**1.** Consider the polynomial of

$$f(x) = -2x^5 + 3x^2 + 2x + 1$$

Use `numpy.polyval()` to plot this curve for the range of $1 < x < 3$.

**2.** Consider the function of

$$f(x) = \sin^5(x) + 4\sin(x)$$

Explain how you would evaluate this function using `polyval()`. Then use `polyval()` to plot this curve for a range of $0 < x < 2$.

**3.** For the polynomial

$$f(x) = x^5 - 3x^4 + 4x + 1$$

**(a)** Utilize `polyder` and `polyint` to derive the polynomial coefficients, and then plot the following four functions on a single graph over the interval $0 < x < 2$:

**i.** $f(x)$

**ii.** $\dfrac{df(x)}{dx}$

**iii.** $\dfrac{d^2f(x)}{dx^2}$

**iv.** $\displaystyle\int_0^x f(t)\,\mathrm{d}t$

**(b)** The roots can be determined of the polynomial of $f(x)$ by using `np.roots()`. Doing this results in a set of complex-valued and real roots. Write a routine identifying the real-valued root of $f(x)$. Verify that the real root is accurately determined `np.roots()` by using `np.polyval()`.

**4.** The convolution of a discrete finite length sequence with another discrete finite length sequence is calculated with the discrete convolution operation of `np.convolve()`. This can be used to determine the product of two polynomials. In this problem, determine the coefficients of the polynomial

$$f(x) = (x^3 + 2x + 1)^5$$

Print out the coefficients of $x$ in descending order of the power of $x$.

©J.Nielsen, 2023

**5.** Suppose we have the points

$$
\begin{aligned}
p_1 &= (x_1, y_1) = (0, 1) \\
p_2 &= (x_2, y_2) = (1, 1.5) \\
p_3 &= (x_3, y_3) = (2, 4) \\
p_4 &= (x_4, y_4) = (4, 7) \\
p_5 &= (x_5, y_5) = (5, 4)
\end{aligned}
$$

Determine the fourth-order polynomial that passes through these points. Then plot the fourth-order interpolated polynomial that passes through these points as determined by `np.polyfit`. Indicate the five points with red x markers and label the axis. List the code to find the coefficients and plot the curve as well as the plot.

**6.** Consider a data file of `meas.npy` that contains 500 measurements of a random process. The measurements themselves are noisy. The following code generates the data file.

```
#program to generate the measurement file
N = 500
x = 0.4 + 4*np.random.normal(loc=0,scale=1,size=N)
a = 0.1; b = 1; c = 2;
y = a + b*np.sin(x) + c*np.sin(x)**2
      + 0.5*np.random.normal(loc=0,scale=1,size=N)
plt.plot(x,y,'r.')
np.save('meas.npy',np.append(x.reshape(N,1),y.reshape(N,1)))
```

**(a)** Read in the data in the binary file of `meas.npy`. The first column is the $x$ values, and the second is the $y$ values. It is known that the data fits the model of

$$y = a + b\sin(x) + c\sin^2(x)$$

in addition to the unknown noise. Use `np.polyfit()` to generate the least square regression curve fit this data determining the unknown coefficients of $a$, $b$ and $c$. Then plot the data points (as a scatter plot) and superimpose the regression curve fit.

**(b)** Determine the standard deviation of the error of the regression curve fit.

©J.Nielsen, 2023