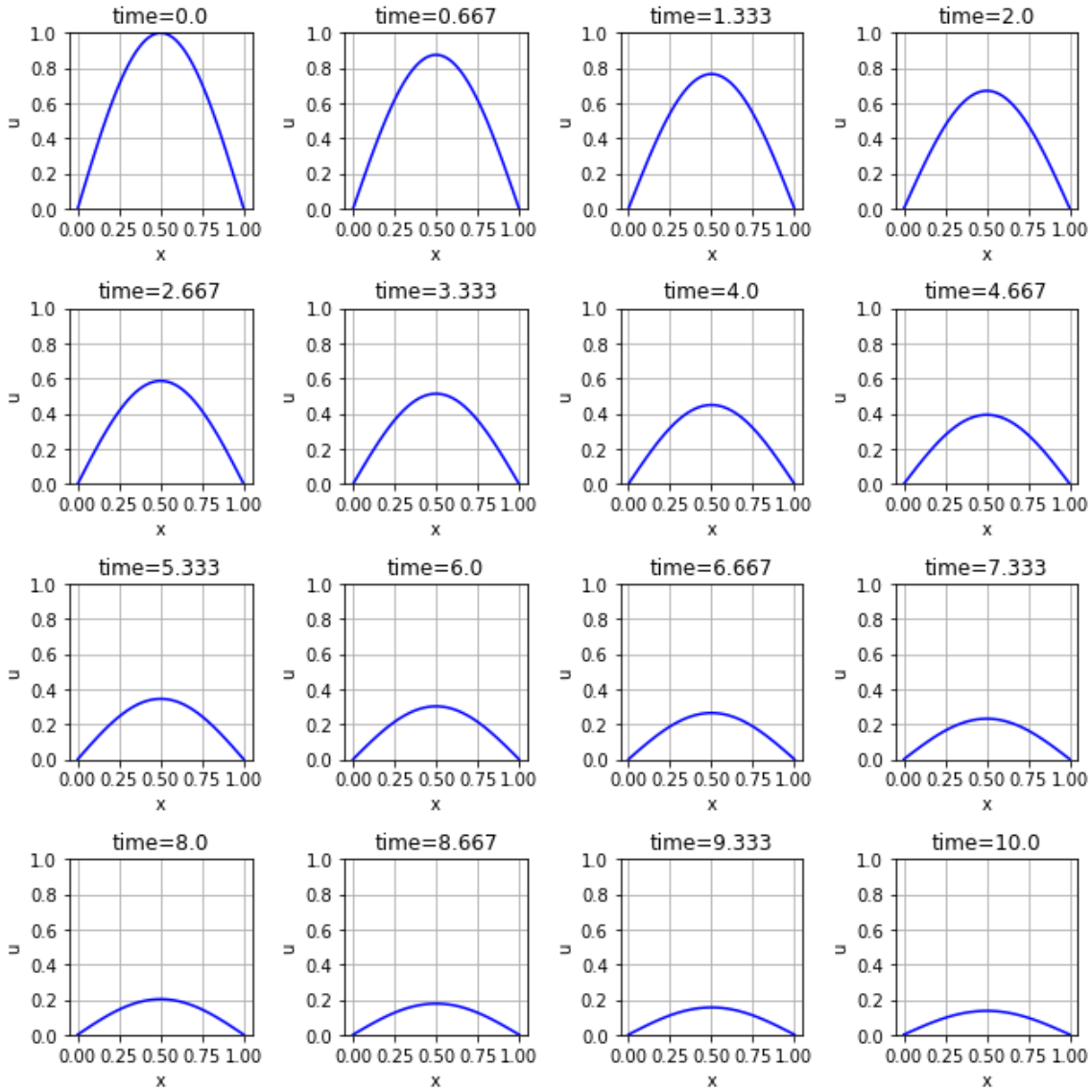
6.10

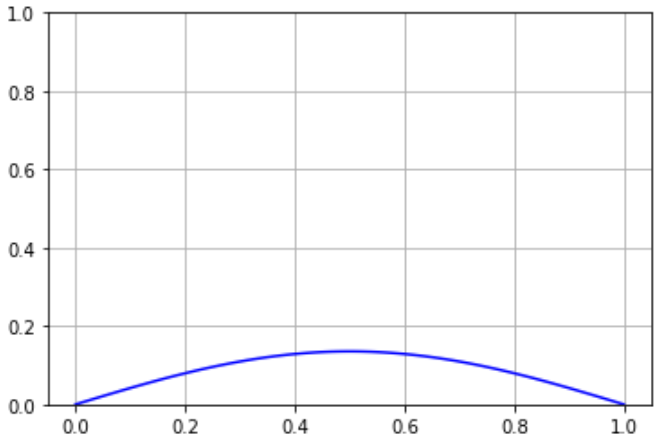
Alice Gee, ag67642 Andrew Yang, ay6764 Mohammad Aga, mba929

In [19]:
```
#a
import numpy as np
import matplotlib.pyplot as plt
u = lambda t, x: np.exp(-0.2*t) * np.sin(np.pi*x)
x = np.linspace(0,1, 100)
t = np.linspace(0, 10, 16)
fig, ax = plt.subplots(nrows=4,ncols=4, figsize = (9,9))
counter = 0 # this counter will count through the times
for n in range(4):
    for m in range(4):
        ax[n,m].plot(x , u(t[counter],x), 'b') # plot x vs u(t[counter],x)
        ax[n,m].grid()
        ax[n,m].set_ylim(0,1) # same axis for every plot
        ax[n,m].set_xlabel('x')
        ax[n,m].set_ylabel('u')
        ax[n,m].set_title("time="+str(round(t[counter], 3)))
        ax[n,m].set_xticks([0,0.25,0.5,0.75, 1])
        counter += 1 # increment the counter
fig.tight_layout()
plt.show()
```



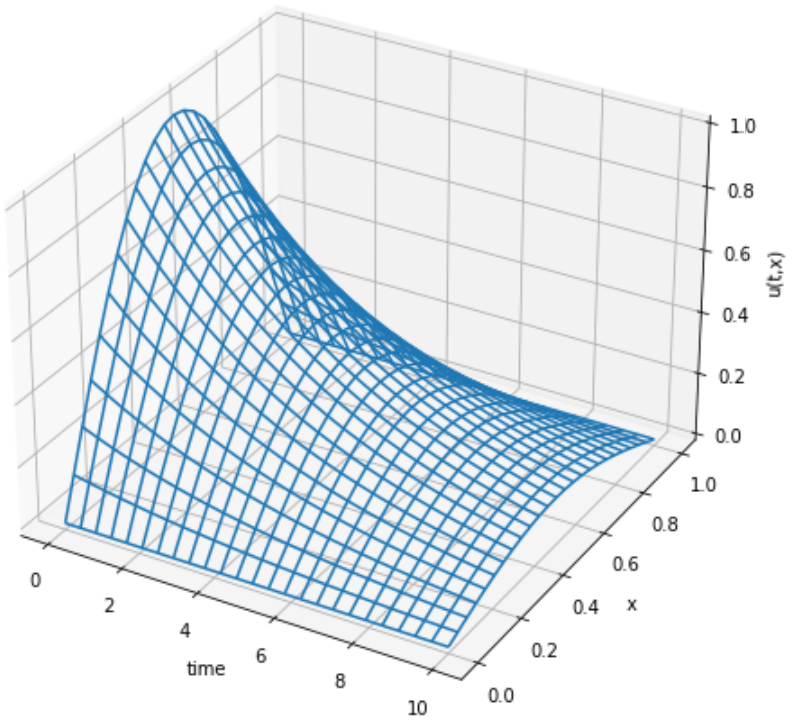
In [21]:
```
#b
import numpy as np
import matplotlib.pyplot as plt
from ipywidgets import interactive

# our function
u = lambda t, x: np.exp(-0.2*t) * (1*np.sin(1*np.pi*x))

# x-axis values to put into `u` function
x = np.linspace(0,1,100)

# plotting function that maps out u(x) for each x value generated from np.linspace()
def plotter(T):
  plt.plot(x , u(T,x), 'b')
  plt.grid()
  plt.ylim(0,1)
  plt.show()

# make slider animation
interactive_plot = interactive(plotter, T=(0,20,0.1))
interactive_plot
```

T 10.00



In [23]:
```
#c
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# set figure size
fig = plt.figure(figsize=(10,8))

# get current axis
ax = fig.gca(projection='3d')

# function `u`
u = lambda t, x: np.exp(-0.2*t)*np.sin(np.pi*x)

# x values to put into `u` function
x = np.linspace(0,1,25)

# t values to put into `u` function
t = np.linspace(0,10,25)

# Return coordinate matrices from coordinate vectors (points for the wireframe functi
T, X = np.meshgrid(t,x)

# plot the 3d wireframe model
ax.plot_wireframe(T,X,u(T,X))

# labels
ax.set_xlabel('time')
ax.set_ylabel('x')
ax.set_zlabel('u(t,x)')
plt.show()
```

```
/var/folders/sx/gw6n5mnj28x21kcmksnp_p240000gn/T/ipykernel_25934/1472606173.py:10:
MatplotlibDeprecationWarning: Calling gca() with keyword arguments was deprecated i
n Matplotlib 3.4. Starting two minor releases later, gca() will take no keyword arg
uments. The gca() function should only be used to get the current axes, or if no ax
es exist, create new axes with default keyword arguments. To create a new axes with
non-default arguments, use plt.axes() or plt.subplot().
  ax = fig.gca(projection='3d')
```


1.0
0.8
0.6
0.4
0.2
0.0
u(t,x)
0
2
4
6
8
10
time
0.0
0.2
0.4
0.6
0.8
1.0
x

In [24]:
```
#d
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import animation, rc
from IPython.display import HTML

# function `u`
u = lambda t, x: np.exp(-0.2*t)*np.sin(np.pi*x)

# x values to put into `u` function
x = np.linspace(0,1,25)

# t values to put into `u` function
t = np.linspace(0,10,101)

# establish the subplot (and parameters)
fig, ax = plt.subplots()
plt.close()
ax.grid()
ax.set_xlabel('x')
ax.set_xlim(( 0, 1))
ax.set_ylim(( 0, 1))
frame, = ax.plot([], [], linewidth=2, linestyle='--')

# creates each frame by calling the `u` function
def animator(N):
  U = u(t[N],x)
  ax.set_title('Time='+str(t[N]))
  frame.set_data(x,U)
  return (frame,)

# animates the function `u`
PlotFrames = range(0,len(t),1)
anim = animation.FuncAnimation(fig,
                               animator,
                               frames=PlotFrames,
                               interval=100,
                              )
rc('animation', html='jshtml') # embed in the HTML for Google Colab
anim
```

Out[24]: