Pages 1–12
2023

# Equivalent Circuits

Daniele Caruana
Group 8
PHY2235
B.Sc.(Hons) in Physics
23rd February 2023

## Abstract

Permittivity describes a material's polarizability. In this experiment, an open-ended coaxial transmission line was approximated using an equivalent circuit for a radiating antenna model. This was used to obtain frequency-dependent values for the reflection coefficients of Methanol and NaCl, and hence obtain frequency-dependent permittivity values of the same materials.

## 1 INTRODUCTION & THEORETICAL BACKGROUND

### 1.1 Permittivity

In electromagnetism, certain materials called dielectrics are used to store electric charges in an applied electric field while simultaneously not letting any electric charges flow through them. The storage of such charges is due to a phenomenon known as dielectric polarisation, which occurs when the dielectric is placed within an electric field, causing the positive charges to shift towards the direction of the field, while the negative charges shift towards the other direction.

Permittivity is a characteristic of a dielectric, describing the magnitude at which it is polarized, and hence its capability to store electric charge by distorting the shape of the charge distribution due to an external field. While the relation between permittivity and the electric field can be considered as

$$D = \varepsilon E,$$

where $D$ is the electric displacement field, $\varepsilon$ is the permittivity, and $E$ is the electric field, permittivity is usually never constant. The variance in permittivity can be caused by several factors such as the frequency of the field, and the temperature. However, the value of the permittivity of free space, $\varepsilon_0 \approx 8.85 \times 10^{-12}\ \mathrm{F\ m^{-1}}$ (Tiesinga et al., 2021), which can be described as the ratio between the electric displacement field and the electric field, is invariant as there are no elements affecting it due to its vacuum property. As permittivity is usually a function of the frequency of the electric field, it is usually considered a complex function.

The dielectric constant, also known as relative permittivity, describes the ratio between the absolute permittivity of a material and the permittivity of free space, thus rendering the number dimensionless. By

considering the complex composition of permittivity, relative permittivity can be expressed as

$$\varepsilon_r'(\omega) - i\varepsilon_r''(\omega) = \frac{\varepsilon(\omega)}{\varepsilon_0},$$

where $\varepsilon_r'$, and $\varepsilon_r''$ are the real and imaginary values of the relative permittivity, and $\varepsilon(\omega)$ is the absolute permittivity of the material (Fujimoto, 2007).

## 1.2 Measuring Techniques

One way of measuring the relative permittivity of a material is to use the dielectric in a capacitor. This is achieved by comparing the capacitance of the capacitor with a vacuum between its plates ($C_0$), and the capacitance of the capacitor with the dielectric between its plates ($C$) such that

$$\varepsilon_r = \frac{C}{C_0}.$$

However, as described in section 1.1, most materials have an electric field with a varying frequency, and thus require a measuring technique which can accommodate this. One technique is the use of scattering parameters measurements, which are the elements $S_{ij}$ of a matrix $\boldsymbol{S}$ describing certain properties of electrical networks, obtained through radio frequency measurements. The S-matrix is denoted as

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nn} \end{bmatrix},$$

where the first index $i$ depicts an output port, where as the second index $j$ depicts the input port. Thus $S_{ij}$, when $i = j$, depicts a reflection. In their paper, Marsland and Evans (1987) proposed a method in which an open-ended coaxial-line probe was used as an approximation of a radiating antenna model, in order to measure the dielectric properties of certain materials. For this purpose, a 1-port network with given parameter $S_{11}$, together with a vector network analyzer can be used to measure the reflection coefficients of lossy materials by pressing the probe against the material, or immersing the probe into the liquids.

# 2 CALCULATIONS

### *2.0.1 Equivalent Circuit*

Equivalent circuits are circuits derived to simplify complex systems. The model proposed by Marsland and Evans (1987) consists of a lumped equivalent circuit as shown in Fig. 1, and can be represented mathematically by the following set of equations, where the admittance $Y$ can be related to the impedance $Z$ as

$$Y = \frac{1}{Z},$$

and by normalising this to the characteristic impedance $Z_0$ using

$$y = YZ_0,$$

the following equation was obtained:

$$y(\omega, \varepsilon_r) = G_0 Z_0 \varepsilon_r^{\frac{5}{2}} + j\omega Z_0(\epsilon_r C_0 + C_f), \tag{1}$$

where $\omega$ is the angular frequency, $G_0$ is the free-space radiation conductance, $C_0$ is the capacitance representing the fringing field, and $C_f$ is the capacitance representing the capacitance representing the fringing field within the dielectric of the cable. Due to the fact that $G_0 \ll C_0$, the $G_0$ term can be ignored, and hence

$$y(\omega, \varepsilon_r) = j\omega Z_0(\epsilon_r C_0 + C_f). \tag{2}$$

In order to calculate the permittivity of a dielectric, the input reflection coefficient is required. This can be acquired by using a two-port embedding network connected to a one-port network such that

$$\Gamma_m = \frac{\rho_m - E_{11}}{\rho_m E_{22} - (E_{11}E_{22} - E_{12}E_{21})}, \tag{3}$$

where $\Gamma_m$ is the true reflection coefficient of the material $m$, $\rho_m$ is the measured reflection coefficient of the material $m$, and $E_{ij}$ are the elements of the scattering matrix $\boldsymbol{E}$.

### *2.0.2 Error Correction*

Using a vector network analyser requires the experimenter to first calibrate the device. This is usually done by measuring at least three terminations, which are the ending of the transmission line; short, open/air, and the match at each frequency level, in order to determine the elements of the matrix $\boldsymbol{E}$. However a device to

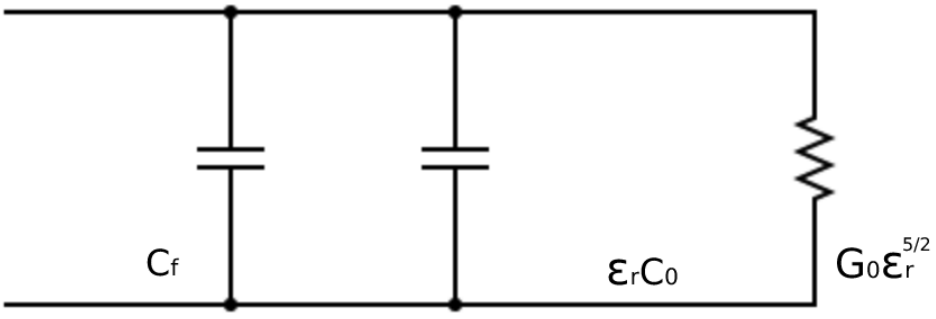


Fig. 1: Equivalent Circuit of probe

replicate the latter termination readings is difficult to devise, due to the varying frequency. A way to deal with this issue is to use a polar liquid such as water or methanol with known characteristics as the match termination, and therefore $\boldsymbol{E}$ can be calculated using the parameters for the equivalent circuit, however this is not accurate due to a number of physical inaccuracies within the probe and the cable. By Marsland and Evans (1987) it is determined that calculating the elements of $\boldsymbol{E}$ is only needed to calculate these errors, and can be replaced by substituting relative values of admittances $y_i$ and reflection coefficients $\rho_i$ into eq. 3, and remove all elements of $\boldsymbol{E}$, the following equation is obtained:

$$y_m = -\frac{\Delta_{m1}\Delta_{32}\ y_3\ y_2 + \Delta_{m2}\Delta_{13}\ y_1\ y_3 + \Delta_{m3}\Delta 21 y_2\ y_1}{\Delta_{m1}\Delta_{32}\ y_1 + \Delta_{m2}\Delta_{13}\ y_2 + \Delta_{m3}\Delta_{21}\ y_3}, \tag{4}$$

where $\Delta_{ij} = \rho_i - \rho_j$. The cross-ratio invariance theorem of bilinear transformations which states that for points $\{z_1, z_2, z_3\}$ on a plane $\Pi_z$, transformed using bilinear transformation into the points $\{w_1, w_2, w_3\}$ on a plane $\Pi_w$, then $\frac{(w-w_1)(w_2-w_3)}{(w_1-w_2)(w_3-w)} = \frac{(z-z_1)(z_2-z_3)}{(z_1-z_2)(z_3-z)}$ (Kreyszig, 2006). Hence, eq. 4 can be written as

$$\frac{\Delta_{m1}\Delta_{32}}{\Delta_{m2}\Delta_{13}} = \frac{(\rho_m-\rho_1)(\rho_3-\rho_2)}{(\rho_m-\rho_2)(\rho_1-\rho_3)} = \frac{(y_m-y_1)(y_3-y_2)}{(y_m-y_2)(y_1-y_3)}, \tag{5}$$

for any number of reflection coefficients $\rho$, and admittances $y$.

### *2.0.3 Deriving the model*

By applying

$$y'(\omega, \varepsilon_r) = \left(\frac{1}{j\omega C_0 Z_0}\right) \cdot y(\omega, \varepsilon_r) - \left(\frac{C_f}{C_0}\right) \tag{6}$$

to 1, the following is obtained:

$$y'(\omega, \varepsilon_r) = \varepsilon_r + (G_0/j\omega C_0)\varepsilon_r^{5/2} \tag{7}$$

$$y'(\omega, \varepsilon_r) = \varepsilon_r + G_n\varepsilon_r^{5/2}, \tag{8}$$

where $G_n = (G_0/j\omega C_0)$. Due to the cross-ratio invariance shown in eq. 5, eq. 8 can be written as follows:

$$\frac{(\rho_m-\rho_1)(\rho_3-\rho_2)}{(\rho_m-\rho_2)(\rho_1-\rho_3)} = \frac{(y_m-y_1)(y_3-y_2)}{(y_m-y_2)(y_1-y_3)}. \tag{9}$$

Hence, for the permittivity of the material $\varepsilon_m$:

$$G_n\varepsilon_m^{5/2} + \varepsilon_m + \left[\frac{\Delta_{m1}\Delta_{32}\ y_3'\ y_2' + \Delta_{m2}\Delta_{13}\ y_1'\ y_3' + \Delta_{m3}\Delta 21 y_2'\ y_1'}{\Delta_{m1}\Delta_{32}\ y_1' + \Delta_{m2}\Delta_{13}\ y_2' + \Delta_{m3}\Delta_{21}\ y_3'}\right] = 0 \tag{10}$$

Simplifying the model using a short-circuit approximation:

$$y_1 \to \infty$$

$$G_n\varepsilon_m^{5/2} + \varepsilon_m + \left[\frac{\Delta_{m1}\Delta_{32}\, y_3'\, y_2' + \Delta_{m2}\Delta_{13}\, y_1'\, y_3' + \Delta_{m3}\Delta_{21} y_2'\, y_1'}{\Delta_{m1}\Delta_{32}\, y_1' + \cancelto{0}{\Delta_{m2}\Delta_{13}\, y_2'} + \cancelto{0}{\Delta_{m3}\Delta_{21}\, y_3'}}\right] = 0 \tag{11}$$

$$G_n\varepsilon_m^{5/2} + \varepsilon_m \left[\frac{\Delta_{m2}\Delta_{13}}{\Delta_{m1}\Delta_{32}} y_3' + \frac{\Delta_{m3}\Delta_{21}}{\Delta_{m1}\Delta_{32}} y_2'\right] = 0 \tag{12}$$

Since $G_n \ll C_0$, then $G_n$ can be omitted, and hence, the permittivity of a material $m$ is given by:

$$\varepsilon_m = -\frac{\Delta_{m2}\Delta_{13}}{\Delta_{m1}\Delta_{32}}\varepsilon_3 - \frac{\Delta_{m3}\Delta_{21}}{\Delta_{m1}\Delta_{32}}\varepsilon_2. \tag{13}$$

# 3 DATA ANALYSIS AND RESULTS

The S11 data was imported into python using the Pandas library, along with known values of the permittivities of water, and known comparable values of the permittivities of Methanol and NaCl. The data was then modelled using equation 13 and fitted using the polyfit function with degree 5.

```
methanol_em_calc = -(e3_arr * (methanol_d_m2 * d_13) / (methanol_d_m1 * d_32)) \
                   - (e2 * (methanol_d_m3 * d_21) / (methanol_d_m1 * d_32))

res_methanol_calc, cov_methanol_calc = np.polyfit(frequency_arr, methanol_em_calc, deg=5,
    cov=True)
p_methanol_calc = np.poly1d(res_methanol_calc)
methanol_extend = np.linspace(5e8, 1.2e10, 201)
tr_x_methanol_calc = p_methanol_calc(methanol_extend)

res_methanol_giv, cov_methanol_giv = np.polyfit(frequency_arr, exp_em_methanol, deg=5, cov=
    True)
p_methanol_giv = np.poly1d(res_methanol_giv)
me_giv_extend = np.linspace(5e8, 1.2e10, 201)
tr_x_methanol_giv = p_methanol_giv(me_giv_extend)
nacl_em_calc = -(e3_arr * (nacl_d_m2 * d_13) / (nacl_d_m1 * d_32)) \
               - (e2 * (nacl_d_m3 * d_21) / (nacl_d_m1 * d_32))

res_nacl_calc, cov_nacl_calc = np.polyfit(frequency_arr, nacl_em_calc, deg=5, cov=True)
p_nacl_calc = np.poly1d(res_nacl_calc)
nacl_extend = np.linspace(5e8, 1.2e10, 201)
tr_x_nacl_calc = p_nacl_calc(nacl_extend)

res_nacl_giv, cov_nacl_giv = np.polyfit(frequency_arr, exp_em_nacl, deg=5, cov=True)
p_nacl_giv = np.poly1d(res_nacl_giv)
na_giv_extend = np.linspace(5e8, 1.2e10, 201)
tr_x_nacl_giv = p_nacl_giv(na_giv_extend)
```

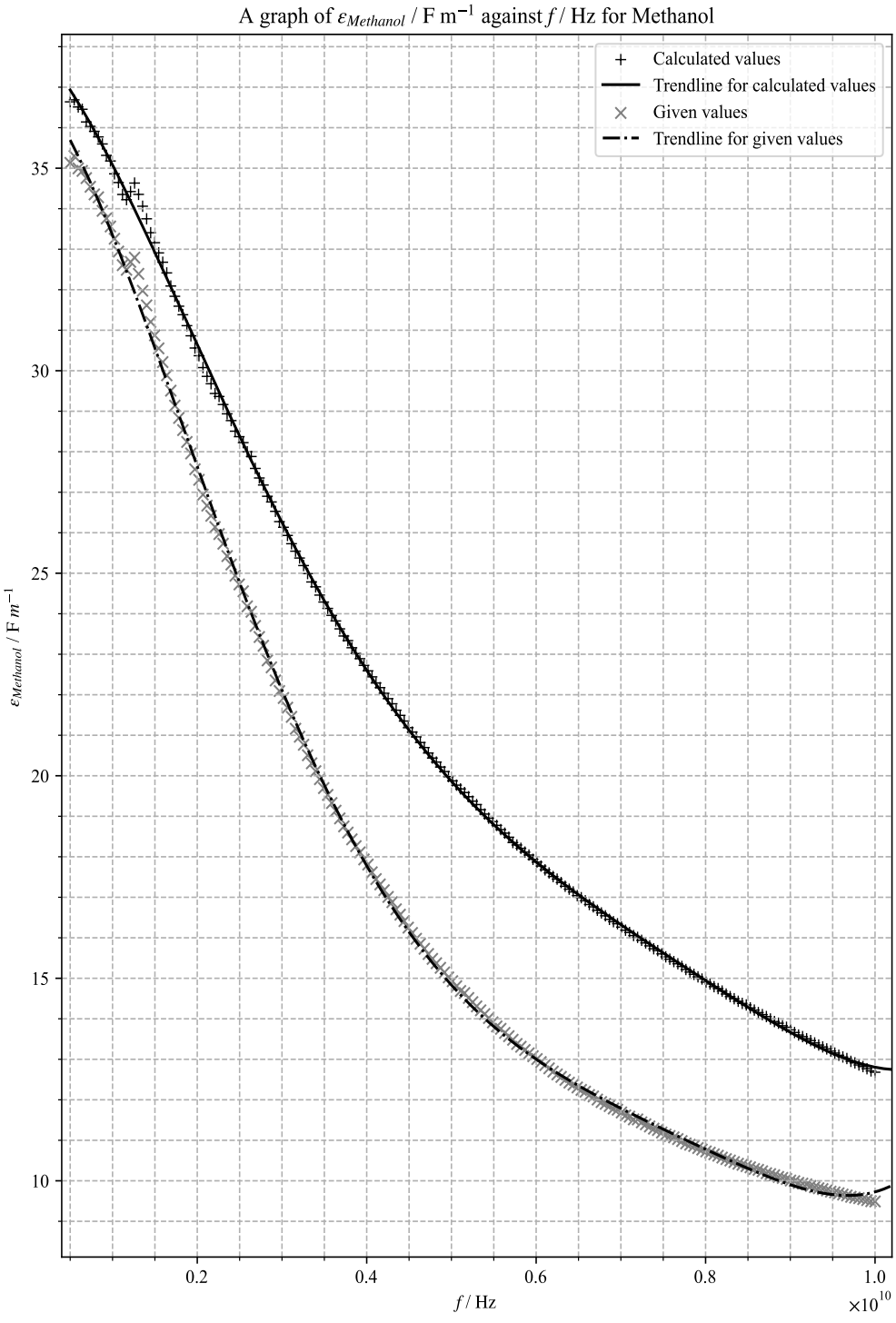


Fig. 2: A graph of the permittivity of Methanol against frequency

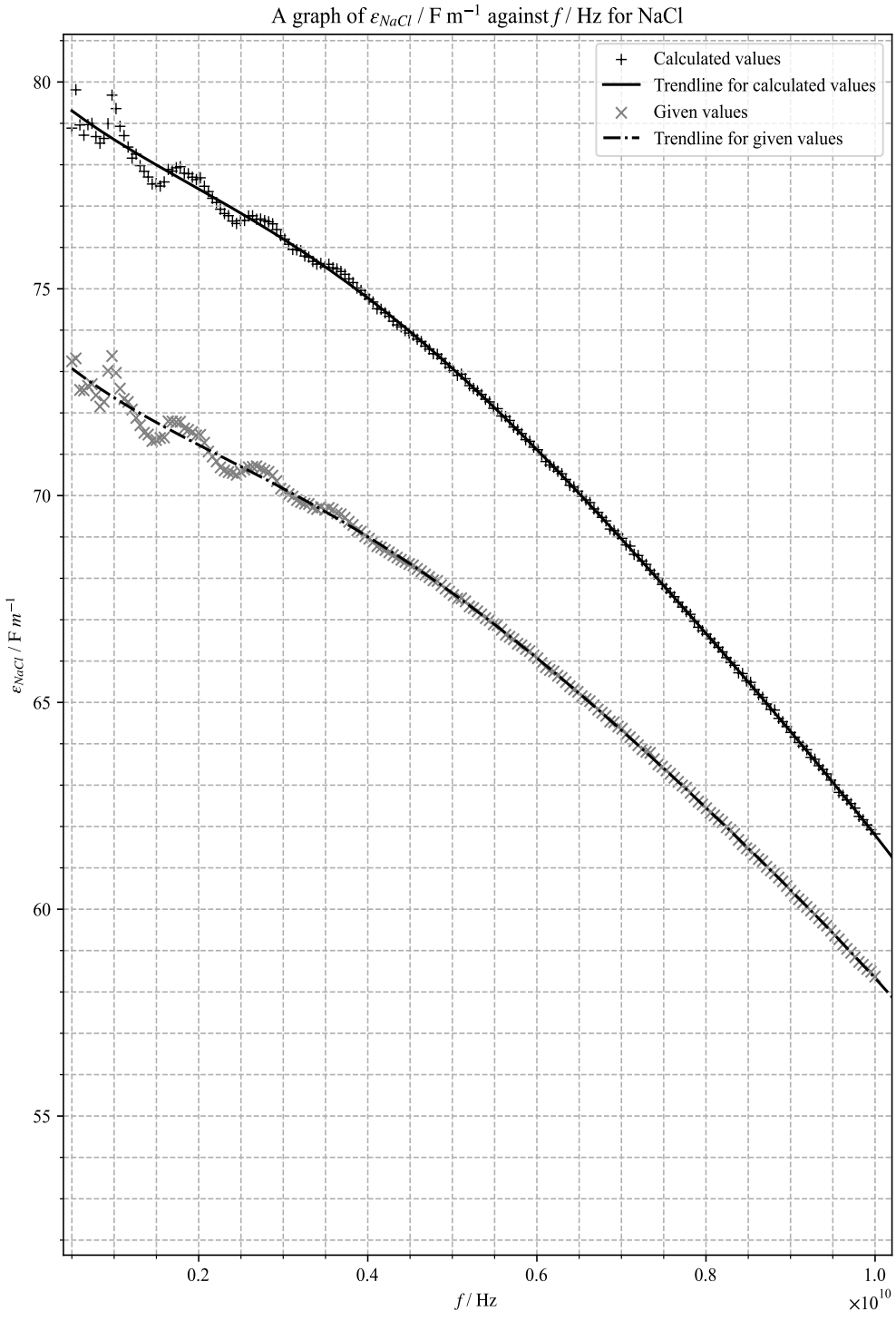


Fig. 3: A graph of the permittivity of NaCl against frequency

# 4 DISCUSSION

In this experiment the method proposed in Marsland and Evans (1987) was followed in order to derive a radiating antenna model which approximates an open-ended coaxial transmission line. Using provided $S_{11}$ data, together with the derived model, the graphs in section 3 were plotted, along with the expected values provided.

It is clear in the graphs that the shape of the calculated plot is very similar to the expected plot, with a slight shift. This shift could have been introduced due to systematic errors which could be caused by improper calibration or issues with the design of the circuit. Theses errors are represented by the scattering matrix $E$ as described in Marsland and Evans (1987). Other errors that could have affected the results include impurities in the calibration materials and the MUTs, instrumental errors in the VNA, and inaccuracies that emerged during the data analysis.

By looking at the lower frequency values in the graphs, it is visible that the probe is more sensitive to systematic errors at low frequencies.

# 5 CONCLUSION

In this experiment, a radiating antenna model was derived in order to approximate an open-ended coaxial transmission line. This was used to obtain values for the reflection coefficients of Methanol and NaCl at different frequencies, which in turn were used to obtain permittivity values of the same materials at different frequencies. The obtained values for permittivity were compared to accepted values, showing that the model was relatively accurate. Through the obtained graphs, it was also shown that the permittivity of a material decreases with an increase in frequency.

# REFERENCES

Fujimoto, M (2007) *Physics of classical electromagnetism* (1st ed.) Springer.

Kreyszig, E (2006) *Advanced engineering mathematics*. John Wiley & Sons.

Marsland, T & Evans, S (1987) Dielectric measurements with an open-ended coaxial probe. *IEE Proceedings H Microwaves, Antennas and Propagation*, *134*(4) 341. https://doi.org/10.1049/ip-h-2.1987.0068

Tiesinga, E Mohr, P Newell, D B & Taylor, B (2021) Codata recommended values of the fundamental physical constants: 2018. https://www.nist.gov/publications/codata-recommended-values-fundamental-physical-constants-2018

# APPENDIX

## 1 CODE LISTING

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as mticker

# Importing Data and defining vars

df_s11 = pd.read_excel('Data.xlsx', 0, header=1)
nacl = pd.read_excel('Data.xlsx', 1)
methanol = pd.read_excel('Data.xlsx', 2)
df_e3 = pd.read_excel('Data.xlsx', 3)

air_s11 = df_s11.iloc[:, 1].to_numpy() + (df_s11.iloc[:, 2].to_numpy() * 1j)
water_s11 = df_s11.iloc[:, 4].to_numpy() + (df_s11.iloc[:, 5].to_numpy() * 1j)
methanol_s11 = df_s11.iloc[:, 7].to_numpy() + (df_s11.iloc[:, 8].to_numpy() * 1j)
nacl_s11 = df_s11.iloc[:, 10].to_numpy() + (df_s11.iloc[:, 11].to_numpy() * 1j)
short_s11 = df_s11.iloc[:, 13].to_numpy()
e3_arr = df_e3.iloc[:, 1].to_numpy() + (df_e3.iloc[:, 2].to_numpy() * 1j)
frequency_arr = df_s11.iloc[:, 0].to_numpy()

exp_em_methanol = methanol.iloc[:, 1].to_numpy() + (methanol.iloc[:, 2].to_numpy() * 1j)
exp_em_nacl = nacl.iloc[:, 1].to_numpy() + (nacl.iloc[:, 2].to_numpy() * 1j)

# Defining reflection coefficients

methanol_d_m2 = methanol_s11 - air_s11
methanol_d_m1 = methanol_s11 - short_s11
methanol_d_m3 = methanol_s11 - water_s11
nacl_d_m2 = nacl_s11 - air_s11
nacl_d_m1 = nacl_s11 - short_s11
nacl_d_m3 = nacl_s11 - water_s11
d_13 = short_s11 - water_s11
d_21 = air_s11 - short_s11
d_32 = water_s11 - air_s11
e2 = 1

# Fitting the function - Methanol

```

```
methanol_em_calc = -(e3_arr * (methanol_d_m2 * d_13) / (methanol_d_m1 * d_32)) \
                   - (e2 * (methanol_d_m3 * d_21) / (methanol_d_m1 * d_32))

res_methanol_calc, cov_methanol_calc = np.polyfit(frequency_arr, methanol_em_calc, deg=5,
    cov=True)
p_methanol_calc = np.poly1d(res_methanol_calc)
methanol_extend = np.linspace(5e8, 1.2e10, 201)
tr_x_methanol_calc = p_methanol_calc(methanol_extend)

res_methanol_giv, cov_methanol_giv = np.polyfit(frequency_arr, exp_em_methanol, deg=5, cov=
    True)
p_methanol_giv = np.poly1d(res_methanol_giv)
me_giv_extend = np.linspace(5e8, 1.2e10, 201)
tr_x_methanol_giv = p_methanol_giv(me_giv_extend)

# Fitting the function - NaCl

nacl_em_calc = -(e3_arr * (nacl_d_m2 * d_13) / (nacl_d_m1 * d_32)) \
               - (e2 * (nacl_d_m3 * d_21) / (nacl_d_m1 * d_32))

res_nacl_calc, cov_nacl_calc = np.polyfit(frequency_arr, nacl_em_calc, deg=5, cov=True)
p_nacl_calc = np.poly1d(res_nacl_calc)
nacl_extend = np.linspace(5e8, 1.2e10, 201)
tr_x_nacl_calc = p_nacl_calc(nacl_extend)

res_nacl_giv, cov_nacl_giv = np.polyfit(frequency_arr, exp_em_nacl, deg=5, cov=True)
p_nacl_giv = np.poly1d(res_nacl_giv)
na_giv_extend = np.linspace(5e8, 1.2e10, 201)
tr_x_nacl_giv = p_nacl_giv(na_giv_extend)

# Plotting Graph - Methanol

rc = {"font.family": "serif",
      "mathtext.fontset": "stix"}
plt.rcParams.update(rc)
plt.rcParams["font.serif"] = ["Times New Roman"] + plt.rcParams["font.serif"]
hfont = {'fontname': 'Times New Roman'}

fig_methanol, ax_methanol = plt.subplots(1)
ax_methanol.figure.set_size_inches(8.27, 11.69)

```

```
78
79 plt.scatter(frequency_arr, np.real(methanol_em_calc), color='black', marker='+', linewidths
       =0.7,
80              label="Calculated values")
81 plt.plot(methanol_extend, np.real(tr_x_methanol_calc), color='black', label='Trendline for
       calculated values')
82
83 plt.scatter(frequency_arr, np.real(exp_em_methanol), marker='x', color='grey', linewidth=1,
       label='Given values')
84 plt.plot(me_giv_extend, np.real(tr_x_methanol_giv), linestyle='-.', color="black", label='
       Trendline for given values')
85
86 plt.grid(which='both', linestyle='--')
87 plt.minorticks_on()
88
89 plt.xlim(4e8, 1.02e10)
90 plt.title(r"A graph of $\varepsilon_{Methanol}$ / $\mathrm{F} \ \mathrm{m}^{-1}$ against $f$
       / $\mathrm{Hz}$ for "
91           r"Methanol")
92 plt.xlabel(r"$f$ / $\mathrm{Hz}$")
93 plt.ylabel(r"$\varepsilon_{Methanol}$ / $\mathrm{F} \ m^{-1}$")
94 formatter = mticker.ScalarFormatter(useMathText=True)
95 ax_methanol.xaxis.set_major_formatter(formatter)
96 plt.legend()
97 plt.savefig('methanol.pdf')
98 plt.show()
99
100 # Plotting graph - NaCl
101
102 fig_nacl, ax_nacl = plt.subplots(1)
103 ax_nacl.figure.set_size_inches(8.27, 11.69)
104
105 plt.scatter(frequency_arr, np.real(nacl_em_calc), color='black', marker='+', linewidths=0.7,
       label="Calculated values")
106 plt.plot(nacl_extend, np.real(tr_x_nacl_calc), color='black', label='Trendline for
       calculated values')
107
108 plt.scatter(frequency_arr, np.real(exp_em_nacl), marker='x', color='grey', linewidth=1,
       label='Given values')
109 plt.plot(na_giv_extend, np.real(tr_x_nacl_giv), linestyle='-.', color="black", label='
       Trendline for given values')
```

```
110
111 plt.grid(which='both', linestyle='--')
112 plt.minorticks_on()
113
114 plt.xlim(4e8, 1.02e10)
115
116 plt.title(r"A graph of $\varepsilon_{NaCl}$ / $\mathrm{F} \ \mathrm{m}^{-1}$ against $f$ / $
        \mathrm{Hz}$ for "
117           r"NaCl")
118 plt.xlabel(r"$f$ / $\mathrm{Hz}$")
119 plt.ylabel(r"$\varepsilon_{NaCl}$ / $\mathrm{F} \ m^{-1}$")
120 formatter = mticker.ScalarFormatter(useMathText=True)
121 ax_nacl.xaxis.set_major_formatter(formatter)
122 plt.legend()
123 plt.savefig('nacl.pdf')
124 plt.show()
125
126 print("Debug")
```