

Homework 15

Problem 1

Find the numerical integral of the following function:

$$I = \int_0^{2\pi} e^{-x} \sin(3x) dx.$$

Problem 2

Given the data x_{given} and y_{given} below. Find the linear interpolate y_{il} corresponding to $x_{\text{i}}=0.54$. Also find the cubic spline value of y_{is} at the same point. The x_{given} and y_{given} values are computed from $y = f(x) = \exp(4x)$. Find the relative error between your interpolants and the exact value. The relative error is given by $\epsilon = |(y - y_{\text{exact}})/y_{\text{exact}}|$. Are you happy with the result?

```
In [ ]: import numpy as np

x_given = np.array([0,0.2,0.4,0.8,1.0])
y_given = np.exp(4.0*x_given)

x_i      = 0.54
y_exact = np.exp(4.0*x_i)
```

Problem 3

In a previous assignment we used thermo data for heat capacities of species. Its kind of annoying that we have two temperature ranges for each species. For the given array of temperatures (K) below, the corresponding cp/Rg array is given for CH4. This is computed using the equations and the coefficients for the two temperature ranges given in previous assignments.

Fit a single 4th order polynomial to the whole range of temperature data. At each point, compare your polynomial fit to the one given.

Report the maximum relative error that occurs over the list of temperatures.

Plot the two versions to visually compare the original data and your polynomial fit.

```
In [ ]: T = np.linspace(300.,3000.,100)

a_lo = np.array([ 5.15,-1.37E-02,4.92E-05,-4.85E-08,1.67E-11])
a_hi = np.array([7.49E-02,1.34E-02,-5.73E-06,1.22E-09,-1.02E-13])
i_lo = np.where(np.logical_and(T>=300.0, T<1000.0))
i_hi = np.where(np.logical_and(T>=1000.0, T<=3000.0))
cpRg = np.zeros(100)
cpRg[i_lo] = a_lo[0] + a_lo[1]*T[i_lo] + a_lo[2]*T[i_lo]**2.0 + \
             a_lo[3]*T[i_lo]**3.0 + a_lo[4]*T[i_lo]**4.0
cpRg[i_hi] = a_hi[0] + a_hi[1]*T[i_hi] + a_hi[2]*T[i_hi]**2.0 + \
             a_hi[3]*T[i_hi]**3.0 + a_hi[4]*T[i_hi]**4.0
```

Problem 4

The following kinetic $R(T)$ were collected. We want to fit this to the following model function: $R = kT^m e^{-E_a/(R_g T)}$. Here, k , m , and (E_a/R_g) , are adjustable parameters that we want to find so that our model best fits the data.

Find the best parameters and plot the data and the model function together on the same graph.

```
In [ ]: T = np.linspace(500.,1000.,8)
R = [105.598, 89.700, 70.768, 66.996, 60.711, 58.992, 55.8328, 53.420]
```

```
In [ ]:
```