python run_optimization.py to run the program in Terminal

pip install CoolProp in Terminal before running the code

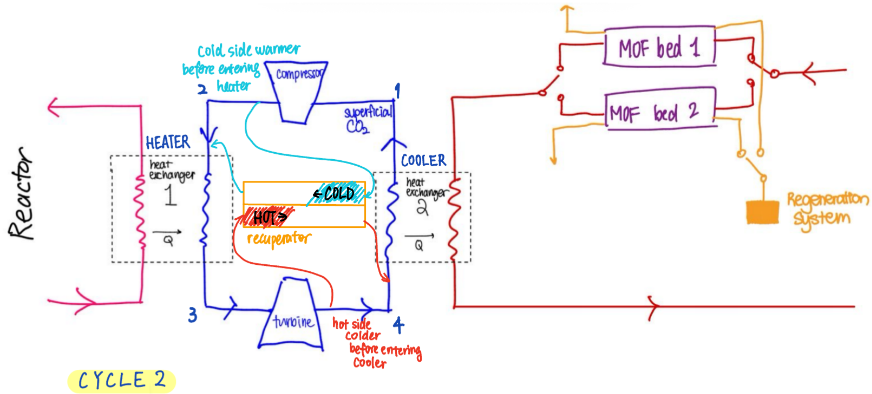
Files / high-level structure

- `properties.py` — thin wrapper around a thermophysical property library (CoolProp). Provides `h(p,T)`, `s(p,T)`, `T(p,s)`, `c_p(p,T)`, `rho(p,T)`.
- `components.py` — component models:
  - `compressor_outlet_state(...)` and `turbine_outlet_state(...)` (isentropic-efficiency based).
  - `recuperator_counterflow(...)` (discretized counterflow HE).
- `cycle_solver.py` — assembles the cycle states and calls components to compute state properties and component works.
- `run` / `optimizer` script — searches (P_high, T_turbine) and solves for mass flow such that the cycle consumes the fixed heat input `Q_available`.

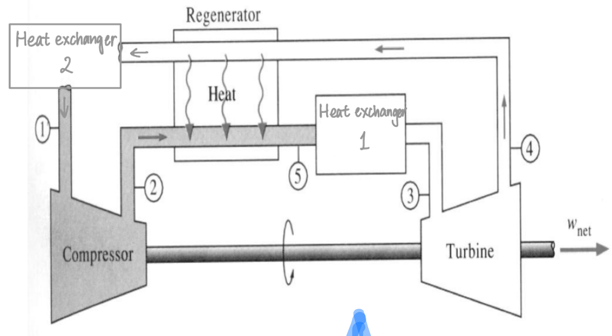
Key assumptions (one sentence each; each assumption used explicitly in the code)

1. **Single-fluid, closed loop:** Working fluid is CO_2 and the cycle is closed (no mass loss) — this isolates thermodynamic performance from leakage and combustion effects.
2. **Low-side pressure fixed (P_low):** P_low is set above the CO_2 critical pressure to avoid two-phase behavior and numerical instability. $P_1 = 8 MPa$
3. **Isentropic-efficiency component model:** Compressors and turbines are modeled by constant isentropic efficiencies (η_c, η_t) — a standard first-order engineering model that isolates aerodynamic losses from thermodynamics. $\eta_c = \eta_t = 0.9$
4. **Single recuperator, nodewise discretized:** Recuperation is represented by a single discretized counterflow heat exchanger with a fixed overall UA (NTU proxy) — this gives physically reasonable preheating while keeping the model simple and robust.
5. **Fixed external heat source:** The helium IHX supplies a known, fixed `Q_available`; the CO_2 mass flow is chosen so the cycle absorbs exactly that heat. $q_{in} = 10MW$, find $\dot{m}$ of CO_2
6. **Simple pressure-drop handling:** Pressure drops are not modeled in detail (or are represented as fixed fractions) — adequate for early-stage thermodynamic sizing

but not detailed mechanical desig



CYCLE 2



Notation (used in formulas)

- p — pressure (Pa)
- T — temperature (K)
- h — specific enthalpy (J/kg)
- s — specific entropy (J/kg·K)
- c_p — isobaric specific heat (J/kg·K
- $\dot{m}$ — mass flow rate (kg/s)
- $\dot{Q}$ — power / heat rate (W)
- W — power (W)
- UA — overall conductance of heat exchanger (W/K)
- $C = \dot{m}\, c_p$ — heat capacity rate (W/K)
- Subscripts: 1 (compressor inlet), 2 (compressor outlet / cold side of recup inlet), 3 (turbine inlet after heater), 4 (turbine outlet / hot side of recup inlet)

Component models — formulas & logic

1. Property calls (library wrapper)

- $h = h(p, T)$
- $s = s(p, T)$
- $T = T(p, s)$
- $c_p = c_p(p, T)$

  These are provided by the `properties` wrapper and are used everywhere.

2. Compressor (isentropic-efficiency formulation)

$T_1 = 35\,°C$ warm ambient temperature

(27~47 °C usually)

Given inlet (p_1, T_1), outlet pressure p_2, isentropic efficiency η_c:

above CO_2 critical temperature $\approx 31°C$

1. Compute inlet enthalpy and entropy:

$$h_1 = h(p_1, T_1), \qquad s_1 = s(p_1, T_1)$$

2. Ideal (isentropic) outlet temperature from entropy and outlet pressure:

$$T_{2,\text{ideal}} = T(p_2, s_1)$$

3. Ideal outlet enthalpy:

$$h_{2,\text{ideal}} = h(p_2, T_{2,\text{ideal}})$$

4. Actual outlet enthalpy using isentropic efficiency:

$$h_2 = h_1 + \frac{h_{2,\text{ideal}} - h_1}{\eta_c}$$

5. Estimate outlet temperature T_2 by inverting $h(p_2, T)$ locally using c_p (or a property inversion):

$$T_2 \approx T_{2,\text{ideal}} + \frac{h_2 - h_{2,\text{ideal}}}{c_p(p_2, T_{2,\text{ideal}})}$$

6. Compressor specific work (input):

$$w_c = h_2 - h_1$$

---

3. Turbine (isentropic-efficiency formulation)

Given inlet (p_3, T_3), outlet pressure p_4, isentropic efficiency η_t:

$T_3 = 800°C$ based on turbine material

1. $h_3 = h(p_3, T_3), \;\; s_3 = s(p_3, T_3)$
2. Ideal (isentropic) outlet temperature:

$$T_{4,\text{ideal}} = T(p_4, s_3)$$

3. Ideal outlet enthalpy:

$$h_{4,\text{ideal}} = h(p_4, T_{4,\text{ideal}})$$

4. Actual outlet enthalpy using efficiency:

$$h_4 = h_3 - \eta_t(h_3 - h_{4,\text{ideal}})$$

5. Estimate outlet temperature T_4:

$$T_4 \approx T_{4,\text{ideal}} + \frac{h_4 - h_{4,\text{ideal}}}{c_p(p_4, T_{4,\text{ideal}})}$$

6. Turbine specific work (output):

$$w_t = h_3 - h_4$$

---

4. Recuperator (discretized counterflow formulation)

The recuperator exchanges heat between the hot stream (turbine exhaust) and the cold stream (compressor discharge). The code marches along `N` discrete nodes.

Per node (small element), with local hot temperature $T_h(x)$ and cold temperature $T_c(x)$:

1. Local heat-transfer rate (node-wise) using global UA distributed evenly:

$$dQ = \frac{UA}{N}\,(T_h - T_c)$$

$UA = NTU \cdot C_{min}$

2. Local heat capacity rates:

$$C_h = \dot{m}_h\, c_{p,h}(p_h, T_h), \qquad C_c = \dot{m}_c\, c_{p,c}(p_c, T_c)$$

3. Update temperatures:

$$T_h \leftarrow T_h - \frac{dQ}{C_h}, \qquad T_c \leftarrow T_c + \frac{dQ}{C_c}$$

4. Repeat for all nodes. Track minimum local temperature difference (pinch):

$$\Delta T_{\min} = \min_x \left(T_h(x) - T_c(x)\right)$$

5. After marching, hot-side outlet temperature is $T_{h,\text{out}}$ and cold-side outlet is $T_{c,\text{out}}$. Convert to enthalpies:

$$h_{h,\text{out}} = h(p_{\text{hot,out}}, T_{h,\text{out}}), \qquad h_{c,\text{out}} = h(p_{\text{cold,out}}, T_{c,\text{out}})$$

Notes on recuperator:

- NTU = 5.0 (chosen), a bit unrealistic, usually around 3.0
- number of nodes = 50
- minimum pinch temperature difference allowed $\Delta T_{pinch} = 3.0$
- no recompression & no mass split ($\dot{m}_h = \dot{m}_c$) limits efficiency
- add more recuperators likely increase efficiency

Cycle assembly — step-by-step thermodynamic calculation

At a high level the solver computes states 1→4 in order (compressor → recup → heater → turbine → recup → cooler) and returns component works and heat needs.

1. **State 1 (compressor inlet):**
   - Known: $p_1 = P_{\text{low}},\ T_1 = T_{\text{cold,in}}$.
   - Compute $h_1 = h(p_1, T_1)$.
2. **Compressor (1 → 2):**
   - Given outlet pressure $p_2 = P_{\text{high}}$ and η_c, compute T_2, h_2, w_c as in compressor formulas.

3. **Recuperator (cold side inlet = state 2):**
   - Cold-side inlet: $p_{\text{cold,in}} = p_2, \; T_{\text{cold,in}} = T_2$.
   - Hot-side inlet: $p_{\text{hot,in}} = p_4, \; T_{\text{hot,in}} = T_4$ (turbine exhaust — initially unknown; the solver uses the turbine calculation first or uses a loop depending on implementation).
   - Run `recuperator_counterflow` to obtain:
     - Cold-side outlet temperature $T_{2,\text{post-recup}}$ (call this T_2') and enthalpy h_2'.
     - Hot-side outlet temperature $T_{4,\text{post-recup}}$ and enthalpy h_4'.
   - In the current simple solver the recup is applied once per cycle evaluation (no iterative inner loop).
4. **Heater (external IHX):**
   - The heater raises the cold-side post-recup state to turbine inlet T_3 at pressure $p_3 = P_{\text{high}}$.
   - Specific heat required per kg:

$$q_{\text{in,spec}} = h_3 - h_2' \quad \text{where } h_3 = h(p_3, T_3)$$

- For a given available heater power $\dot{Q}_{\text{available}}$, $\dot{m}$ must satisfy:

$$\dot{m} \cdot q_{\text{in,spec}} = \dot{Q}_{\text{available}}$$ = 10MW from Helium cycle

- The code solves this equation for $\dot{m}$ (bisection or root find).

5. **Turbine (3 → 4):**
   - Given $p_4 = P_{\text{low}}$, compute $T_4, \; h_4, \; w_t$ using turbine formulas.
6. **Recuperator (hot side inlet = state 4):**
   - The hot stream entering the recup is the turbine exhaust (state 4); run recup discretization as above to get $h_{4,\text{out}}$ and the cold preheat h_2' that is fed to the heater.
7. **Net work and efficiency:**
   - Net specific work:

$$w_{\text{net, spec}} = w_t - w_c$$

- Net power:

$$W_{\text{net}} = \dot{m} \cdot w_{\text{net, spec}}$$

- Thermal efficiency (reference to IHX heat):

$$\eta = \frac{W_{\text{net}}}{\dot{Q}_{\text{available}}}$$

Other Parameters Chosen & Reasonings

run_optimization.py

run_optimization.py > solve_for_mdot > residual

```python
# ========================
# USER INPUTS
# ========================
Q_available = 10e6          # 10 MW from helium IHX cycle
P_low_MPa = 8.0             # Chosen above CO2 critical pressure to avoid two-phase behavior
P_low = P_low_MPa * 1e6

eta_c = 0.90                # Chosen isentropic efficiency for compressor
eta_t = 0.90                # Chosen isentropic efficiency for turbine
T_cold_in_C = 35.0          # Chosen above CO2 critical temperature (31C) to avoid two-phase behavior (typically 27-47C)

# Design space
P_high_list_MPa = [21, 24, 27, 30, 33, 35]      # try different compressor outlet pressures (P2)
                                                # Chosen based on mechanical design limits (typical range 20-35MPa)

T_turbine_list_C = [650, 700, 750, 800, 850, 900]    # try different turbine inlet temperatures (T3)
                                                # Chosen based on mechanical design limits

mdot_min = 0.1
mdot_max = 300.0        # large range to ensure actual mdot lies within the range during bisection search
```

Results saved in .csv document, Best point printed out

```
Best design point:
P_high_MPa          : 35.0
P_low_MPa           : 8.0
T_turbine_in_C      : 750
mdot_kg_s           : 19.02406208463653
net_power_MW        : 3.533930486503994
eta                 : 0.3533930486503994
T1_C                : 35.0
T2_C                : 110.2813763884734
T3_C                : 750.0
T4_C                : 551.6223903822238
```

$P_2 = P_3 = 35\,MPa$

$P_1 = P_4 = 8\,MPa$

$35.34\% = \eta$

Goals $\left\{ \begin{array}{l} \text{prove that current conditions are practical ?} \\ \text{add recompression + recuperation} \end{array} \right.$

Short answer: for a **closed sCO_2 Brayton cycle with a recuperator**, typical thermal efficiency usually falls roughly in these bands:

- **Simple recuperated Brayton (practical, engineering-grade): ~30–42%** (lower end for moderate turbine inlet temps and imperfect recuperators; upper end for higher temps and good components). ScienceDirect +1
- **Advanced layouts (recompression / recompression + recuperation): ~40–55%** (this is what literature and recent designs report when peak cycle temperatures are high ≈650–800 °C and recuperators are highly effective). MDPI +1
- **Lab / ideal-case numbers can be higher (≫50%)**, but real systems are penalized by recuperator effectiveness, pressure losses, and non-ideal turbomachinery; those losses often reduce practical efficiencies by many percentage points. MDPI +1

Why the spread? The big drivers are:

- **Maximum cycle temperature** (higher Tmax → higher Carnot limit → higher efficiency). SpringerLink
- **Recuperator effectiveness and pressure drop** (a poor or highly-lossy recuperator cuts efficiency substantially). ScienceDirect +1
- **Cycle layout** (simple recuperated vs recompression vs reheated/recompressed variants). MDPI
- **Real component performance** (isentropic efficiencies of compressor/turbine, leakages, sealing). MDPI