CME502 Final Presentation DSC Data Analysis

Aiden Gauer

Introduction

- Polylactic acid (PLA) is one of the most promising biopolymers due to its natural biodegradability and good mechanical properties
- It is especially promising for single use plastics

Structure of Poly(L-lactic acid):

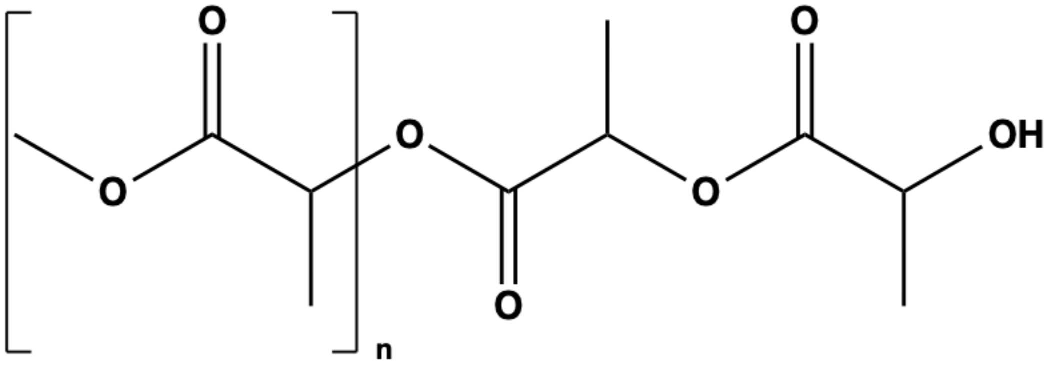
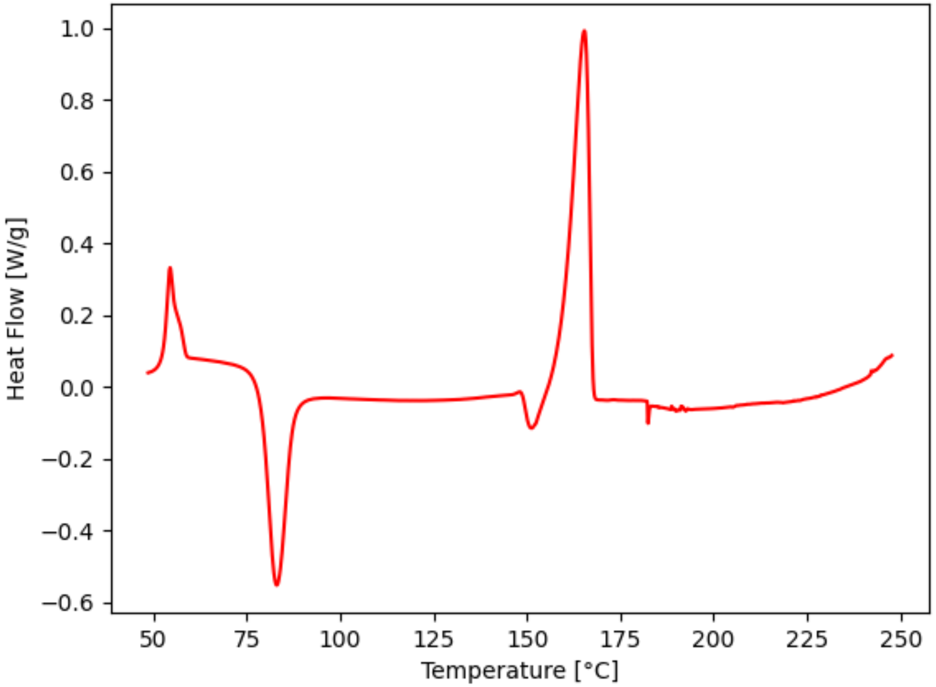


Figure 1. Chemical structure of PLLA (made by ChemDraw)

Our sample:
Mw ~ 160kDa
D content: 0.83%
T_m = 168 °C
T_g = 55 °C

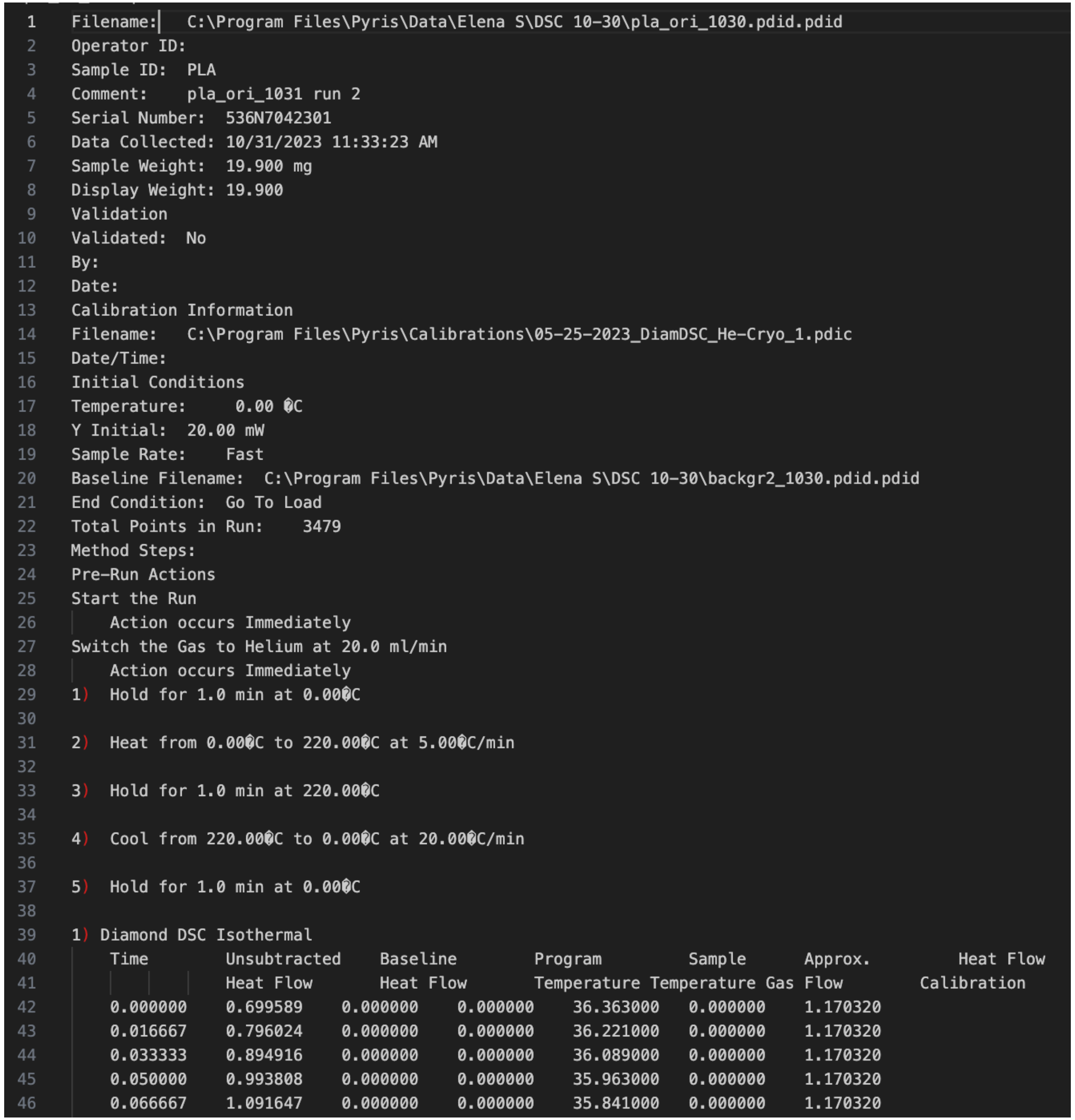
What is DSC?

- Differential scanning calorimetry (DSC) is a thermal analysis instrument which measures the heat flow from a sample as the temperature is ramped
- This gives information about a sample such as its glass transition temperature (T_g), melting temperature (T_m), and crystallinity



Data Formatting

- Data is output into a formatted text file
- There are always 41 lines of headers
- Additionally, there are intermediate text lines that indicate each step in the DSC process

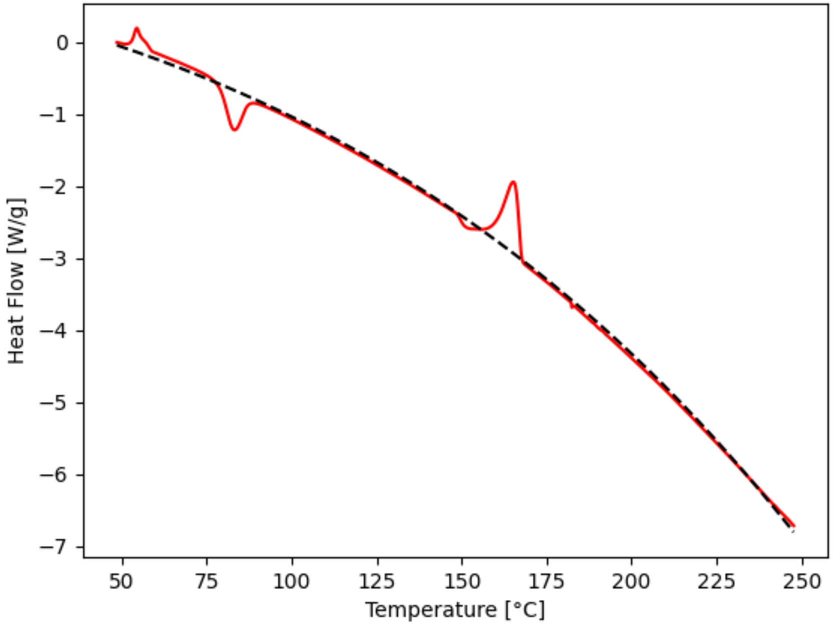


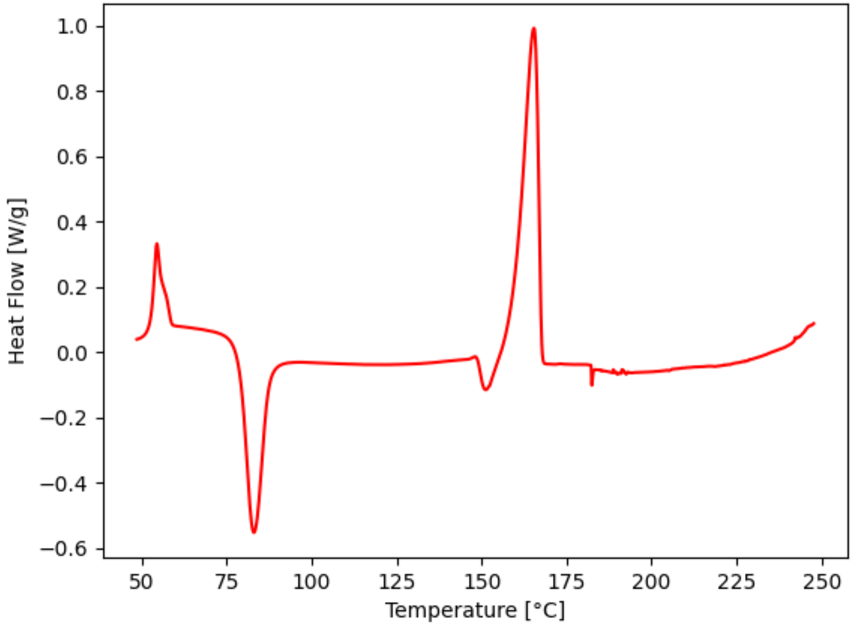
```
1   Filename:   C:\Program Files\Pyris\Data\Elena S\DSC 10-30\pla_ori_1030.pdid.pdid
2   Operator ID:
3   Sample ID:  PLA
4   Comment:    pla_ori_1031 run 2
5   Serial Number:  536N7042301
6   Data Collected: 10/31/2023 11:33:23 AM
7   Sample Weight:  19.900 mg
8   Display Weight: 19.900
9   Validation
10  Validated:  No
11  By:
12  Date:
13  Calibration Information
14  Filename:   C:\Program Files\Pyris\Calibrations\05-25-2023_DiamDSC_He-Cryo_1.pdic
15  Date/Time:
16  Initial Conditions
17  Temperature:     0.00 �C
18  Y Initial:  20.00 mW
19  Sample Rate:    Fast
20  Baseline Filename:  C:\Program Files\Pyris\Data\Elena S\DSC 10-30\backgr2_1030.pdid.pdid
21  End Condition:  Go To Load
22  Total Points in Run:    3479
23  Method Steps:
24  Pre-Run Actions
25  Start the Run
26      Action occurs Immediately
27  Switch the Gas to Helium at 20.0 ml/min
28      Action occurs Immediately
29  1)  Hold for 1.0 min at 0.00�C
30
31  2)  Heat from 0.00�C to 220.00�C at 5.00�C/min
32
33  3)  Hold for 1.0 min at 220.00�C
34
35  4)  Cool from 220.00�C to 0.00�C at 20.00�C/min
36
37  5)  Hold for 1.0 min at 0.00�C
38
39  1) Diamond DSC Isothermal
40      Time        Unsubtracted    Baseline        Program         Sample      Approx.         Heat Flow
41                  Heat Flow       Heat Flow       Temperature Temperature Gas Flow        Calibration
42      0.000000    0.699589    0.000000    0.000000    36.363000   0.000000    1.170320
43      0.016667    0.796024    0.000000    0.000000    36.221000   0.000000    1.170320
44      0.033333    0.894916    0.000000    0.000000    36.089000   0.000000    1.170320
45      0.050000    0.993808    0.000000    0.000000    35.963000   0.000000    1.170320
46      0.066667    1.091647    0.000000    0.000000    35.841000   0.000000    1.170320
```

Normalization

- Heat flow data is normalized by the sample weight
- Unstable data points from the start of the run were truncated from the data set
- Additionally, the DSC curve is normalized by the heat flow trend

$$f(T) = ax^2 + bx + c$$

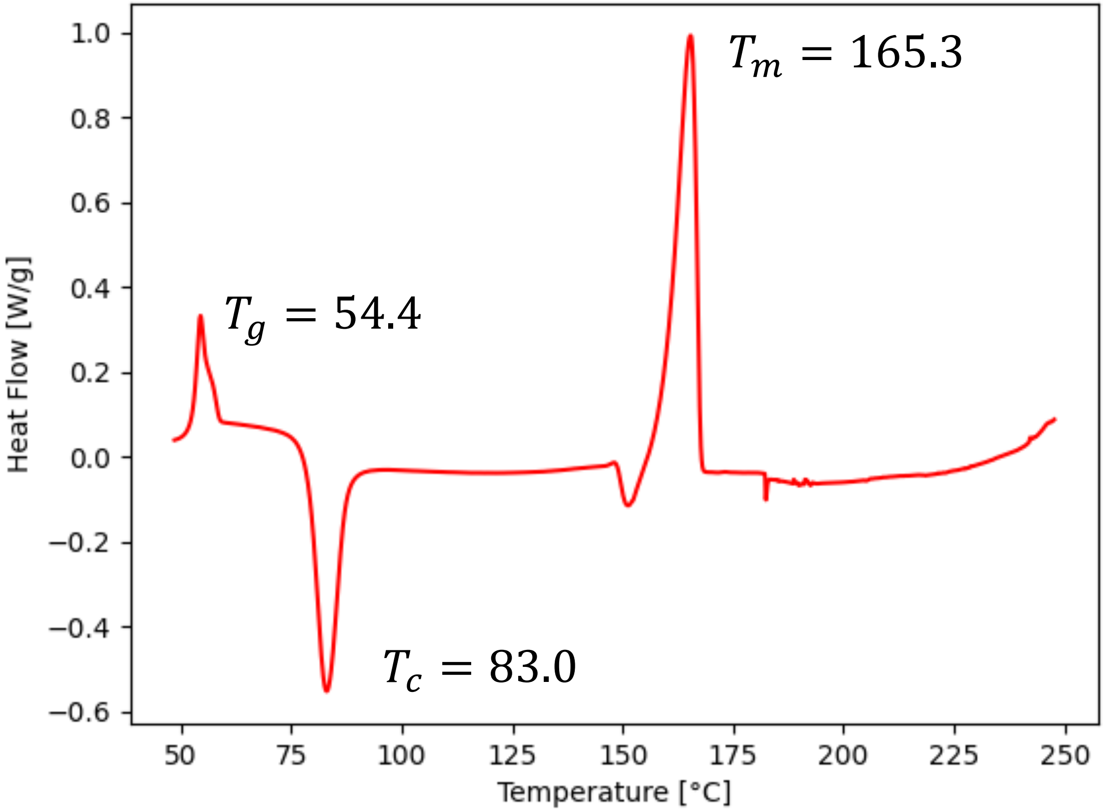




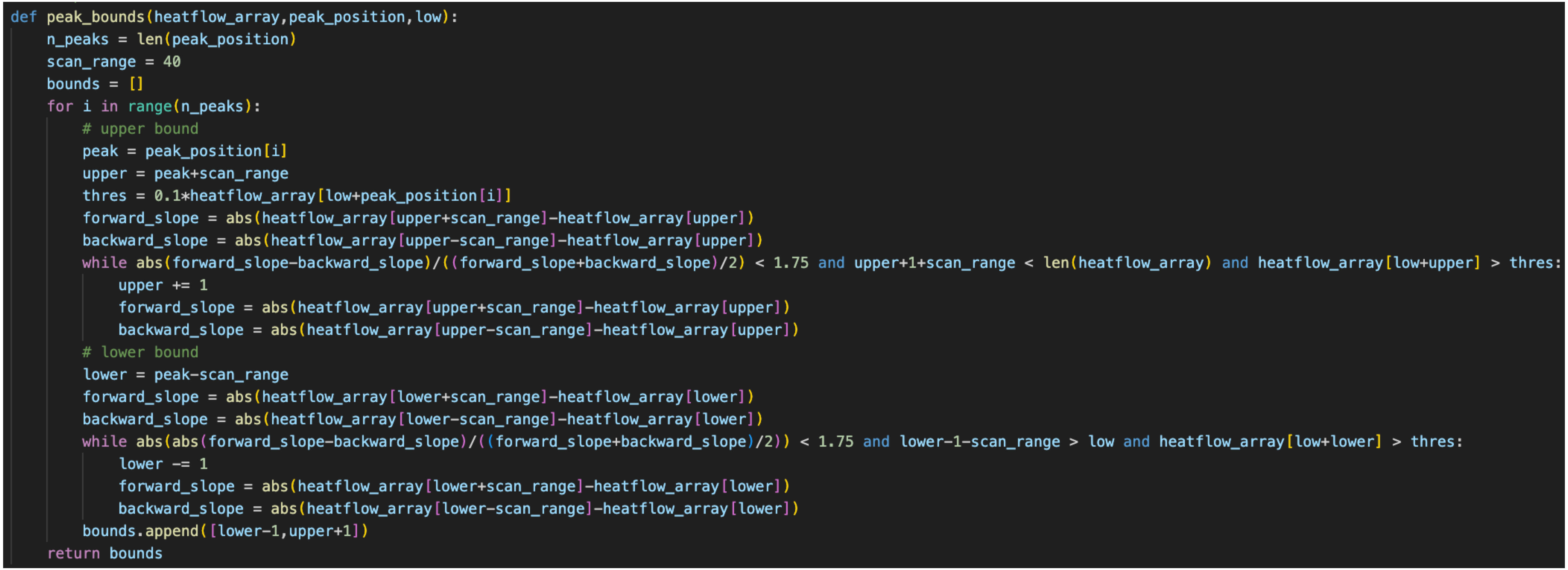
Finding Peak Positions

- Peak positions show phase transition temperatures

```
scipy.signal.find_peaks(x,height=)
```

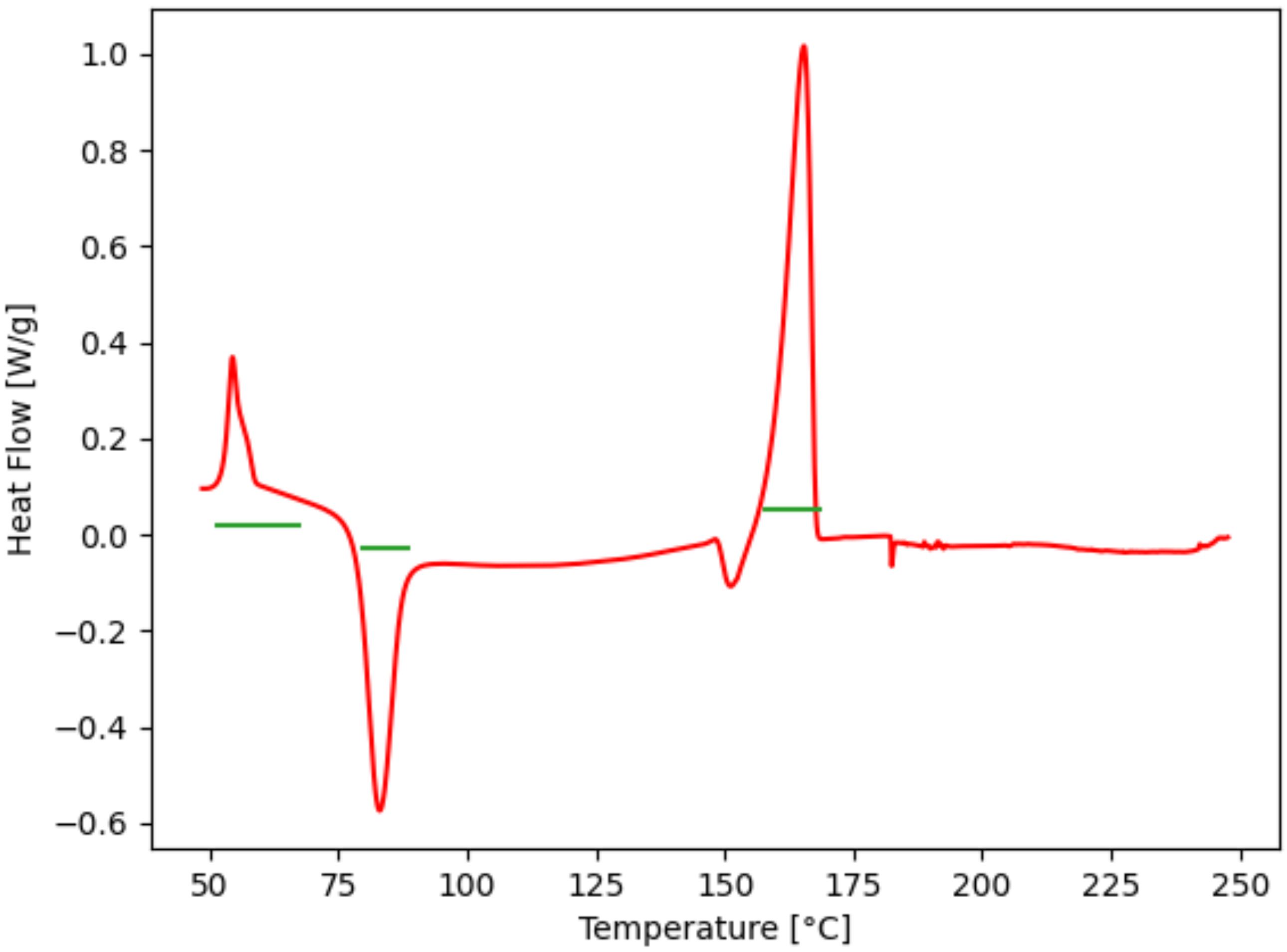


Finding Peak Bounds



```python
def peak_bounds(heatflow_array,peak_position,low):
    n_peaks = len(peak_position)
    scan_range = 40
    bounds = []
    for i in range(n_peaks):
        # upper bound
        peak = peak_position[i]
        upper = peak+scan_range
        thres = 0.1*heatflow_array[low+peak_position[i]]
        forward_slope = abs(heatflow_array[upper+scan_range]-heatflow_array[upper])
        backward_slope = abs(heatflow_array[upper-scan_range]-heatflow_array[upper])
        while abs(forward_slope-backward_slope)/((forward_slope+backward_slope)/2) < 1.75 and upper+1+scan_range < len(heatflow_array) and heatflow_array[low+upper] > thres:
            upper += 1
            forward_slope = abs(heatflow_array[upper+scan_range]-heatflow_array[upper])
            backward_slope = abs(heatflow_array[upper-scan_range]-heatflow_array[upper])
        # lower bound
        lower = peak-scan_range
        forward_slope = abs(heatflow_array[lower+scan_range]-heatflow_array[lower])
        backward_slope = abs(heatflow_array[lower-scan_range]-heatflow_array[lower])
        while abs(abs(forward_slope-backward_slope)/((forward_slope+backward_slope)/2)) < 1.75 and lower-1-scan_range > low and heatflow_array[low+lower] > thres:
            lower -= 1
            forward_slope = abs(heatflow_array[lower+scan_range]-heatflow_array[lower])
            backward_slope = abs(heatflow_array[lower-scan_range]-heatflow_array[lower])
        bounds.append([lower-1,upper+1])
    return bounds
```

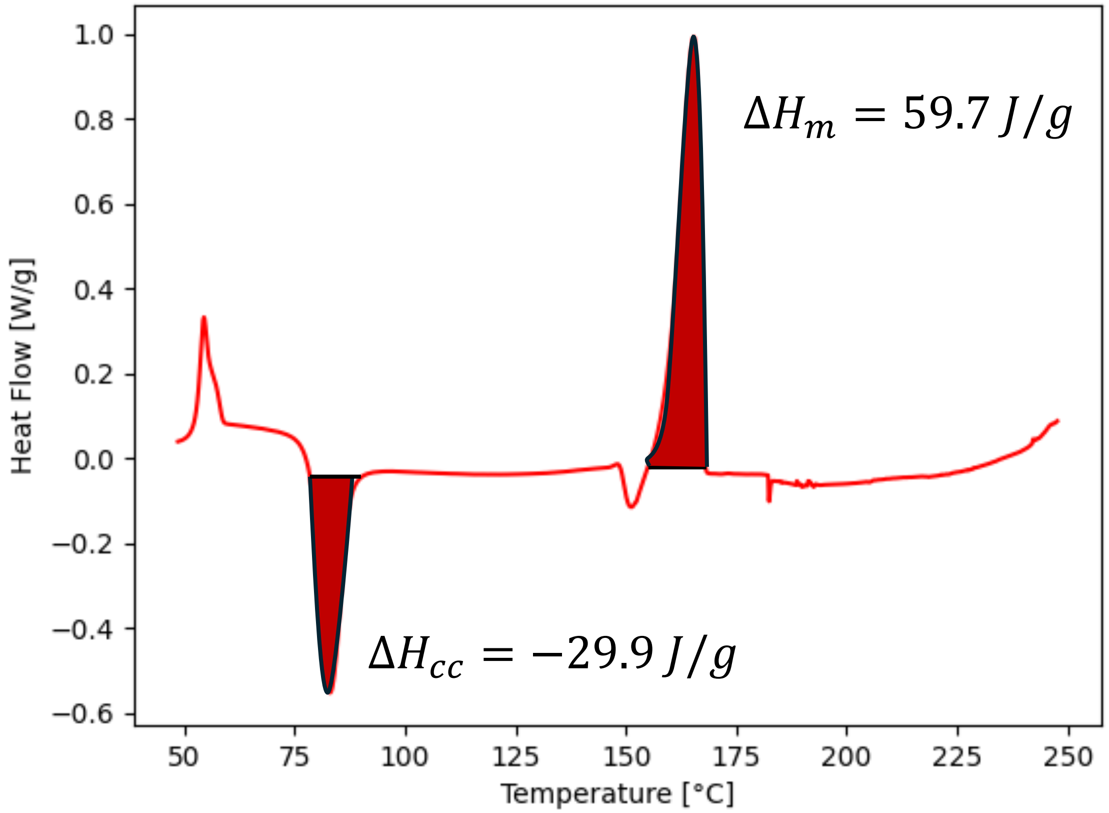
Finding Peak Bounds (cont.)



Calculating Enthalpy

- Integrate heat flow [W/g] over time to get enthalpy [J/g]
- Numerical integration can be used to do this

```
np.trapz(y[lower:upper]
,x[lower:upper])
```



$\Delta H_m = 59.7\,J/g$

$\Delta H_{cc} = -29.9\,J/g$

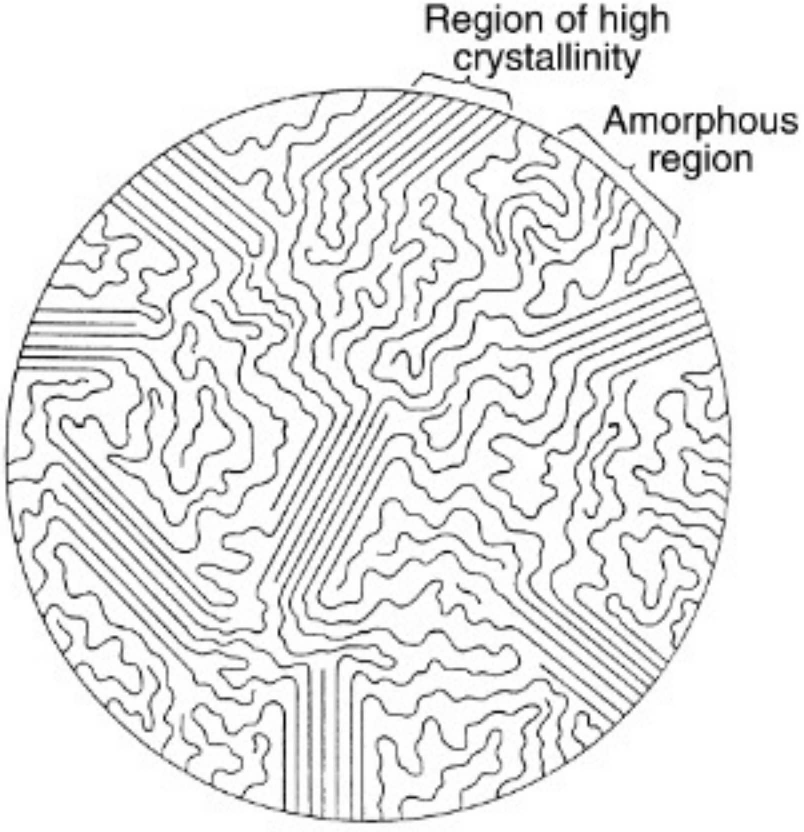
Calculating Crystallinity

- Percent crystallinity of PLA can be calculated by the following formula [Tisserat et al. *Ind. Crops & Prod.*, 2012]

$$\chi_c = \frac{\Delta H_\mathrm{m} - \Delta H_\mathrm{cc}}{\Delta H_\mathrm{m}^0} \times \frac{100}{w}$$

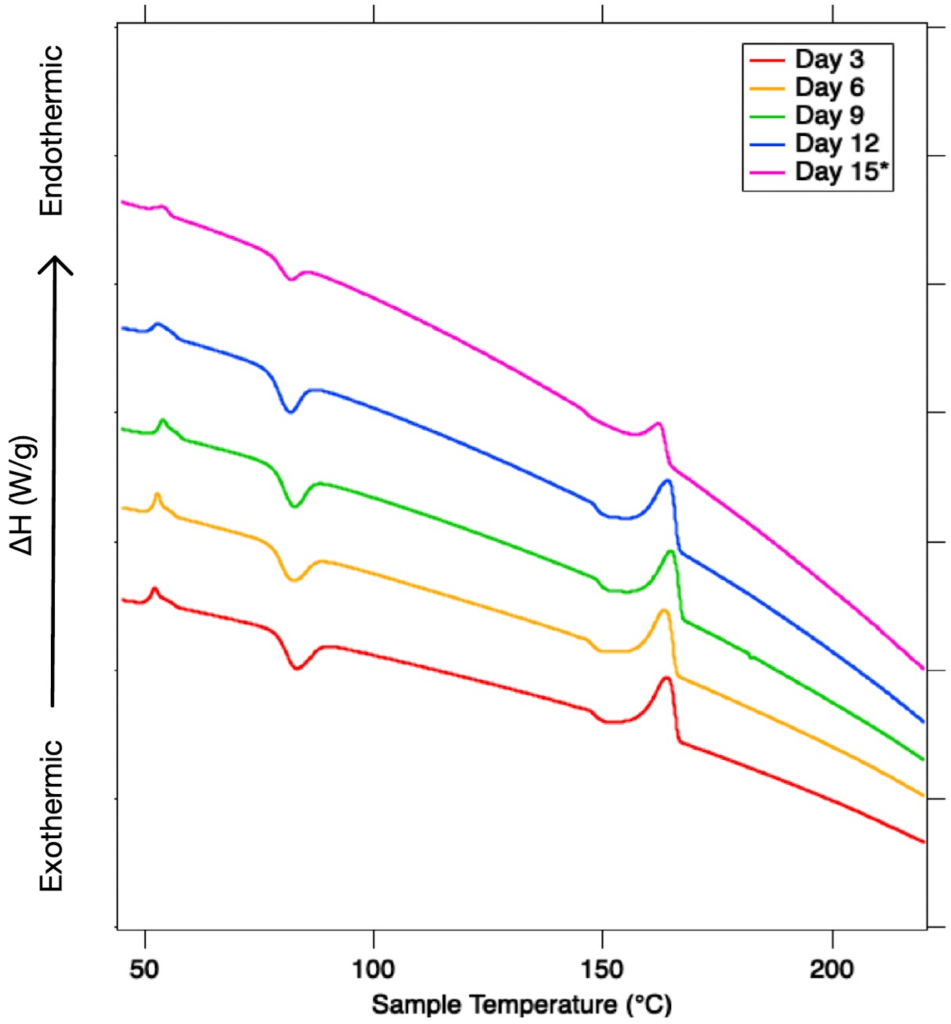
- For the example

$$\chi_c = 31.8\%$$



Conclusion

Sample: Bulk PLA (1mm) with 5 degradation times



Sample (Degraded PLA)	Xc (%)
Day 3	25.2
Day 6	26.2
Day 9	31.8
Day 12	33.5
Day 15	54.3