

Thermo-Calc Workflow Guide

Extracting CALPHAD Data for Python Plotting

Anthony DiMascio

MSE 4381 | Honda CALPHAD Project | Spring 2026

1 File Organization

Create this folder structure **before** starting:

```
honda-calphad/
  data/
    thermocalc/
      raw/           <- Direct exports from TC
      processed/     <- Cleaned CSVs for Python
    literature/      <- Reference values from papers
    simulations/
    pycalphad/       <- Your existing Marimo notebook
```

1.1 File Naming Convention

```
[system]_[property]_[conditions].txt
```

Examples:

cu2o_dGf_1273-1873K.txt	Gibbs energy of Cu ₂ O formation
al2o3_dGf_1273-1873K.txt	Gibbs energy of Al ₂ O ₃ formation
fe-cu_activity_1873K.txt	Cu activity in Fe at 1873K
cu-al-o_isotherm_1873K.txt	Cu-Al-O phase diagram section

2 Calculation 1: Oxide Gibbs Energies (Ellingham Data)

Goal: Get accurate ΔG_f° vs T for each oxide to replace our approximations.

2.1 Setup

1. Open Thermo-Calc → File → New Project
2. Select database: **SSUB** (SGTE Substance Database)
3. Define system: For Cu₂O, select elements Cu, O

2.2 For Each Oxide

Repeat the calculation (Section 2.3-2.4) for each oxide below. Use the **exact mole fractions** shown to get stoichiometric composition.

Oxide	Elements	X (mole frac)	Phase	Output file
Cu ₂ O	Cu, O	X(Cu)=0.6667, X(O)=0.3333	CU2O_S	cu2o_dGf.txt
CuO	Cu, O	X(Cu)=0.5, X(O)=0.5	CUO_S	cuo_dGf.txt
Al ₂ O ₃	Al, O	X(Al)=0.4, X(O)=0.6	CORUNDUM	al2o3_dGf.txt
MgO	Mg, O	X(Mg)=0.5, X(O)=0.5	PERICLASE	mgo_dGf.txt
SiO ₂	Si, O	X(Si)=0.3333, X(O)=0.6667	QUARTZ*	sio2_dGf.txt
TiO ₂	Ti, O	X(Ti)=0.3333, X(O)=0.6667	RUTILE	tio2_dGf.txt
FeO	Fe, O	X(Fe)=0.5, X(O)=0.5	HALITE**	feo_dGf.txt

Table 1: *

*At 1273–1873K, SiO₂ is in the CRISTOBALITE or TRIDYMITE region (QUARTZ only stable below 846K); TC handles phase selection automatically. **FeO uses HALITE structure (NaCl-type) in SSUB.

2.2.1 Stoichiometry Reference

The mole fractions come from the oxide formula (atoms of each element / total atoms):

$$\begin{aligned}
 \text{Cu}_2\text{O} : \quad & \frac{2}{2+1} = 0.6667 \text{ Cu}, \quad \frac{1}{2+1} = 0.3333 \text{ O} \\
 \text{Al}_2\text{O}_3 : \quad & \frac{2}{2+3} = 0.4 \text{ Al}, \quad \frac{3}{2+3} = 0.6 \text{ O} \\
 \text{CuO, MgO, FeO} : \quad & \frac{1}{1+1} = 0.5 \text{ metal}, \quad \frac{1}{1+1} = 0.5 \text{ O} \\
 \text{SiO}_2, \text{TiO}_2 : \quad & \frac{1}{1+2} = 0.3333 \text{ metal}, \quad \frac{2}{1+2} = 0.6667 \text{ O}
 \end{aligned}$$

2.3 Running the Calculation

In Thermo-Calc Graphical Mode:

1. Calculate → Property Diagram
2. Axis Variable:
 - Variable: T (Temperature)
 - Min: 1273 K
 - Max: 1873 K
 - Step: 25 K (gives 25 data points)
3. Conditions:
 - P = 101325 (1 atm, in Pa)
 - N = 1 (1 mole total)
 - X(element) = values from Table 1
4. Output:
 - Add: GM (Gibbs energy of entire system, J/mol)
5. Click Calculate

Console Mode (alternative, example for Cu₂O):

```

go poly
s-sys Cu 0
s-c t=1273 p=101325 n=1 x(cu)=0.6667 x(o)=0.3333
c-e
lis,,,,

```

2.4 Exporting

1. Results → Table Renderer
2. Select columns: T, GM
3. File → Export Table → Tab-separated
4. Save to: data/thermocalc/raw/cu2o_dGf_1273-1873K.txt

Repeat for each oxide. Total: 7 files.

3 Calculation 2: Cu Activity in Liquid Fe

Goal: How “active” is 0.3% Cu in molten steel?

3.1 Setup

1. Database: **TCFE** (Steels/Fe-alloys database)
2. Define System: select Fe, Cu
3. Reject all phases except LIQUID (we only want liquid steel)

3.2 Calculation A: Activity vs Composition at 1873K

Shows how Cu activity changes with Cu content at fixed temperature.

1. Calculate → Property Diagram
2. **Axis Variable:**
 - Variable: X(Cu)
 - Min: 0.0001 (avoid zero)
 - Max: 0.05 (5 mol%)
 - Step: 0.001
3. **Conditions:**
 - T = 1873 K
 - P = 101325 Pa
 - N = 1
4. **Output:** AC(Cu) (activity of Cu, reference: pure liquid Cu)
5. Export: fe-cu_activity-vs-xcu_1873K.txt

3.3 Calculation B: Activity vs Temperature at 0.3% Cu

Shows how temperature affects Cu activity at typical contamination level.

1. Calculate → Property Diagram

2. Axis Variable:

- Variable: T
- Min: 1773 K (1500°C)
- Max: 1973 K (1700°C)
- Step: 10 K

3. Conditions:

- $X(\text{Cu}) = 0.003$ (0.3 mol% \approx 0.3 wt%)
- $P = 101325$ Pa
- $N = 1$

4. Output: AC(Cu)

5. Export: fe-cu_activity-vs-T_xcu003.txt

4 Calculation 3: Cu-Al-O Phase Diagram

Goal: What phases exist when Cu contacts Al_2O_3 ? Are spinels (CuAl_2O_4) stable?

4.1 Check Database Availability

1. In Thermo-Calc: Database \rightarrow Database Manager
2. Look for: **TCOX** (Metal Oxide Solutions Database)
3. If unavailable: use **SSUB** (has stoichiometric oxides only, no solutions)

4.2 Setup (if TCOX available)

1. Database: **TCOX**
2. Define System: select Cu, Al, O
3. Keep all phases (to see what's stable)

4.3 Isothermal Section at 1873K

1. Calculate \rightarrow Phase Diagram \rightarrow Ternary
2. Conditions:
 - $T = 1873$ K (fixed)
 - $P = 101325$ Pa (fixed)
3. Axes: Cu-Al-O composition triangle (automatic)
4. Click Calculate
5. Look for:
 - CORUNDUM (Al_2O_3) — does it have Cu solubility?
 - SPINEL (CuAl_2O_4) — is this phase stable?
 - LIQUID regions — where does Cu melt?

4.4 Alternative: Pseudo-Binary Cu– Al_2O_3

If ternary is complex, try a 1D cut:

1. Calculate \rightarrow Property Diagram
2. Fix O content at Al_2O_3 stoichiometry: $X(\text{O}) = 0.6$

3. Axis: vary $X(\text{Cu})$ from 0 to 0.3, with $X(\text{Al}) = 0.4 - X(\text{Cu})$
4. This walks from pure Al_2O_3 toward Cu-rich compositions while maintaining 60% oxygen

Note: If TCOX unavailable, this calculation is limited. Document what database was actually used.

5 Calculation 4: Cu Solubility in Al_2O_3

Goal: Maximum Cu that dissolves in alumina vs temperature. This is the key data for predicting ceramic capture capacity.

5.1 Check if Modeled

First, check if CORUNDUM is a *solution phase* or *stoichiometric*:

1. Database \rightarrow List Phase Status
2. Find CORUNDUM
3. If it shows constituent species including Cu \rightarrow solubility is modeled
4. If only Al:O \rightarrow stoichiometric, no Cu solubility (limitation)

5.2 If Cu Solubility is Modeled

1. Database: **TCOX**
2. System: Cu, Al, O
3. Calculate \rightarrow Property Diagram
4. **Axis:** T from 1273 to 1873 K, step 25 K
5. **Conditions:**
 - Set system in two-phase region: CORUNDUM + LIQUID or CORUNDUM + CU₂O
 - P = 101325 Pa
6. **Output:** $X(\text{CORUNDUM}, \text{Cu})$ — mole fraction Cu in corundum phase
7. Export: al2o3-cu-solubility-1273-1873K.txt

5.3 If NOT Modeled (Stoichiometric)

Document this limitation:

- The database treats Al_2O_3 as a line compound with zero Cu solubility
- Real Al_2O_3 may dissolve small amounts of Cu at high T (literature values needed)
- Search literature for experimental Cu solubility data in Al_2O_3
- Add to data/literature/ folder

5.4 Repeat for Other Ceramics

If time permits, check Cu solubility in:

- MgO (PERICLASE phase)
- TiO_2 (RUTILE phase)
- Spinels (SPINEL phase) — may have higher Cu capacity

6 Data Processing

6.1 Raw Export Format

Thermo-Calc exports look like:

```
T      GM
1273   -245678.3
1298   -246012.1
...
```

6.2 Processing Script

Save as `data/process_tc_data.py`:

```
import pandas as pd
from pathlib import Path

RAW = Path("thermocalc/raw")
OUT = Path("thermocalc/processed")
OUT.mkdir(exist_ok=True)

def process_oxide(filename, o2_factor):
    """Convert GM to dGf per mol O2 for Ellingham."""
    df = pd.read_csv(RAW/filename, sep='\t')
    df.columns = ['T_K', 'GM_J']

    # GM is in J/mol, convert to kJ/mol O2
    df['dGf_kJ_per_molO2'] = (df['GM_J'] / 1000) / o2_factor
    df['T_C'] = df['T_K'] - 273.15

    out_name = filename.replace('.txt', '_processed.csv')
    df.to_csv(OUT/out_name, index=False)
    print(f"Saved: {out_name}")

# Process each oxide (filename, O2 factor in reaction)
process_oxide('cu2o_dGf_1273-1873K.txt', 0.5)    # 2Cu + 1/2 O2 -> Cu2O
process_oxide('al2o3_dGf_1273-1873K.txt', 1.5)    # 4/3 Al + O2 -> 2/3 Al2O3
process_oxide('mgo_dGf_1273-1873K.txt', 0.5)      # Mg + 1/2 O2 -> MgO
process_oxide('sio2_dGf_1273-1873K.txt', 1.0)     # Si + O2 -> SiO2
process_oxide('tio2_dGf_1273-1873K.txt', 1.0)     # Ti + O2 -> TiO2
process_oxide('feo_dGf_1273-1873K.txt', 0.5)      # Fe + 1/2 O2 -> FeO
```

7 Plotting in Python

Add to your Marimo notebook or create new script:

```
import pandas as pd
import matplotlib.pyplot as plt
from pathlib import Path

DATA = Path("data/thermocalc/processed")
```

```
# Load all oxides
oxides = {
    'Cu2O': ('cu2o_dGf_1273-1873K_processed.csv', '#0077BB', '--'),
    'Al2O3': ('al2o3_dGf_1273-1873K_processed.csv', '#AA3377', ':'),
    'MgO': ('mgo_dGf_1273-1873K_processed.csv', '#009988', '--'),
    'SiO2': ('sio2_dGf_1273-1873K_processed.csv', '#CC3311', '--'),
    'TiO2': ('tio2_dGf_1273-1873K_processed.csv', '#E69F00', ':'),
    'FeO': ('feo_dGf_1273-1873K_processed.csv', '#EE7733', '--'),
}

fig, ax = plt.subplots(figsize=(10, 7))

for name, (file, color, ls) in oxides.items():
    df = pd.read_csv(DATA/file)
    ax.plot(df['T_C'], df['dGf_kJ_per_molO2'],
            color=color, ls=ls, lw=2.5, label=name)

ax.set_xlabel('Temperature (C)')
ax.set_ylabel('dG (kJ/mol O2)')
ax.set_title('Ellingham Diagram (Thermo-Calc Data)')
ax.legend()
ax.grid(True, alpha=0.3)
plt.tight_layout()
plt.savefig('ellingham_thermocalc.png', dpi=150)
```

8 Checklist

Task	Done
Create folder structure	<input type="checkbox"/>
Export Cu ₂ O ΔG_f vs T	<input type="checkbox"/>
Export Al ₂ O ₃ ΔG_f vs T	<input type="checkbox"/>
Export MgO ΔG_f vs T	<input type="checkbox"/>
Export SiO ₂ ΔG_f vs T	<input type="checkbox"/>
Export TiO ₂ ΔG_f vs T	<input type="checkbox"/>
Export FeO ΔG_f vs T	<input type="checkbox"/>
Export Cu activity in Fe vs X(Cu)	<input type="checkbox"/>
Export Cu activity in Fe vs T	<input type="checkbox"/>
Cu-Al-O isothermal section (if TCOX available)	<input type="checkbox"/>
Run processing script	<input type="checkbox"/>
Generate new Ellingham plot	<input type="checkbox"/>
Update Marimo notebook with TC data	<input type="checkbox"/>

After completing, commit data to GitHub: `git add data/ && git commit -m "Add TC data" && git push`