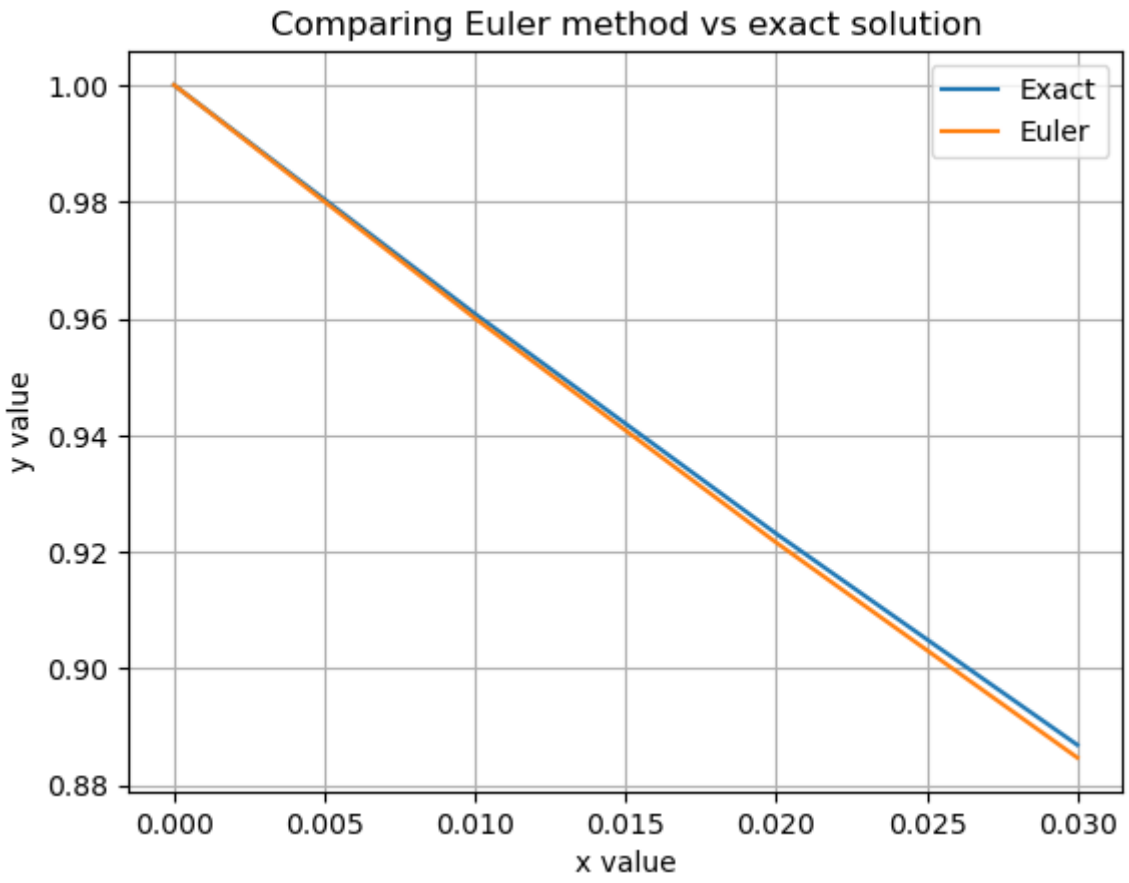
In [4]:

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


# Here will will do example 7.1 page 250, Chapter 7 in the book
# of Kiusalaas J. Numerical methods in Engineering with Pyhton 3


# The first order differential equation is dy/dx+4y=x^2, y(0)= 1.
# dy/dx = f(x,y) where f(x,y)=x^2-4y
# We will let the spacing h = 0.01 and we will step from x = 0 to 0.03
# using the Euler methods and study
# the truncation error and the accumulated error when comparing with the exact
# solution.


h = 0.01
xstart = 0
xend =0.03


X=[]
Yexact=[]
Yeuler=[]

# Save the initial condition y(0)=1 in the arrays

x = xstart
yexact = 1
yeuler = 1

X.append(x)
Yexact.append(yexact)
Yeuler.append(yeuler)

while (x<xend):

    h = min(h,xend-x)  # In case we have to reduce the timestep in the end to
                       # fit the simulation time interval.

    yeuler =yeuler+h*(x**2-4*yeuler)  # Note this is an efficient way
                                      # to write ynew = yold + h*f(xold,yold)

    # update x and also find exact solution.
    x = x+h
    yexact= 31/32*math.exp(-4*x)+1/4*x**2-1/8*x+1/32

    erroraccumulated = yeuler-yexact

    print(f'x verdi: {x}')
    print(f'Euler: {yeuler}')
    print(f'Exact: {yexact}')
    print(f'Accumulated error: {erroraccumulated}')

    X.append(x)
    Yexact.append(yexact)
    Yeuler.append(yeuler)

print('-------------------------------------------------------')
print('We observe that the accummulated error increases with approx 0.0007'
 ' for each step when we simulated from 0 to 0.03 s with steps of 0.01 s')
```

```
plt.plot(X,Yexact,X,Yeuler)
plt.title('Comparing Euler method vs exact solution')
plt.grid(True)
plt.xlabel('x value')
plt.ylabel('y value')
plt.legend(['Exact', 'Euler'])
plt.show()


print(f'Change xend to 0.5. Try first with h = 0.01. Then h = 0.1. What is \n'
      'the impact of increasing h ?')
```

```
x verdi: 0.01
Euler: 0.96
Exact: 0.9607897691788132
Accumulated error: -0.0007897691788132377
x verdi: 0.02
Euler: 0.921601
Exact: 0.9231189605620534
Accumulated error: -0.0015179605620534442
x verdi: 0.03
Euler: 0.88474096
Exact: 0.8869291730697463
Accumulated error: -0.0021882130697462587
------------------------------------------------------------
We observe that the accummulated error increases with approx 0.0007 for each step
when we simulated from 0 to 0.03 s with steps of 0.01 s
```




```
Change xend to 0.5. Try first with h = 0.01. Then h = 0.1. What is
the impact of increasing h ?
```

In [ ]: