**Algorithms background:**

**Cubic splines:**

In many engineering problems, we normally have measured values for a specific value collected from precise experiments. However, as shown in this problem, we might need to evaluate the function on a value other than the ones used in the experiment. In order to do this, we use splines and for accurate results we use cubic spline. The method is based on cutting the measured value to intervals. Each interval has its own equation based on coefficients and their second derivatives. The main advantage of this method is that it provides high accuracy for the values within each interval.

**Thomas algorithm:**

**Numerical integration:**

Usually, analytical integration can be used for simple problems. However, there are problems where integrating the given problem’s function analytically would be time consuming and might be too difficult. Therefore, using numerical integration would be more time efficient and easier to work with. In order to evaluate the numerical integration, there are 3 known methods for doing so, trapezoidal rule, 1/3 Simpson and 3/8 Simpson. In this project, we chose to use the trapezoidal method for its efficiency in data given as points.

**Numerical differentiation:**

Similar to the numerical integration, evaluating the derivative at a certain value can be complex and difficult especially if the values are given in tabular form without knowing the function beforehand. In this case, using numerical differentiation methods can make the process much easier with accurate results depending on the error type of the equation used. The main method for evaluating the derivative is through the absolute difference between the value of the function at two points surrounding the wanted value divided by the difference between the two points. For higher derivatives, the expression of the result is obtained from the Taylor expansion of the function.

**Code functionality:**

numerical\_int( x, y):

The function takes the values of the volumes as x and the corresponding pressures as y, with the first and last value being calculated from the splines equations. Afterwards, it calculated the values of the integral using the trapezoidal method by calculating the partitioning the values into segments and calculating each one of them separately and add them up in the end to get the integral.

numerical\_diff(splines,values, x):

The function takes the coefficients of each spline interval, the values of the volume and the point at which the derivative needs to be calculated and return the derivative. The function uses a difference of 1e-12 and handles three cases. The first case is if the given point is the last in the dataset, in this case it uses the backward difference. The second case is if the given point is the first in the dataset, in this case it uses the forward difference. Otherwise, it uses the centered difference method.

range\_diff( splines,values,x1,x2):

The function takes the coefficients of each spline interval, the values of the volume and the two points between which the derivative needs to be calculated and return a vector containing the derivative at each point. The function loops over the given interval with a step of 0.1 and calculated the derivative for each value.