Outgassing of Electronegative Impurities within Materials Used in the nEXO Neutrinoless Double Beta Decay Detector

Moore Lab
Principal Investigator (left): Prof. David Moore, Ph.D
Graduate Student Mentor (right): Sierra Wilde, Ph.D Student

Wright Laboratory

Ronald Thorpe
Undergraduate Researcher
Yale University
ronald.thorpe@yale.edu
Goals and Rationale

● How can we model the diffusion of relevant atmospheric gasses from various polymers that are beings used within nEXO?
● Can we model a diffusion constant ($D_0$) and activation energy ($E_a$) for these polymers?
● Will we uncover other relevant species other than O2?
● Are there other metrics that are making our fits successful or unsuccessful?
Theoretical Model

Flow rate of particles out of material model

$$J = D \left( \frac{\partial c(x, t)}{\partial x} \right)_{x=0}$$

$$= \frac{4c_0 D}{d} \sum_0^\infty \exp \left\{ - \left( \frac{\pi (2n + 1)}{d} \right)^2 \cdot Dt \right\}$$

Short timescale approximation

$$J = c_0 \sqrt{\frac{D}{t}}$$

Long timescale approximation (1st term of series)

$$J = \frac{4c_0 D}{d} \exp \left\{ - \left( \frac{\pi}{d} \right)^2 \cdot Dt \right\}$$

Temperature dependence

$$D(T) = D_0 \cdot \exp \left( - \frac{E_a}{k_B T} \right)$$

***All sourced from:
def fitfunction_short_temp_offset(self, tT, a, b, c, tau):
    dt = tT[1] - tT[0]  # Assuming t is evenly spaced
    decay_t = np.arange(0, len(tT[0])) * dt  # Define time for the decay function
    decay = np.exp(-decay_t/tau) * (decay_t >= 0)
    Normdecay = decay/np.sum(decay)  # Normalize decay function
    padded_data = np.pad(tT[1], (len(decay)-1, 0), mode='edge')  # Pad data with values similar to edge to resolve boundary issues
    smooth_step = np.convolve(padded_data, Normdecay, mode='valid')  # Convolve two functions
    return a*(np.exp((-b)/(smooth_step)))/(tT[0]+c)**(0.5) + np.exp((-1.0*tT[0]))

def fitfunction(self,tT,a,b,c, tau):
    factor = np.exp(-1.0*b/(smooth_step))
    pi = 3.141592653589793
    function = 0
    d = .01
    for m in [0, 1, 2, 3, 4, 5]:
        function += np.exp(-1.0*(((2*m+1)*(pi)/d)**2)*tT[0]*c*factor)
    return ((4*a)/d)**(c) * factor * function

```python
from IPython.display import display, Math
CI = 10  # L/5
Area = 0.01  # m^2
mbar_to_Pa = 100
K = 0.22
p_atm = 1.01e5  # Pa
m_to_ce = 100
L_to_m = 1e-3
kB = 4.14e-21 #7
sccm_DO = np.percentile(flat_samples[:,0]@('C'+mbar_to_Pa@'L_to_m')@('K'+p_atm@Area)@'m_to_ce')**2, [16, 50, 90])
sccm_Ea = np.percentile(flat_samples[:,1], [16, 50, 90])
print("DO [ce^2/s]:", sccm_DO)
print("Activation energy [e]": sccm_Ea)
print(flat_samples[:,0])
DO [ce^2/s]: [0.00001630 0.00219349 0.00001330 0.00001330]
Activation energy [e]: [7160.2278465 7872.3714613 8593.34487626 7872.3714613 8593.34487626 0.00851097 0.122441 0.08407665 0.07596504 0.13876767 0.1136024]
```
Current Data

Successful VESPEL fits - achieved by Ishan Narra (Undergraduate)

Epoxy Measurement anomalies

Teflon fitting analysis
Current Pitfalls

- Data is not fitting to our model
- Potential background interference?
- Confounding factors in our setup?
Revisiting the Experimental Setup

*Image sourced from:

*Image sourced from:

\[ V(t) = V_0 \frac{1 - e^{-t/RC}}{1} \]

Step Function Smoothing with Different Tau Values

- Smoothed Step Function (tau=1000)
- Smoothed Step Function (tau=2000)
- Smoothed Step Function (tau=5000)
- Smoothed Step Function (tau=7500)
- Smoothed Step Function (tau=10000)

Mean temp data

\[ V(t) = V_0 e^{-t/2RC} \]
Misunderstanding the Timescale of the Sample?

- Implementing the entire diffusion model to the code.
Questions?