



# Thermodynamics Group Project

---

## Column Pressure Optimization: A Modeling and Cost Analysis Approach

---

### Authors

Rane, A.<sup>a</sup>, Rasal, Y.<sup>a</sup>, Rathod, K.<sup>a</sup>, Raut, S.<sup>a\*</sup>, Singh, R.<sup>a</sup>, Ramji, R.<sup>a</sup>, Anand, R.<sup>a</sup>, Patel, R.<sup>a</sup>

### Affiliations

a- Department of Chemical Engineering, Institute of Chemical Technology, Mumbai, Maharashtra, 400019, India.

### Corresponding Author

*Email:* 23chesn.raut@ug.ictmumbai.edu.in (S. Raut)

---

**Abstract:** This study presents a systematic methodology for the optimization of a distillation column's operating pressure to minimize the total annualized cost (TAC) for the separation of Acetone and Acetonitrile binary system. The Experimental VLE data were regressed using the NRTL Model and the column was simulated at different operating pressures.

---

## 1. Introduction

Distillation is one of the most common separation processes used in chemical and petrochemical industries used to separate fluid mixtures either binary or multiple components. According to an article posted by Douglas C. White, it was noted that nearly 40-50% of the total energy consumed is consumed by distillation column<sup>[1]</sup>. Distillation separations account for about 3% of the total U.S. energy consumption. The distillation of crude oil into its fractions is very energy-intensive, requiring about 40% of the total energy used in crude-oil refining.

To achieve an optimized distillation column, the vapor-liquid equilibrium (VLE) data must be well described. Different types of thermodynamic models such as Margules, Redlich-kister, Van Laar, NRTL and Wilson are frequently used to correlate the experimental and simulated

data. Simulations are guaranteed to approximate the real separation as nearly as feasible when the model is fitted correctly.

Pressure optimization of distillation columns is essential as: if the operational pressure is too low, the driving force for heat exchange for the condenser and the heat duty for the reboiler decreases. In addition to this, lower pressures increase column height but decrease reboiler area.

On the flip side, if the operational pressure increases, condensation becomes easier but reboiler duty increases because of higher boiling points. Furthermore, higher pressures may reduce column height but increase reboiler area. Thus, to reach the best operating conditions, optimization is required.

In our work we have developed a simulation model to evaluate column performance under different operating pressures and the total annualized costs were calculated at each of the operating pressures. The optimization of this distillation column was done using python scripts and a thorough analysis was done to choose the optimum pressure of the distillation column design.

## **2. Literature Review**

For the binary mixture of Acetone and Acetonitrile only limited experimental data was available for the vapor-liquid equilibrium (VLE) data which has been cited <sup>[2]</sup>. In addition to this, no scientific literature is available which explores the optimisation of a distillation column of this given binary mixture. Thus, this report would be of importance and possibly contribute to literature as a preliminary first-pass distillation column design and thereby extending the limited literature available for this binary mixture.

The methodology of this report takes into account experimental data for thermodynamic model fitting, specifically the NRTL model, and running pressure-dependent column simulations to predict and optimise the total annualised cost of running the column.

### 3. Experimental Data

Liquid-vapour equilibrium data are given for the systems acetone ~ nitromethane and acetone + acetonitrile at 45.00 °C. The data are used to calculate the excess free energy of mixing for these systems. The former system has a negative value of the excess free energy of mixing and shows a change in sign of the deviation of the activity of nitromethane from the modified Raoult's law equation value at a mole fraction of acetone of 0.36

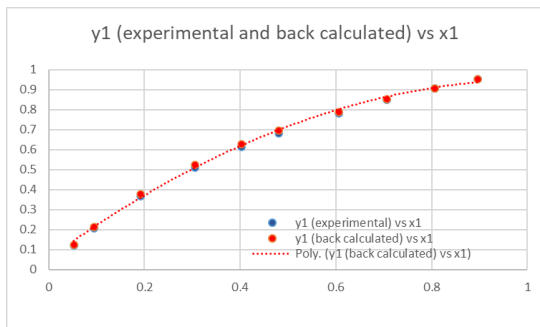
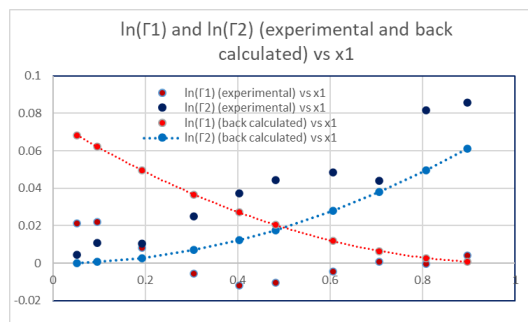
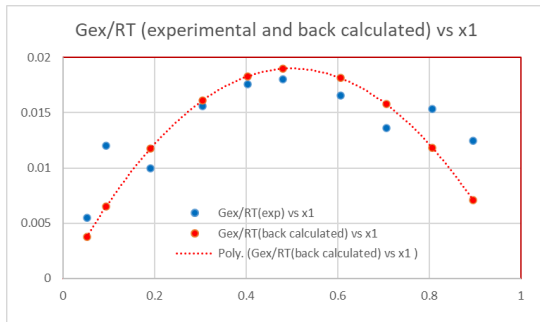
$x_1$	$y_1$	P (mmHg)	$\alpha$
0.052	0.120	225.3	2.48
0.095	0.206	239.9	2.47
0.192	0.367	268.5	2.44
0.305	0.510	302.7	2.37
0.403	0.611	331.7	2.33
0.481	0.682	355.2	2.31
0.606	0.781	393.2	2.31
0.706	0.849	423.6	2.33
0.807	0.904	454.1	2.25
0.896	0.951	481.4	2.25

## 4. Thermodynamic Models

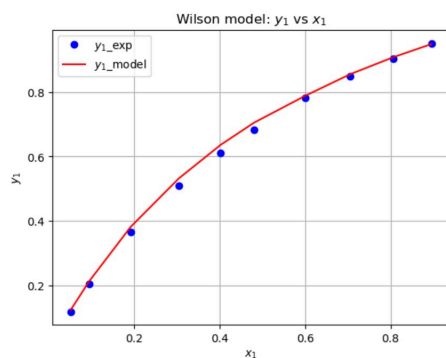
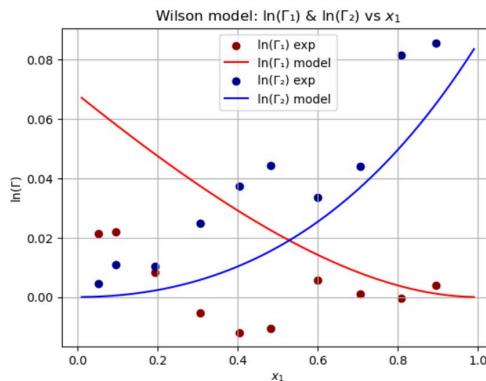
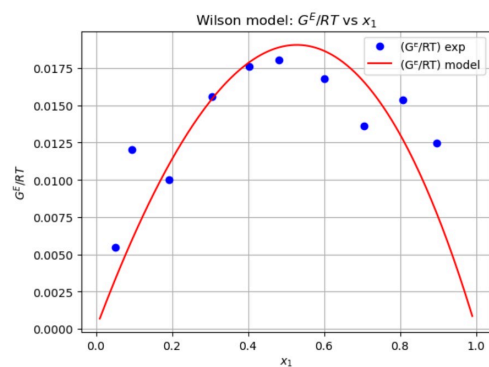
Fitted Model	Goodness of fit ( $R^2$ value)	Model Parameters
Margules one constant	0.9568 For $\ln(\Gamma_1)$ fit=0.4528 For $\ln(\Gamma_2)$ fit=0.9135 For $G^{\text{Ex}}/RT=0.7318$ $y_1=0.9997$ Decent fit	$A=0.076$
Redlich Kister	0.002 For $\ln(\Gamma_1)$ fit=0.4658471  For $\ln(\Gamma_2)$ fit=0.9159  For $G^{\text{Ex}}/RT=0.72392$ $y_1=0.9996$ Poor fit	$B=-0.0022$ , $A=0.0899$
Van Laar	0.0008 For $\ln(\Gamma_1)$ fit=0.4599  For $\ln(\Gamma_2)$ fit=0.9149  For $G^{\text{Ex}}/RT=0.7277$ $y_1=0.99989$ Decent fit	$\alpha=0.08427$ , $\beta=0.08208$
Wilson	For $\ln(\Gamma_1)$ fit=-6.6707  For $\ln(\Gamma_2)$ fit=0.4889  For $G^{\text{Ex}}/RT=0.3849$ $y_1=0.9973$ Decent fit	$\lambda_{12} = 1.1623$ $\lambda_{21} = 0.7809$
NRTL	For $\ln(\Gamma_1)$ fit=-0.20805  For $\ln(\Gamma_2)$ fit=0.23117  For $G^{\text{Ex}}/RT=0.7315$ $y_1=0.999$ Good fit	$\alpha = 12.391$ $\tau_{12} = 0.21393$ $\tau_{21} = 0.1298$

# Thermodynamic Model Fitting:

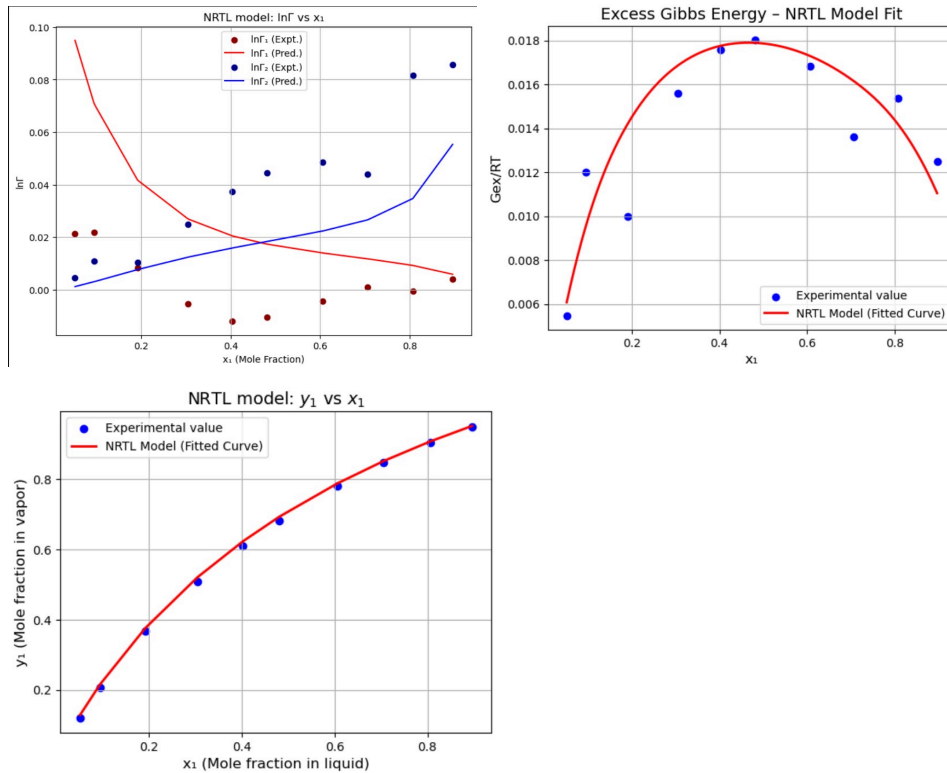
## 4.1 Margules



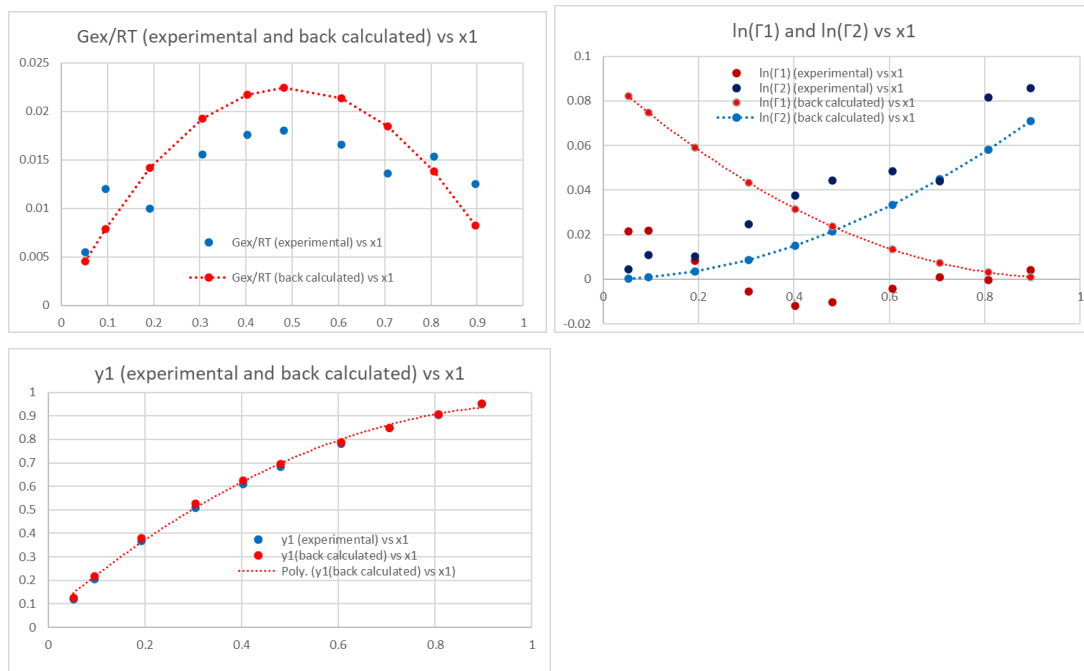
## 4.2 Wilson



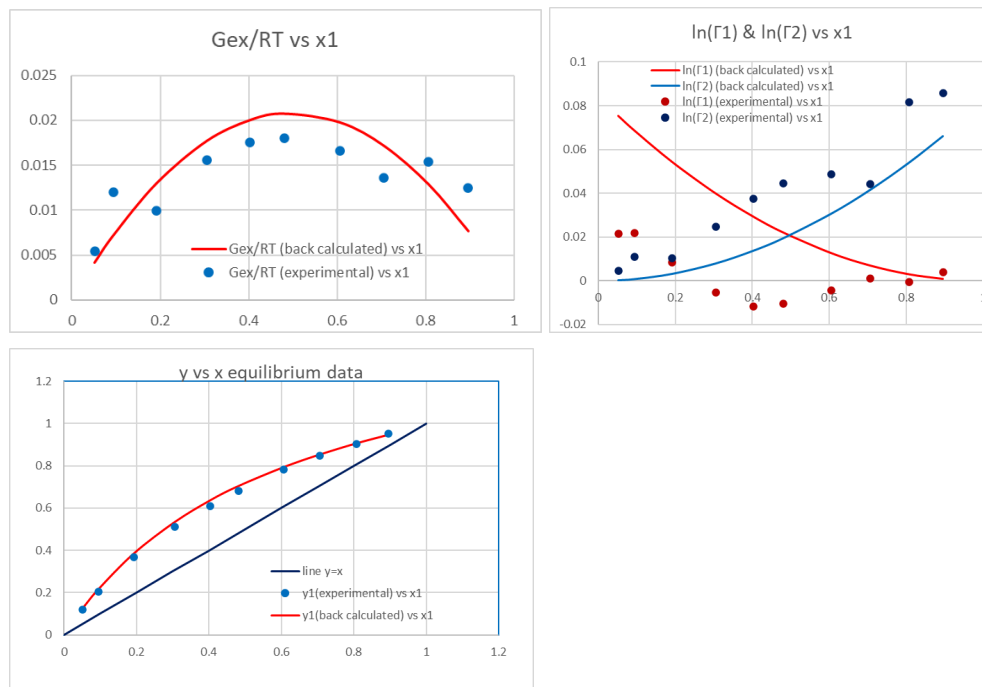
## 4.3 NRTL



## 4.4 Redlich Kister



## 4.5 Van Laar



### Notes and assumptions:

Based on the comparison of experimental data with model predictions, it is evident that none of the models tested provide an excellent fit to the data. However, among them, the NRTL model shows the relatively best agreement with experimental values. Although deviations are still present, the NRTL model captures the system behavior better than the other models considered, making it the most suitable choice in this context.

## 5. Column Design and Methodology: Bottom top approach calculations

### 5.1 Bottom top approach calculations

Objective: Given a liquid mole fraction  $x$  and an operating (stage) pressure  $P$  (isobaric stage), find the bubble-point temperature  $T_b$  and the corresponding vapour composition  $y$ . The activity model  $\Gamma_i(x, T)$  is provided by a fitted model (parameters obtained by regression to experimental  $G^{Ex}/RT$  data) and is treated here as a temperature-dependent function.

$$\ln \gamma_1 = x_2^2 \left[ \tau_{21} \left( \frac{G_{21}}{x_1 + x_2 G_{21}} \right)^2 + \frac{\tau_{12} G_{12}}{(x_2 + x_1 G_{12})^2} \right]$$

$$\ln \gamma_2 = x_1^2 \left[ \tau_{12} \left( \frac{G_{12}}{x_2 + x_1 G_{12}} \right)^2 + \frac{\tau_{21} G_{21}}{(x_1 + x_2 G_{21})^2} \right]$$

1. Saturation pressure (Antoine's) for each pure component:

$$(\log_{10} P_i^{sat} = A_i - \frac{B_i}{T(K) + C_i}) \text{ (in bar)}^{[9]}$$

	Acetone	Acetonitrile
A	4.42448	4.27873
B	1312.253	1355.374
C	-32.445	-37.853

2. Bubble-point (isobaric) condition:

- a.  $P = \sum_{i=1,2} x_i \Gamma_i(x, T) P_i^{sat}(T).$

- b. Equivalently, defining the function

$$F(T) = \sum_i x_i \Gamma_i(x, T) P_i^{sat}(T) - P,$$

- c. and solving  $F(T) = 0$  for  $T = T_b$ .

- d. Vapour composition (once  $T_b$  is found):

$$y_i = \frac{x_i \Gamma_i(x, T_b) P_i^{sat}(T_b)}{P}, \quad \sum_i y_i = 1.$$

3. Relation to experimental  $G^{Ex}/RT$  : experimental data gives

$$(G^{Ex}/RT)_{exp} = \sum_{i=1,2} x_i \ln(\Gamma_i^{exp}),$$

and the fitted model returns  $\Gamma_i(x, T)$  with explicit T-dependence (e.g. parameters converted to

$$\frac{U_{ij} \sim U_{jj}}{R} \text{ constants so } \tau_{ij}(T) = \frac{U_{ij} \sim U_{jj}}{RT}). \text{ Put values of } \Gamma_1 \text{ and } \Gamma_2$$



Numerical procedure:

- Build the function  $F(T)$  from the chosen  $\Gamma_i(x, T)$  model and  $P_i^{sat}(T)$ .
- Use a scalar root solver (fsolve from scipy.optimize is used) to find  $T_b$ . Practical choices:
  - Initial guess: use the previous stage temperature (if stepping stage-by-stage) or a tabulated pure-component bubble point as fallback.
  - Tolerances: require  $\frac{|F(T)|}{P} < 10^{-6}$  or  $|\Delta T| < 10^{-4} K$  for convergence.
  - Bounds: confine search between the pure-component bubble points (min and max of  $P_i^{sat}(T) = P$  solutions) to avoid unphysical roots.
- After convergence, compute  $\Gamma_i$  using (2) and check  $0 \leq y_i \leq 1$  and  $\sum_i y_i \approx 1$ .

Stage temperature profile (how  $T_{stage}$  is obtained):

- In a McCabe–Thiele style numerical stepping, for each liquid composition  $x$  encountered the bubble-point solve yields  $T(x; P)$  and  $y(x; P)$ . Collect  $T_{stage} = T(x_{stage}; P)$  for each stage. Using the previous stage solution as the initial guess substantially reduces solver iterations.
- Model temperature dependence: when the fitted model expresses parameters as Interaction energy constants ( $\frac{U_{ij} \sim U_{jj}}{R} = \tau_{ij \exp} \times T_{exp}$ ), implement  $\tau_{ij}(T) = \frac{U_{ij} \sim U_{jj}}{RT}$  to obtain  $\Gamma_i(x, T)$ . Treat the fitted parameters as calibrated at experimental temperature range and avoid excessive extrapolation.

### Algorithm:

Compute  $T_b$  by solving  $P = \sum_i x_i \Gamma_i(x, T) P_i^{sat}(T)$ ; then obtain vapour composition by  $y_i = x_i \Gamma_i P_i^{sat} / P$ ; repeat for each stage composition to build the stage temperature profile used in energy and height calculations.

**Notes/assumptions:**

We assume that the interaction energies used for  $\tau$  calculations of the are constant over the given temperature range thus making  $\Gamma_i$  a function of temperature.<sup>[3]</sup>

---

## 5.2 Material balance

Compute the two outlet flows and compositions when the feed flow/composition and any two of  $\{D, x_D, B, x_B\}$  are given. Used to close mass and component balances before stage/energy calculations.<sup>[6]</sup>

**Governing relations (linear):**

$$D + B = F; \quad D \times x_D + B \times x_B = F x_f$$

Solve the 2×2 linear system above. In matrix form:

$$\begin{bmatrix} 1 & 1 \\ x_D & x_B \end{bmatrix} \begin{bmatrix} D \\ B \end{bmatrix} = \begin{bmatrix} F \\ F x_f \end{bmatrix}$$

**Algorithm:**

1. Accept inputs:  $F, x_f$  and any two of  $\{D, x_D, B, x_B\}$  (must provide at least two independent quantities).
  2. Solve algebraically (closed-form) or by direct linear solve for the two unknowns.
  3. Validate physicality: require  $D \geq 0, B \geq 0, 0 \leq x \leq 1$  Raise error if violated.
-

## 5.3 Operating lines and Theory - $R_{min}$ and RR

Determine the minimum reflux and the rectifying-line intersection with the feed (q-line). These locate the feed entry / operating lines used by the stage synthesis routine and set the reflux ratio used for subsequent design.[6]

### Elements used

- q-line (feed thermal line) — linear in  $x$ : slope and intercept from feed thermal condition  $q$ .
- Equilibrium point on q-line: solve VLE at feed composition and the stage temperature to get  $(x_q, y_q)$ .
- Rectifying operating line (for reflux  $R$ ):

$$y = \left(\frac{R}{R+1}\right)x + \frac{x_D}{R+1}. [6]$$

- Minimum reflux (analytic relation): At the intersection of the q-line and the rectifying line the minimum reflux  $R_{min}$  satisfies

$$y_q = \left(\frac{R_{min}}{R_{min}+1}\right)x_q + \frac{x_D}{R_{min}+1}. [6]$$

Solving for  $R_{min}$  gives (algebraically)

$$R_{min} = \frac{x_D - y_q}{y_q - x_q} = \frac{y_q - x_D}{x_q - y_q}$$

**Selected operating reflux.** Use a design multiplier, e.g.  $R = 1.2 R_{min}$  (assumed margin).

**Intersection point of operating line and q-line.** For chosen  $R$  the intersection  $x_i$  satisfies

$$\frac{R}{R+1}x_i + \frac{x_D}{R+1} = m_q x_i + b_q,$$

where  $m_q, b_q$  are the q-line slope/intercept. Solve the single linear equation for  $x_i$ ; then  $y_i$  is obtained by substituting back into the operating line.

### Algorithm:

1. Evaluate VLE at feed (and stage temperature) to obtain  $(x_q, y_q)$ .
2. Compute  $R_{min}$  from the analytic relation above.

3. Set design reflux  $R = \kappa R_{min}$  (e.g.  $\kappa = 1.2$ ).
4. Solve the linear equation between rectifying line (with  $R$ ) and q-line to find  $x_i$  ;  
compute  $y_i$
5. Return  $\{x_q, y_q, R_{min}, R, x_i, y_i\}$ .

#### Notes/assumptions.

- Equilibrium  $y_q$  values require a VLE routine at the correct temperature/pressure; this is the only thermodynamic input.
  - The analytic expression for  $R_{min}$  assumes the intersection point lies on the equilibrium locus.
  - The intersection  $(x_i, y_i)$  is used as the starting point for stage counting (rectifying operating line meets q-line at feed location).
- 

## 5.4 Calculating the number of stages - composition and temperature on each stage at equilibrium

Stage counting produces the number of theoretical stages (total, rectifying, stripping) required to achieve a given separation for fixed feed and product specifications at a chosen top pressure. The routine implemented follows McCabe–Thiele style graphical stepping but performed numerically by iterating along the liquid–vapour equilibrium curve using the VLE routine `y_equm(x, P)` (returns vapour composition  $y$  and temperature  $T$  for a liquid mole fraction  $x$  at pressure  $P$ ). The computed stage list (liquid compositions `xs`, vapour compositions `ys`, and stage temperatures `Tstage`) is used downstream for packing height estimates, energy calculations and graphical estimation and further analysis.[7]

### A. Thermodynamic and design fundamentals

- **Equilibrium law:** Each theoretical stage achieves equilibrium; the vapour composition leaving stage is  $y_i = y_{eq}(x_i; P)$ , where  $y_{eq}$  is obtained from the VLE model(NRTL) at column pressure  $P$ . Temperature at that stage  $T_i$  is also

returned by the VLE routine.[4]

- Operating lines: The column is partitioned into rectifying and stripping sections separated at the feed (or feed intersection point). The rectifying line is governed by the reflux ratio and distillate composition  $x_D$ , while the stripping line is constructed from mass balances relative to the bottoms. In numerical stepping the code uses linear expressions connecting the known intersection point  $(x_i, y_i)$  to the top and bottom endpoints to map between vapour and liquid coordinates during each step.[6]
- Stepping (McCabe–Thiele principle): Starting from the bottom (composition  $x_B$ ), the algorithm alternately moves vertically to the equilibrium curve (compute  $y = y_{eq}(x)$ ) and horizontally to the current operating line (map  $y \rightarrow x$  using rectifying or stripping line) until the distillate composition  $x_D$  is reached. Each pair of vertical+horizontal moves corresponds to one theoretical stage.[6]
- Minimum stages (total reflux): The minimum theoretical stages  $N_{min}$  under total reflux are obtained by stepping on the equilibrium curve using the limiting operating line  $y = x$  (i.e., vapour and liquid compositions equal on the operating line). Numerically this is implemented by iterating  $x \leftarrow y_{eq}(x)$  starting from  $x_B$  until  $x \geq x_D$ . The number of iterations gives  $N_{min}$ . [6]

## B. Numerical control flow and logic (explanation of loop behaviour)

The numerical routine implements the following logic (here described without code):

Algorithm — Stage counting

1. **Input:** bottoms composition  $x_B$ , distillate composition  $x_D$ , feed intersection point  $(x_i, y_i)$ , column pressure  $P$ , and VLE callable  $y_{eq}(x; P)$ .
2. **Define mapping lines:**
  - a. Construct the rectifying mapping  $x = f_{rect}(y)$  as the linear function that passes through the two known rectifying-line points:  $(x_i, y_i)$  and  $(x_D, x_D)$  (the top operating point). This mapping takes vapour

composition  $y_{stage}$  and returns the corresponding liquid composition  $x_{stage}$  on the rectifying line.

- b. Construct the *stripping* mapping  $x = f_{strip}(y)$  as the linear function through  $(x_i, y_i)$  and  $(x_B, x_B)$  (the bottom reference).

These linear maps are the algebraic equivalent of the operating-line equations; they allow mapping from vapour composition to the liquid composition to step horizontally in the McCabe–Thiele sense.

3. **Initialize:** set current liquid composition  $x \leftarrow x_B$ . Set counters **total**, **rect**, **strip** = 0. Prepare empty lists **xs**, **ys**, **Tstage**.

4. **Stage stepping loop (for design reflux):** while  $x < x_D$ :

- Evaluate equilibrium: compute  $y, T = y_{eq}(x; P)$ . Append  $y$  (vapour) to **ys** and  $x$  (liquid) to **xs**; append  $T$  to **Tstage**.

**Decide which side to use:**

- a. If  $y < y_i$ : the vapour point lies below the feed intersection; use the *stripping* mapping: set new  $x \leftarrow f_{strip}(y)$ ; increment the **strip** counter.
  - b. Else (vapour at or above the feed intersection): use the *rectifying* mapping: set  $x \leftarrow f_{rect}(y)$ ; increment **rect**.
  - c. There is a small special-case guard: if  $y > x_D$  (vapour composition beyond distillate), treat as quick convergence by setting  $x \leftarrow y$  (this handles numerical overshoot).
  - d. Append new **x** to **xs**, update **total** += 1, and continue.
5. **Minimum stage count (total reflux):** initialize  $x \leftarrow x_B$  and loop: while  $x < x_D$ : compute  $y = y_{eq}(x; P)$ ; set  $x \leftarrow y$ ; increment **Nmin**. This loop counts stages under the limiting case when the operating line is  $y = x$ .

6. **Return** lists `xs`, `ys`, `Tstage` and stage counts (`total`, `rect`, `strip`, `Nmin`).

The resulting `Tstage` list therefore contains repeated stage temperatures for plotting.

- The feed intersection  $(x_i, y_i)$  partitions the McCabe–Thiele diagram: below the intersection the operating line used to map vapour to liquid is the stripping-line; above it the rectifying-line applies. The `y < y_i` test therefore selects the correct horizontal mapping.

### C. Double appending of temperatures and compositions (practical note).

- In a numerical implementation one often appends pairs of points per stage (liquid before equilibrium and liquid after operating-line horizontal move) so that the recorded arrays can be plotted as the classical stairs

### D. Assumptions, limitations and numerical safeguards

- **Equilibrium assumption:** Each theoretical stage is assumed to achieve vapour–liquid equilibrium (efficiency:  $\eta = 1$ ). Real packed stages deviate (efficiency:  $\eta \leq 1$ ); one must convert theoretical stages to packed height using stage/efficiency models (HETP or Murphree efficiencies).
- **Round-off & stage discretization effects:** Stage counts are integers; small parameter changes can produce step changes in `total` due to discrete stage rounding..

### E. Computational optimization recommendations

- Cache VLE calls (memoization) or use vectorized/batched evaluations.
  - Keep the number of nested loops to the minimum and ensure the inner loop is preferred by python.
  - Consider the `if else` - control flow & `Loop` based tradeoff for addressing computational and memory complexity.
  - For packed columns prefer an HETP-based height calculation rather than discrete stage stepping if height is the final objective — it avoids integer stage jumps.
-

## 5.5 Energy Balance of Column - internal and external flow rates, both duties

Purpose: The energy routine computes vapour and liquid flows needed by the column thermal balance, the duties on condenser and reboiler, the mass flows of utilities (steam, cooling water), and the associated hourly costs. These results feed the sizing (area, packing height), utility infrastructure design and economic objective.[5][6][7]

### Assumptions

- Constant molar (or mass) overflow in the column (radial): heat effects across stages are small relative to latent energy so vapour and liquid molar flows are approximately constant in each section. (no heat spots are formed).
- Negligible pressure drop and its effect on temperature across stages for thermal calculations (pressure fixed as **Pop**).
- VLE and temperatures returned by **y\_equm** are thermodynamically consistent and used to set **Tstage** profile.
- $\lambda$  of the fluid mixture is constant for and is taken at the reboiler composition and is a function temperature
- $c_p$  constant for the fluid mixture regardless of composition and is taken at feed conditions

### Energy balances:

#### 1. Reflux and boil-up flows

- Reflux liquid  $L = R \times D$
- Boilup vapour leaving reboiler  $V = L + D$  (constant molar overflow assumption in the rectifying section).
- Total vapour throughput through the condenser (mass condensed per hour) equals the sum of vapour generated plus any vapour present in feed (if feed is partially vapor):  $m_{vap,total} = V + (q) F_{vapor\ equivalent}$  where  $qF$  represents the vapor fraction of feed (mass basis) entering the column as vapor.

#### 2. Sensible heating of feed

- If the feed is sub-cooled or not at the boiling point, the reboiler must supply sensible heat to raise the feed to its bubble-point:  
$$Q_{sensible} = F C_p (T_{reboil} - T_{feed}).$$



### 3. Latent heat for vaporization

- Net latent energy required to generate boilup equals amount vaporized times the mixture latent heat:  $Q_{latent} = \dot{m}_{vaporized} \cdot \lambda_{mixture}$

### 4. Reboiler duty

- The reboiler duty supplies the sensible heat for feed and the latent energy to maintain boilup; a balanced expression commonly used:

$$Q_R = Q_{sensible} + Q_{latent} - Q_{feed\ latent}$$

Here  $Q_{feed\ latent} = F \times q \times \lambda_{acetonitrile}$  is the latent heat brought in by a vapor feed (if  $q < 1$ ) and is subtracted because that latent heat does not need to be supplied by the reboiler.

### 5. Steam mass flow (utility)

- Steam mass required ( $\text{kg} \cdot \text{hr}^{-1}$ ) is estimated from:  $m_{steam} = \frac{Q_R}{\lambda_{steam}}$

(accounting for steam condensate enthalpy and any flash losses; in preliminary design latent-to-latent conversion suffices).

### 6. Condenser duty

- The condenser removes the latent heat of the vapour condensed at the top:

$$Q_c = \dot{m}_{vap, total} \cdot \lambda_{acetone}$$

(If partial condensation or reflux subcooling is included, add sensible change terms.)

### 7. Cooling-water mass flow

- The required coolant mass flow ( $\text{kg} \cdot \text{hr}^{-1}$  or  $\text{kg} \cdot \text{s}^{-1}$ ) is obtained from energy balance:

$$m_{cw} = \frac{Q_c}{C_{p,w} \Delta T_{lm}}$$

where  $\Delta T_{lm}$  is the log-mean temperature difference between vapour saturation and coolant inlet/outlet:

$$\Delta T_{lm} = \frac{\Delta T_1 - \Delta T_2}{\ln(\Delta T_1 / \Delta T_2)}, \quad \Delta T_{1,2} = T_{sat} - t_{in,out}$$

## 8. Utility costs

- Steam cost (Rs·hr<sup>-1</sup>):  $cost_{steam} = \left(\frac{m_{steam}}{1000}\right) \times \frac{Rs}{MT\ of\ steam}$ .
- Cooling-water cost (Rs·hr<sup>-1</sup>): similar conversion from  $m_{cw}$  and water cost basis.

## 9. Reporting units

- The implementation uses J·hr<sup>-1</sup> as the internal energy unit and converts to kW for output: 1 kW = 1 kJ·s<sup>-1</sup> = 3.6×10<sup>3</sup> kJ·hr<sup>-1</sup>; so

$$Q\ (kW) = \frac{Q\ (J \cdot hr^{-1})}{3.6 \times 10^6}.$$

## Algorithm

**Input:** parameter object containing  $F, D, B, q, R$  and temperature / property constants.

**Output:** vector of energy results: vapour flow, condenser & reboiler duties (kW), utility mass flows and hourly costs.

Algorithm — compute\_energy (conceptual)

1. Compute reflux  $L = R \cdot D$ . Compute boilup  $V = L + D$ .
2. Estimate total vapour throughput  $V_{total} = V + q \cdot F$  (interpretation: add vaporized portion of feed if  $q < 1$  or subtract if  $q > 1$ ; exact sign depends on the chosen  $q$  convention).
3. Compute sensible heat to raise feed to reboil temperature:

$$Q_{sensible} = F C_p (T_{reboil} - T_{feed})$$

4. Compute latent heats: evaluate  $\lambda_{mixture}$  at appropriate temperature (or use an empirical correlation if mixture latent heat varies with  $T$ ). Then compute latent energy to generate the boilup and to condense at the top.
5. Assemble reboiler duty  $Q_R$  from sensible and latent terms, subtracting any latent carried in by feed vapour:

$$Q_R = Q_{sensible} + (m) \lambda_{mixture} - Fq \lambda_{mixture}$$

6. Compute steam mass:  $m_{steam} = Q_R / \lambda_{steam}$ .
  7. Compute condenser duty  $Q_C = \dot{m}_{vap, total} \cdot \lambda_{mixture}$ . Compute cooling-water mass by solving  $m_{cwc, w} \Delta T_{lm} = Q_C$ .
  8. Compute hourly costs from utility cost rates. Return a structured table of results (flows, duties, utility masses, costs).
- 

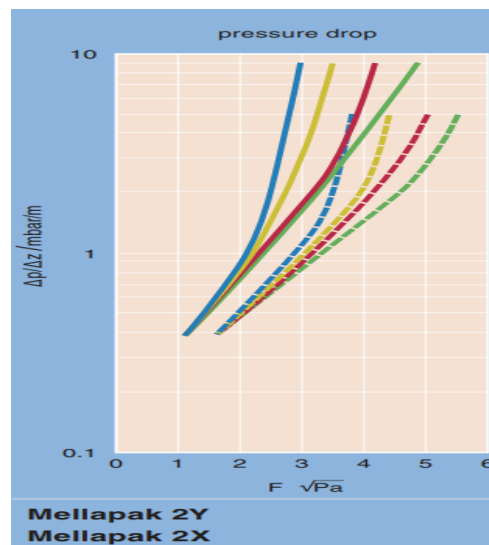
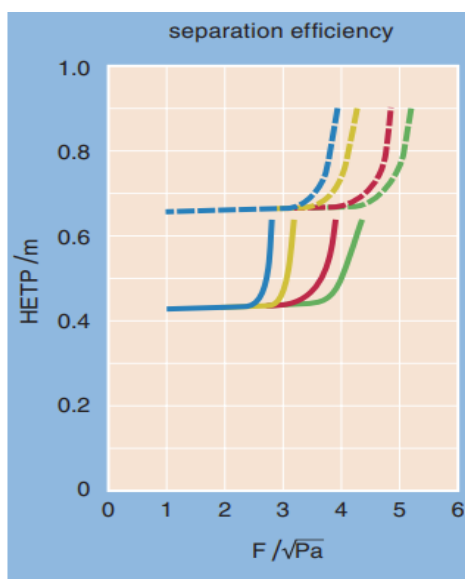
## 5.6. 2-D interpolation and data extraction for packing

### Data and objective:

Experimental curves for Mellapak-2X were digitized from Sulzer graphs at four nominal pressure (or operating) levels: **960, 400, 100, 50** (same units as the original plots). For each level we extracted two relationships as tabulated arrays of points: (1) pressure-drop per unit height  $\Delta p / \Delta z$  versus capacity factor  $F$ , and (2) **HETP** versus capacity factor  $F$ . The aim is to obtain continuous surfaces[8]

$$F = f(\Delta p / \Delta z, P), \quad HETP = g(F, P)$$

so design queries at arbitrary  $(\Delta p / \Delta z, P)$  &  $(F, P)$  inside the experimental range return interpolated values.



Mellapak 2X structured packing was chosen for this system because it offers high separation efficiency with very low HETP, while maintaining minimal pressure drop per stage—an essential factor since the column operates under sub-atmospheric conditions. Its corrugated geometry promotes efficient vapor–liquid contact, enhancing mass transfer and reducing column height requirements compared to trays or random packings. At the same time, its low pressure drop prevents excessive reboiler and condenser duties, making it highly suitable for energy-sensitive separations like this one, where operating pressure directly influences the overall Total Annualized Cost (TAC).

### Why RegularGridInterpolator (RGI)?

- The digitized measurements lie on a *regular*  $P$  grid (the four  $P$ -levels) and for each  $P$  we have values on an ordered set of  $F$  points *\implies* data that fits the regular-grid assumption.
- **RegularGridInterpolator** implements a tensor-product interpolation (linear, cubic, ...) that is vectorised, numerically robust, and supports ND interpolation. It is the modern, supported approach for gridded data (unlike deprecated 2-D helpers - `interp2d` from `scipy.interpolate`).
- RGI is simple, fast, and gives physically plausible interior values; it also allows controlled extrapolation or nearest-neighbour fallback when requested.

### Mathematics (bilinear / tensor-product linear interpolation)

For clarity we state the bilinear case (linear in each dimension) — RGI generalises this to ND.

Let the grid in  $\mathbf{F}$  be  $F_i$  and in  $\mathbf{P}$  be  $P_j$ . For query point  $(F, P)$  locate indices  $i, j$  such that  $F_i \leq F \leq F_{i+1}$  and  $P_j \leq P \leq P_{j+1}$ . Define local weights

$$\alpha = \frac{F - F_i}{F_{i+1} - F_i}, \quad \beta = \frac{P - P_j}{P_{j+1} - P_j}, \quad 0 \leq \alpha, \beta \leq 1.$$

If the four corner values are

$$v_{00} = v(F_i, P_j), v_{10} = v(F_{i+1}, P_j), v_{01} = v(F_i, P_{j+1}), v_{11} = v(F_{i+1}, P_{j+1}),$$

the bilinear interpolant is

$$v(F, P) = (1 - \alpha)(1 - \beta)v_{00} + \alpha(1 - \beta)v_{10} + (1 - \alpha)\beta v_{01} + \alpha\beta v_{11}.$$

This equals performing linear interpolation in  $\mathbf{F}$  at  $P_j$  and  $P_{j+1}$ , then linear interpolation in  $\mathbf{P}$  between those two results. RGI implements this tensor-product interpolation (and higher-order variants if chosen).

### Pre-processing & practical notes used in the work.

- Points were digitized from Sulzer performance charts (Sulzer Pumps Ltd.) graphs using webplotdigitizer, sorted and resampled so each  $\mathbf{P}$  level shares a common ordered  $\mathbf{F}$  vector (regular grid). If original  $\mathbf{F}$  nodes differed, values were regridded by 1-D interpolation along  $\mathbf{F}$  per  $\mathbf{P}$  to create the regular array.
- For strongly nonlinear responses (orders-of-magnitude changes) a log transform (e.g. interpolate  $\log(\text{HETP})$  vs  $\mathbf{F}, \mathbf{P}$ ) often improves accuracy; inverse transform the result. Use judiciously and report transform.

### Purpose and advantages of this approach.

- Produces continuous, differentiable surfaces for  $\mathbf{F}(\Delta p / \Delta z, \mathbf{P})$  and  $\text{HETP}(\mathbf{F}, \mathbf{P})$  required in hydraulic and height calculations.
- Avoids ad-hoc curve fits for each  $\mathbf{P}$  level (which can introduce bias); the tensor product interpolant respects the measured grid structure.
- Fast and vectorised: supports many design queries (sweeps, optimisation) with negligible overhead.

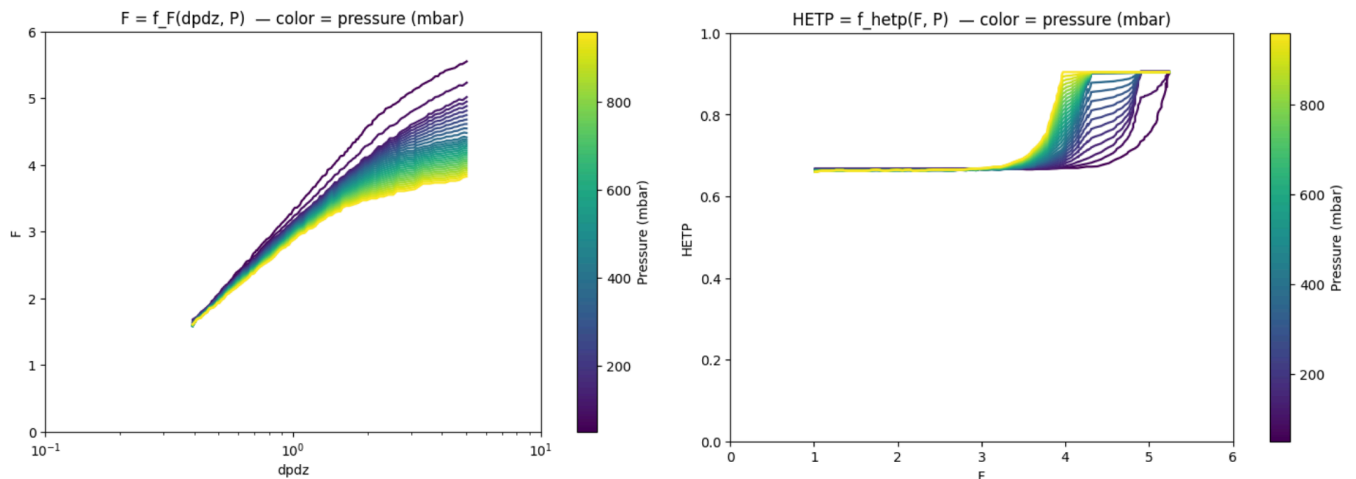
### Minimal algorithm / steps implemented

1. Digitize Sulzer plots for Mellapack-2X  $\rightarrow$  arrays  $\mathbf{F\_vals}$  (common  $\mathbf{F}$  grid) and  $\mathbf{P\_levels}=[960, 400, 100, 50]$ .
2. For each  $\mathbf{P\_j}$  collect  $\mathbf{F\_j}(\text{dpdz})$  and  $\text{hetp\_j}(\mathbf{F})$  on  $\mathbf{F\_vals}$ .

Construct interpolators:

```
from scipy.interpolate import RegularGridInterpolator
f_F_2d = RegularGridInterpolator((dpdz_grid, pressures),
F_grid, bounds_error=False, fill_value=None)
f_hetp_2d = RegularGridInterpolator((F_axis, pressures),
H_grid, bounds_error=False, fill_value=None)
```

3. (note ordering: axis0 = **P**, axis1 = **F**).
4. Query: given scalar or vector (**dpdz**, **P\_q**) produce **F** = **f\_F((dpdz, F\_q))** and **hetp** = **f\_hetp((F\_q, Pop))**.
5. Validate: cross-validate with withheld points or compute RMSE vs digitized data to quantify interpolation error.



## 5.7 All the other Column specification & internals - design and costs

### Area sizing (vapour capacity → cross-section)

#### Purpose

Convert a vapour mass flow rate into a column cross-sectional area and an equivalent circular diameter using a chosen design superficial vapour velocity (design velocity = **eff** \* **vmax**). This is the standard preliminary sizing used to avoid flooding and to set packing/tube areas .[6]

#### Governing relations (units explicit)

Convert mass flow to volumetric flow ( $\text{m}^3 \cdot \text{s}^{-1}$ ):  $V = \frac{m}{\rho_v \cdot 3600}$

Design superficial velocity used:  $u_{design} = eff \times v_{max}$

#### Algorithm (stepwise)

1. Compute volumetric flow  $\dot{V}$  from  $\dot{m}$  and  $\rho_v$ .
2. Choose design superficial velocity  $u_{design} = \text{eff} * v_{max}$ .
3. Compute  $A = \dot{V} / u_{design}$ .
4. Compute  $D = \sqrt{4A/\pi}$ .
5. Return **A** and **D**.

### Design notes & rationale (C&R)

- **v\_max** should be tied to flooding charts or allowable entrainment velocity for the considered system; C&R suggests using a fraction of the flooding velocity or literature correlation for allowable superficial velocity.
- **eff** (0.8–0.95) is a safety reduction to avoid operating close to the limiting velocity.
- Use vapour density at the appropriate local saturation temperature and pressure (Pop). Small errors in  $\rho_v$  linearly affect the area.

### Checks / sensitivity

- If **A** or **D** is unrealistically small, re-evaluate **rho\_v** and **v\_max**.
- For packed columns consider additional clearance for distributors and bypass; use effective hydraulic area rather than gross shell CSA if internals occupy space.

### Height calculation (HETP × stages + allowances)

#### Purpose

Estimate packed-bed height (and total column height) from the number of theoretical stages and packing performance (HETP). Also add practical allowances for distributors, freeboard and maintenance access. This follows Coulson & Richardson guidance for packed columns and distributor spacing.[6]

#### Algorithm (stepwise)

1. Compute capacity factor  $F = f_F(dp dz, P)$ .
  2. Compute  $HETP = f_{hetp}(F, P)$ .
  3. Compute  $H_{packing} = HETP * N_{total} * safety\_factor$   
( $safety\_factor \approx 1.05-1.2$ ; 1.1 used as conservative modest allowance).
  4. Compute distributor and allowance heights and sum to  $H_{column}$ .
  5. Return both heights.
- 

### **Packing costs (volume + distributor cost)**

#### **Purpose**

Estimate internals CAPEX for structured packings: packing material cost proportional to the packing volume + distributor cost scaled by cross-sectional area.[6]

#### **Algorithm (stepwise)**

1. Compute  $pack\_vol = Area * Height$ .
  2. Compute  $cost\_pack = pack\_vol * cost\_per\_m3$ .
  3. Compute  $cost\_dist = n\_distributors * distributor\_cost\_each * Area$ .
  4. Return  $cost\_pack + cost\_dist$ .
- 

### **Shell cost (cylindrical shell mass $\rightarrow$ cost)**

#### **Purpose**

Estimate shell thickness by pressure criteria, compute shell mass, and estimate base material + fabrication cost. The approach follows the thin-cylinder formula in C&R with a conservative minimum thickness for vacuum/column shells.[6]

#### **Governing formulae (thin cylinder, internal/external pressure)**

Design pressure (simple approximate delta used in the code):



$$P_{design} \approx |1.01325 - P_{oper}| \quad (bar \rightarrow convert\ to\ Pa\ for\ units) \quad [6]$$

Thin cylinder formula (internal pressure case; rearranged for thickness):

$$t_{calc} = \frac{P_{design} D}{2 f e} + c \quad [6]$$

where  $f$  is allowable stress (Pa),  $e$  is weld efficiency ( $0 < e \leq 1$ ),  $c$  is corrosion allowance (m).

Enforce practical minimum thickness (C&R guidance):

$$t = \max(t_{calc}, t_{min})$$

with  $t_{min} = 5$  mm for small shells and larger if diameter  $> 1$  m (e.g., 7 mm), per typical C&R/ASME practical minimums.

Shell volume and mass:

$$V_{shell} = \pi D H t, \quad m_{shell} = V_{shell} \rho_{ss}$$

Base material cost:

$$Cost_{material} = m_{shell} \times cost\_per\_kg$$

### Algorithm (stepwise)

1. Compute design pressure in Pa from **Pop** difference metric (or use site design pressure).
  2. Compute **t\_calc** from thin-cylinder relation and add corrosion allowance.
  3. Apply **t\_min** rule: set **t** = **max(t\_calc, t\_min)**.
  4. Compute **V\_shell** =  $\pi D H t$  and **m\_shell** = **V\_shell** \* **rho\_ss**.
  5. Compute **base\_cost** = **m\_shell** \* **cost\_per\_kg**; **fabrication\_cost** = **fabrication\_multiplier** \* **base\_cost**.
  6. Return thickness, mass and costs.
-

## Miscellaneous / contingency cost

### Purpose

Add a pragmatic allowance to cover transport, installation, instrumentation, piping and miscellaneous site works not explicitly estimated. This is a project-level contingency used in early screening.

### Formula

$$Misc = \alpha \times (packing\ cost + shell\ cost)$$

where  $\alpha$  is a chosen fraction (in the code 0.5 or 50%).

---

## 5.8 Condenser

The correlations and design equations applied for determining heat-transfer area, film condensation coefficients, bundle corrections, and Boyko–Kruzhilin shear-controlled condensation were taken from **Coulson and Richardson's Chemical Engineering Design (Vol. 6)**. [6]

- Log-mean temperature difference (isothermal vapour)
- Heat-transfer area from duty, U and LMTD
- Condensate mass flow (from duty), and heat flux
- Nusselt film condensation — single tube (horizontal/vertical forms)
- Horizontal / outside (single tube, Nusselt form)
- Vertical tube (single tube): vertical loading
- Film Reynolds number for the condensate film
- Bundle (tube-bank) correction
- Conservative bundle mean coefficient
- Boyko–Kruzhilin shear-controlled condensation:

Where  $h_i$  is the single-phase condensate (tube-side) coefficient for the liquid flow alone and FFF depends on inlet/outlet quality terms. A practical explicit correlation for  $h_i$  is:

where  $u_v$  and  $u_L$  are the phase velocities for vapour and liquid (phase alone in the tube);  $d_i$  in meters

- Shell : bundle guidance (selection ranges):

$$D_{shell} / D_{bundle} \approx$$

$$1.2 - 1.5 \text{ for low } q''$$

1.4 – 1.8 *for medium  $q''$*

1.7 – 2.0 *for high  $q''$*

The condenser cost/m<sup>2</sup> was defined to be as INR 5,00,000. The method used for calculating the area of condenser has been shown above and the Total Condenser Cost was calculated to be INR 1,09,11,182.

---

## 5.9 Kettle Reboiler

The design equations and empirical correlations used for nucleate boiling heat transfer, overall heat-transfer resistance, critical heat flux, vapor velocity constraints, and shell–tube bundle selection were all taken from standard references in **Coulson and Richardson's Chemical Engineering Design** (Vol. 6).[6]

- Area from U and  $\Delta T$
- Mostinski nucleate-boiling (empirical form used)
- Overall U from resistances (series)
- Modified Zuber (Palen & Small) — bundle critical heat flux
- Vapour velocity constraint at liquid surface.
- Shell : bundle guidance (selection ranges).

$$\frac{D_{\text{shell}}}{D_{\text{bundle}}} = 1.2 - 1.5 \text{ (low heat flux),}$$

1.4 – 1.8 (medium heat flux)

1.7 – 2.0 (high heat flux)

The Reboiler cost/m<sup>2</sup> was defined to be as INR 5,00,000. The method used for calculating the area of the reboiler has been shown above and the Total Reboiler Cost was calculated to be INR 84,54,053.

---

## 5.10 Objective Function - TAC to be optimized

### 1. Total annualized cost (TAC) — definition and objective function

Let  $P_{op}$  denote the chosen operating pressure (single scalar decision variable). We define TAC as the sum of annual operating cost and annualized capital (annuity of CAPEX):

$$TAC(P) = AnnualOperating(P) + AnnualizedCapital(P).[6]$$

$AnnualOperating$  commonly includes fuel/steam and cooling-water costs:

$$AnnualOperating(P) = 24 \times 365 \times Operating\ Cost / hr$$

$AnnualizedCapital$  may be computed from CAPEX via a capital recovery factor (CRF) or, in a simplified engineering model, as a fixed fraction  $\alpha$  of CAPEX:

- precise annuity form:

$$AnnualizedCapital(P) = CAPEX(P) \times CRF(i, n), \quad CRF(i, n) = \frac{i(1+i)^n}{(1+i)^n - 1}$$

with  $i$  = real discount rate,  $n$  = lifetime (years).

- simplified form used in code:

$$AnnualizedCapital(P) \approx \alpha CAPEX(P),$$

(e.g.  $\alpha = 0.2$  used as rule-of-thumb in the submitted code).

Objective (scalar) to minimize:

$$\min_{P \in [P_{min}, P_{max}]} TAC(P).$$

### 2. The objective as a composition of nested maps (mathematical structure)

The TAC function is computed via a chain of submodels:

$$P \rightarrow Intersections \rightarrow Stages \rightarrow Energy \rightarrow \{A, H\} \rightarrow CAPEX$$

Thus,

$$TAC(P) = T(C(S(E(N(I(P))))))$$

Notation:  $I$  = intersection solver (VLE geometry),  $N$  = stage counting (McCabe-Thiele / stage models),  $E$  = column energy (vapour/reboiler loads),  $S$  = area/height geometry functions,  $C$  = cost aggregation,  $T$  = annualization and aggregation.

**Differentiability.** Each mapping may be:

- continuous and differentiable (thermo properties, energy balances) or
- non-smooth/discontinuous (stage truncation, discrete tube counts, feasibility cutoffs, piecewise shell sizing, if/else failures).

If the whole composition were smooth, chain rule would allow analytic derivatives:

$$\frac{dTAC}{dP} = \frac{dT}{dC} \cdot \frac{dC}{dS} \cdot \frac{dS}{dE} \cdot \frac{dE}{dN} \cdot \frac{dN}{dI} \cdot \frac{dI}{dP}$$

In practice, many submodels include discrete operations (ceil, int, conditional bounds) or black-box solvers; thus derivatives are not available or noisy. For that reason, derivative-free scalar optimization is appropriate.

### 3. Integration: stage counts → height → energy → cost (control flow)

In a full design loop (used in the objective function), the execution order is:

1. **Material closure:** solve materialBalance for D, B,  $x_D$ ,  $x_B$ .
2. **Thermodynamics:** for a candidate top pressure **Pop**, compute intersection (q-line / operating lines) and determine reflux ratio **R** (design multiple of  $R_{min}$ ).
3. **Stage Calculations:** call the stage counting routine to obtain theoretical stage counts and temperature profile **Tstage**.
4. **Column energy:** build **ColumnEnergyParams** using **Tstage** and process flows; compute **compute\_energy** to obtain vapour flow and duties.
5. **Hydraulics & sizing:** size cross-section **area(Vflow)**; compute packing height via **height(Nstages, f\_F, f\_hetp, P)**.
6. **Costs:** compute packing, shell, condenser and reboiler costs from sizes/duties; combine CAPEX and OPEX to produce the economic objective.
7. **Repeat:** for optimization the objective is evaluated multiple times at different **Pop**; therefore the above pipeline must be numerically robust and performant.

### Performance control & concurrency

- Precompute VLE tables and interpolators if the optimizer queries many pressures.
  - Use vectorized or compiled VLE for speed.
  - Cache intermediate results keyed by `Pop` to avoid recomputation in local optimizer steps.
  - Use coarse-to-fine search: a coarse pressure sweep to find candidate minima followed by local refined search.
- 

## 5.11 Optimization and sensitivity analysis

### 1. Why single-variable scalar optimization (`minimize_scalar`) is appropriate

- Decision space is one dimensional (operating pressure  $P$ ).
- TAC is computed as an expensive black-box scalar evaluation; gradient availability is doubtful.
- Univariate derivative-free algorithms (golden section, Brent, bounded Brent) are mature, efficient, and require only function evaluations and bracketing.
- Minimizing a scalar avoids the complexity of multi-dimensional search and lets one use targeted 1-D global / local strategies and sensitivity analysis around the minimizer.

Thus the chosen design is to evaluate:

$$\text{objective}(P) = TAC(P),$$

and call a robust 1-D scalar minimizer.

---

## 2. Local 1-D algorithms (mathematics / idea)

### Golden-section search

- Works with a bracketing interval  $[a, b]$ .
- At each iteration it evaluates the function at two interior points determined by the golden ratio  $\varphi = (1 + \sqrt{5})/2$  and shrinks the interval while keeping a unimodal minimum inside.
- Convergence is linear; it requires only function values (no interpolation).

### Brent's method (bracketing + inverse parabolic interpolation)

- Hybrid of golden section and inverse parabolic interpolation.
- Uses parabolic fits when possible (superlinear convergence near smooth minima) and falls back to the golden section if interpolation is unreliable.
- Doesn't require derivatives; needs a bracketing interval or can attempt to bracket.

### Bounded Brent (bounded)

- Brent's ideas adapted to a bounded domain; enforces  $P \in [P_{min}, P_{max}]$ .
- Good for well-behaved unimodal problems and is commonly used for expensive scalar objectives.

### Why these for TAC?

- If TAC is unimodal (single well) on the domain, Brent gives fast convergence.
  - If TAC is noisy, non-smooth or multi-modal, these methods may converge to local minima — must be combined with global search or restarts.
-

### 3. Alternatives & when to use them

#### Gradient-based methods

- Require derivatives (analytic or accurate finite differences). If the model is smooth and cheap to evaluate, one can use Newton/trust-region/quasi-Newton methods.
- Not suitable if the function contains discrete steps as in this case, solver failures or noise.

#### Global deterministic methods (univariate method)

- Brute-force grid search followed by local refinement (practical for 1-D). Guarantees locating global minimum if grid is fine enough.
- Ternary/golden over multiple initial brackets.

#### Stochastic global optimizers

- Differential Evolution (DE), Particle Swarm Optimization (PSO), Simulated Annealing (SA) — useful if TAC is multi-modal. More function evaluations required.
- Recommended if you suspect multiple minima and the objective is cheap enough or you have a parallel compute.

#### Why `minimize_scalar` / bounded Brent is in practice

- Balanced: few function evaluations, robust for unimodal curves, easy to apply with bounds.
- The code's plotting/experience (flat right side, steep left) suggests a single interior minimum (near  $P_{opt} \approx 0.71$  bar). Brent/bounded is efficient to accurately locate that minimum.

### 4. Mathematical behavior of nested evaluations and why solvers still work

- Minimizers only require function values  $f(P) = TAC(P)$ . They do not need internal structure. Repeated scalar sampling yields bracket reduction or interpolation.



- Even with nested black-boxes, as long as calls return finite scalars (and failures are penalized), bracketing/interpolation algorithms can proceed.
  - If function evaluations are **noisy** or **discontinuous**, the parabolic interpolation step in Brent can fail — the method then falls back to bracket-shrinking (golden section) which still reduces the interval, hence robustness.
- 

## 5. Sensitivity analysis — mathematical measures & procedures

### Local (derivative based) sensitivity

Estimate derivative numerically (central finite difference):

$$\left. \frac{dTAC}{dP} \right|_{P^*} \approx \frac{TAC(P^* + \Delta) - TAC(P^* - \Delta)}{2\Delta}$$

Normalized (relative) sensitivity (elasticity):

$$S_P = \frac{P^*}{TAC(P^*)} \left. \frac{dTAC}{dP} \right|_{P^*}$$

Curvature (approx. second derivative):

$$\left. \frac{d^2TAC}{dP^2} \right|_{P^*} \approx \frac{TAC(P^* + \Delta) - 2TAC(P^*) + TAC(P^* - \Delta)}{\Delta^2}$$

- If  $|S_P|$  small near  $P^*$ , TAC is *insensitive* to pressure changes (flat region).
  - Large negative slope left of  $P^*$  and small slope right explains the observation: steep cost increase for  $P < P^*$ , flat increase for  $P > P^*$ .
- 

## 6. Recommendation for further study

1. Evaluate TAC on a coarse uniform grid of  $P \in [Pmin, Pmax]$  (e.g., 21 points). Record best few minima.
2. For the best bracket  $(a, b, c)$  or nearest minima, call a bounded local method (Brent/bounded) to refine the minimum.

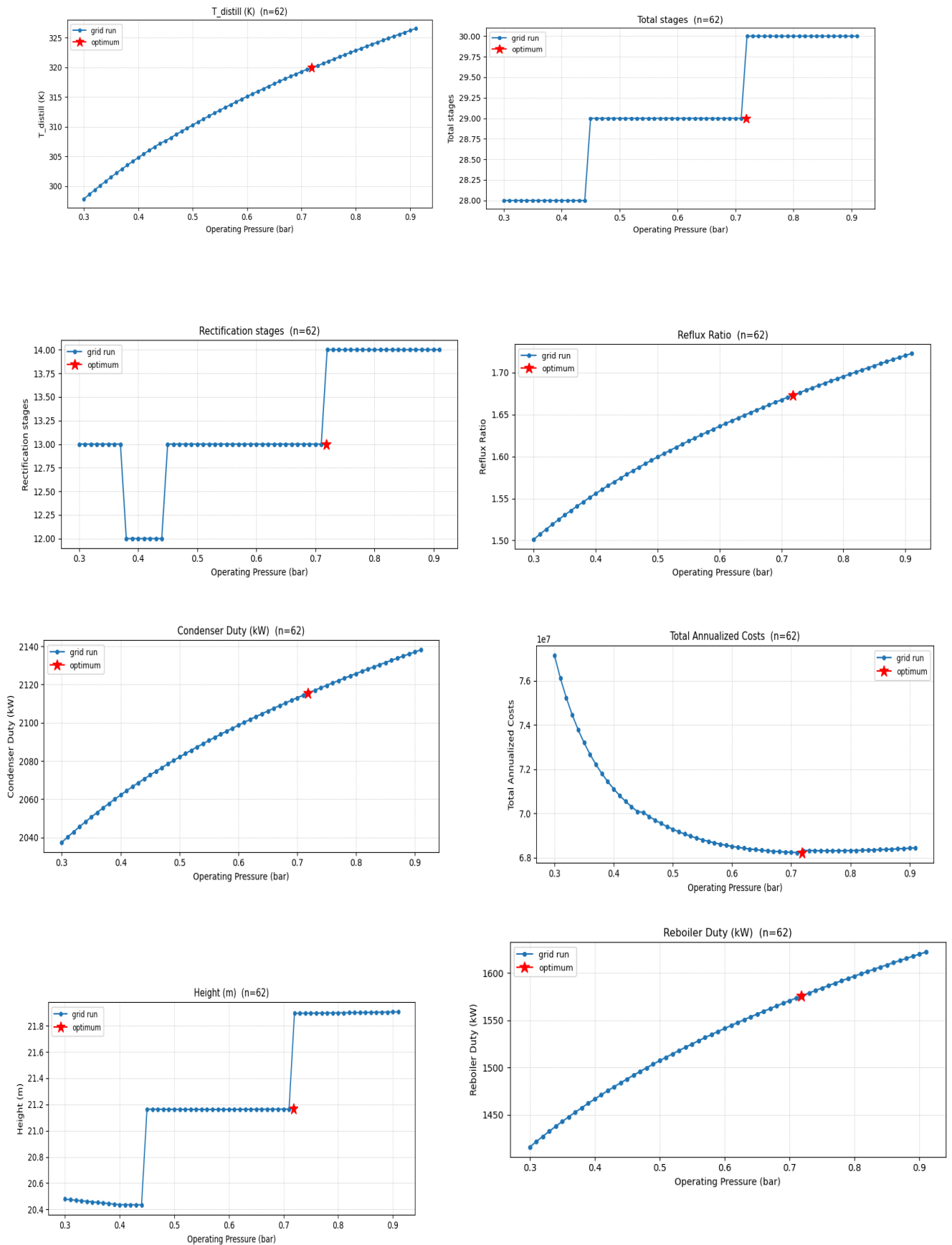
3. Compute central finite-difference derivative and elasticity at optimum. Compute TAC at  $P^* \pm \delta$  for several  $\delta$  (1%, 5%, 10%) to quantify sensitivity.
  4. Report  $P^*$ ,  $\text{TAC}(P^*)$ , elasticity  $S_p$ , and robustness intervals  $[P^* - \Delta, P^* + \Delta]$  where TAC increases by less than a chosen threshold (e.g., 5%).
- 

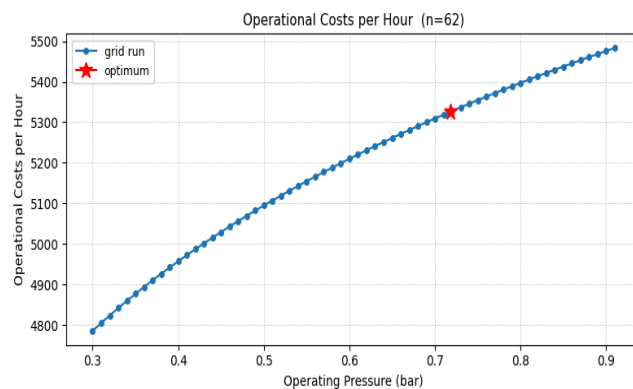
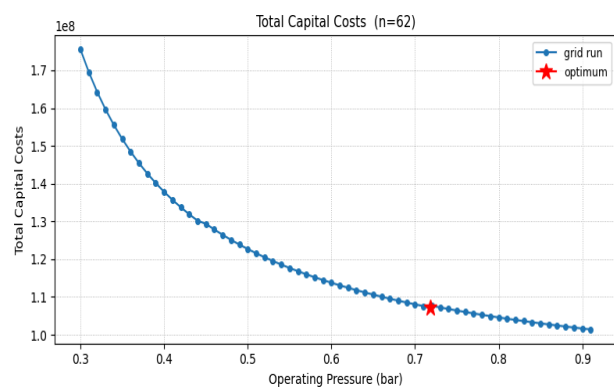
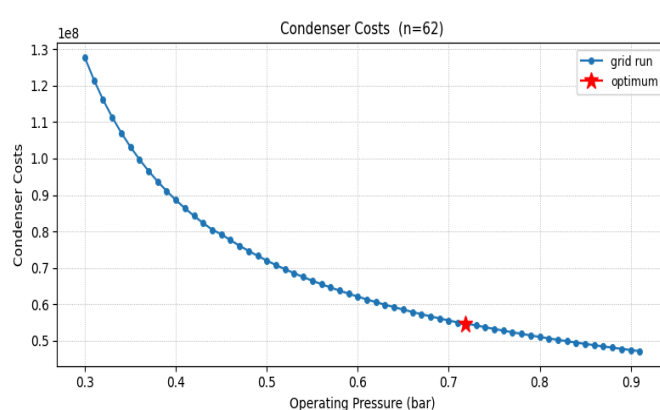
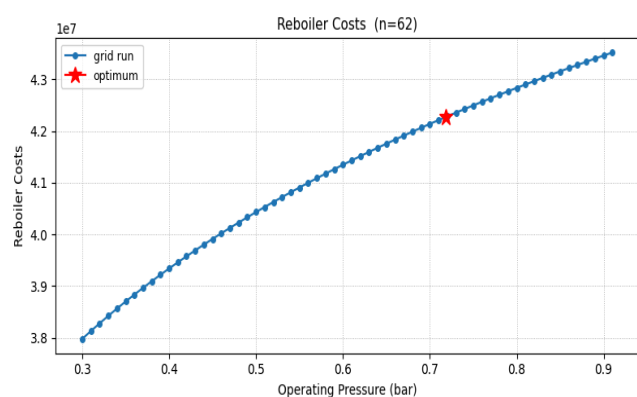
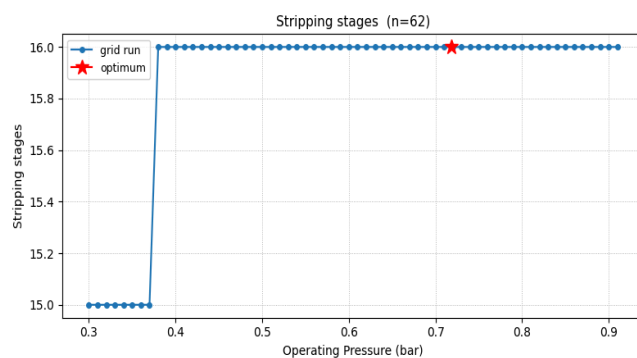
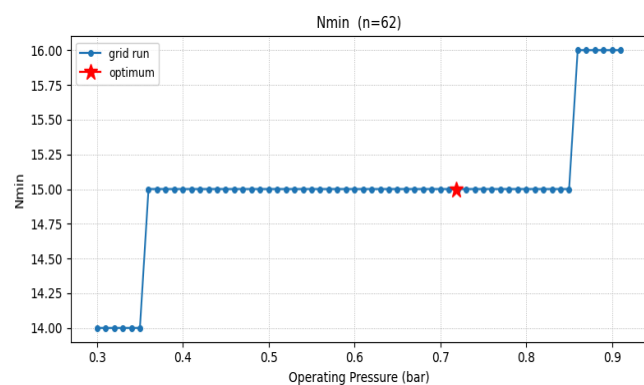
## 7. Why not other approaches

- **Full multivariate gradient methods:** unnecessary complexity because only one design variable is optimized here. Also derivatives are absent or unreliable.
  - **Heuristics (PSO / GA):** overkill for 1-D; large number of function calls and slow convergence relative to Brent or Bayesian strategies. Use these only if TAC is multi-modal and cheap to evaluate.
  - **Adjoint / analytic derivatives:** powerful if you can make the entire chain differentiable (unlikely here due to discrete steps and black-box solvers).
-

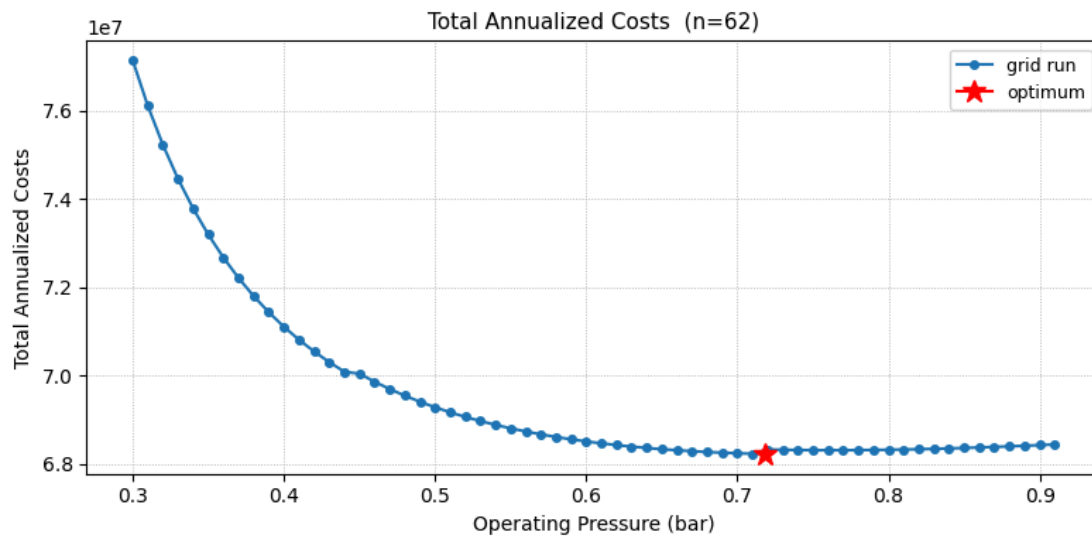
# 6. Results

## 14.2 Column Simulation Results



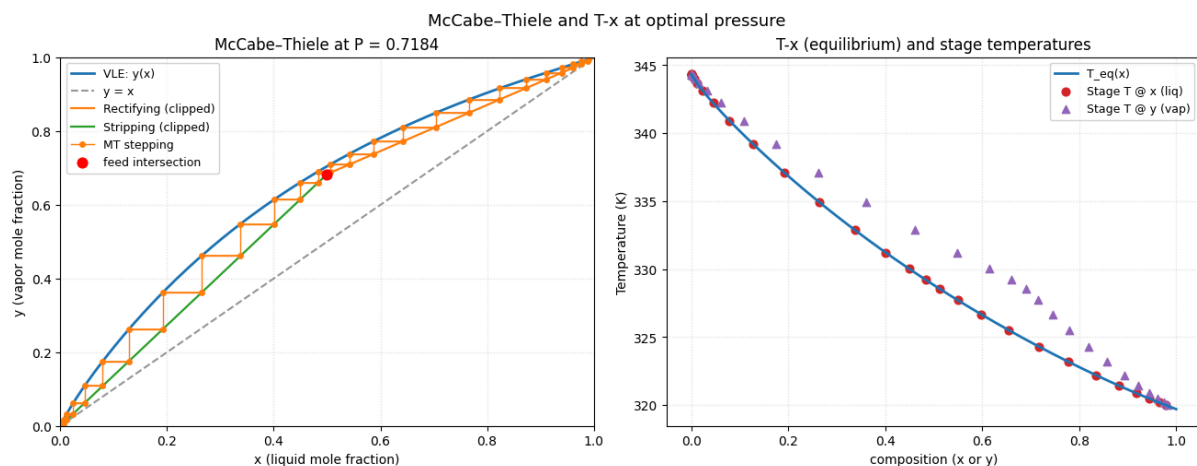


## 7. Cost vs Pressure Results



From the cost–pressure analysis, the **optimum operating pressure** for the designed distillation column was determined to be **0.718 bar**, corresponding to a **minimum total annualized cost (TAC) of INR 6.82 Crores**. At this pressure, the operating and capital cost contributions reach their most economical balance.

## 8. Optimal Pressure



At the optimum pressure of 0.718 bar, the McCabe–Thiele diagram shows the equilibrium curve along with the operating lines, confirming the number of theoretical stages required for separation. The stepping procedure indicates the feed stage location and the feasibility of the design at this pressure. The corresponding T–x–y diagram

illustrates the equilibrium temperature profile across the composition range, together with the calculated stage temperatures for both liquid and vapor phases, demonstrating close agreement with the equilibrium curve

## 9. References

- (1) White, D. C. Optimize Energy Use in Distillation. *Chem. Eng. Prog.* 2012, *March*, 35–36. (Original work published by Emerson Process Management as “Optimize Energy Use in Distillation,” reprinted from *CEP*, March 2012.)
- (2) Brown, I.; Smith, F. Liquid–Vapour Equilibria. X. The Systems Acetone + Nitromethane and Acetone + Acetonitrile at 45 °C. *Aust. J. Chem.* 1960, *13* (1), 29–40.
- (3) Narasigadu, C. *Phase Equilibrium Investigation of the Water and Acetonitrile Solvent with Heavy Hydrocarbons*; University of KwaZulu-Natal, Durban, South Africa, 2006.
- (4) Sandler, S. I. *Chemical, Biochemical, and Engineering Thermodynamics*, Fifth edition.; Wiley: Hoboken, NJ, 2017.
- (5) Sinnott, R. *Chemical Engineering Design: Chemical Engineering Volume 6*, 4th ed.; Chemical Engineering Ser; Elsevier Science & Technology: Jordan Hill, 2005.
- (6) Coulson, J. M.; Richardson, J. F. *Chemical Engineering. 2: Particle Technology and Separation Processes*; Butterworth Heinemann: Amsterdam, 2010.
- (7) Seader, J. D.; Henley, E. J.; Roper, D. K. *Separation Process Principles: Chemical and Biochemical Operations*, Third edition.; John Wiley & Sons: Chichester, 2011.
- (8) Python 3.12.11 (main, Jun 4 2025, 08:56:18) [GCC 11.4.0] on linux Colab google
  - Hardware Architecture:
    - Architecture: x86\_64
    - CPU op-mode(s): 32-bit, 64-bit
    - Address sizes: 46 bits physical, 48 bits virtual
    - Byte Order: Little Endian
    - CPU(s): 2
    - Vendor ID: GenuineIntel
    - Model name: Intel(R) Xeon(R) CPU @ 2.20GHz
    - CPU family: 6
    - Model: 79
    - Thread(s) per core: 2(3) Prausnitz, J. M. *Molecular Thermodynamics of Fluid-Phase Equilibria*, 3rd ed.; Prentice-Hall international series in the physical and chemical engineering sciences; Prentice Hall PTR: Upper Saddle River, N.J, 1999.
- [9] *NIST Chemistry WebBook*; Mallard, W. G.; Linstrom, P. J., Eds.; *NIST Standard Reference Database 69*; National Institute of Standards and Technology: Gaithersburg, MD, (accessed 2025-08-21)

(6) Python 3.12.11 (main, Jun 4 2025, 08:56:18) [GCC 11.4.0] on linux Colab google. (Note: Specific ACS style for this type of system information is not readily available, so it is presented here as a descriptive note.)

(7) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, İ.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P.; SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 2020, 17 (3), 261–272. DOI: 10.1038/s41592-019-0686-2. (This citation covers both Scipy.optimize and Scipy.interpolate as they are part of the SciPy library).

(8) Green, D. W., Perry, R. H., Eds. *Perry's Chemical Engineers' Handbook*, 8th ed.; McGraw-Hill Professional: New York, 2007.

#### Acknowledgements

These are the paper which we referred for help while making this report and doing this project

(1) Prausnitz, J. M. *Molecular Thermodynamics of Fluid-Phase Equilibria*, 3rd ed.; Prentice-Hall international series in the physical and chemical engineering sciences; Prentice Hall PTR: Upper Saddle River, N.J, 1999.

(2) White, D. C. Optimize Energy Use in Distillation.

(2) Fredenslund, A.; Gmehling, J.; Michelsen, M. L.; Rasmussen, P.; Prausnitz, J. M. Computerized Design of Multicomponent Distillation Columns Using the UNIFAC Group Contribution Method for Calculation of Activity Coefficients. *Ind. Eng. Chem. Proc. Des. Dev.* 1977, 16 (4), 450–462.

## 10. Appendices:

Google Drive Link: [Kindly Click here](#)

Google Colab File: [Kindly Click here](#)

## 11. Assumption:

The plant is in Baddi himachal pradesh where the average annual temperature is assumed to be 15 °C.

## 12. Notations

P - Pressure

T - Temperature

F - Feed Flow Rate

D - Distillate Flow Rate

B - Bottom Flow Rate

$x_D$  - Mole Fraction of More volatile component in Distillate

$x_B$  - Mole Fraction of More volatile component in Bottom Product

$G^{Ex}/RT$  - Gibbs Excess Energy

$R_{min}$  - Minimum Reflux Ratio

TAC - Total Annualized Cost

VLE - Vapour Liquid Equilibrium

$\Gamma_1$  - Activity Coefficient of Component 1

$\Gamma_2$  - Activity Coefficient of Component 2

A - Area

HETP - Height Equivalent to Theoretical Plate

NTU - Number of Transfer Units

$\rho_L$  - Liquid Density

$\rho_G$  - Vapour Density

Re - Reynolds Number

Pr - Prandtl Number

R - Actual Reflux Ratio

$\lambda$  - Latent Heat

V- Boilup vapour leaving reboiler

L - Reflux Liquid entering column

$c_p$  - Heat Capacity



$m_{vapourized}$  - Amount vaporized

$\lambda_{mixture}$  - latent Heat of mixture

$Q_{sensible}$  - Sensible Heat Energy

$Q_R$  - Reboiler Duty

$Q_C$  - Condenser Duty

$m_{steam}$  - Mass of Steam

$T_{reboil}$  - Reboil Temperature

$T_{feed}$  - Feed Temperature

$u_{design}$  - Design superficial Velocity

$v_{max}$  - Maximum velocity

V - Volumetric Flow Rate

$x_1$  - mole fraction of more volatile component in liquid phase

$y_1$  - mole fraction of more volatile component in vapor phase

$P^{sat}$  - Saturation Pressure

N - No. of Theoretical stages

$N_{min}$  - Minimum Number of Theoretical Stages