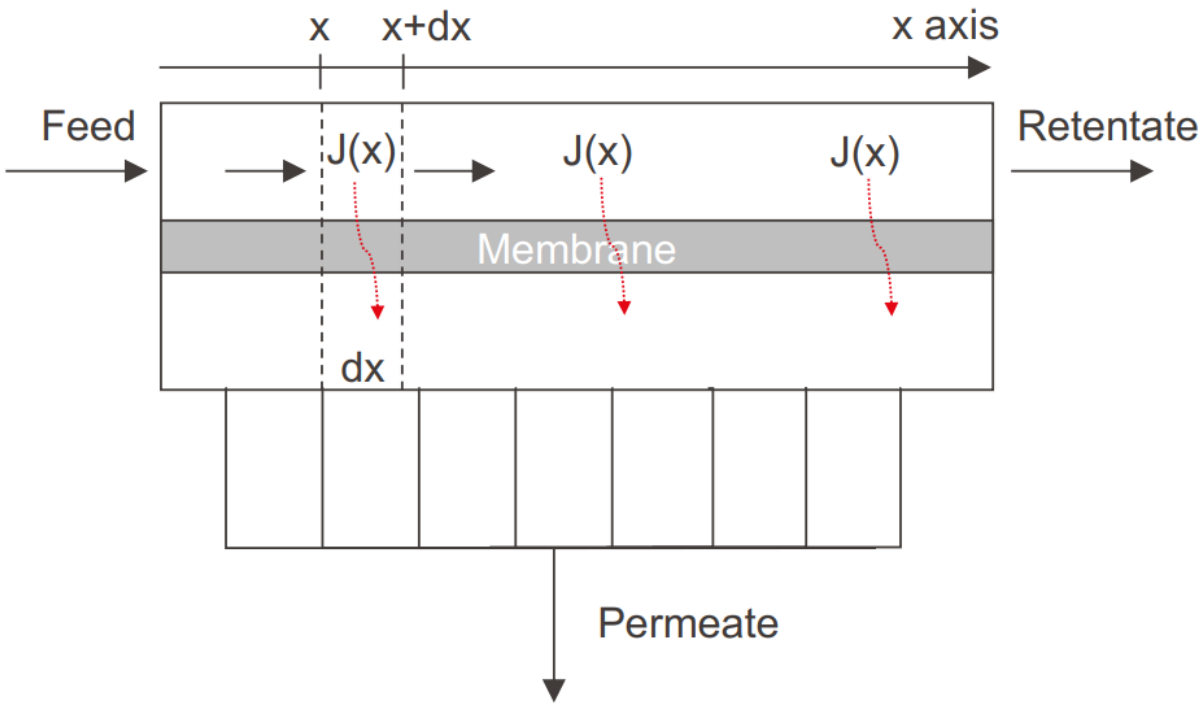
Modelling Permeate Pressure Effects on Performance of Gas Separation

Introduction

Influence of **permeate pressure** on single stage and

permeate pressure 2 on double stage in cross-current flow arrangement



Introduction

$$X'_{\mathrm{CO}_2,p}[i] = \frac{1 + (\alpha - 1)\left(\beta + X_{\mathrm{CO}_2,f}[i]\right) - \sqrt{\left[1 + (\alpha - 1)\left(\beta + X_{\mathrm{CO}_2,f}[i]\right)\right]^2 - 4 \cdot \alpha\beta(\alpha - 1)X_{C_{\mathrm{CO}_2,f}}[i]}}{2\beta(\alpha - 1)}$$

$$JX'_{\mathrm{CO}_2,p}[i] = J_{\mathrm{CO}_2}[i] = \boldsymbol{P}_{\mathrm{CO}_2}\left(P_{feed}\ X_{\mathrm{CO}_2,f}[i] - P_{perm}\, X'_{\mathrm{CO}_2,\mathrm{p}}[i]\right)$$

$$J\left(1 - X'_{CO_2,p}[i]\right) = J_{N_2}[i] = \boldsymbol{P}_{N_2}\left(P_{feed}\ \left(1 - X_{CO_2,f}[i]\right) - P_{perm}\ \left(1 - X'_{CO_2,p}[i]\right)\right)$$

$$Q_r[i] = Q_f[i] - \left(J_{\mathrm{CO}_2}[i] + J_{N_2}[i]\right)dA$$

$$X_{CO_2,r}[i]Q_r[i] = X_{CO_2,f}[i]Q_f[i] - J_{\mathrm{CO}_2}[i]dA$$

Introduction

Mass balance

$$Q_P + Q_r = Q_f \Longrightarrow Q_p X_{\mathrm{CO}_2,p} + Q_r X_{\mathrm{CO}_2,r} = Q_f X_{\mathrm{CO}_2,f}$$

$$dQ_r = -J dA \Longrightarrow d\left(Q_r X'_{\mathrm{CO}_2,r}\right) = -J X'_{\mathrm{CO}_2,p} dA$$

$$\alpha = \text{ selectivity } = \frac{P_{\mathrm{CO}_2}}{P_{N_2}}$$

$$\beta = \text{pressure ratio} = \frac{P_{\mathrm{perm}}}{P_{\mathrm{feed}}}$$

$$J X'_{\mathrm{CO}_2,p}[i] = J_{\mathrm{CO}_2}[i] = P_{\mathrm{CO}_2}\left(P_{feed}\ X_{\mathrm{CO}_2,f}[i] - P_{perm}\, X'_{\mathrm{CO}_2,\mathrm{p}}[i]\right)$$

$$J\left(1 - X'_{CO_2,p}[i]\right) = J_{N_2}[i] = P_{N_2}\left(P_{feed}\ \left(1 - X_{CO_2,f}[i]\right) - P_{perm}\ \left(1 - X'_{CO_2,p}[i]\right)\right.$$

$$\frac{J_{\mathrm{CO}_2}[i]}{J_{\mathrm{N}_2}[i]} = \frac{J X'_{\mathrm{CO}_2,p}[i]}{J\left(1 - X'_{\mathrm{CO}_2,p}[i]\right)} = \frac{P_{\mathrm{CO}_2}\left(P_{\mathrm{feed}}\ X_{\mathrm{CO}_2,f}[i] - P_{\mathrm{perm}}\ X'_{\mathrm{CO}_2,p}[i]\right)}{P_{\mathrm{N}_2}\left(P_{\mathrm{feed}}\ \left(1 - X_{\mathrm{CO}_2,f}[i]\right) - P_{\mathrm{perm}}\ \left(1 - X'_{\mathrm{CO}_2,p}[i]\right)\right.}$$

$$= \frac{\alpha\left(P_{\mathrm{feed}}\ X_{\mathrm{CO}_2,f}[i] - P_{\mathrm{perm}}\ X'_{\mathrm{CO}_2,p}[i]\right)}{P_{\mathrm{feed}}\ \left(1 - X_{\mathrm{CO}_2,f}[i]\right) - P_{\mathrm{perm}}\ \left(1 - X'_{\mathrm{CO}_2,p}[i]\right)}$$

Introduction

$$X'_{\mathrm{CO}_2,p}[i](P_{\mathrm{feed}}\ (1-X_{\mathrm{CO}_2,f}[i])-P_{\mathrm{perm}}(1-X'_{\mathrm{CO}_2,p}[i]))=\alpha(1-X'_{\mathrm{CO}_2,p}[i])\left(P_{\mathrm{feed}}X_{\mathrm{CO}_2,f}[i]-P_{\mathrm{perm}}X'_{\mathrm{CO}_2,p}[i]\right)$$

$$X'_{\mathrm{CO}_2,p}[i]\left(1-X_{\mathrm{CO}_2,f}[i]\right)-\beta\left(1-X'_{\mathrm{CO}_2,p}[i]\right))=\alpha(1-X'_{\mathrm{CO}_2,p}[i])\left(X_{\mathrm{CO}_2,f}[i]-\beta X'_{\mathrm{CO}_2,p}[i]\right)$$

$$\beta X'^{2}_{\mathrm{CO}_2,p}[i]+\left(1-X_{\mathrm{CO}_2,}[i]-\beta\right)X'_{\mathrm{CO}_2,p}[i]=\alpha\beta X'^{2}_{\mathrm{CO}_2,p}[i]+\left(-\alpha\beta-\alpha X_{\mathrm{CO}_2\mathrm{f}}[i]\right)X'_{\mathrm{CO}_2,p}[i]+\alpha X_{\mathrm{CO}_2,f}[i]$$

$$\beta(\alpha-1)X'^{2}_{\mathrm{CO}_2,p}[i]-\left(1+(\alpha-1)\left(\beta+X_{CO_2,f}[i]\right)\right)X'_{\mathrm{CO}_2,p}[i]+\alpha X_{\mathrm{CO}_2,f}[i]=0$$

$$X'_{\mathrm{CO}_2,p}[i]=\frac{1+(\alpha-1)\left(\beta+X_{\mathrm{CO}_2,f}[i]\right)-\sqrt{\left[1+(\alpha-1)\left(\beta+X_{\mathrm{CO}_2,f}[i]\right)\right]^2-4\cdot\alpha\beta(\alpha-1)X_{C_{\mathrm{CO}_2,f}}[i]}}{2\beta(\alpha-1)}$$

Building the model for a single-stage

Assumptions

- No variation along the width
- Permeance independent of length element
- Negligible pressure drops and concentration polarization
- Isothermal process
- Well mixed feed and no mixing in permeate channels
- CO_2-N_2 binary mixture

Building the model for a single-stage

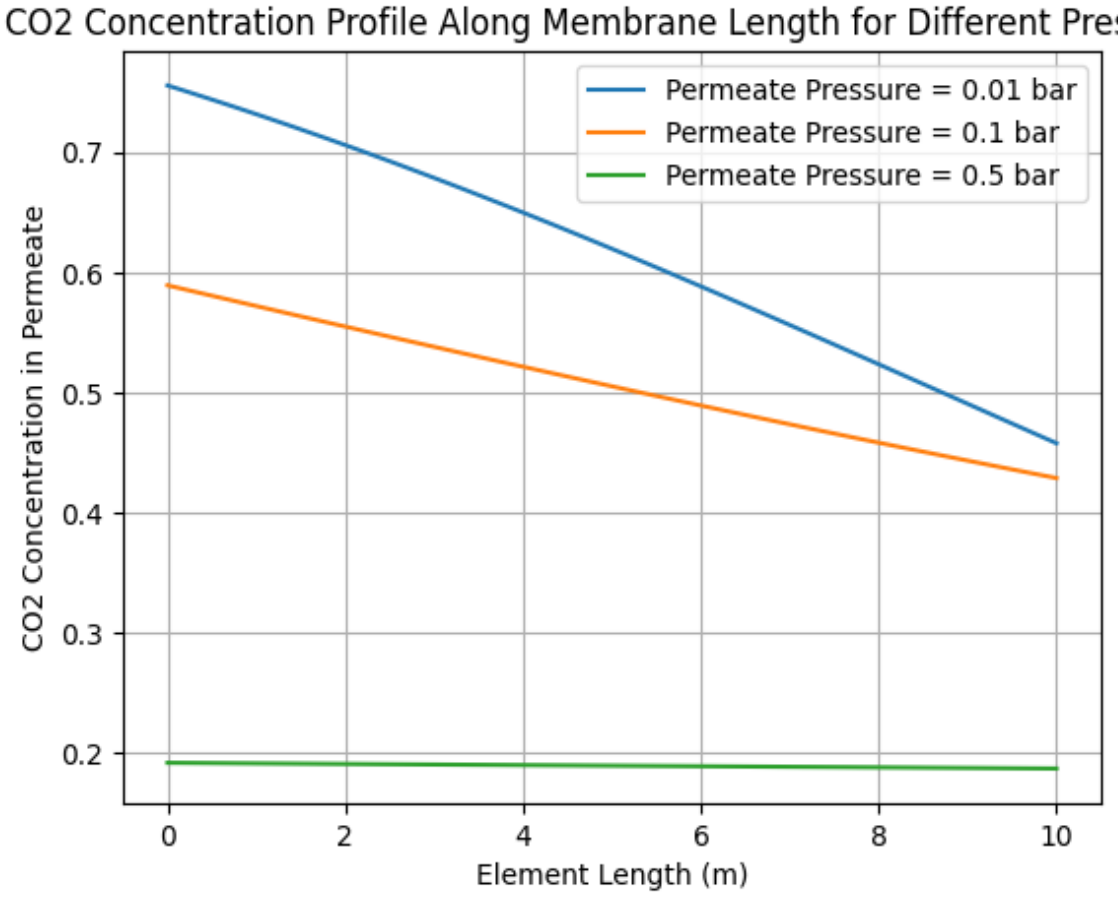
Inputs and parameters

- Permeance: 10000 GPU = 10000 * 3.3464e-10 mol/(m^2sPa)
- Selectivity α: 30
- Feed pressure P_f: 1 bar
- **Permeate pressure P_p: 0.01 bar ~ 0.5 bar**
- Feed concentration: 0.1
- Feed flow rate: 2.5 mol/s
- Membrane area A: 10 m^2

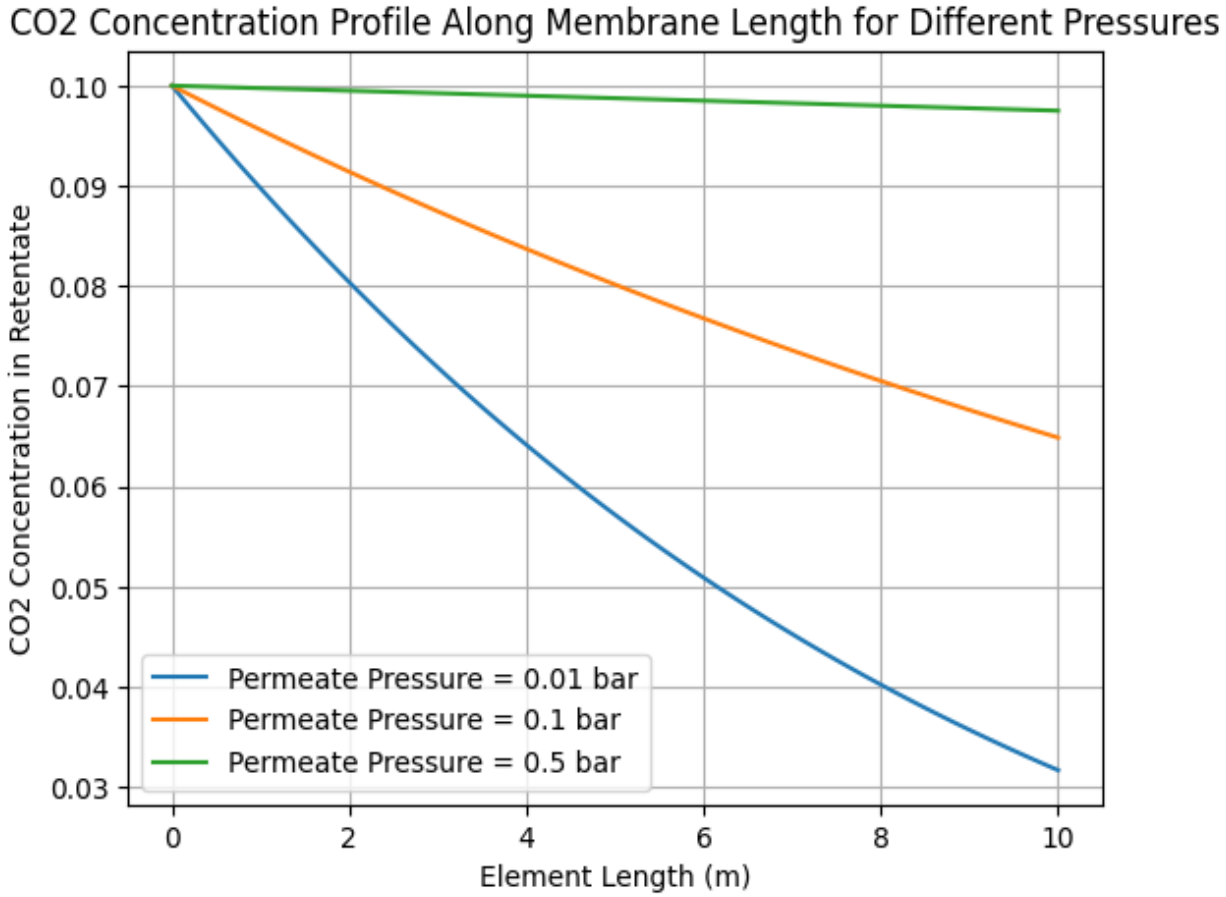
Building the model for a single-stage

```
N_elem = 100
Permeance = 10000 * 3.3464e-10  # mol/(m^2·s·Pa)
P_f, P_perm = 1e5, 10000  # Pa
X_f, Q_f_init, A = 0.1, 2.5, 10  # mol/s, m²
alpha = 30  # Selectivity
beta = P_perm / P_f

# Define single stage process
def single_stage(N_elem, Permeance, P_f, P_perm, X_f, Q_f_init, alpha, beta):
    # Initialize arrays
    X_CO2_f, Q_r, Q_f = np.zeros(N_elem), np.zeros(N_elem), np.zeros(N_elem)
    X_CO2_f[0] = X_f
    Q_f[0] = Q_f_init
    J_CO2, J_N2 = np.zeros(N_elem), np.zeros(N_elem)
    X_prime_CO2_perm_locations = np.zeros(N_elem)
    dA = A / N_elem

    # Single-stage process solver
    for i in range(N_elem):
        term = (alpha - 1) * (beta + X_CO2_f[i])
        discriminant = max(0, (1 + term)**2 - 4 * alpha * beta * (alpha - 1) * X_CO2_f[i])
        X_prime_CO2_perm = (1 + term - np.sqrt(discriminant)) / (2 * beta * (alpha - 1) + 1e-8)
        X_prime_CO2_perm = np.clip(X_prime_CO2_perm, 0, 1)
        X_prime_CO2_perm_locations[i] = X_prime_CO2_perm

        J_CO2[i] = Permeance * (P_f * X_CO2_f[i] - P_perm * X_prime_CO2_perm)
        J_N2[i] = (Permeance / alpha) * (P_f * (1 - X_CO2_f[i]) - P_perm * (1 - X_prime_CO2_perm))
        J_CO2[i], J_N2[i] = max(0, J_CO2[i]), max(0, J_N2[i])

        Q_r[i] = Q_f[i] - (J_CO2[i] + J_N2[i]) * dA
        Q_r[i] = max(0, Q_r[i])
        if i < N_elem - 1:
            X_CO2_f[i + 1] = (X_CO2_f[i] * Q_f[i] - J_CO2[i] * dA) / (Q_r[i] + 1e-8)
            Q_f[i + 1] = Q_r[i]
    # Outputs
    Q_ret_out, X_CO2_ret_out = Q_r[-1], X_CO2_f[-1]
    Q_perm_out = Q_f[0] - Q_ret_out
    X_CO2_perm_out = (X_f * Q_f[0] - X_CO2_ret_out * Q_ret_out) / (Q_perm_out + 1e-8)

    # Purity and recovery
    purity = X_CO2_perm_out
    recovery = Q_perm_out * X_CO2_perm_out / (Q_f[0] * X_f + 1e-8)
    return Q_ret_out, X_CO2_ret_out, Q_perm_out, X_CO2_perm_out, purity, recovery, X_prime_CO2_perm_locations, X_CO2_f, J_N2, J_CO2, Q_f

# Run single stage model
Q_ret_out, X_CO2_ret_out, Q_perm_out, X_CO2_perm_out, purity, recovery, X_prime_CO2_perm_locations, X_CO2_f, J_N2, J_CO2, Q_f = single_stage(
    N_elem, Permeance, P_f, P_perm, X_f, Q_f_init, alpha, beta)
```

Results of single-stage

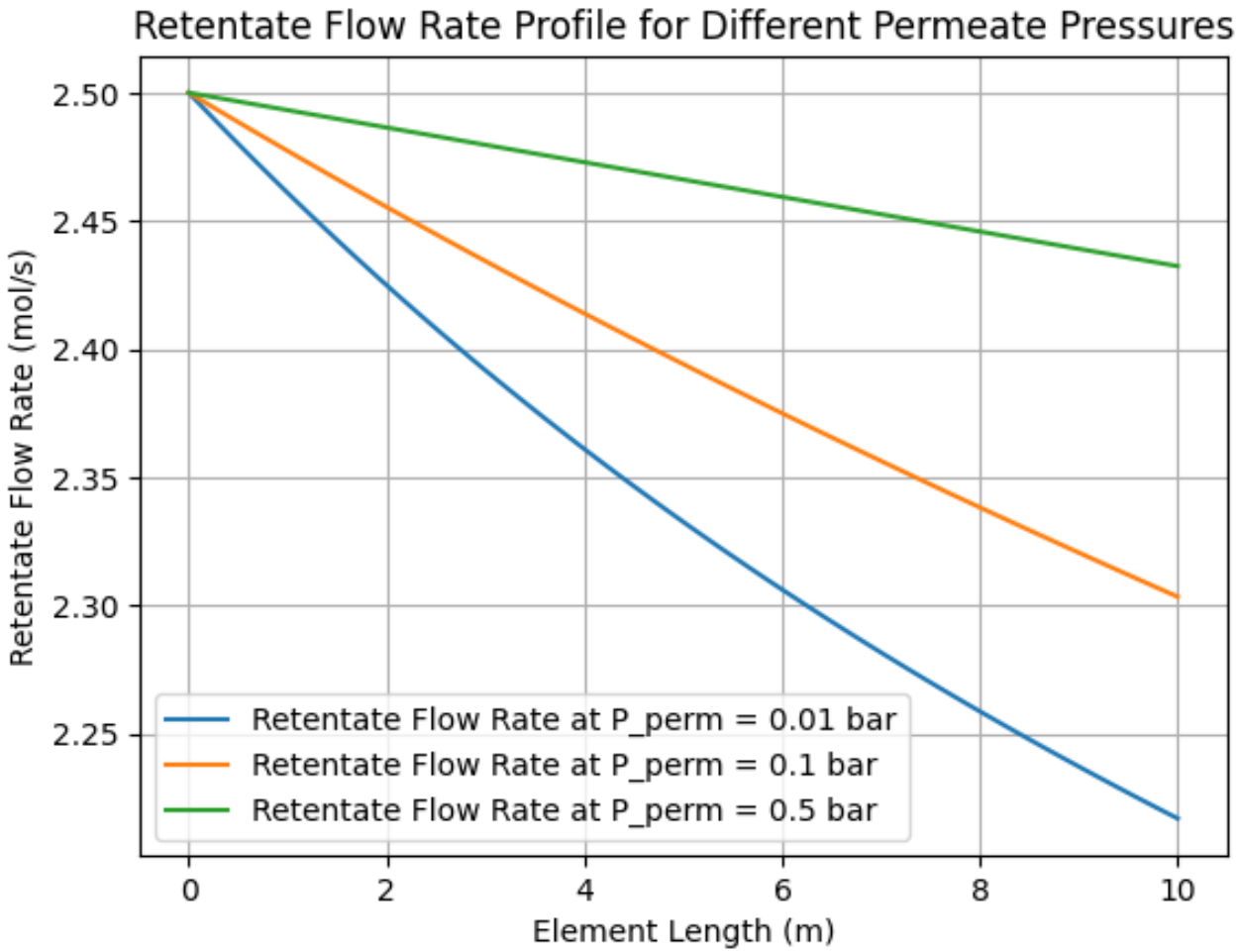


Permeate Concentration Profile
With different permeate pressures



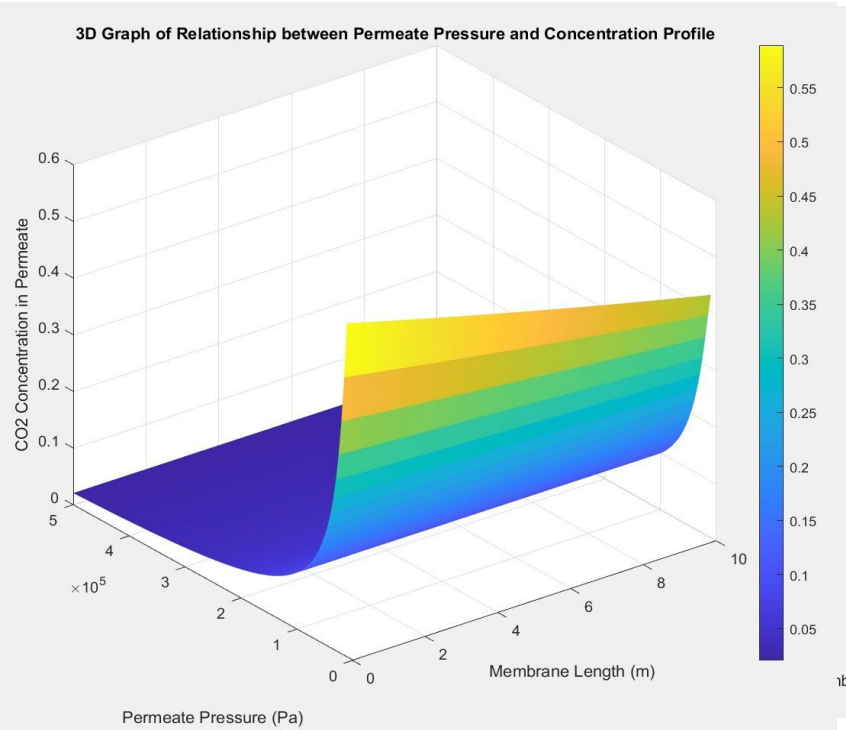
Retentate Concentration Profile
With different permeate pressures

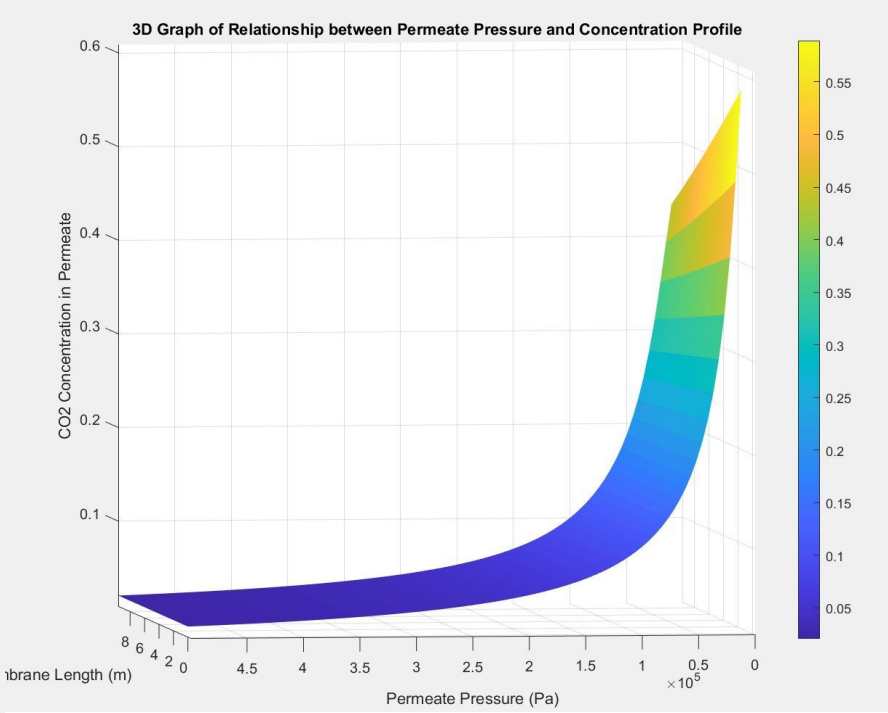
Results of single-stage

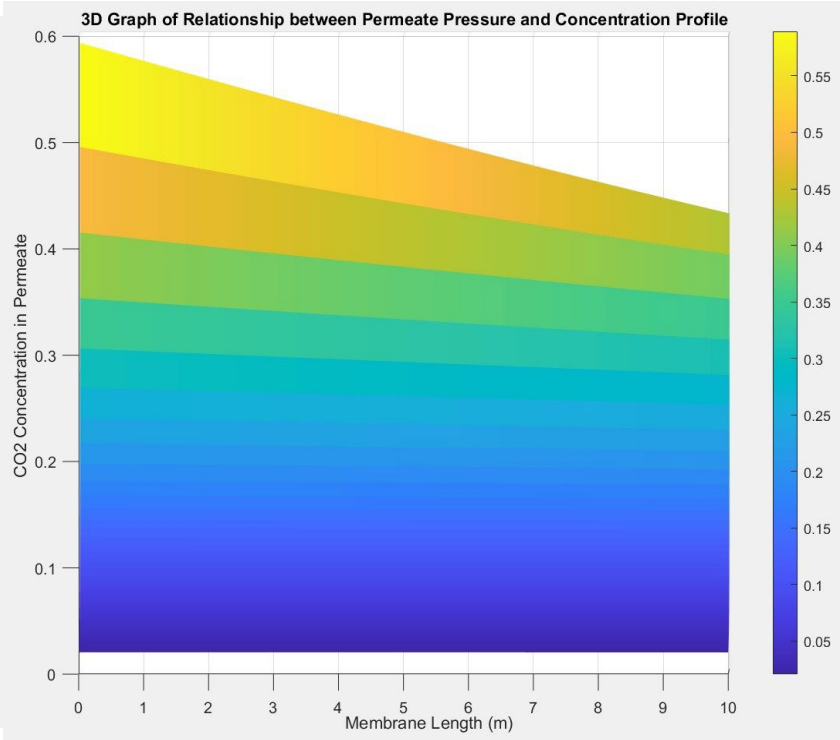


Retentate Flow Rate Profile for Different Permeate Pressure

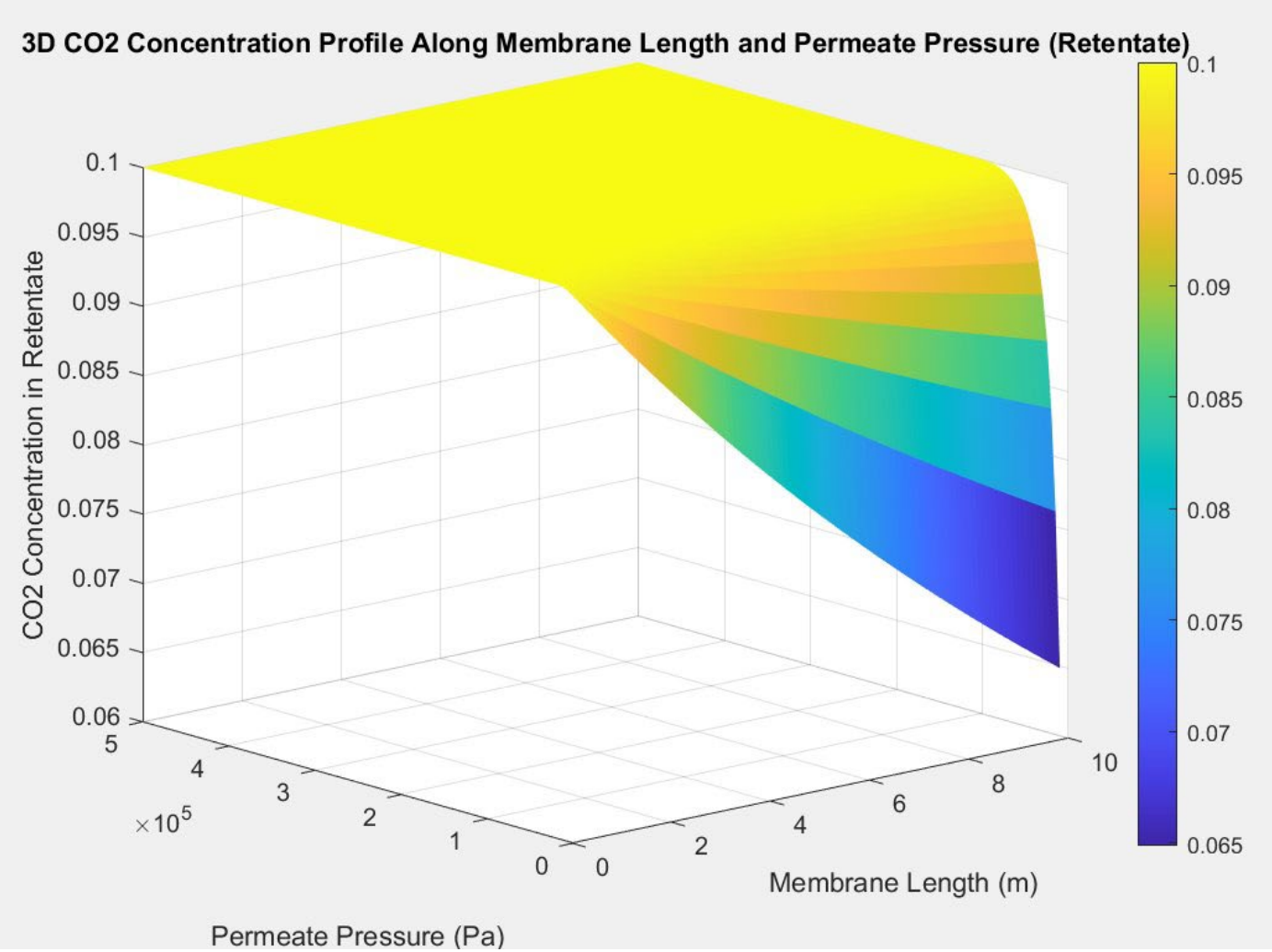
Results of single-stage



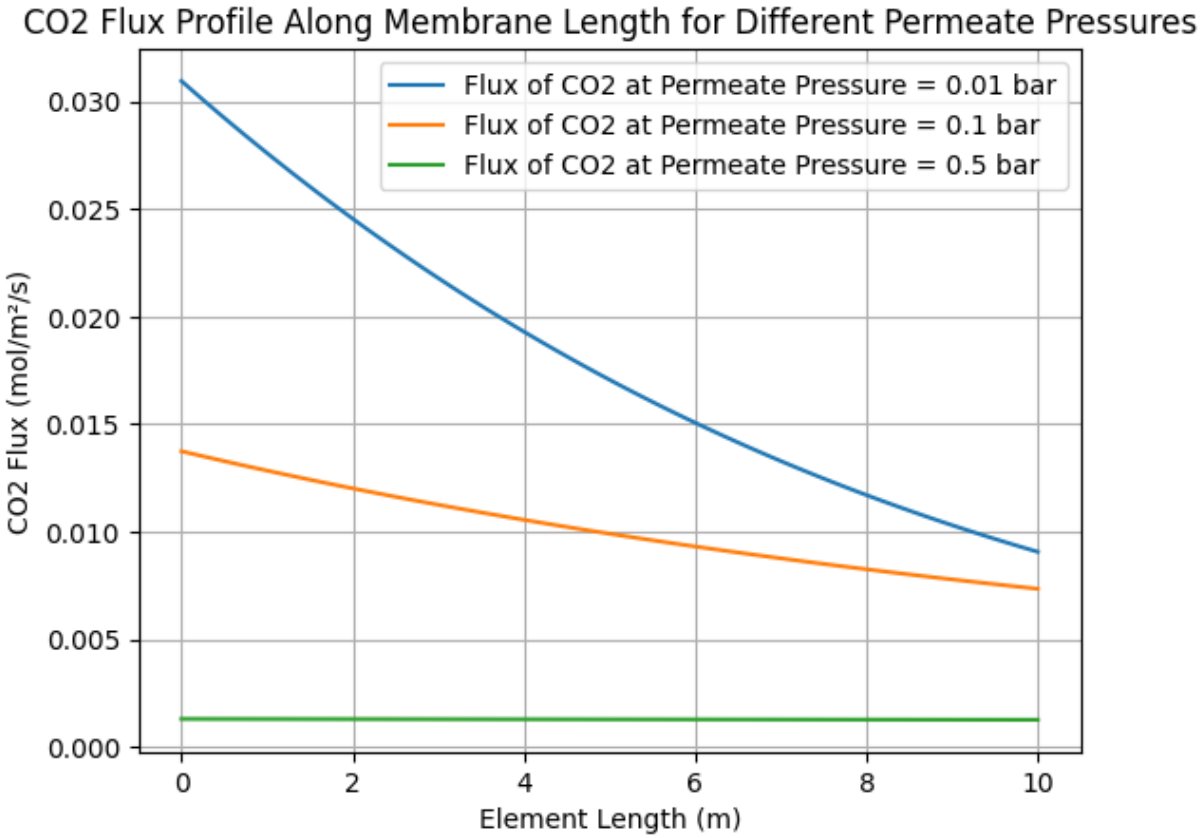




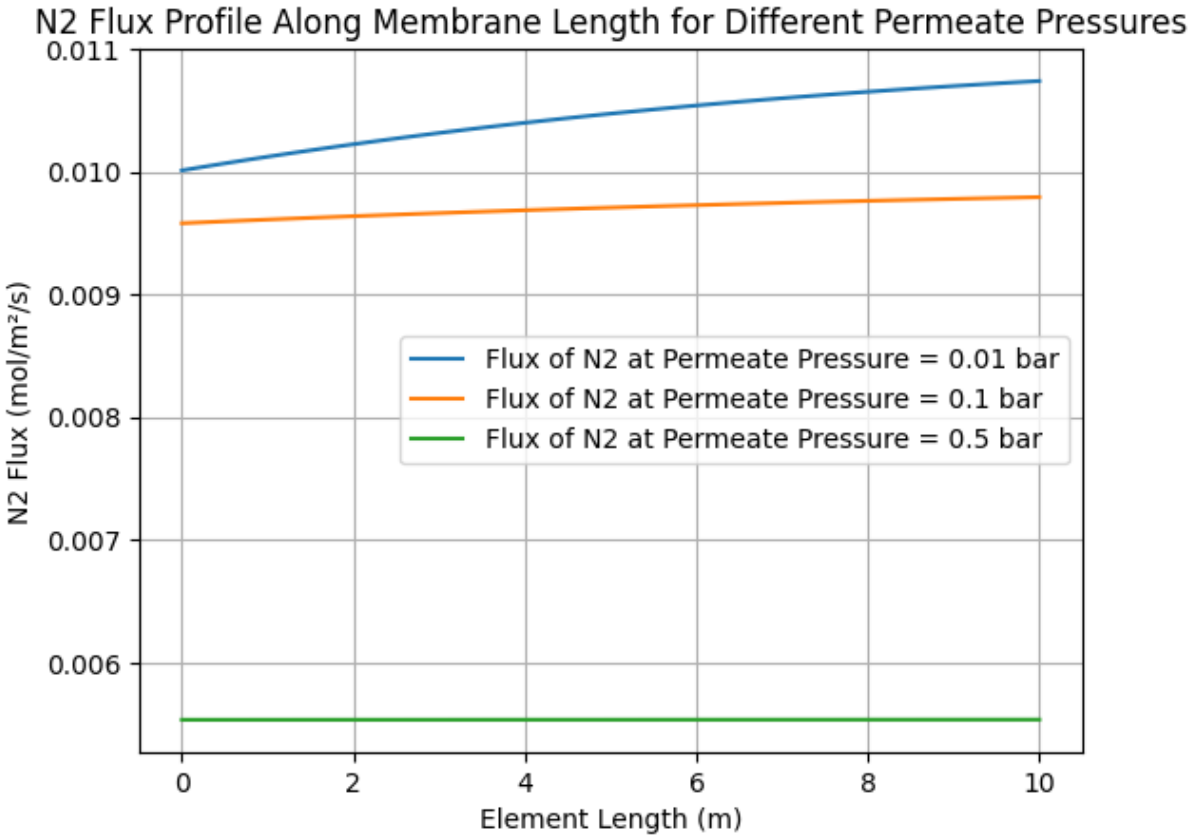
Results of single-stage



Results of single-stage

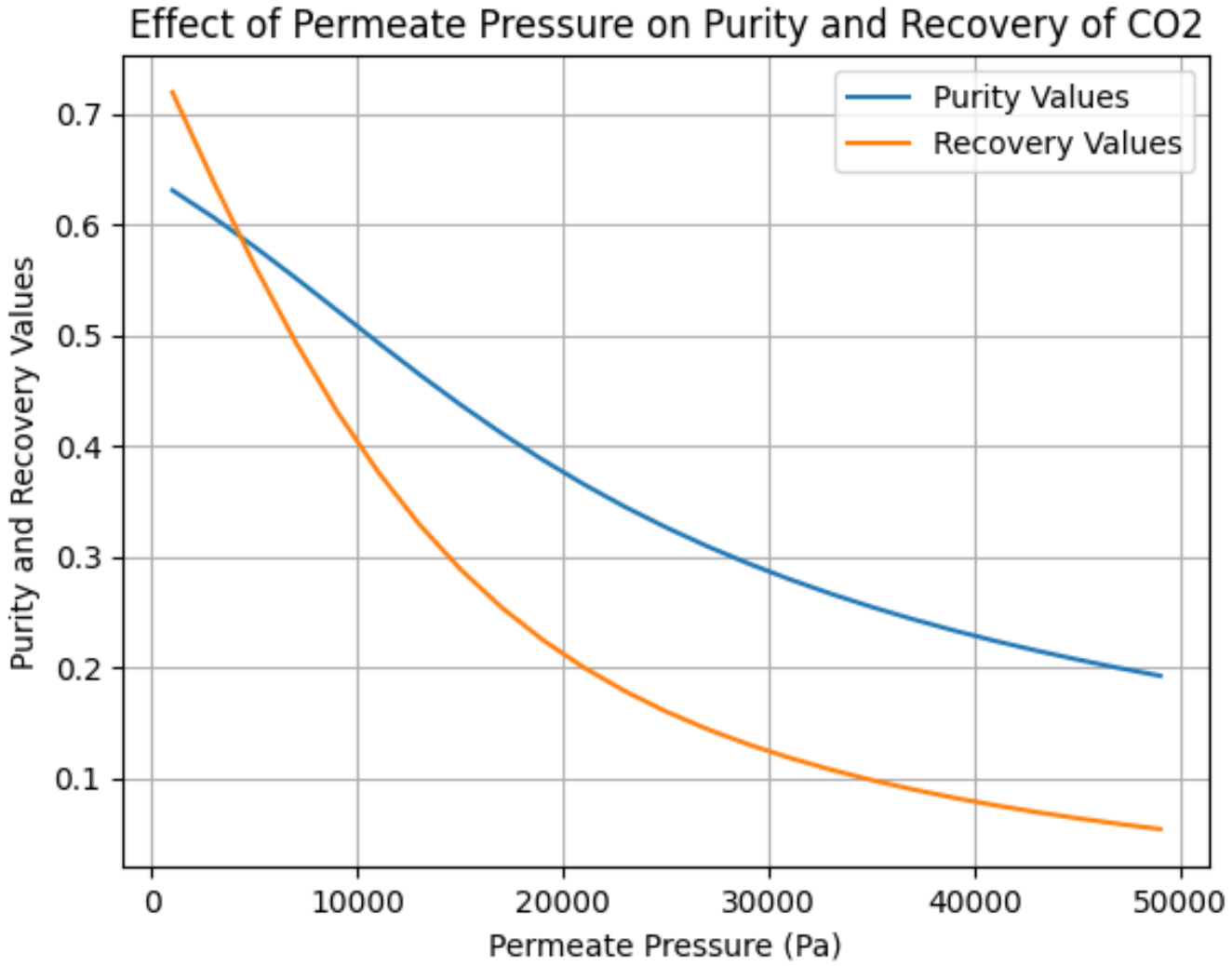


CO_2 Flux Profile Along Membrane Length for Different Permeate Pressure



N_2 Flux Profile Along Membrane Length for Different Permeate Pressure

Results of single-stage



Effect of Permeate Pressure on Purity and Recovery of CO_2

Building the model for a double-stage

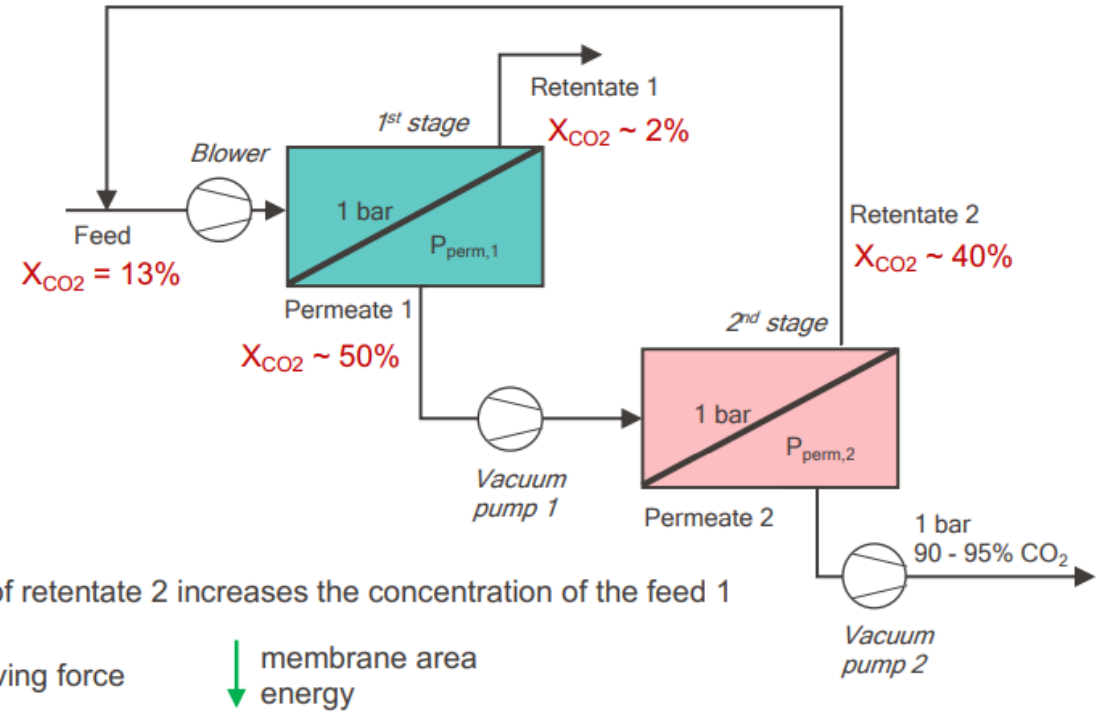
Inputs and parameters

- Permeance: 10000 GPU = 10000 * 3.3464e-10 mol/(m^2sPa)
- Selectivity α: 30
- Feed pressure 1 P_{f1}: 1 bar, Permeate Pressure 1 P_{p1}: 0.1 bar,
- Feed pressure 2 P_{f2}: 1 bar,
- **Permeate pressure 2 P_{p2}: 0.01 bar ~ 0.5 bar**
- Feed concentration 1: 0.1
- Feed flow rate 1: 2.5 mol/s
- Membrane area 1 A_1: 10 m^2, Membrane area 2 A_2: 5 m^2
- For Feed concentration 2 and Feed flow rate 2, We put Purity 1 and Permeate flow rate 1

Building the model for a double-stage

What's Different?

1. The output of Stage 1 is the input of Stage 2.
2. Parameters Changed: P_perm2 & A2.



Building the model for a double-stage

A Simple Validation Code

```
# Double Stage Process

N_elem = 100
Permeance = 10000 * 3.3464e-10  # mol/(m^2·s·Pa)
P_f, P_perm = 1e5, 20000  # Pa
X_f, Q_f_init, A = 0.50765, 0.1983136 , 5  # mol/s, m²
alpha = 30  # Selectivity
beta = P_perm / P_f

Q_ret_out2, X_CO2_ret_out2, Q_perm_out2, X_CO2_perm_out2, purity2, recovery2
, X_prime_CO2_perm_locations, X_CO2_f, J_N2, J_CO2, Q_f = single_stage(
    N_elem, Permeance, P_f, P_perm, X_f, Q_f_init, alpha, beta)

print("Retentate flow rate (Q_ret_out):", Q_ret_out2, "mol/s")
print("Retentate CO2 molar fraction (X_CO2_ret_out):", X_CO2_ret_out2)
print("Permeate flow rate (Q_perm_out):", Q_perm_out2, "mol/s")
print("Permeate CO2 molar fraction (X_CO2_perm_out):", X_CO2_perm_out2)
print("Purity of CO2 in permeate:", purity2)
print("Recovery of CO2:", recovery2)
print("Global Recovery: ",recovery2*0.402695)
```

```
Retentate flow rate (Q_ret
_out): 0.05784974607977469
4 mol/s Retentate CO2 mola
r fraction (X_CO2_ret_out)
: 0.02712650089660375 Perm
eate flow rate (Q_perm_out
): 0.1404638539202253 mol/
s Permeate CO2 molar fract
ion (X_CO2_perm_out): 0.70
55525534124844 Purity of C
O2 in permeate: 0.70555255
34124844 Recovery of CO2:
0.9844122647129623 Global
Recovery: 0.39641789693858
637
```

Building the model for a double-stage

Making it work:

```
Q_ret_out1, X_CO2_ret_out1, Q_perm_out1, X_CO2_perm_out1, purity1, recovery1, \
X_CO2_f1, X_CO2_p_prime1, J_CO2_1, J_N2_1, Q_f1 = single_stage(
    N_elem1, Permeance_CO2, Permeance_N2, P_feed1, P_perm1, X_f0, Q_f0, Selectivity, beta1, A1)
```

We will have to NUMBER the variables in each stage because now we have two stages, so here we are just using the same f unction single_stage() and then defining new variables with it.

The next step we print Q_ret_out1, X_CO2_ret_out, Purity, Recovery...etc to verify if our stage 1 calculation matches the pre vious calculations.

```
P_perm2_values = [1e3, 2e4, 5e4]  # 0.01 bar, 0.1 bar, and 0.2 bar 0.5 bar in Pa
for P_perm2 in P_perm2_values:
    beta2 = P_perm2 / P_feed2

    # Second Stage
    Q_f0_2 = Q_perm_out1  # Feed flow rate for second stage is permeate from first stage
    X_f0_2 = X_CO2_perm_out1  # Feed CO2 fraction for second stage is permeate fraction from first stage

    Q_ret_out2, X_CO2_ret_out2, Q_perm_out2, X_CO2_perm_out2, purity2, recovery2, \
    X_CO2_f2, X_CO2_p_prime2, J_CO2_2, J_N2_2, Q_f2 = single_stage(
        N_elem2, Permeance_CO2, Permeance_N2, P_feed2, P_perm2, X_f0_2, Q_f0_2, Selectivity, beta2, A2)
```

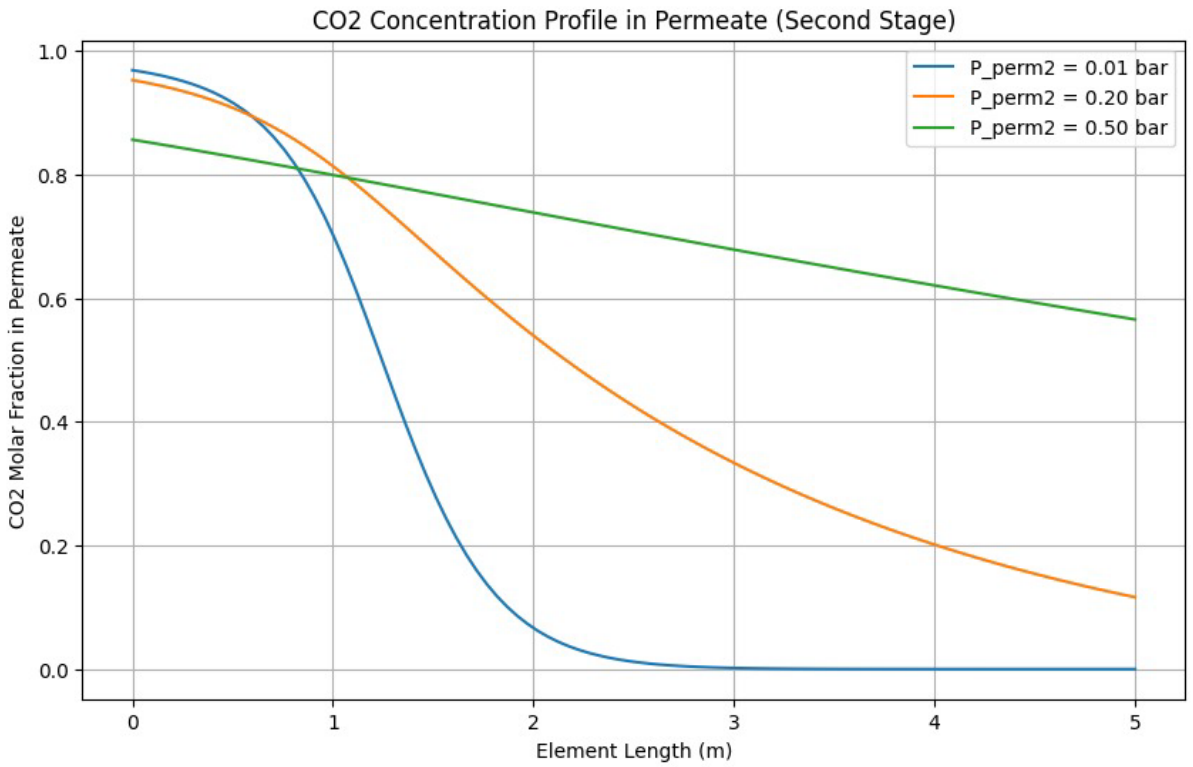
Building the model for a double-stage
Making it Work (Continued)

Here's some final touches we would need so that we have a list of values that vary with the distance. And th at for loop was meant for us to calculate conditions at different permeate pressures, so now we are all set, w e have the concentration at different location under different permeate pressures.

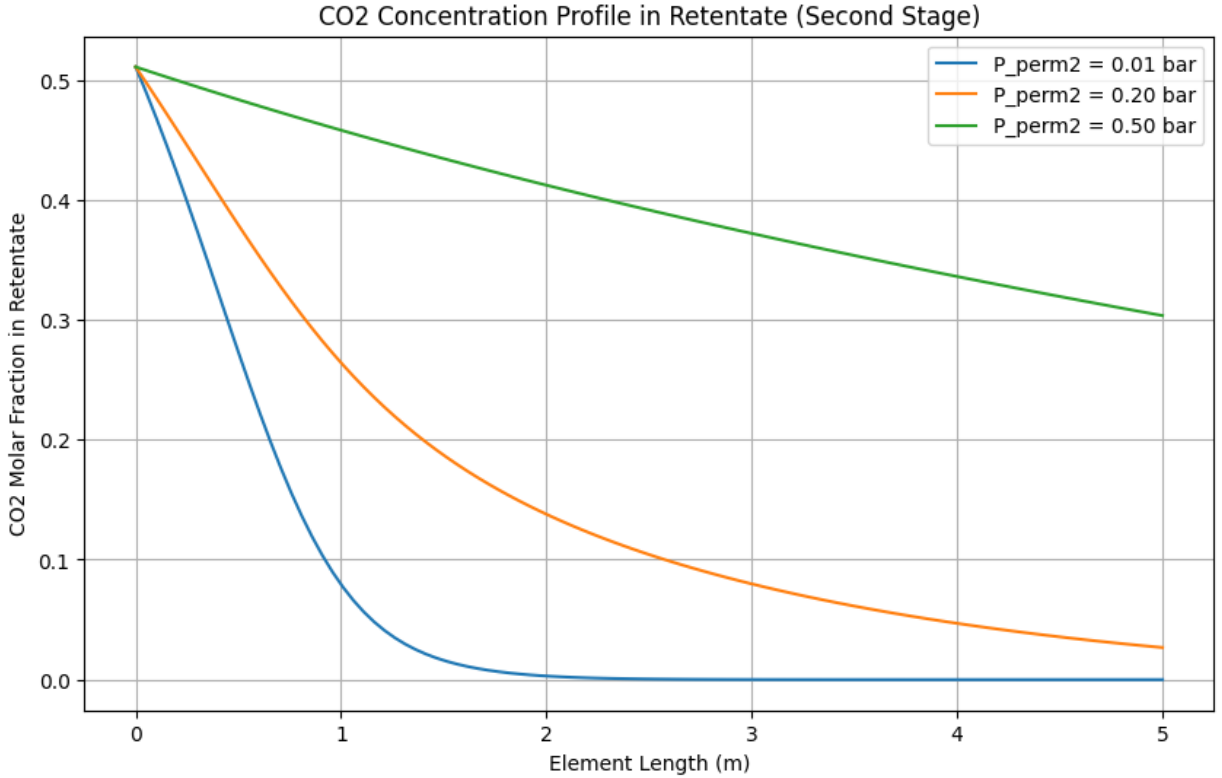
```
purity_values.append(overall_purity)
    recovery_values.append(overall_recovery)

    # Store profiles for plotting
    X_CO2_p_permeate_list.append(X_CO2_p_prime2)
    X_CO2_f_retentate_list.append(X_CO2_f2)
    J_CO2_list.append(J_CO2_2)
    J_N2_list.append(J_N2_2)
    Q_f_list.append(Q_f2)
```

```
  # Print Stage 2 results
    print(f"Second Stage Results with P_perm2 = {P_perm2/1e5:.2f} bar:
")
    print(f"Retentate flow rate (Q_ret_out2): {Q_ret_out2:.4f} mol/s")
    print(f"Retentate CO2 molar fraction (X_CO2_ret_out2): {X_CO2_ret_
out2:.4f}")
    print(f"Permeate flow rate (Q_perm_out2): {Q_perm_out2:.4f} mol/s"
)
    print(f"Permeate CO2 molar fraction (X_CO2_perm_out2): {X_CO2_perm
_out2:.4f}")
    print(f"Purity of CO2 in permeate (Stage 2): {purity2:.4f}")
    print(f"Recovery of CO2 (Stage 2): {recovery2:.4f}")
    print(f"Overall Purity: {overall_purity:.4f}")
    print(f"Overall Recovery: {overall_recovery:.4f}")
    print("-" * 40)
```

Results of double-stage

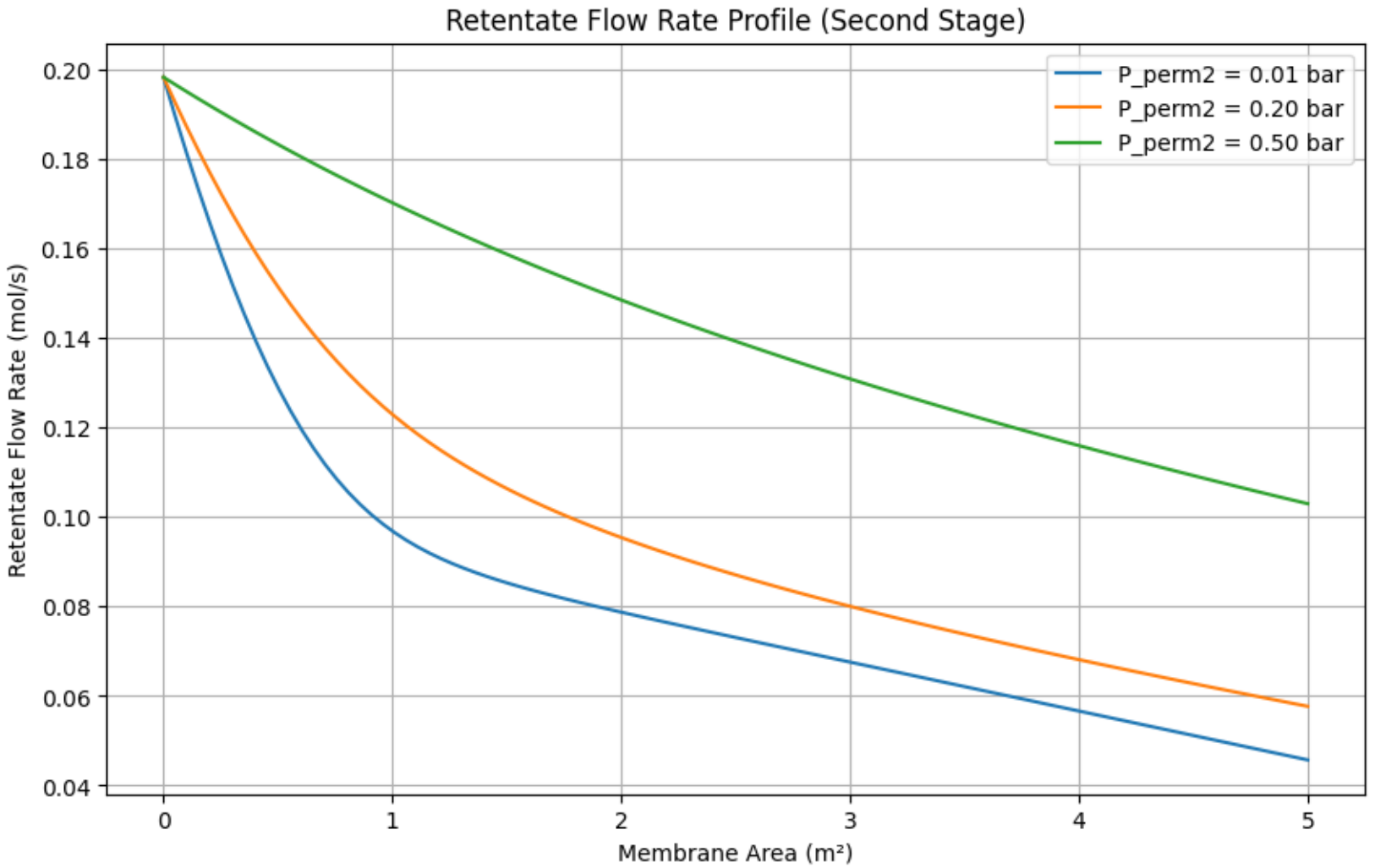


Permeate Concentration Profile
With different permeate pressures in Second Stage



Retentate Concentration Profile
With different permeate pressures in Second Stage

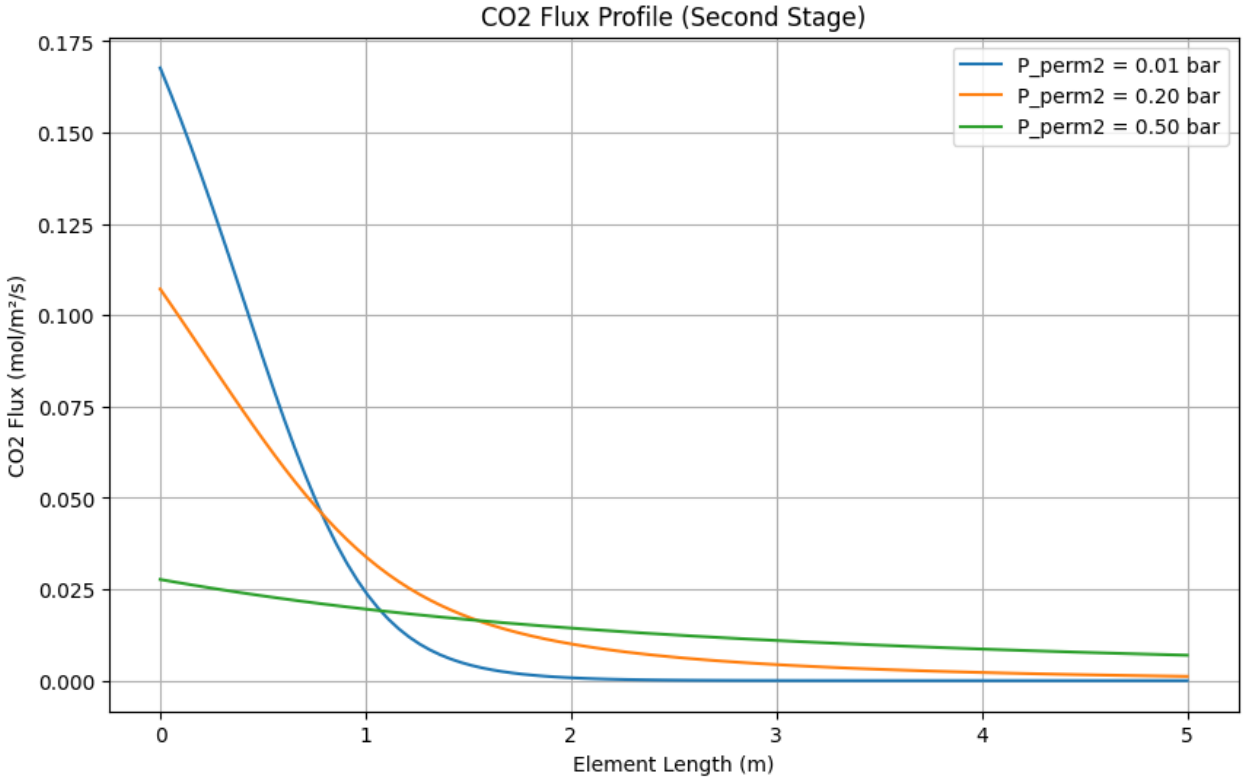
Results of double-stage



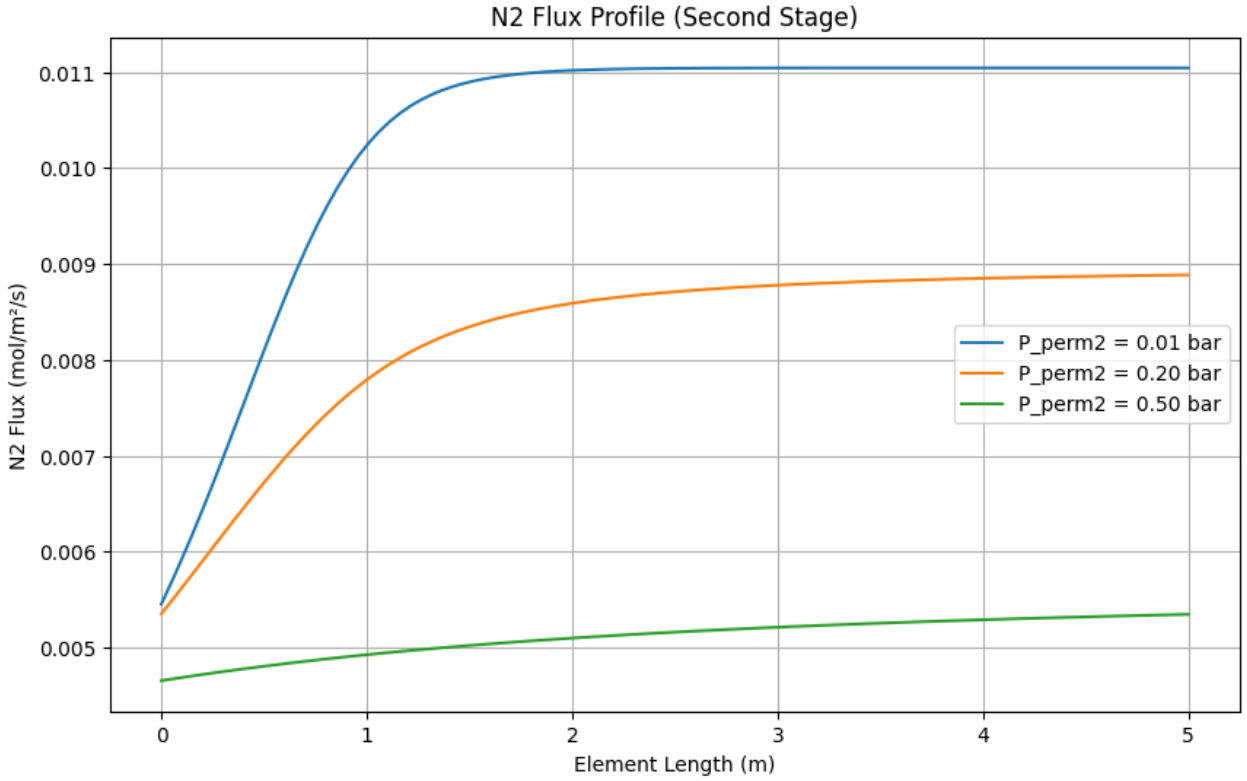
Membrane area ->

Retentate Flow Rate Profile for Different Permeate Pressure in Second stage

Results of double-stage

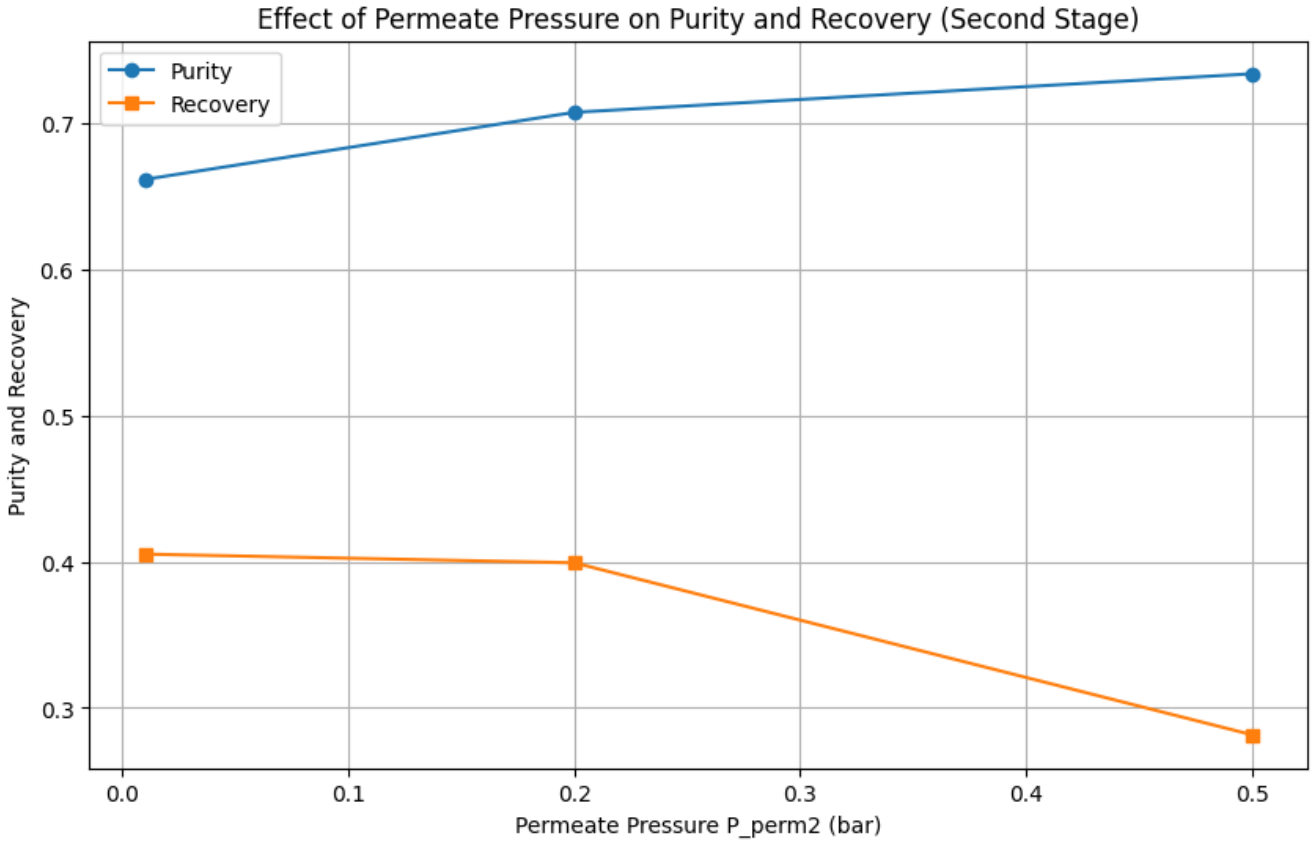


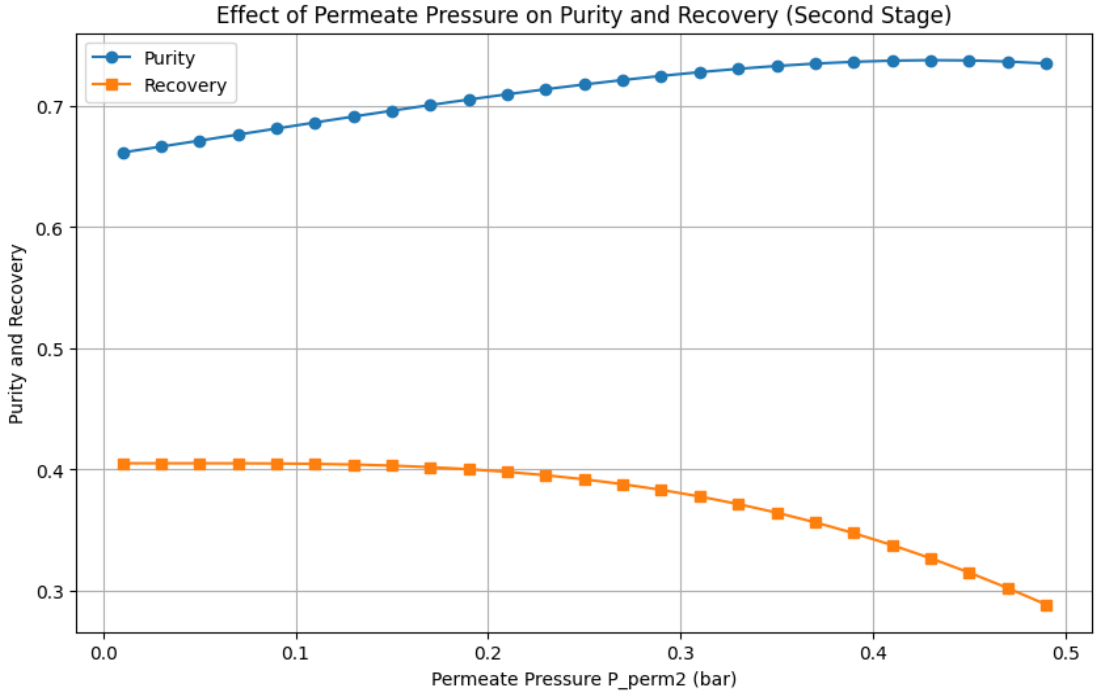
CO_2 Flux Profile Along Membrane Length for Different Permeate Pressure in Second Stage



N_2 Flux Profile Along Membrane Length for Different Permeate Pressure in Second Stage

Results of double-stage





Here we added a graph with more perme ate pressure values to get a better visualiz ation result. We can see a trade off betwe en recovery and purity which is an effect studied by Gabrielli et al.

APPENDIX (P_perm2 --> 100+ values)