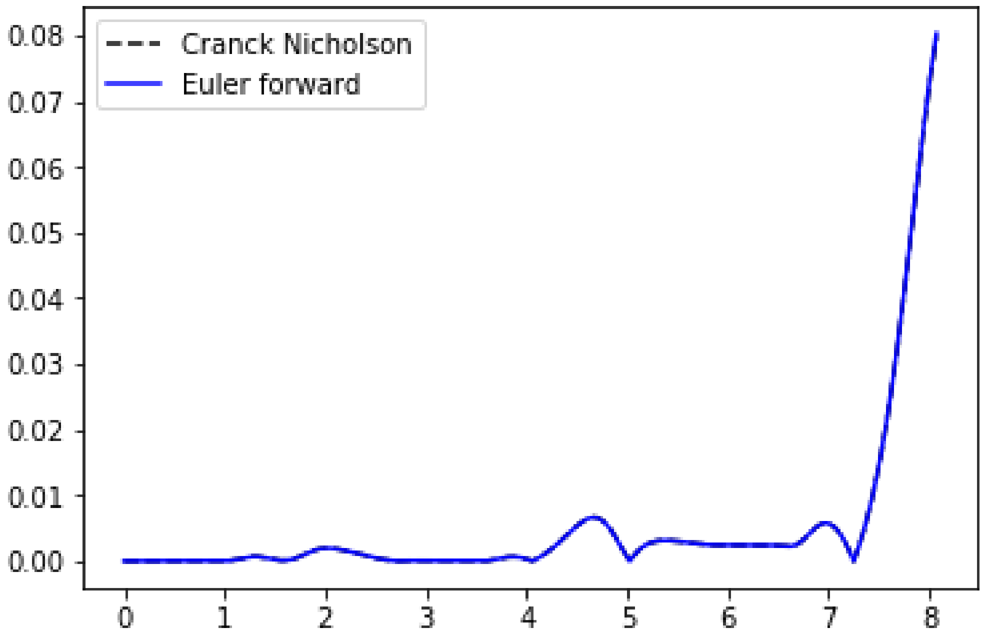

```
In [11]: plt.plot(D[0],D[15],"k--", label = "Cranck Nicholson")
         plt.plot(D[0],D[16],"b-", label = "Euler forward")
         plt.legend()
```

```
Out[11]: <matplotlib.legend.Legend at 0x11d0d1c90>
```



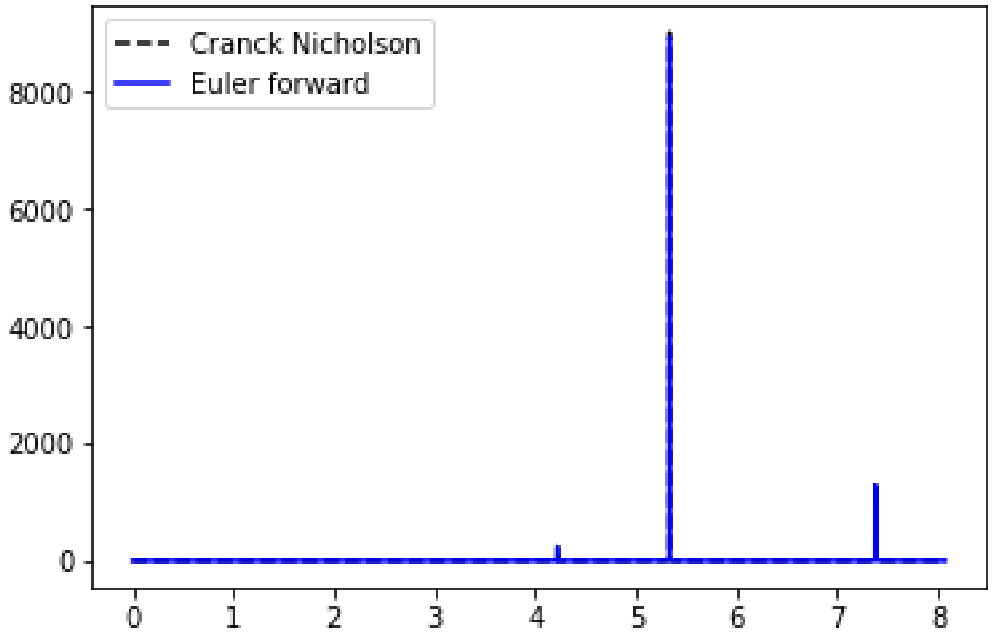
```
In [12]: plt.plot(D[0],D[17],"k--", label = "Cranck Nicholson")
         plt.plot(D[0],D[18],"b-", label = "Euler forward")
         plt.legend()
```

```
Out[12]: <matplotlib.legend.Legend at 0x11859ec10>
```



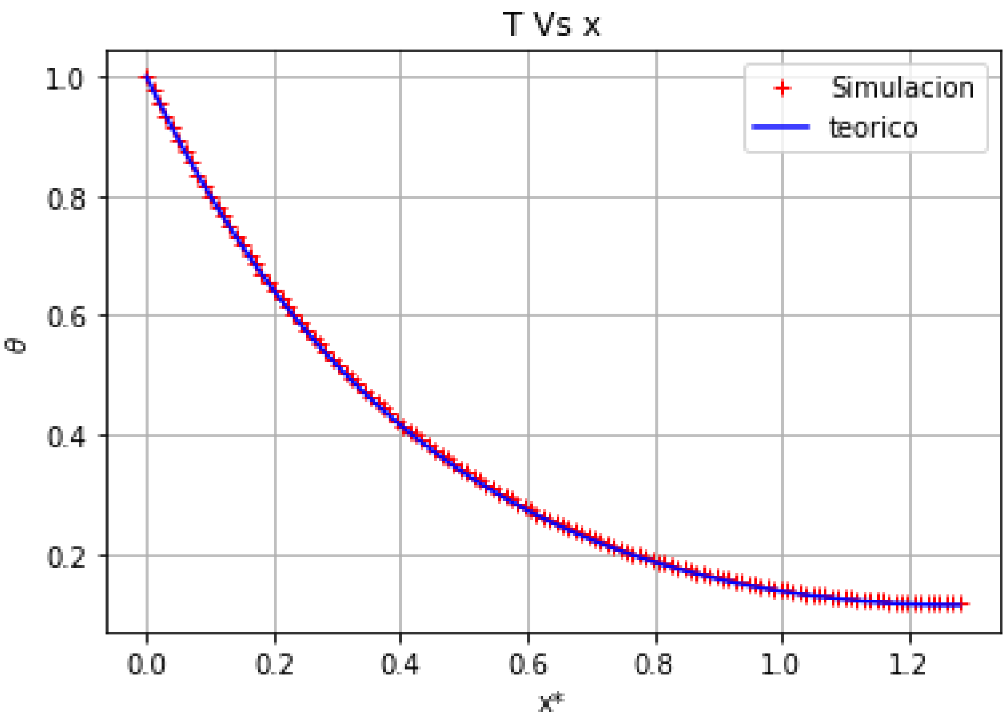

```
In [13]: plt.plot(D[0],D[19],"k--", label = "Cranck Nicholson")
         plt.plot(D[0],D[20],"b-", label = "Euler forward")
         plt.legend()
```

```
Out[13]: <matplotlib.legend.Legend at 0x11b17c6d0>
```



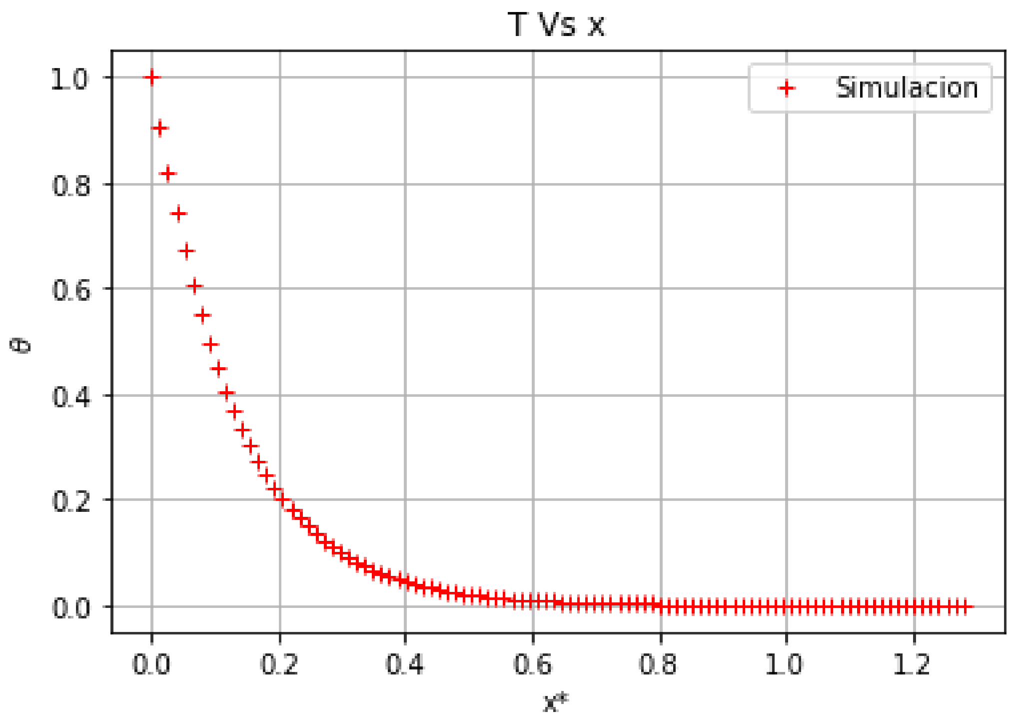
6 Problema de ecuación de difusión estacionario

```
In [14]: theta = np.loadtxt('theta.dat', unpack = True)
         dx = 0.01
         L = 2**7*dx
         l = 5.
         x = np.linspace(0,L,len(theta))
         plt.plot(x,theta,"r+",label = "Simulacion")
         plt.plot(x,np.cosh(np.sqrt(l)*(L-x))/np.cosh(np.sqrt(l)*L),"b-",label = "teorico")
         plt.title("T Vs x")
         plt.xlabel("x*")
         plt.ylabel("$\\theta$")
         plt.grid(True)
         plt.legend()

Out[14]: <matplotlib.legend.Legend at 0x11d2a0e10>
```

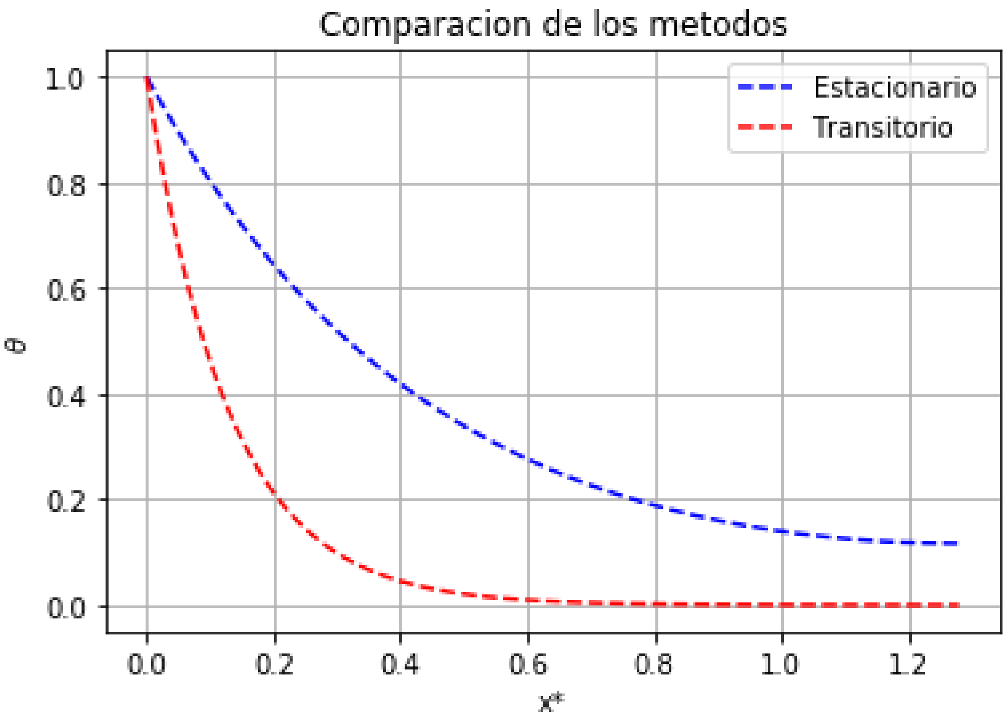


7 Ecuación de difusión transitorio

```
In [15]: thetat = np.loadtxt('theta_transitorio.dat', unpack = True)
         dx = 0.01
         L = 2**7*dx
         l = 5.
         xt = np.linspace(0,L,len(thetat))
         plt.plot(xt,thetat,"r+",label = "Simulacion")
         #plt.plot(x,np.cosh(np.sqrt(l)*(L-x))/np.cosh(np.sqrt(l)*L),"b-",label = "teorico")
         plt.title("T Vs x")
         plt.xlabel("x*")
         plt.ylabel("$\\theta$")
         plt.grid(True)
         plt.legend()

Out[15]: <matplotlib.legend.Legend at 0x110290290>
```

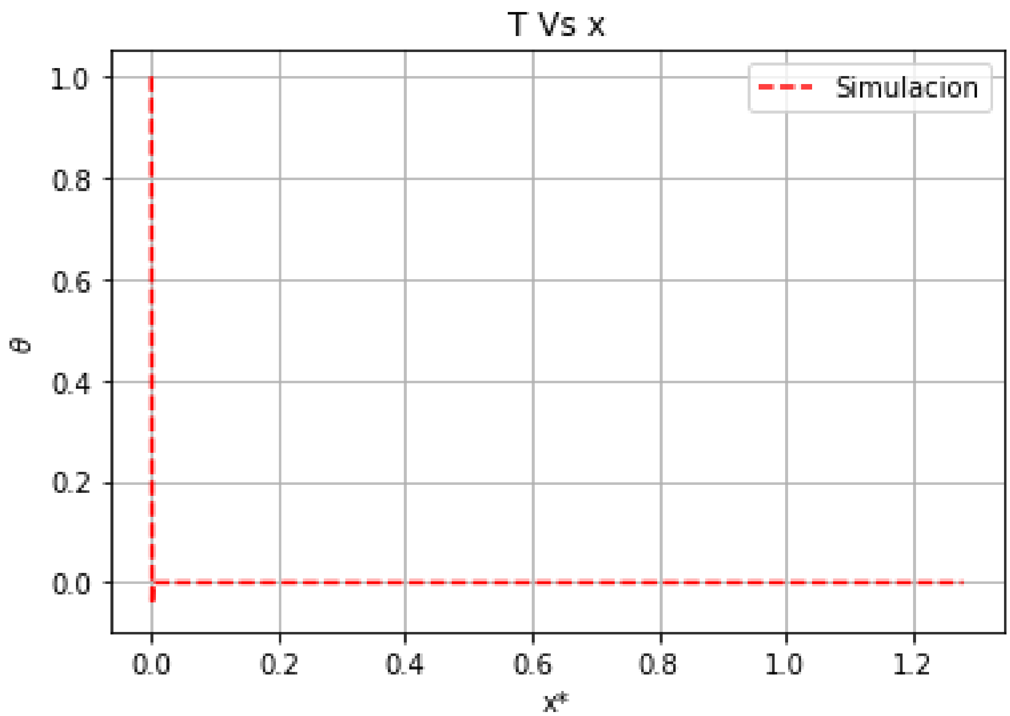



```
In [16]: plt.plot(x,theta,"b--",label = "Estacionario")
         plt.plot(xt,thetat,"r--",label = "Transitorio")
         plt.grid(True)
         plt.legend()
         plt.title("Comparacion de los metodos")
         plt.xlabel("x*")
         plt.ylabel("$\\theta$")

Out[16]: Text(0,0.5,u'$\\theta$')
```



In [17]:
```
T = np.loadtxt('CrackNicholson_transitoria.dat', unpack = True)
dx = 0.01
L = 2**7*dx
xT = np.linspace(0,L,len(T))
plt.plot(xT,T,"r--",label = "Simulacion")
plt.title("T Vs x")
plt.xlabel("x*")
plt.ylabel("$\\theta$")
plt.grid(True)
plt.legend()
```

Out[17]:
```
<matplotlib.legend.Legend at 0x11d09c390>
```



8 Ecuacion de difusión convección 1D

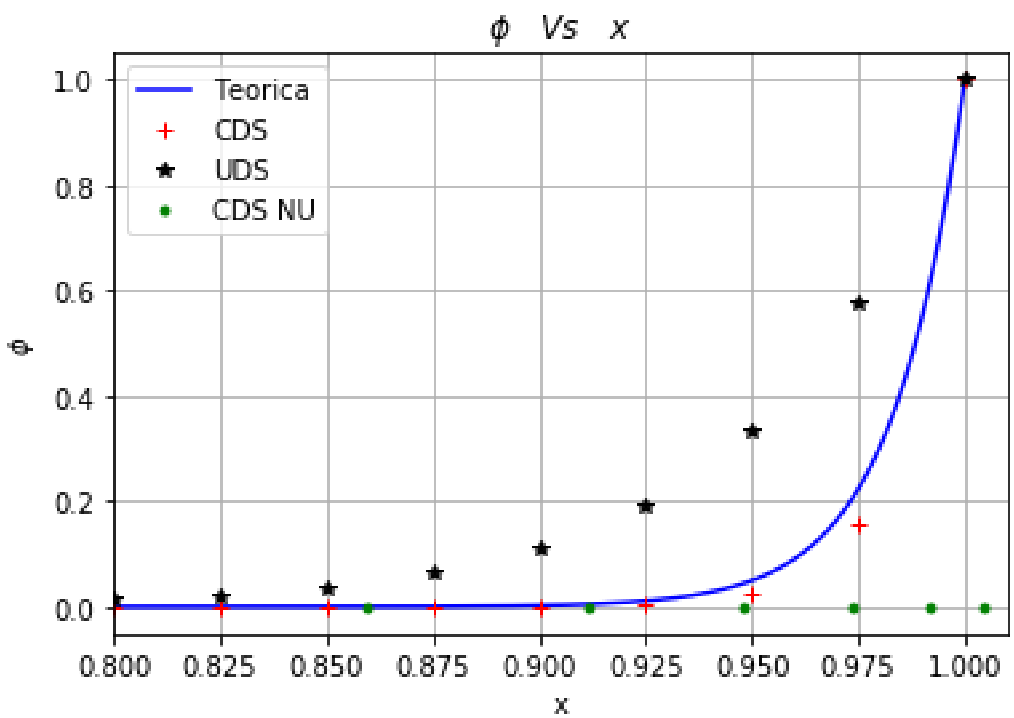
8.1 (Gráficas)

In [18]:
```
l=1.0; rho=1.0; u=1.0; gamma=0.01667
Pe = rho*u*l/gamma
print Pe
def phiT(x):
    return (np.exp(x*Pe/l)-1.0)/(np.exp(Pe)-1.0)


phi = np.loadtxt('DC_CDS.dat', unpack = True)
phiuds = np.loadtxt('DC_UDS.dat', unpack = True)
phicds = np.loadtxt('CDS_NU.dat', unpack = True)

xuds = np.linspace(0,l,len(phiuds))
xcds = np.linspace(0,l,len(phicds))
x = np.linspace(0,1.0,1000)

phix = np.linspace(0,l,len(phi))
plt.plot(x,phiT(x),"b-",label = "Teorica")
plt.plot(phix,phi,"r+",label = "CDS")
plt.plot(xuds,phiuds,"k*",label = "UDS")
```

```
plt.plot(phicds[:,0],phicds[:,1],"g.",label = "CDS NU")

plt.title("$\\phi \quad Vs \quad  x$")
plt.xlabel("x")




plt.ylabel("$\\phi$")
plt.xlim(0.8,1.01)
plt.grid(True)
plt.legend()
#plt.savefig("DC1D.png")
```

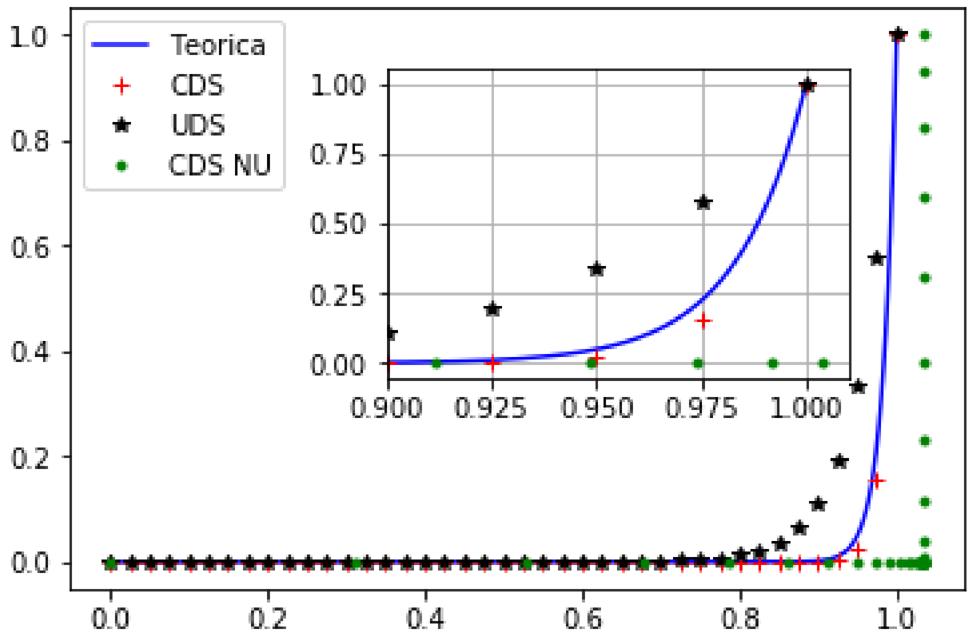
```
59.9880023995
```

Out[18]: <matplotlib.legend.Legend at 0x118382390>




```
In [19]: import matplotlib.pyplot as plt

         fig, ax1 = plt.subplots()

         # These are in unitless percentages of the figure size. (0,0 is bottom left)
```

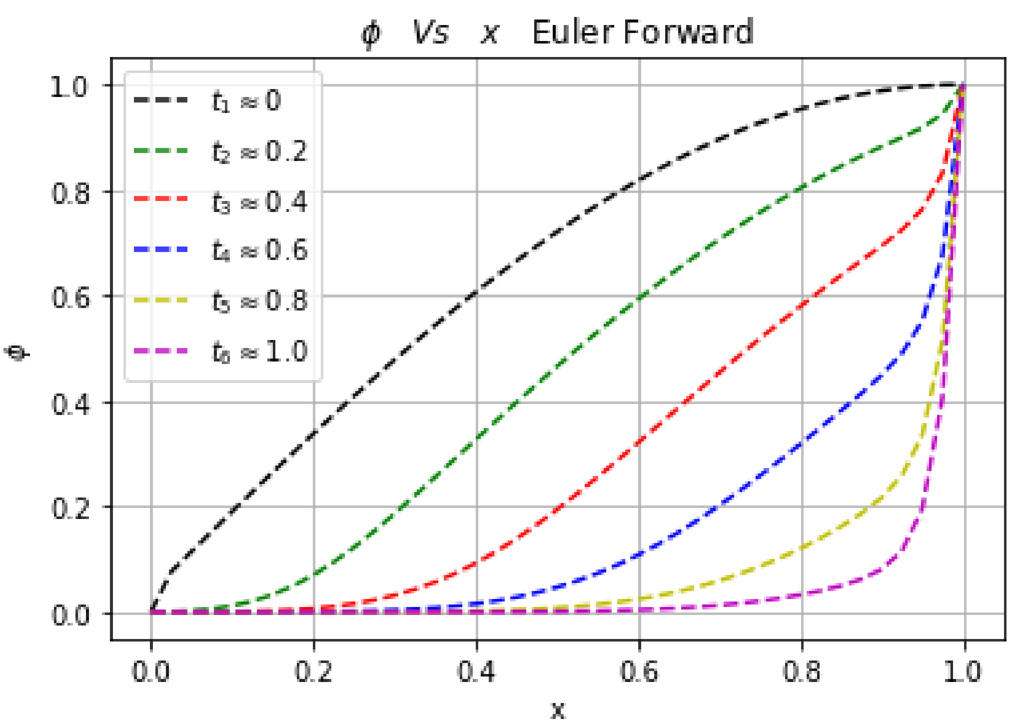
```
left, bottom, width, height = [0.4, 0.4, 0.4, 0.4]
ax2 = fig.add_axes([left, bottom, width, height])


ax1.plot(x,phiT(x),"b-",label = "Teorica")
ax1.plot(phix,phi,"r+",label = "CDS")
ax1.plot(xuds,phiuds,"k*",label = "UDS")
ax1.plot(phicds[:,0],phicds[:,1],"g.",label = "CDS NU")
plt.grid(True)

ax2.plot(x,phiT(x),"b-",label = "Teorica")
ax2.plot(phix,phi,"r+",label = "CDS")
ax2.plot(xuds,phiuds,"k*",label = "UDS")
ax2.plot(phicds[:,0],phicds[:,1],"g.",label = "CDS NU")
ax2.set_xlim(0.9,1.01)

ax1.legend()
plt.grid(True)
plt.savefig("DC1D.eps",format = "eps", dpi =400)
plt.show()
```



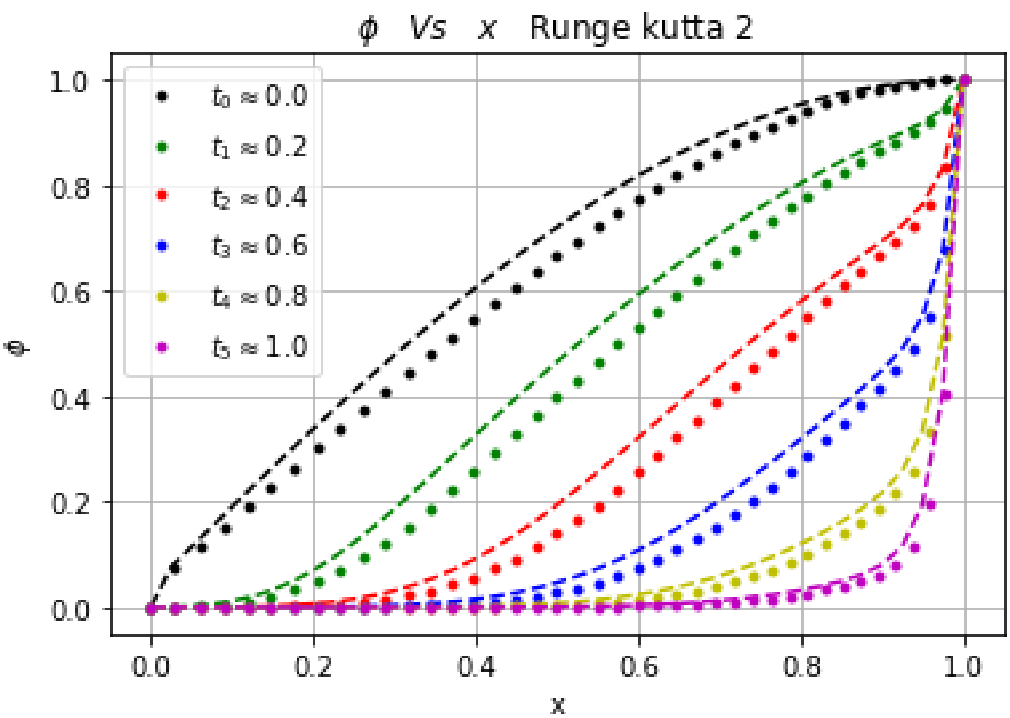
9 Difusion Convección Transitoria

9.1 Euler Forward

In [20]:
```
phiT = np.loadtxt('DCT_CDS.dat', unpack = True)
T = np.loadtxt('DCT_CDS_t.dat', unpack = True)

xT = np.linspace(0,1,len(phiT))
plt.plot(xT,phiT[:,0],"k--", label = "$t_{1} \\approx 0 $")
plt.plot(xT,phiT[:,1],"g--",label = "$t_{2} \\approx 0.2$")
plt.plot(xT,phiT[:,2],"r--",label = "$t_{3} \\approx 0.4$")
plt.plot(xT,phiT[:,3],"b--",label = "$t_{4} \\approx 0.6$")
plt.plot(xT,phiT[:,4],"y--",label = "$t_{5} \\approx 0.8$")
plt.plot(xT,phiT[:,5],"m--",label = "$t_{6} \\approx 1.0$")


plt.title("$\\phi\quad Vs \quadx $   Euler Forward")
plt.ylabel("$\\phi$")
plt.legend()
plt.grid(True)
plt.xlabel("x")
plt.savefig("punto3_euler.eps",format = "eps", dpi =400)
#plt.xlim(0.8,1.1)
```



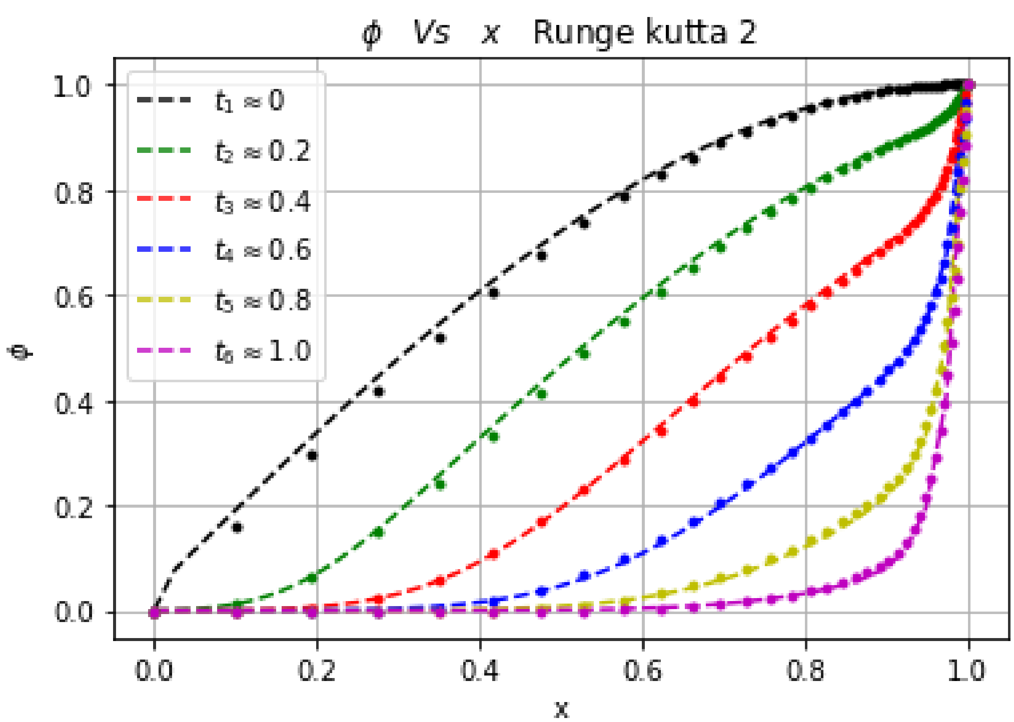
```
In [21]: phiRK2NU = np.loadtxt('DCT_CDS_NU.dat', unpack = True)
         phiT = np.loadtxt('DCT_CDS.dat', unpack = True)
         xRK2NU = np.loadtxt('malla_CDS.dat', unpack = True)
         xT = np.linspace(0,1,len(phiT))

         markers1 = ["k.","g.","r.","b.","y.","m."]
         markers2 = ["k--","g--","r--","b--","y--","m--"]
         for i in range(np.shape(phiRK2NU)[1]):
             plt.plot(xRK2NU,phiRK2NU[:,i],markers1[i], label = "$t_{"+str(i)+"} \\approx "+st
             plt.plot(xT,phiT[:,i],markers2[i])


         plt.title("$\\phi\quad Vs \quadx $   Runge kutta 2 ")
         plt.ylabel("$\\phi$")
         plt.legend()
         plt.grid(True)
         plt.xlabel("x")
         #plt.xlim(0.0,0.1)
         plt.savefig("punto3_euler.eps",format = "eps", dpi =400)
```



9.2 Runge kutta

```
In [24]: phiRK = np.loadtxt('DCT_RK2.dat', unpack = True)
         xRK = np.linspace(0,1,len(phiRK))
```

```
plt.plot(xRK,phiRK[:,0],"k--", label = "$t_{1} \\approx 0 $")
plt.plot(xRK,phiRK[:,1],"g--", label = "$t_{2} \\approx 0.2 $")
plt.plot(xRK,phiRK[:,2],"r--", label = "$t_{3} \\approx 0.4 $")
plt.plot(xRK,phiRK[:,3],"b--", label = "$t_{4} \\approx 0.6 $")
plt.plot(xRK,phiRK[:,4],"y--", label = "$t_{5} \\approx 0.8 $")
plt.plot(xRK,phiRK[:,5],"m--", label = "$t_{6} \\approx 1.0 $")
plt.title("$\\phi\quad Vs \quadx $   Runge kutta 2 ")
plt.ylabel("$\\phi$")
plt.legend()
plt.grid(True)
plt.xlabel("x")
#plt.xlim(0.0,0.1)


plt.plot(xRKNU,phiRKNU[:,0],"k.", label = "$t_{1} \\approx 0 $")
plt.plot(xRKNU,phiRKNU[:,1],"g.", label = "$t_{2} \\approx 0.2 $")
plt.plot(xRKNU,phiRKNU[:,2],"r.", label = "$t_{3} \\approx 0.4 $")
plt.plot(xRKNU,phiRKNU[:,3],"b.", label = "$t_{4} \\approx 0.6 $")
plt.plot(xRKNU,phiRKNU[:,4],"y.", label = "$t_{5} \\approx 0.8 $")
plt.plot(xRKNU,phiRKNU[:,5],"m.", label = "$t_{6} \\approx 1.0 $")



plt.savefig("punto3_RK2.eps",format = "eps", dpi =400)
```



In [23]:
```
phiRKNU = np.loadtxt('DCT_RK2_NU.dat', unpack = True)
xRKNU = np.loadtxt('malla_RK2.dat', unpack = True)

plt.plot(xRKNU,phiRKNU[:,0],"k.", label = "$t_{1} \\approx 0 $")
plt.plot(xRKNU,phiRKNU[:,1],"g.", label = "$t_{2} \\approx 0.2 $")
plt.plot(xRKNU,phiRKNU[:,2],"r.", label = "$t_{3} \\approx 0.4 $")
plt.plot(xRKNU,phiRKNU[:,3],"b.", label = "$t_{4} \\approx 0.6 $")
plt.plot(xRKNU,phiRKNU[:,4],"y.", label = "$t_{5} \\approx 0.8 $")
plt.plot(xRKNU,phiRKNU[:,5],"m.", label = "$t_{6} \\approx 1.0 $")
plt.title("$\\phi\quad Vs \quadx $   Runge kutta 2 ")
plt.ylabel("$\\phi$")
plt.legend()
plt.grid(True)
plt.xlabel("x")
#plt.xlim(0.0,1)
```

Out[23]: Text(0.5,0,u'x')