Thermodynamics Steam Saturation Interpolation

# Abstract

Allow the user to calculate Steam Saturation based on temperature entered by the user. If the temperature is not listed it will be interpolated based on the temperature index.

# Problem Statement

Tables in higher math are wonderful things, but sometimes they don’t list the value we need. When that happens we either calculate all the values by hand or we interpolate the table. Either way we get our answer. The interpolation process we are using is linear interpolation.

# Methodology

Utilizing Python with pandas and numpy we are able handle all the necessary calculations to retrieve our answer. First thing we do is grab the table from the url and have pandas create a data frame inside python. Then we drop any columns we do not need such as; **vg, ug, hfg, hg, sfg,** and **sg**. The user will enter a temperature to reference. If the temperature is not listed then the remaining columns are interpolated with the users input. It then lists all the variables needed.

import pandas as pd

import numpy as np

from pandas.io.sql import table\_exists

url=r'https://thermo.pressbooks.com/chapter/saturation-properties-temperature-table/'

t\_table = pd.read\_html(url, header=1)

t\_table = t\_table[0]

cols = t\_table.columns

t\_table = t\_table.rename({'°C': 'degC'}, axis=1)

t\_table = t\_table.drop(columns=['vg', 'ug', 'hfg', 'hg', 'sfg', 'sg'])

continue\_yn ='y'

while continue\_yn=='y':

t\_table = t\_table

temp = float(input('Enter a temperature from 0.01 to 373.95: '))

if temp < 0.01 or temp > 373.95:

print(f'The temperature is not valid {temp}.')

print(f'Enter a valid value 0.01 to 373.95')

else:

if temp in t\_table.degC.values:

t\_table\_temp = t\_table[t\_table['degC']==temp]

print(t\_table\_temp)

continue\_yn = input('Do you wish to continue? Y or N ').lower()

else:

t\_table\_ap = t\_table

table\_columns = list(t\_table\_ap.columns)

degC\_list = t\_table\_ap['degC'].to\_list()

inter\_values = []

t\_tablenew = t\_table

for value in table\_columns:

column\_list = t\_table\_ap[value].to\_list()

inter = np.interp(temp,degC\_list,column\_list)

inter\_values.append(inter)

t\_tablenew.loc[99] = inter\_values

t\_tablenew = t\_tablenew.sort\_values('degC')

t\_tablenew = t\_tablenew.reset\_index(drop=True)

t\_tablenew = t\_tablenew[t\_tablenew['degC']==temp]

print(t\_tablenew)

continue\_yn = input('Do you wish to continue? Y or N ').lower()

# Solution

The user will enter a temperature within a specified range. If the user enters a temperature outside of a predetermined range they will be prompted to enter a new temperature. If the temperate is listed in the table then it will output those variables in the table, if it is not listed it will interpolate the correct vales and list them. Below is a table of Temperatures with the values listed:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Temperature | Pressure (Mpa) | Volume (vf) | Energy (uf) | Enthalpy (hf) | Entropy (sf) |
| 6.0 | 0.000944 | 0.001 | 25.20 | 25.22 | 0.0913 |
| 25.5 | 0.003278 | 0.01003 | 106.89 | 106.92 | 0.3742 |
| 123 | 0.22018 | 0.001063 | 516.35 | 516.58 | 1.5599 |
| 245 | 3.66155 | 0.001241 | 1057.10 | 1061.70 | 2.7478 |
| 373 | 21.818684 | 0.002892 | 1974.40 | 2073.70 | 4.3359 |

# Conclusion

Not being limited to answers within the table we can further expand our understanding how various forces operate together. There are many different types of interpolation that give just as varied answers for the missing data. We were wanting to get it as accurate as possible which is the reason we did the interpolation based on index and not wither it was just between two variables.