Thermodynamics of Materials AD19:
Class Activity 02

Team:

Antonio Osamu Katagiri Tanaka **A01212611@itesm.mx**

Angel Manuel Villalba Rodríguez **A00828035@itesm.mx**

S.Leharne, "The physical chemistry of high-sensitivity differential scanning calorimetry of biopolymers" ChemTexts (2017) 3:1

In [69]:

```
# PYTHON LIBRARIES
%matplotlib inline

import numpy as np
np.seterr(divide='ignore', invalid='ignore')
import pandas as pd
import matplotlib.pyplot as plt
plt.rc('xtick', labelsize=15)
plt.rc('ytick', labelsize=15)

from scipy import special, optimize

# DATA FIG 1
data_df = pd.read_csv("./fig1_data.txt", delimiter=",");
data_df = data_df.sort_values(by=['T']);
data_T  = data_df.iloc[:]['T'];
data_c  = data_df.iloc[:]['C'];
T_1 = np.array(data_T);
C_1 = np.array(data_c);

# DATA FIG 7a
data_df = pd.read_csv("./fig7a_data.txt", delimiter=",");
data_df = data_df.sort_values(by=['T']);
data_T  = data_df.iloc[:]['T'];
data_c  = data_df.iloc[:]['C'];
T_7a = np.array(data_T);
C_7a = np.array(data_c);

# DATA FIG 11
data_df = pd.read_csv("./fig11_data.txt", delimiter=",");
data_df = data_df.sort_values(by=['T']);
data_T  = data_df.iloc[:]['T'];
data_c  = data_df.iloc[:]['C'];
T_11 = np.array(data_T);
C_11 = np.array(data_c);
```

Equation 21:

$$K(T) = e^{\frac{\Delta H_{vH,ref}}{R}\left(\frac{1}{T_{ref}}-\frac{1}{T}\right)+\frac{\Delta C_P}{R}\left(ln\left(\frac{T}{T_{ref}}\right)+\frac{T_{ref}}{T}-1\right)}$$

In [78]:

```
#eq 21
def K_(T, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp):
    # x = [T, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp]

    R = 8.314/1000;

    pwr1 = (Delta_v_ref/R)*((1/T_ref)-(1/T));
    pwr2 = (Delta_Cp/R) * (np.log(T/T_ref) + (T_ref/T) - 1);

    res = np.exp(pwr1 + pwr2);
    return res
```

Equation 18:

$$f_D = f(T) = \frac{K(T)}{1+K(T)}$$

In [79]:

```
#eq 18
def f_(T, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp):

    nume = K_(T, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp);
    deno = 1 + K_(T, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp);

    res = nume/deno;
    return res;
```

$$C_P = \frac{\Delta_{cal}H\Delta_{vH}H}{RT^2}f(T)(1-f(T)) + f(T)\frac{\Delta_{cal}H_{ref}}{\Delta_{vH}H_{ref}}\Delta C_P$$

In [80]:
```python
#eq
def C_(T, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp):
    # x = [Delta_cal, Delta_v, T_ref, Delta_Cp]

    R = 8.314/1000;

    fact1_1 = (Delta_cal_ref * Delta_v_ref)/(R * T**2)
    fact1_2 = f_(T, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp)
    fact1_3 = 1 - f_(T, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp)
    sum1 = fact1_1 * fact1_2 * fact1_3

    fact2_1 = f_(T, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp)
    fact2_2 = Delta_cal_ref/Delta_v_ref
    sum2 = fact2_1 * fact2_2 * Delta_Cp

    res = sum1 + sum2
    return res;
```

In [81]:
```python
def T_ref_(T, C):
    maxC = max(C);
    for i in range(len(C)):
        if C[i] == maxC:
            return int(T[i]+20);
```

In [125]:
```python
# reasonable initial guesses for EOS parameters
Delta_cal_ref = 150
Delta_v_ref = 150
Delta_Cp = 3
p0 = Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp

# PRINT table with fittig parameters
tmplt_str_l = "  ".join(["{:<15}"] + ["{:<15}"]*4);
tmplt_str_s = "  ".join(["{:<15}"] + ["{:<15.7}"]*4);
print('Fitting')
print(tmplt_str_l.format(*['parameters:', 'Delta_cal_H_ref', 'Delta_v_H_ref', 'T_ref', 'Delta_Cp']));
print("-" * 76);

# FIT FIG 1 -------------------------------------------------------------------
T_ref = T_ref_(T_1, C_1); # reasonable initial guesses for T_ref EOS parameter
C_prev_1 = C_(T_1, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp);

results = optimize.curve_fit(C_, T_1, C_1, p0)

Delta_cal_H_ref_1 = results[0][0];
Delta_v_H_ref_1 = results[0][1];
T_ref_1 = results[0][2];
Delta_Cp_1 = results[0][3];

C_fit_1 = C_(T_1, Delta_cal_H_ref_1, Delta_v_H_ref_1, T_ref_1, Delta_Cp_1);
print(tmplt_str_s.format('Figure 1', *results[0], 0));

# FIT FIG 7a -------------------------------------------------------------------
T_ref = T_ref_(T_7a, C_7a); # reasonable initial guesses for T_ref EOS parameter
C_prev_7a = C_(T_7a, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp);

results = optimize.curve_fit(C_, T_7a, C_7a, p0)

Delta_cal_H_ref_7a = results[0][0];
Delta_v_H_ref_7a = results[0][1];
T_ref_7a = results[0][2];
Delta_Cp_7a = results[0][3];

C_fit_7a = C_(T_7a, Delta_cal_H_ref_7a, Delta_v_H_ref_7a, T_ref_7a, Delta_Cp_7a);
print(tmplt_str_s.format('Figure 7a', *results[0], 0));

# FIT FIG 11 -------------------------------------------------------------------
T_ref = T_ref_(T_11, C_11); # reasonable initial guesses for T_ref EOS parameter
C_prev_11 = C_(T_11, Delta_cal_ref, Delta_v_ref, T_ref, Delta_Cp);

results = optimize.curve_fit(C_, T_11, C_11, p0)

Delta_cal_H_ref_11 = results[0][0];
Delta_v_H_ref_11 = results[0][1];
T_ref_11 = results[0][2];
Delta_Cp_11 = results[0][3];

C_fit_11 = C_(T_11, Delta_cal_H_ref_11, Delta_v_H_ref_11, T_ref_11, Delta_Cp_11);
print(tmplt_str_s.format('Figure 11', *results[0], 0));
```

```
Fitting
parameters:      Delta_cal_H_ref  Delta_v_H_ref    T_ref            Delta_Cp
----------------------------------------------------------------------------
Figure 1         195.9418         219.8923         330.2103         3.333808
Figure 7a        505.3589         343.8413         332.1958         14.64676
Figure 11        540.9507         222.163          324.601          1.42091
```

In [131]:

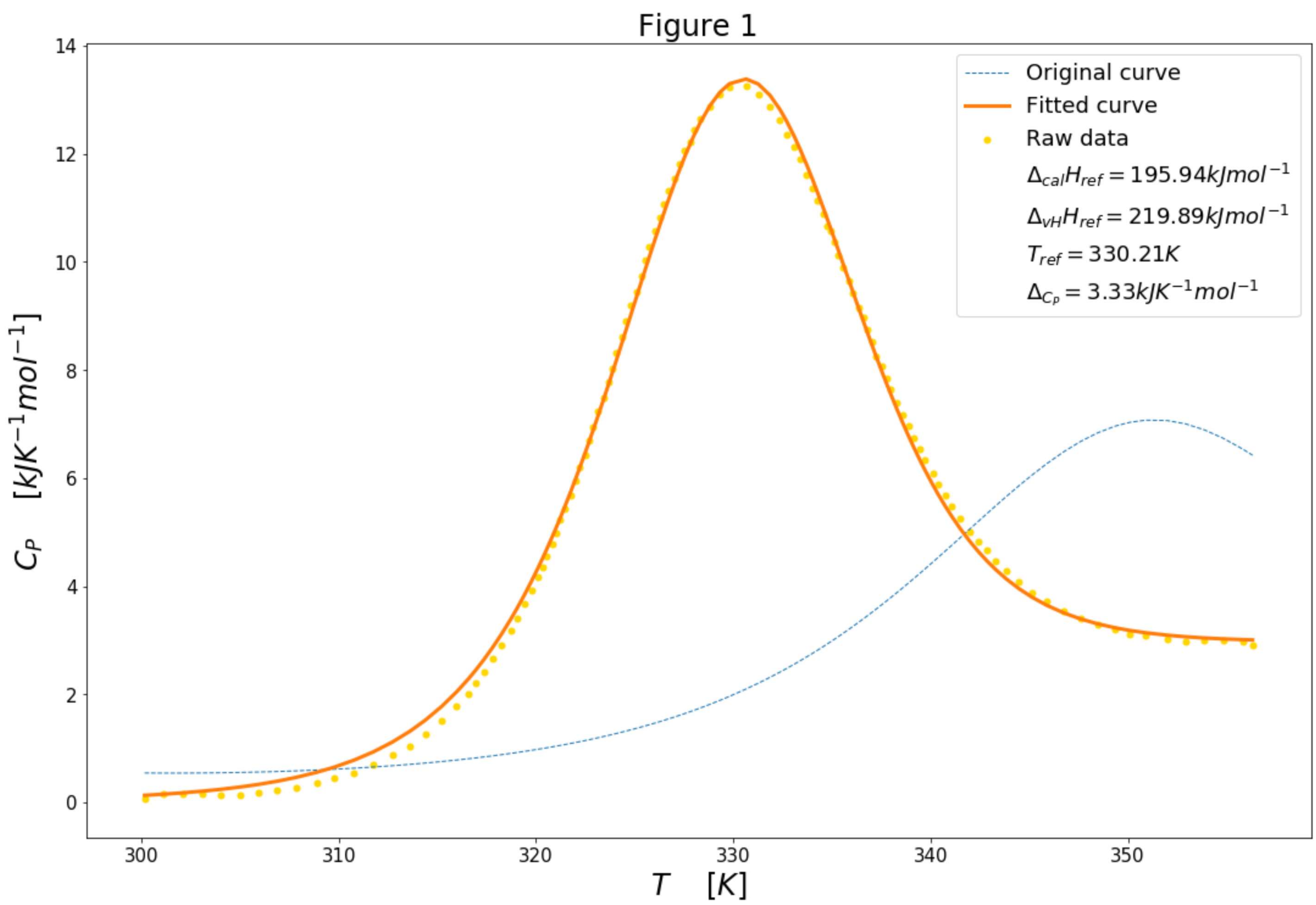
```
# PLOT FIG 1
scale = 6;
plt.subplots(figsize=(3*scale, 2*scale));

# Plot
plt.plot(T_1, C_prev_1, '--', linewidth=1, label='Original curve')
plt.plot(T_1, C_fit_1, '-', linewidth=3, label='Fitted curve')
plt.scatter(T_1, C_1, s=25, color='gold', label='Raw data');

# Print fitting parameters as plot legends
x = T_1[0]
y = C_1[0]
plt.scatter(x, y, s=0, label=r'$\Delta_{cal} H_{ref} = $' + str(round(Delta_cal_H_ref_1, 2)) + r'$kJ {mol}^{-1}$')
plt.scatter(x, y, s=0, label=r'$\Delta_{vH} H_{ref} = $' + str(round(Delta_v_H_ref_1, 2)) + r'$kJ {mol}^{-1}$')
plt.scatter(x, y, s=0, label=r'$T_{ref} = $' + str(round(T_ref_1, 2)) + r'$K$')
plt.scatter(x, y, s=0, label=r'$\Delta_{C_P} = $' + str(round(Delta_Cp_1, 2)) + r'$kJ K^{-1} {mol}^{-1}$')

# Display plots
plt.yscale('linear');
plt.xlabel(r'$T$' + '    ' + r'$[K]$', fontsize=24);
plt.ylabel(r'$C_P$' + '    ' + r'$[kJ K^{-1} mol^{-1}]$', fontsize=24);
plt.title('Figure 1', size=24);
plt.legend(prop={'size': 18});
display(plt);
```

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



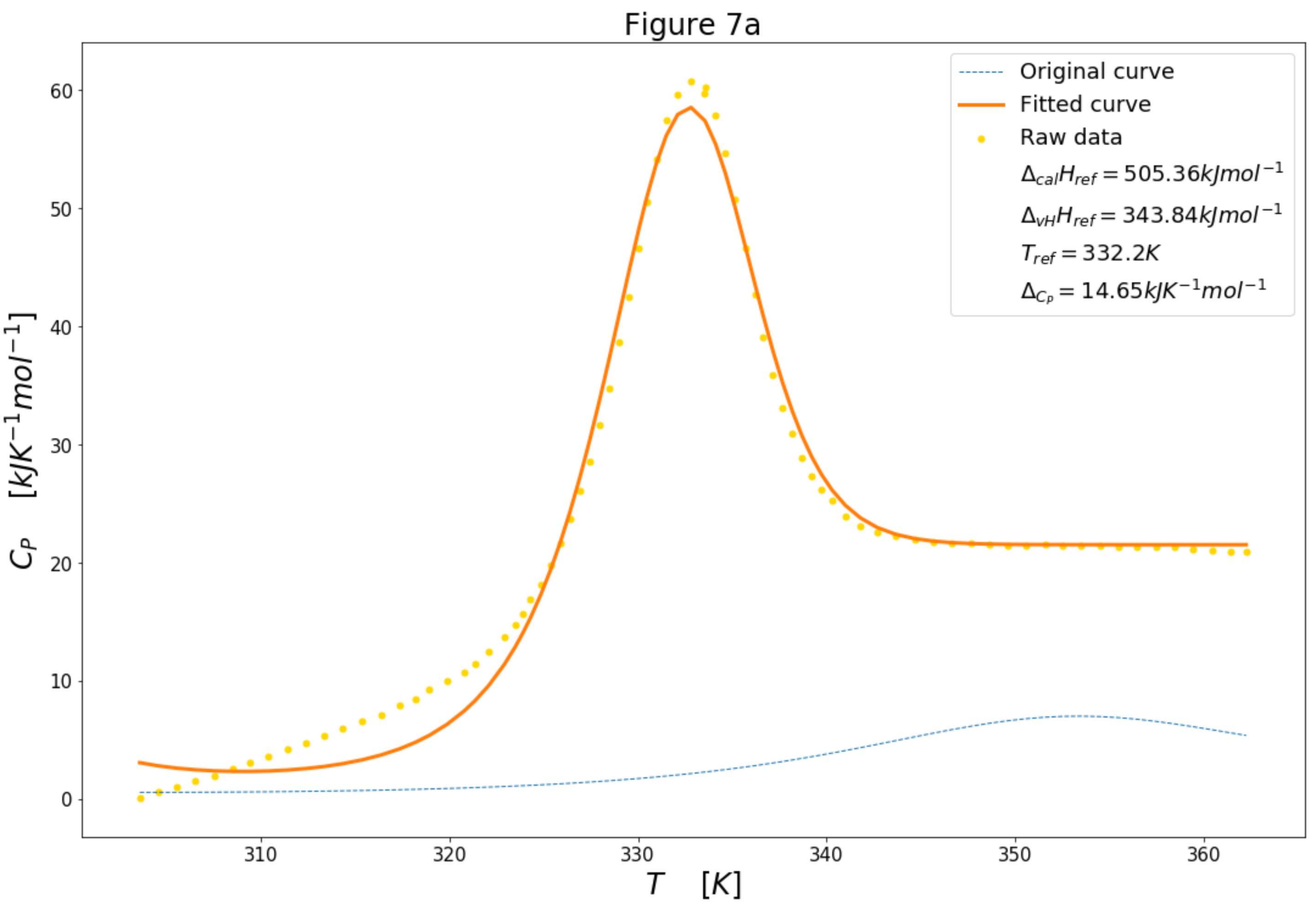
In [132]:
```
# PLOT FIG 7a
scale = 6;
plt.subplots(figsize=(3*scale, 2*scale));

# Plot
plt.plot(T_7a, C_prev_7a, '--', linewidth=1, label='Original curve')
plt.plot(T_7a, C_fit_7a, '-', linewidth=3, label='Fitted curve')
plt.scatter(T_7a, C_7a, s=25, color='gold', label='Raw data');

# Print fitting parameters as plot legends
x = T_7a[0]
y = C_7a[0]
plt.scatter(x, y, s=0, label=r'$\Delta_{cal} H_{ref} = $' + str(round(Delta_cal_H_ref_7a, 2)) + r'$kJ {mol}^{-1}$')
plt.scatter(x, y, s=0, label=r'$\Delta_{vH} H_{ref} = $' + str(round(Delta_v_H_ref_7a, 2)) + r'$kJ {mol}^{-1}$')
plt.scatter(x, y, s=0, label=r'$T_{ref} = $' + str(round(T_ref_7a, 2)) + r'$K$')
plt.scatter(x, y, s=0, label=r'$\Delta_{C_P} = $' + str(round(Delta_Cp_7a, 2)) + r'$kJ K^{-1} {mol}^{-1}$')

# Display plots
plt.yscale('linear');
plt.xlabel(r'$T$' + '     ' + r'$[K]$', fontsize=24);
plt.ylabel(r'$C_P$' + '     ' + r'$[kJ K^{-1} mol^{-1}]$', fontsize=24);
plt.title('Figure 7a', size=24);
plt.legend(prop={'size': 18});
display(plt);
```

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



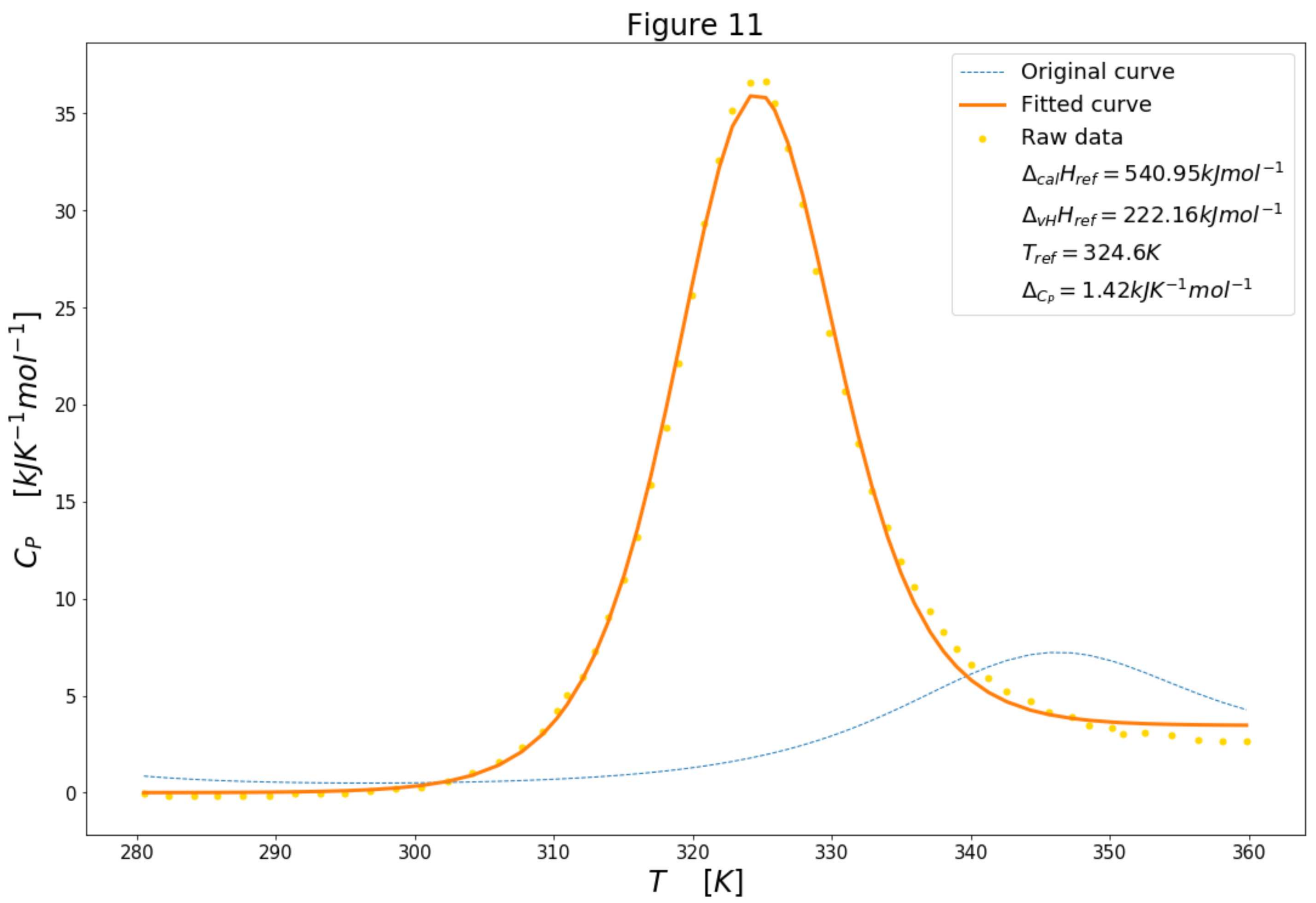
In [134]:
```
# PLOT FIG 11
scale = 6;
plt.subplots(figsize=(3*scale, 2*scale));

# Plot
plt.plot(T_11, C_prev_11, '--', linewidth=1, label='Original curve')
plt.plot(T_11, C_fit_11, '-', linewidth=3, label='Fitted curve')
plt.scatter(T_11, C_11, s=25, color='gold', label='Raw data');

# Print fitting parameters as plot legends
x = T_11[0]
y = C_11[0]
plt.scatter(x, y, s=0, label=r'$\Delta_{cal} H_{ref} = $' + str(round(Delta_cal_H_ref_11, 2)) + r'$kJ {mol}^{-1}$')
plt.scatter(x, y, s=0, label=r'$\Delta_{vH} H_{ref} = $' + str(round(Delta_v_H_ref_11, 2)) + r'$kJ {mol}^{-1}$')
plt.scatter(x, y, s=0, label=r'$T_{ref} = $' + str(round(T_ref_11, 2)) + r'$K$')
plt.scatter(x, y, s=0, label=r'$\Delta_{C_P} = $' + str(round(Delta_Cp_11, 2)) + r'$kJ K^{-1} {mol}^{-1}$')

# Display plots
plt.yscale('linear');
plt.xlabel(r'$T$' + '     ' + r'$[K]$', fontsize=24);
plt.ylabel(r'$C_P$' + '     ' + r'$[kJ K^{-1} mol^{-1}]$', fontsize=24);
plt.title('Figure 11', size=24);
plt.legend(prop={'size': 18});
display(plt);
```

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



In [ ]: