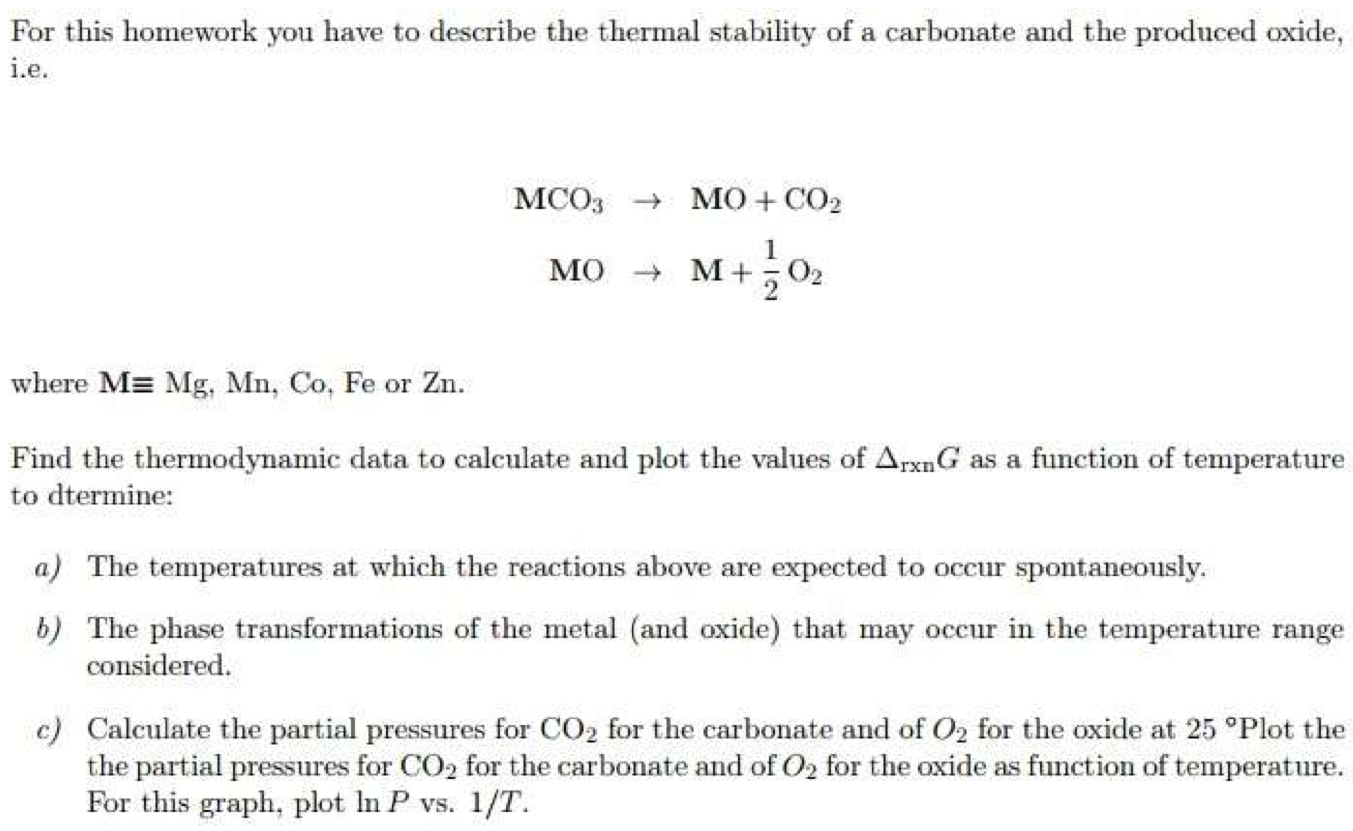
Q4001 Thermodynamics of Materials

Homework 07

Osamu Katagiri - A01212611@itesm.mx - CEM
Diego Sebastián - A01373414@itesm.mx - CEM
Carlos Cardoso - A01750267@itesm.mx - CEM

In [97]:
```
#******************************************************************************#
# Futures
%matplotlib inline
# from __future__ import unicode_literals
# from __future__ import print_function

# Generic/Built-in
import datetime
import argparse

# Other Libs
from IPython.display import display, Image
#
from sympy import *
#
import matplotlib.pyplot as plt
plt.rc('xtick', labelsize=15)
plt.rc('ytick', labelsize=15)
#
import numpy as np
np.seterr(divide='ignore', invalid='ignore')
#
from pint import UnitRegistry
u = UnitRegistry()
u.default_format = 'P'
#
import math

# Owned
pfont = {'fontname':'Consolas'};
# from nostalgia_util import log_utils
# from nostalgia_util import settings_util
__authors__   = ["Osamu Katagiri - A01212611@itesm.mx"]
__copyright__ = "None"
__credits__   = ["Marcelo Videa - mvidea@itesm.mx"]
__license__   = "None"
__status__    = "Under Work"
#******************************************************************************#
display(Image(filename='./directions/0.jpg'))
```



For this homework you have to describe the thermal stability of a carbonate and the produced oxide, i.e.

$$\begin{aligned} \mathrm{MCO_3} &\rightarrow \mathrm{MO + CO_2} \\ \mathrm{MO} &\rightarrow \mathrm{M} + \frac{1}{2}\mathrm{O_2} \end{aligned}$$

where **M**≡ Mg, Mn, Co, Fe or Zn.

Find the thermodynamic data to calculate and plot the values of $\Delta_{\mathrm{rxn}}G$ as a function of temperature to dtermine:

a) The temperatures at which the reactions above are expected to occur spontaneously.

b) The phase transformations of the metal (and oxide) that may occur in the temperature range considered.

c) Calculate the partial pressures for CO_2 for the carbonate and of O_2 for the oxide at 25 °Plot the the partial pressures for CO_2 for the carbonate and of O_2 for the oxide as function of temperature. For this graph, plot $\ln P$ vs. $1/T$.

Thermal decomposition of $MgCO_{3(s)}$ to $Mg_{(s)}$

Thermodymanic Data	$MgCO_{3(s)}$	$\rightarrow$	$MgO_{(s)}$	$+$	$CO_{2(g)}$	$\rightarrow$	$Mg_{(s)}$	$+$	$\frac{1}{2}O_{2(g)}$
$\Delta H^{\theta}{}_{formation}$ $[kJmol^{-1}]$	-1111.689		-601.241		-393.522		0		0
$\Delta G^{\theta}{}_{formation}$ $[kJmol^{-1}]$	-1028.166		-568.961		-394.389		0		0
$S^{\theta}{}_{species}$ $[JK^{-1}mol^{-1}]$	66.009		26.945		213.911		32.661		205.152

Thermal decomposition of $MgCO_{3(s)}$ to $Mg_{(g)}$

Thermodymanic Data	$MgCO_{3(s)}$	$\rightarrow$	$MgO_{(s)}$	$+$	$CO_{2(g)}$	$\rightarrow$	$Mg_{(g)}$	$+$	$\frac{1}{2}O_{2(g)}$
$\Delta H^{\theta}{}_{formation}$ $[kJmol^{-1}]$	-1111.689		-601.241		-393.522		147.10		0
$\Delta G^{\theta}{}_{formation}$ $[kJmol^{-1}]$	-1028.166		-568.961		-394.389		112.50		0
$S^{\theta}{}_{species}$ $[JK^{-1}mol^{-1}]$	66.009		26.945		213.911		148.60		205.152

Yaws, C. L. (2003). Yaws' Handbook of Thermodynamic and Physical Properties of Chemical Compounds - Knovel. Knovel. Retrieved from https://app.knovel.com/web/toc.v/cid:kpYHTPPCC4/viewerType:toc//root_slug:yaws-handbook-thermodynamic/url_slug:yaws-handbook-thermodynamic?b-q=Yaws%27%20Handbook%20of%20Thermodynamic&sort_on=default&b-subscription=true&b-group-by=true&b-sort-on=default&b- (https://app.knovel.com/web/toc.v/cid:kpYHTPPCC4/viewerType:toc//root_slug:yaws-handbook-thermodynamic/url_slug:yaws-handbook-thermodynamic?b-q=Yaws%27%20Handbook%20of%20Thermodynamic&sort_on=default&b-subscription=true&b-group-by=true&b-sort-on=default&b-)

Calculate the enthalpy change

For $MgCO_{3(s)} \rightarrow MgO_{(s)} + CO_{2(g)}$

$$\Delta H^{\theta}{}_{sys} = \sum \Delta H^{\theta}{}_{f(products)} - \sum \Delta H^{\theta}{}_{f(reactants)}$$
$$\Delta H^{\theta}{}_{sys} = (-601.241) + (-393.522) - (-1111.689)$$
$$\Delta H^{\theta}{}_{sys} = 116.926 kJmol^{-1}$$

For $MgO_{(s)} \rightarrow Mg_{(s)} + \frac{1}{2}O_{2(g)}$

$$\Delta H^{\theta}{}_{sys} = (0) + (0) - (-601.241)$$
$$\Delta H^{\theta}{}_{sys} = 601.241 kJmol^{-1}$$

For $MgO_{(s)} \rightarrow Mg_{g)} + \frac{1}{2}O_{2(g)}$

$$\Delta H^{\theta}{}_{sys} = (147.10) + (0) - (-601.241)$$
$$\Delta H^{\theta}{}_{sys} = 748.341 kJmol^{-1}$$

Calculate the free change

For $MgCO_{3(s)} \rightarrow MgO_{(s)} + CO_{2(g)}$

$$\Delta G^{\theta}{}_{sys} = \sum \Delta G^{\theta}{}_{f(products)} - \sum \Delta G^{\theta}{}_{f(reactants)}$$
$$\Delta G^{\theta}{}_{sys} = (-568.961) + (-394.389) - (-1028.166)$$
$$\Delta G^{\theta}{}_{sys} = 64.816 kJmol^{-1}$$

For $MgO_{(s)} \rightarrow Mg_{(s)} + \frac{1}{2}O_{2(g)}$

$$\Delta G^{\theta}{}_{sys} = (0) + (0) - (-568.961)$$
$$\Delta G^{\theta}{}_{sys} = 568.961 kJmol^{-1}$$

For $MgO_{(s)} \rightarrow Mg_{(g)} + \frac{1}{2}O_{2(g)}$

$$\Delta G^{\theta}{}_{sys} = (112.50) + (0) - (-568.961)$$
$$\Delta G^{\theta}{}_{sys} = 681.461 kJmol^{-1}$$

Calculate the entropy change

For $MgCO_{3(s)} \rightarrow MgO_{(s)} + CO_{2(g)}$

$$\Delta S^{\theta}{}_{sys} = \sum S^{\theta}{}_{s(products)} - \sum S^{\theta}{}_{s(reactants)}$$
$$\Delta S^{\theta}{}_{sys} = (26.945) + (213.911) - (66.009)$$
$$\Delta S^{\theta}{}_{sys} = 174.847 JK^{-1}mol^{-1}$$

For $MgO_{(s)} \rightarrow Mg_{(s)} + \frac{1}{2}O_{2(g)}$

$$\Delta S^{\theta}{}_{sys} = (32.661) + \left(\frac{1}{2}205.152\right) - (26.945)$$
$$\Delta S^{\theta}{}_{sys} = 108.292 JK^{-1}mol^{-1}$$

For $MgO_{(s)} \rightarrow Mg_{(g)} + \frac{1}{2}O_{2(g)}$

$$\Delta S^{\theta}{}_{sys} = (148.60) + \left(\frac{1}{2}205.152\right) - (26.945)$$
$$\Delta S^{\theta}{}_{sys} = 224.2626 JK^{-1}mol^{-1}$$

Sustitute into the Gibbs free energy eq.

For $MgCO_{3(s)} \rightarrow MgO_{(s)} + CO_{2(g)}$

$$\Delta G^{\theta}{}_{sys} = \Delta H^{\theta}{}_{sys} - T\Delta S^{\theta}{}_{sys}$$
$$\Delta G^{\theta}{}_{sys} = 116.926 - 0.174847T$$

For $MgO_{(s)} \rightarrow Mg_{(s)} + \frac{1}{2}O_{2(g)}$

$$\Delta G^{\theta}{}_{sys} = 601.241 - 0.108292T$$

For $MgO_{(s)} \rightarrow Mg_{(g)} + \frac{1}{2}O_{2(g)}$

$$\Delta G^{\theta}{}_{sys} = 748.341 - 0.2242626T$$

Part A

The reaction is spontaneous when $\Delta G^{\theta}{}_{sys} = 0$

$$\Delta G^{\theta}{}_{sys} = \Delta H^{\theta}{}_{sys} - T\Delta S^{\theta}{}_{sys}$$

$$\Delta H^{\theta}{}_{sys} - T\Delta S^{\theta}{}_{sys} = 0$$

$$T = \frac{\Delta H^{\theta}{}_{sys}}{\Delta S^{\theta}{}_{sys}}$$

For $MgCO_{3\,(s)} \rightarrow MgO_{(s)} + CO_{2\,(g)}$

$$T = \frac{116.926}{0.174847}$$

$$\boxed{T = 668.733K}$$

For $MgO_{(s)} \rightarrow Mg_{(s)} + \frac{1}{2}O_{2\,(g)}$

$$T = \frac{601.241}{0.108292}$$

$$\boxed{T = 5552.035K}$$

For $MgO_{(s)} \rightarrow Mg_{(g)} + \frac{1}{2}O_{2\,(g)}$

$$T = \frac{748.341}{0.2242626}$$

$$\boxed{T = 3336.896K}$$

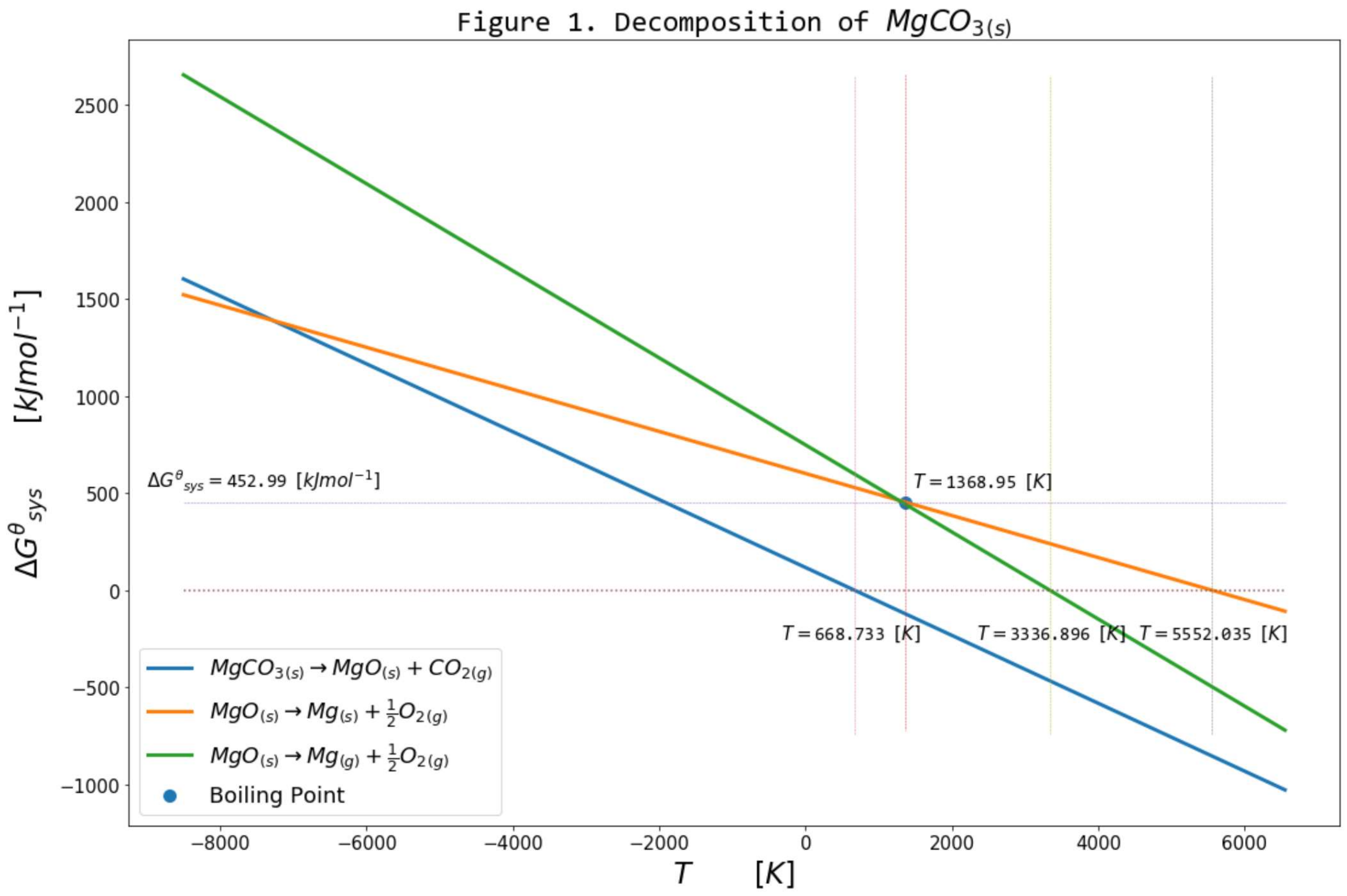
In [36]:

```
# Function to compute the coefficient of linear expansion of rubber at constant tensile stress and volume takes
def G_(H, S, T):
    return H - T*S;

def find_nearest(array, value):
    array = np.asarray(array)
    idx = (np.abs(array - value)).argmin()
    return array[idx]

def plotDeltaG_(plot_name, Tmin, Tmax, H1, S1, G1_name, H2, S2, G2_name, H3, S3, G3_name):
    # Draw the plot's workspace
    scale = 6;
    plt.subplots(figsize=(3*scale, 2*scale));

    # Define constants
    Tmin = Tmin - 1000;
    Tmax = Tmax + 1000;
    T = np.linspace(Tmin, Tmax, 1000);

    # Plot
    G1 = G_(H1, S1, T);
    plt.plot(T, G1, '-', linewidth=3, label=G1_name);

    G2 = G_(H2, S2, T);
    plt.plot(T, G2, '-', linewidth=3, label=G2_name);

    G3 = G_(H3, S3, T);
    plt.plot(T, G3, '-', linewidth=3, label=G3_name);

    # find index of T where T = 1363
    T_boi = find_nearest(T, 1363)
    # find boiling point (Delta G)
    for i in range(len(T)):
        if T[i] == T_boi:
            G_boi = G2[i];
    # plot
    plt.scatter(T_boi,G_boi,s=100, label='Boiling Point')
    plt.plot(np.linspace(T_boi, T_boi, 1000), G3, '--', linewidth=0.5);
    plt.plot(T, np.linspace(G_boi, G_boi, 1000), '--', linewidth=0.5);
    plt.annotate(r'${{\Delta G}^\theta}_{sys} = $' + str(round(G_boi,2)) + ' ' + r'$[kJ mol^{-1}]$', xy=(-9000,G_boi+80), xycoords='data', fontsize=15, **pfont)
    plt.annotate(r'$T = $' + str(round(T_boi,2)) + ' ' + r'$[K]$', xy=(T_boi+100,G_boi+80), xycoords='data', fontsize=15, **pfont)

    Tzeros = np.linspace(0, 0, 1000);
    plt.plot(T, Tzeros, ':', linewidth=1.5);

    # Display plots
    plt.yscale('linear')
    plt.xlabel(r'$T$' + '    ' + r'$[K]$', fontsize=24, **pfont);
    plt.ylabel(r'${{\Delta G}^\theta}_{sys}$' + '    ' + r'$[kJ mol^{-1}]$', fontsize=24, **pfont);
    plt.title(plot_name, size=24, **pfont);
    plt.legend(prop={'size': 18});
    display(plt);
```

Part B

In [55]:
```
plot_name = 'Figure 1. Decomposition of ' + r'$ {Mg C O_3}_{(s)} $';
Tmin = -7500;
Tmax = 5552.035;
H1 = 116.926;
S1 = 0.174847;
G1_name = r'$ {Mg C O_3}_{(s)} \rightarrow {Mg O}_{(s)} + {C O_2}_{(g)} $';
H2 = 601.241;
S2 = 0.108292;
G2_name = r'$ {Mg O}_{(s)} \rightarrow {Mg}_{(s)} + \frac{1}{2}{O_2}_{(g)} $';
H3 = 748.341;
S3 = 0.2242626;
G3_name = r'$ {Mg O}_{(s)} \rightarrow {Mg}_{(g)} + \frac{1}{2}{O_2}_{(g)} $';
plotDeltaG_(plot_name, Tmin, Tmax, H1, S1, G1_name, H2, S2, G2_name, H3, S3, G3_name);
plt.plot(np.linspace(668.733, 668.733, 1000), np.linspace(-740, 2650, 1000), '--', linewidth=0.5);
plt.plot(np.linspace(5552.035, 5552.035, 1000), np.linspace(-740, 2650, 1000), '--', linewidth=0.5);
plt.plot(np.linspace(3336.896, 3336.896, 1000), np.linspace(-740, 2650, 1000), '--', linewidth=0.5);
plt.annotate(r'$T = $' + '668.733 ' + r'$[K]$', xy=(668-1000,0-250), xycoords='data', fontsize=15, **pfont)
plt.annotate(r'$T = $' + '5552.035 ' + r'$[K]$', xy=(5552-1000,0-250), xycoords='data', fontsize=15, **pfont)
plt.annotate(r'$T = $' + '3336.896 ' + r'$[K]$', xy=(3336-1000,0-250), xycoords='data', fontsize=15, **pfont)
display(plt);
```

```
<module 'matplotlib.pyplot' from 'C:\\Users\\oskat\\Anaconda3\\lib\\site-packages\\matplotlib\\pyplot.py'>

<module 'matplotlib.pyplot' from 'C:\\Users\\oskat\\Anaconda3\\lib\\site-packages\\matplotlib\\pyplot.py'>
```



Part C

Clapeyron Eq.

The pressure-temperature boundary at the solid-liquid phase boundary is approximately

$$P(T) = P^* + \frac{\Delta \bar{H}_{melt}}{\Delta \bar{V}_{melt}} Ln\left(\frac{T}{T^*}\right)$$

where: Ln is the natural logarithm, T^* is a reference tempearture, and P^* is the pressure at $T = T^*$. Let use the triple point pressure and temperature as the reference pressure P^* and temperature T^*, respectively, and calculate $P(T)$ at temperatures ranging from $5K$ below the triple point up to the triple point.

Clausius-Clapeyron Eq.

The pressure at the liquid-gas boundary is approximated by

$$P(T) = Pe^{\frac{\Delta \bar{H} vap}{R}\left(\frac{1}{T^*} - \frac{1}{T}\right)}$$

$$ln(P(T)) = ln(P_0) - \frac{\Delta \bar{H}_{vap}}{R}\left(\frac{1}{T^*} - \frac{1}{T}\right)$$

T^* is a reference temperature, and P^* is the pressure at $T = T^*$, assuming that the presure of the gas is described by the ideal-gas law. Look at temperature extending from the triple point up to the critical point. Plot using the critical point as the reference point.

In [194]:

```
def plotPhaseDiagram_(T_inv, P0, H, pName):

    R = 8.3144621 #*u.J/(u.mol*u.K)

    constant_sub = H/R                              # constant
    T_sub_reduced = constant_sub*(T_inv - T_inv[0]) # reduced temperature -> the exponent eqn above
    P = -T_sub_reduced + np.log(P0);

    minP = min(P);
    maxP = max(P);
    for i in range(len(P)):
        if P[i] == minP:
            minP_index = i;
        if P[i] == maxP:
            maxP_index = i;

    if minP_index > maxP_index:
        plot_index = minP_index;
    else:
        plot_index = maxP_index;

    print(P[plot_index])

    T_inv = T_inv[plot_index:]
    P = P[plot_index:]

    # PLOT
    scale = 6;
    fig, ax = plt.subplots(figsize=(3*scale, 2*scale));
    plt.plot(T_inv, P, '-', linewidth=3);

    # Triple and critical points
    #plt.scatter([T_crit/u.K],[P_crit/u.Pa],s=100, label='Critical Point')
    #plt.scatter([T_trip/u.K],[P_trip/u.Pa],s=100, color='c', label='Triple Point')
    #plt.scatter([T_boil/u.K],[P_boil/u.Pa],s=100, color='g', label='Boiling Point')

    # Plot config
    plt.xscale('linear')
    plt.yscale('log')
    plt.xlabel(r'$T^{-1}$' + '    ' + r'$[K^{-1}]$', fontsize=24, **pfont);
    plt.ylabel(r'$ln(P)$' + '    ' + r'$[Pa]$', fontsize=24, **pfont);
    plt.title(pName + ' - calculated Van\'T Hoff', size=24, **pfont);
    #plt.legend(prop={'size': 18});
    #plt.annotate('hola', xy=(T_inv[0],P0), xycoords='data', fontsize=20, **pfont)
    #plt.annotate('liquid', xy=(1100,-250e6), xycoords='data', fontsize=20, **pfont)
    #plt.annotate('gas', xy=(850,50e6), xycoords='data', fontsize=20, **pfont)
    #plt.ylim(-2.5e9, 0.125e9)
    #display(plt);
```

CO_2 Partial Pressures

$$ln(P(T)) = ln(P_0) - \frac{\Delta \bar{H}_{vap}}{R}\left(\frac{1}{T^*} - \frac{1}{T}\right)$$

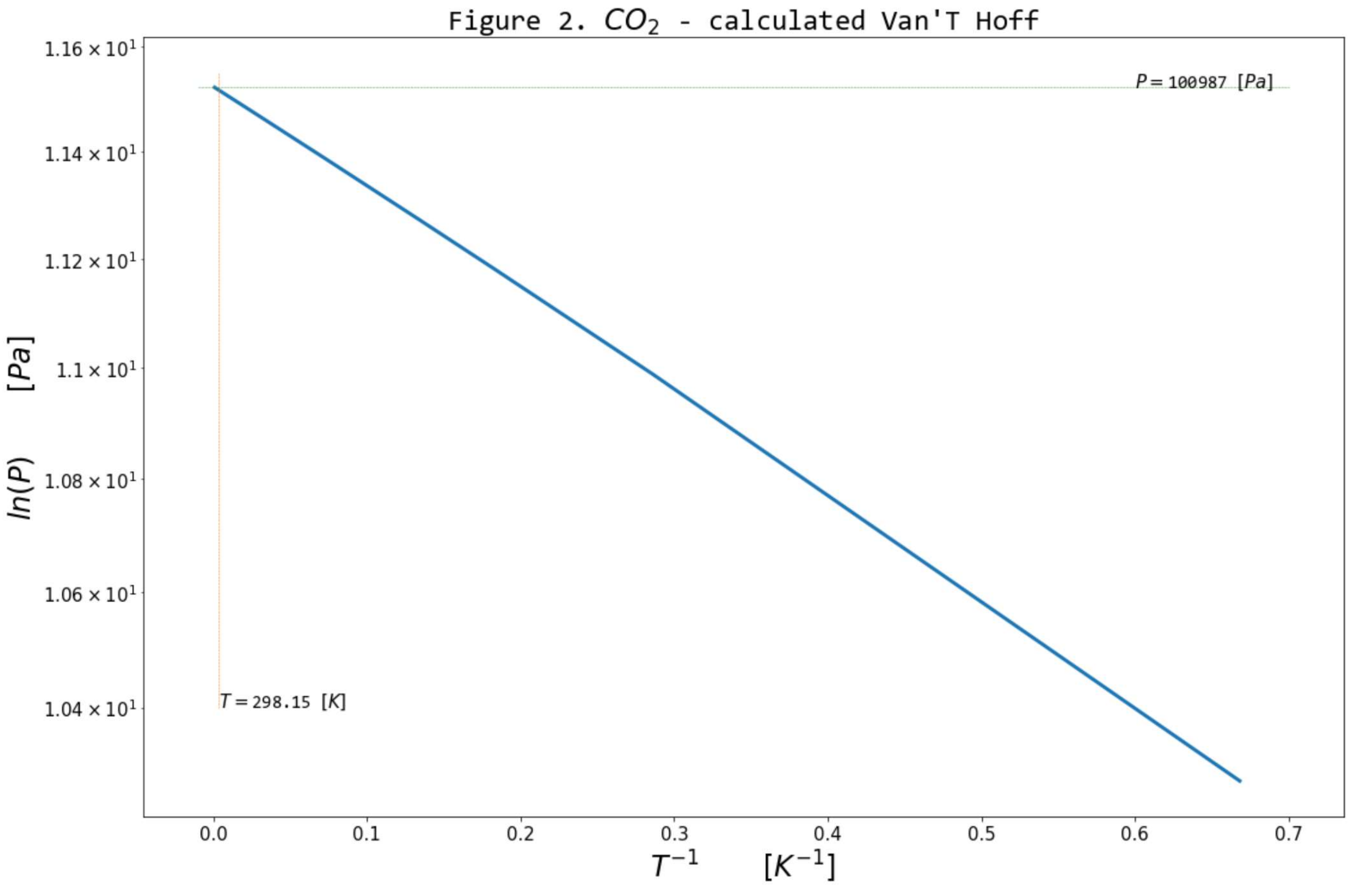
where:
$P_0 = 1atm = 101325Pa$: reference pressure
$\bar{H}_{vap} = 15.55kJ/mol$: enthalpy of evaporation
$R = 8.314J/(molK)$: gas constant
$T^* = 25C = 298.15K$: reference temperature (boiling point)
T : final temperature

$$ln(P(T)) = ln(101325Pa) - \frac{15.55kJ/mol}{8.314J/(molK)}\left(\frac{1}{194.69K} - \frac{1}{T}\right)$$

In [296]:

```
# the temperature and pressure at three important reference points in the water phase
# diagram http://marohn-public.site44.com/Marohn-20180418-005000-Phase-Diagram.html
# (T,P) at the triple point

# array of temp points and cosntants
P0 = 101325 #*u.Pa
H_vap = 15.55 #*u.kJ/u.mol
T_ref = 194.69 #*u.K
T_inv = 1/np.linspace(T_ref-1000, T_ref+1000, 1000) #*u.K

# plot
plotPhaseDiagram_(T_inv, P0, H_vap, 'Figure 2. ' + r'$C O_2$')
plt.plot(np.linspace((1/298.15), (1/298.15), 1000), np.linspace(1.04e1, 1.155e1, 1000),
'--', linewidth=0.5);
plt.annotate(r'$T = $' + '298.15 ' + r'$[K]$', xy=((1/298.15),1.04e1), xycoords='data',
fontsize=15, **pfont);
plt.plot(np.linspace(-0.01, 0.7, 1000), np.linspace(11.52275, 11.52275, 1000), '--', lin
ewidth=0.5);
plt.annotate(r'$P = $' + '100987 ' + r'$[Pa]$', xy=(0.6,11.52275), xycoords='data', font
size=15, **pfont);
display(plt);
```

```
10.274282686035173

<module 'matplotlib.pyplot' from 'C:\\Users\\oskat\\Anaconda3\\lib\\site-packages\\matp
lotlib\\pyplot.py'>
```



O_2 Partial Pressures

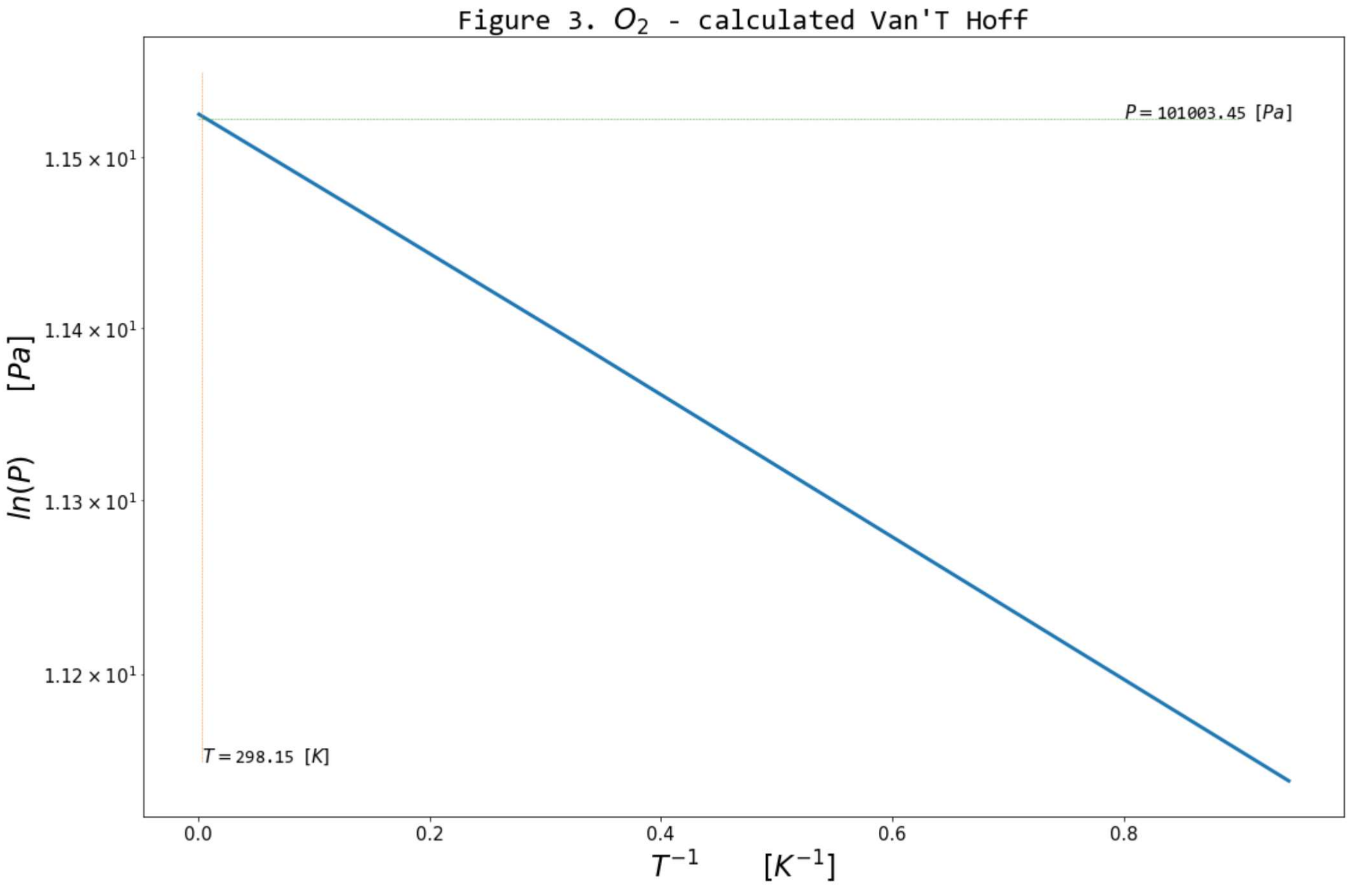
$$ln(P(T)) = ln(P_0) - \frac{\Delta\bar{H}_{vap}}{R}\left(\frac{1}{T^*} - \frac{1}{T}\right)$$

where:
$P_0 = 1atm = 101325Pa$: reference pressure
$\bar{H}_{vap} = 3.4099kJ/mol$: enthalpy of evaporation
$R = 8.314J/(molK)$: gas constant
$T^* = 25C = 298.15K$: reference temperature (boiling point)
T : final temperature

$$ln(P(T)) = ln(101325Pa) - \frac{3.4099kJ/mol}{8.314J/(molK)}\left(\frac{1}{90.15K} - \frac{1}{T}\right)$$

```
In [304]: # the temperature and pressure at three important reference points in the water phase
          # diagram http://marohn-public.site44.com/Marohn-20180418-005000-Phase-Diagram.html
          # (T,P) at the triple point

          # array of temp points and cosntants
          P0 = 101325 #*u.Pa
          H_vap = 3.4099 #*u.kJ/u.mol
          T_ref = 90.15 #*u.K
          T_inv = 1/np.linspace(T_ref-1000, T_ref+1000, 1000) #*u.K

          # plot
          plotPhaseDiagram_(T_inv, P0, H_vap, 'Figure 3. ' + r'$O_2$');
          plt.plot(np.linspace((1/298.15), (1/298.15), 1000), np.linspace(1.115e1, 1.155e1, 1000),
          '--', linewidth=0.5);
          plt.annotate(r'$T = $' + '298.15 ' + r'$[K]$', xy=((1/298.15),1.115e1), xycoords='data',
          fontsize=15, **pfont);
          plt.plot(np.linspace(0.0, 0.9, 1000), np.linspace(11.52291, 11.52291, 1000), '--', linew
          idth=0.5);
          plt.annotate(r'$P = $' + '101003.45 ' + r'$[Pa]$', xy=(0.8,11.52291), xycoords='data', f
          ontsize=15, **pfont);
          display(plt);
          #11.52291
```

```
11.139067324146362

<module 'matplotlib.pyplot' from 'C:\\Users\\oskat\\Anaconda3\\lib\\site-packages\\matp
lotlib\\pyplot.py'>
```



```
In [ ]:
```