# Solution for Problem DN

Source Filename: /solution.py

Rico A. R. Picone

```
"""Solution to Chapter 3 problem DN"""
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
import sys

sys.path.append("../")
import engcom
```

## Introduction

This program defines several mathematical functions as vectorized functions that can handle NumPy array inputs and plots them over the given domain using Matplotlib.

## Define Mathematical Functions

Define $f(x) = x^2 + 3x + 9$:

```
def f(x: np.ndarray) -> np.ndarray:
    return np.tanh(4 * np.sin(x))
```

Define $g(x) = 1 + \sin^2 x$:

```
def g(x: np.ndarray) -> np.ndarray:
    return np.sin(np.sqrt(x))
```

Define $h(x, y) = e^{-3x} + \ln y$:

```
def h(x: np.ndarray) -> np.ndarray:
    return np.where(x >= 0, np.exp(-x) * np.sin(2 * np.pi * x), 0)
```

## Plotting

Define a plotting function:

```
def plotter(fig, fun, limits, labels):
    x = np.linspace(limits[0], limits[1], 201)
    fig.gca().plot(x, fun(x))
    fig.gca().set_xlabel(labels[0])
    fig.gca().set_ylabel(labels[1])
    return fig
```

Plot $f(x)$:

```
fig, ax = plt.subplots()
plotter(fig, fun=f, limits=(-5, 8), labels=("$x$", "$f(x)$"))
```

```
engcom.show(fig)
```

```
<IPython.core.display.Markdown object>
```

Plot $g(x)$:

```
fig, ax = plt.subplots()
plotter(fig, fun=g, limits=(0, 100), labels=("$x$", "$g(x)$"))
```

```
engcom.show(fig)
```

```
<IPython.core.display.Markdown object>
```

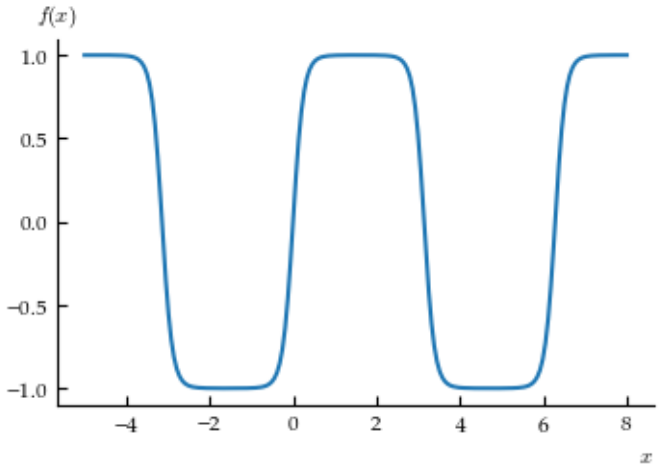
Plot $h(x)$:

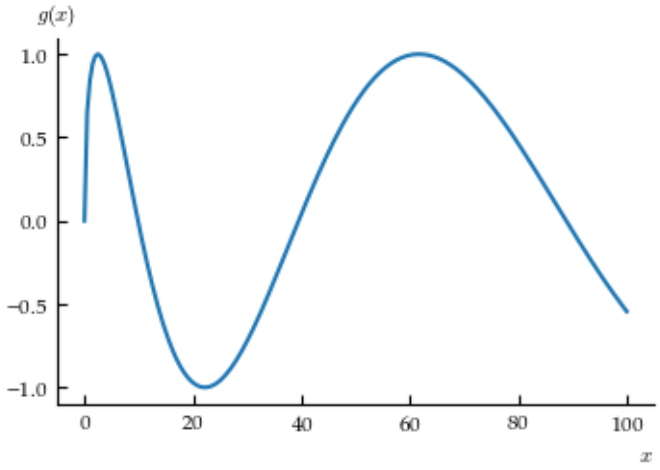
```
fig, ax = plt.subplots()
plotter(fig, fun=h, limits=(-2, 6), labels=("$x$", "$h(x)$"))
```

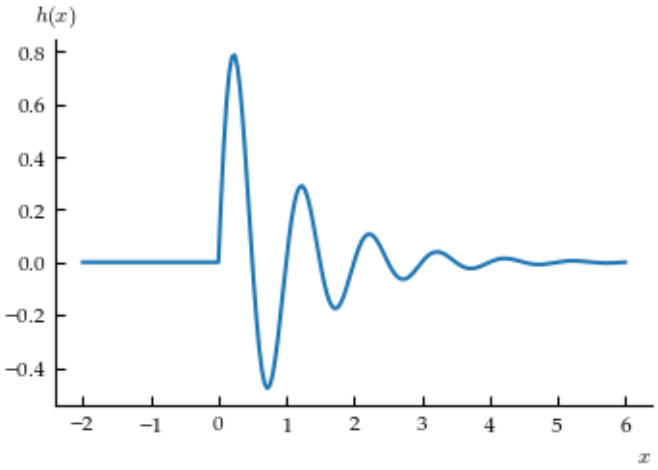
```
engcom.show(fig)
```

```
<IPython.core.display.Markdown object>
```

```
plt.show()
```


f(x)
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4
6
8
x


g(x)
1.0
0.5
0.0
−0.5
−1.0
0
20
40
60
80
100
x


h(x)
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−2
−1
0
1
2
3
4
5
6
x