Science & Technology Facilities Council

Technical Report
RAL-TR-2018-011

Tests and Development of an in situ Calorimeter for IRIS

S Postorino

November 2018

©2018 Science and Technology Facilities Council

This work is licensed under a Creative Commons Attribution 4.0 Unported License.

Enquiries concerning this report should be addressed to:

RAL Library
STFC Rutherford Appleton Laboratory
Harwell Oxford
Didcot
OX11 0QX

Tel: +44(0)1235 445384
Fax: +44(0)1235 446403
email: libraryral@stfc.ac.uk

Science and Technology Facilities Council reports are available online at: http://epubs.stfc.ac.uk

ISSN 1358-6254

Neither the Council nor the Laboratory accept any responsibility for loss or damage arising from the use of information contained in any of their reports or in any communication about their tests or investigations.

Summer Intern
Sara Postorino

Tests and development of an in situ calorimeter for IRIS

Supervisors

Dr. **Victoria Garcia Sakai**

Dr. **Ian Silverwood**

11 June 2017 - 1 September 2017

Contents

Introduction **1**

1 Overview on thermal analysis tecniques **3**
1.1 DTA:Differential Thermal Analysis 4
1.2 DSC:Differential Scanning Calorimetry 4
1.3 Why a DSC system for this instrument? 5

2 First instrument setup and previous results **7**
2.1 First instrument setup (1 heater per can) 7
2.2 Previous results 7

3 DSC Measurements **9**
3.1 Toluene 10
3.2 Glycerol 12
3.3 Buthanol 14
3.4 Ethylene Glycol 16
3.5 Cyclohexanol 18
3.6 Dimethylsulfoxide 20

4 Linkers **21**
4.1 Honeycomb Linkers 22
4.1.1 All temperatures 22
4.1.2 Gradient 23
4.1.3 Heaters and currents 23
4.1.4 Temperature, current and heating power subtractions 25
4.2 Aluminium Linkers 27
4.2.1 All temperatures 27
4.2.2 Gradient 28
4.2.3 Heaters and currents 28
4.2.4 Temperature, current and heating power subtractions 30
4.3 Stainless Steel Linkers 32
4.3.1 All temperatures 32
4.3.2 Gradient 33
4.3.3 Heaters and currents 33
4.3.4 Temperature, current and heating power subtractions 35

4.4 G-10 Linkers . . . 37
4.4.1 All temperatures . . . 37
4.4.2 Gradient . . . 38
4.4.3 Heaters and currents . . . 38
4.4.4 Temperature, current and heating power subtractions . . . 40
4.5 Comments on results . . . 42

5 Environment **43**
5.1 Helium vs Vacuum . . . 43
5.2 Ramping CCR vs fixed temperature CCR . . . 46

6 Other tests **49**
6.1 Reversed sensors tests . . . 50
6.2 Heaters off tests . . . 57

7 Empty-empty characterization **59**
7.1 Empty-empty tests with fixed CCR . . . 60
7.2 Empty-empty tests with ramping CCR . . . 62
7.3 Empty-emptytests with heaters off . . . 63

8 New Instrument setup with two heaters per can **65**
8.1 New measurements showing glass transitions . . . 65
8.1.1 Toluene . . . 65
8.1.2 Glycerol . . . 72

9 Future developement **79**
9.1 Neutron measurements . . . 79
9.2 Limitations and possible improvements . . . 80

Appendix A **81**

A List of measurements **81**
A.1 Linkers tests . . . 81
A.2 Further tests with G-10 linkers . . . 82
A.3 Empty-Empty characterization . . . 82
A.4 Reversed heaters (Empty, Toluene, Glycerol) . . . 83
A.5 Two heaters per can (Toluene, Glycerol, attempts with steps) . . . 84
A.6 Two heaters per can steps separating data collection . . . 85

Appendix B **87**

B Python scripts **87**
B.1 Data extraction for one-heaters-per-can setup . . . 87
B.2 Data extraction for two-heaters-per-can setup . . . 89
B.3 Moving averages and graphs for one-heaters-per-can setup . . . 92
B.4 Moving averages and graphs for two-heaters-per-can setup . . . 97

Introduction

Glass transitions can be studied from a thermodynamical point of view using thermal analysis, but we can also study the molecular-scale structure and dynamics using neutrons. This project aims to develop a new instrument capable of doing thermal analysis during neutron experiments.

The set up on which I worked is suitable only for low temperature transitions, precisely in a range between 15K (CCR low limit) and 325K (temperature sensors upper limit). Another limit of the system used is given by the can used to hold the sample; due to its structure one can use only liquids and it is better if the viscosity is not much more than water-like. These constrains affected the choice of the glass formers.

It was necessary to change the instrument setup, so half of the measurements has been taken with the old one-heater-per-can set up while the second half has been taken using two heaters per can. The last modification to the instrument allowed to clearly see glass transitions

This report is divided in the following parts:

- Overview on thermal analysis techniques
- DSC analysis of glass formers
- Brief description of instrument, environment and previous tests
- Tests on the linkers to achieve the best thermal insulation of sample and reference can
- Tests on the environment (helium, vacuum, CCR Temperature)
- Other tests (reversed sensors, heaters off test)
- Characterization of the system using only empty cans
- New set up with two heaters per can and relative measurements showing glass transitions
- Future developments

In the appendixes at the end of the report there are:

- List of measurements done during the summer 2017
- Python scripts for data analysis

CHAPTER 1

Overview on thermal analysis tecniques

In thermal analysis techniques, a physical property is measured as a function of temperature while the sample is subjected to a predefined heating or cooling program (usually linear). The two analyzed in the following pages are :

- DTA :Differential Thermal Analysis
- DSC :Differential Scanning Calorimetry

These two tecniques are usually confused because of the similar way in which the experiments are performed.

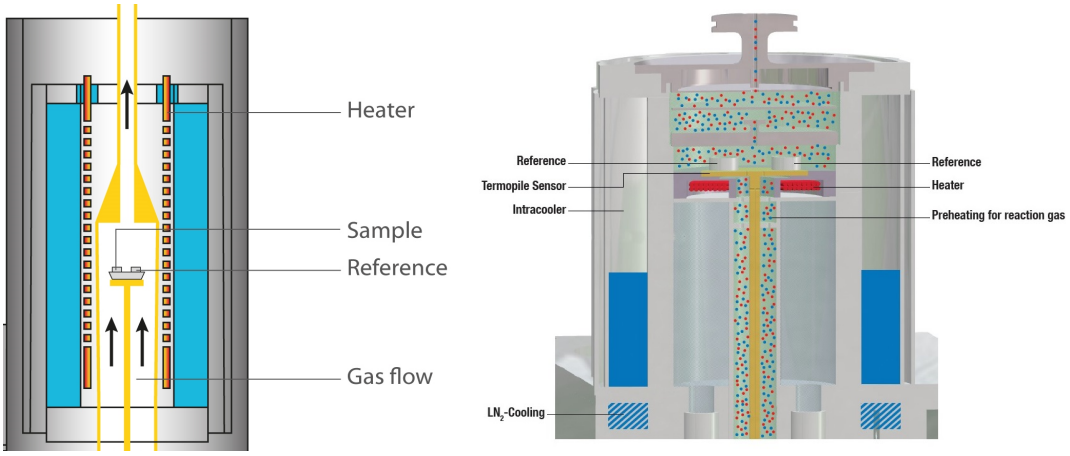


Figure 1.1: On the left there is a DTA in which one heater works for both sample and reference, while on the left there is a DSC with to separated heat sources

1.1 DTA:Differential Thermal Analysis

DTA is a technique in which the difference in temperature between the sample and a thermally inert reference is continuously recorded as a function of temperature or time. Both sample and inert reference are under a linear temperature ramp heated by a single source. If any physical or chemical process occurs in the sample it will be accompanied by an additional enthalpic change that can be either endothermic or exothermic; respectively the temperature of the sample will be lower or higher than that of the reference and a difference ΔT will show in the graph. If there is zero temperature difference between sample and reference, sample does not undergo any chemical or physical change.

When we compare the two temperatures we get graph like the following in which may appear steps and peaks. Peaks indicates melting and crystallization, while the step are for the glass transitions.The area under a DTA peak is proportional to enthalpy and independent of heat capacity.

In the usual DTA the sample environment is kept homogeneous using a gas. As it will be shown in next chapters this method of analysis is not suitable for our instrument because of the big inhomogeneity of the environment.

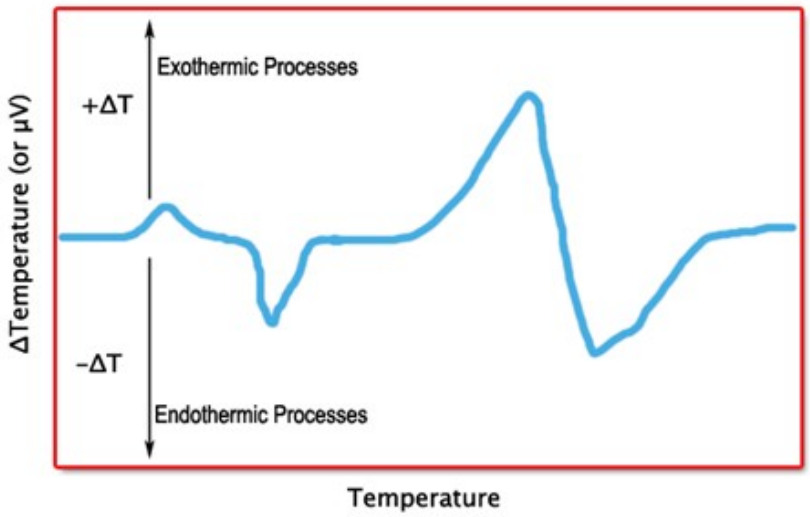


Figure 1.2: Thermogram from a DTA measurement

1.2 DSC:Differential Scanning Calorimetry

Differential scanning calorimetry, or DSC, is a thermoanalytical technique in which the difference in the amount of heat required to increase the temperature of sample and reference is measured as a function of temperature. Both the sample and reference are maintained at nearly the same temperature throughout the experiment. Generally, the temperature program for a DSC analysis is designed such that the sample holder temperature increases linearly as a function of time. The reference should have a well-defined heat capacity over the range of temperatures to be scanned.

When the sample undergoes a physical transformation such as phase transitions, more or less heat will need to flow to it than the reference to maintain both at the same temperature. Whether less or more heat must flow to the sample depends on whether the process is exothermic or endothermic. By observing the difference in heat flow between the sample and reference, differential scanning calorimeters are able to measure the amount of heat absorbed

or released during such transitions. DSC may also be used to observe glass transitions in which we have a change in the specific heat.

The result of a DSC experiment is a curve of heat flux versus temperature or time. This curve can be used to calculate enthalpies of transitions. This is done by integrating the peak corresponding to a given transition. It can be shown that the enthalpy of transition can be expressed using the following equation:

$$\Delta H = KA \tag{1.2.1}$$

where ΔH is the enthalpy of transition, K is the calorimetric constant, and A is the area under the curve. The calorimetric constant will vary from instrument to instrument, and can be determined by analyzing a well-characterized sample with known enthalpies of transition.

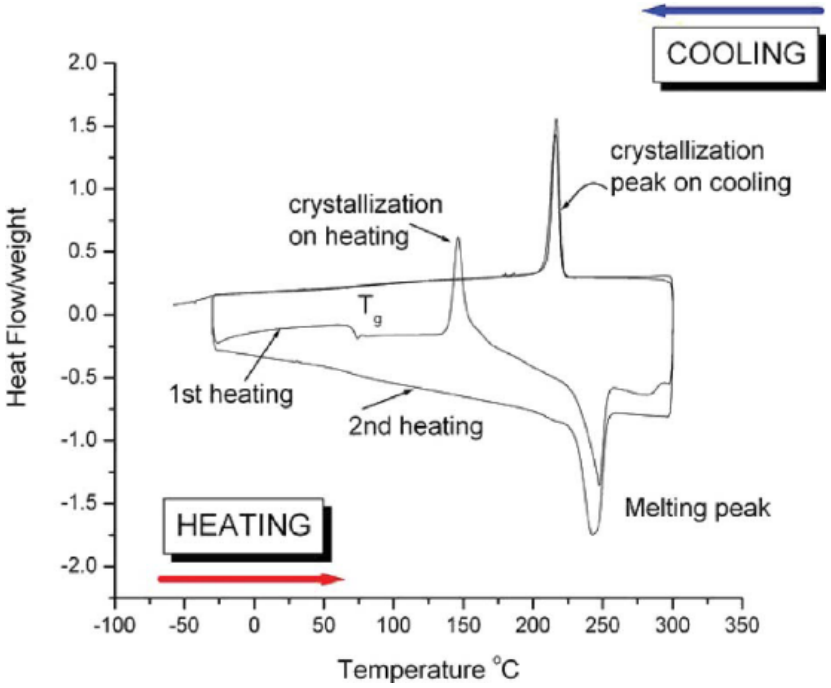


Figure 1.3: Heat flow curve from a DSC Measurement

1.3 Why a DSC system for this instrument?

In the next chapters it will be clear that now it is not possible achive the homogeneity of the environment inside our cryostat due to its big dimension and its structure, but one can try to reach a good control of temperature locally around the cans using separate heaters.

Anyway the temperature analysis can be used beacause there are small temperature changes that are faster than the control system to respond. This is not a proper DTA because of the different experimental procedure and system, but the data used and their analysis is the same.

CHAPTER 2

First instrument setup and previous results

2.1 First instrument setup (1 heater per can)

The instrument is composed of two identical cans, one for the sample and the other empty used as reference for the subtractions. They are both connected to a bracket,the sample can is connected downward so it can be in the in the neutron beam, while the reference can is connected upward so it can stay in similar condition. On the reference there is a cap in order to keep the reference more insulated.

Cans are $3D$ printed, the main component is Aluminium but there is some Silicon in it. The choice of aluminium is due to its good thermal conductivity and this aims to get a small temperature gradient along the can although heating from one end, but the presence of silicon may be a problem.
The geometry used is anular because it is the best for IRIS:the body of the can is made fo two cilinders one inside the other with a spacing of $0.5mm$ which is the limit of this kind of $3D$ printing.

The first linkers used are made of stainless steel and have an hexagonal pattern to reduce themal conduction because it is important for reference and sample cans to be thermally insulated .

The heaters are on the upper end of reference and lower end of sample, while the temperature sensors are on both ends of each can. Cernox sensors are in use and these give an upper limitation on temperatures $(325K)$, but they have very good sensitivity that can be exploited to make thermal analysis.

All the measurements are performed in a long neck CCR with a lower temperature limit of about $15K$.

2.2 Previous results

Previous measurements done with water, glycerol and empty cans show the following results:

- The empty can test shows that linearity in temperature during the heating can be achieved
- The melting transitions of water is clearly visible in the heater output and there is a

peak also in temperature signal. The baseline in this case is good and the reference seems to be well insulated from the sample.

- Glycerol measurement shows a quite different situation: the glass transition is not clearly visible but the main problem is that reference and sample behave in a similar way and the linearity is lost in both signals
- Other tests show that it could be interesting to perform next experiments with 30*mbar* of helium rather than vacuum and using the sensor opposite to the heater for the temperature control.

CHAPTER 3

DSC Measurements

Measurements have been performed on the TAinstrument DSC present in the instrument room in R80. To choose the glass formers there are some constraints:

- Water-like viscosity or lower at room temperature ($\eta \leq 1Pa \cdot s$)
- Glass transition temperature between 113K (due to the limits of the DSC used) and 300K (due to the limits of the sensors of our instrument)
- Availability in the ISIS chemical laboratories

Good candidates are listed next:

- Toluene
- Glycerol
- Buthanol
- Etylene Glycol
- Cycloheaxanol
- Dymethilsulfoxide

3.1 Toluene

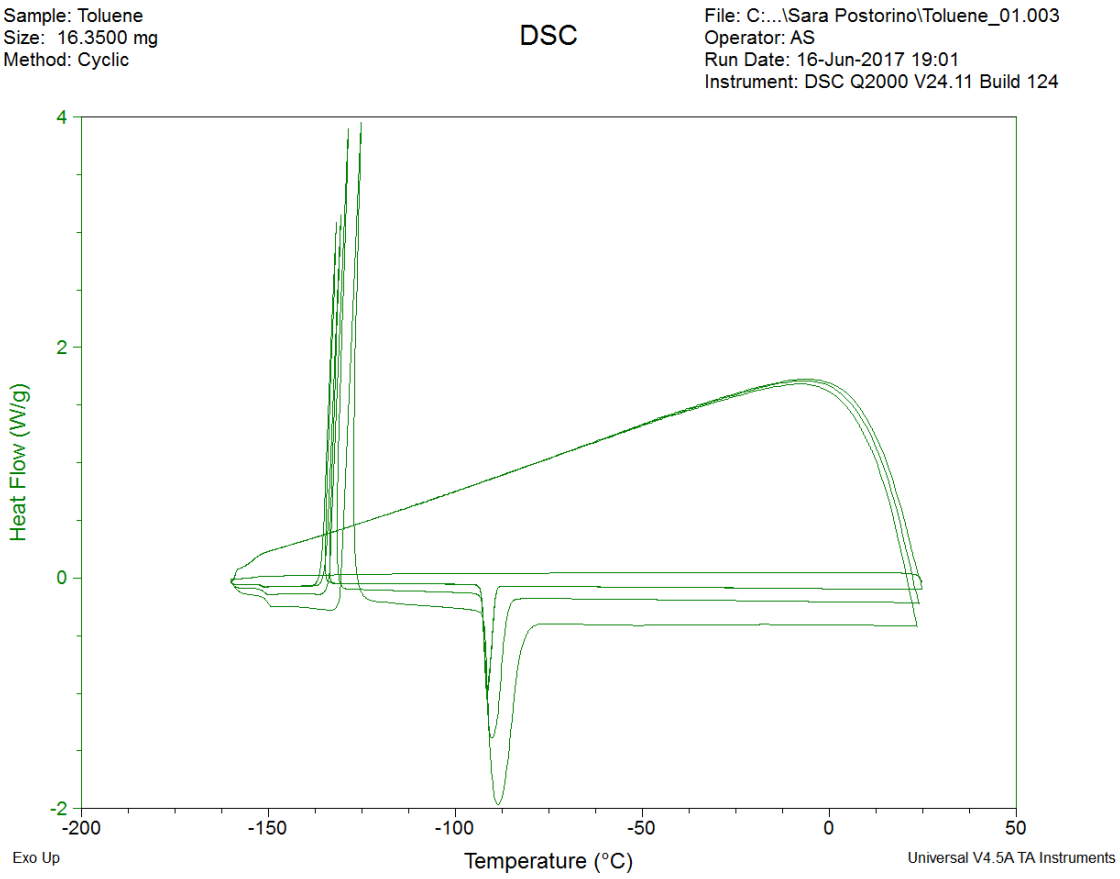


Figure 3.1: Complete Heat flow curve of Toluene showing different ramp rates: cooling at maximum rate and then at $2K/min$; heating at $10K/min$, $5K/min$ and $2K/min$

Toluene shows melting, crystallization and glass transition well defined and with a good separation between them. This made it to be the best candidate for the measurements with the calorimeter prototype.
Using very slow rate the glass transition becomes smaller but it is still visible.

The analysis of the transitions is reported in the following page.

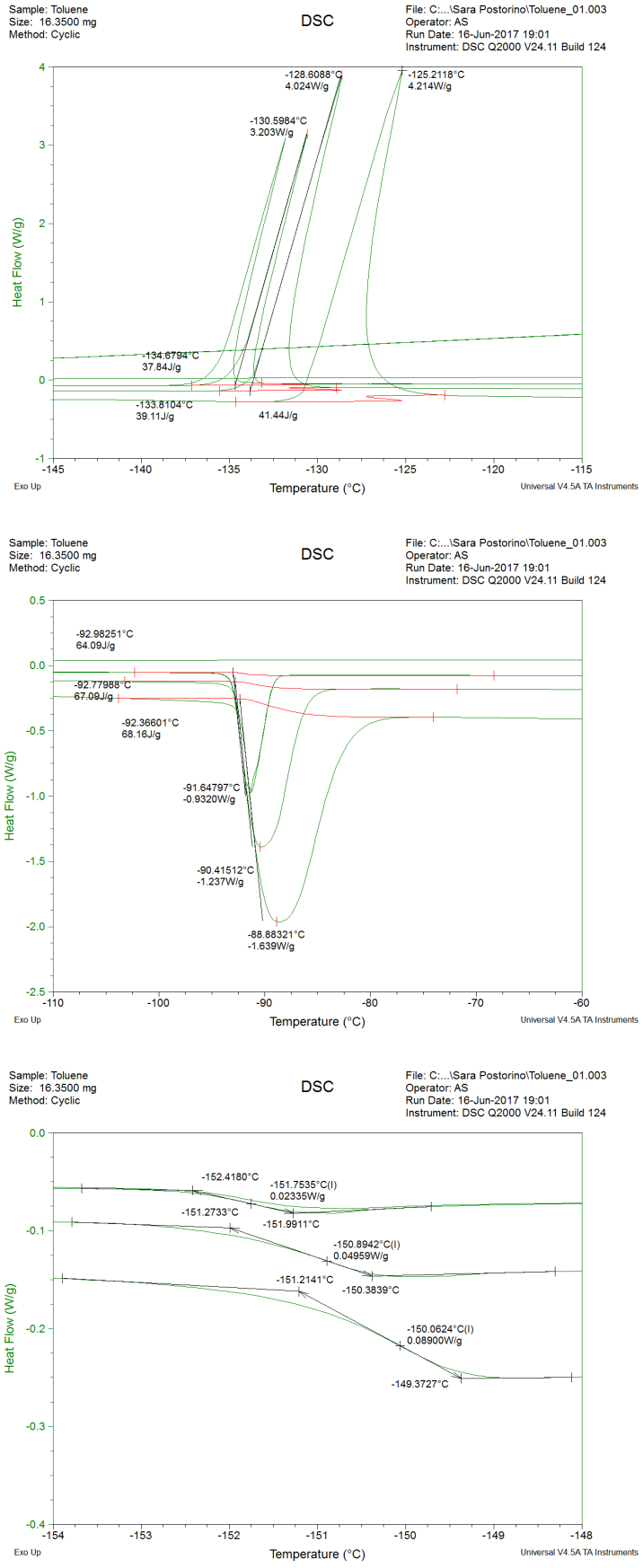


Figure 3.2: a)Crystallization peaks at different rates, b) Melting peaks at different rates, c) Glass transitions at different rates

3.2 Glycerol

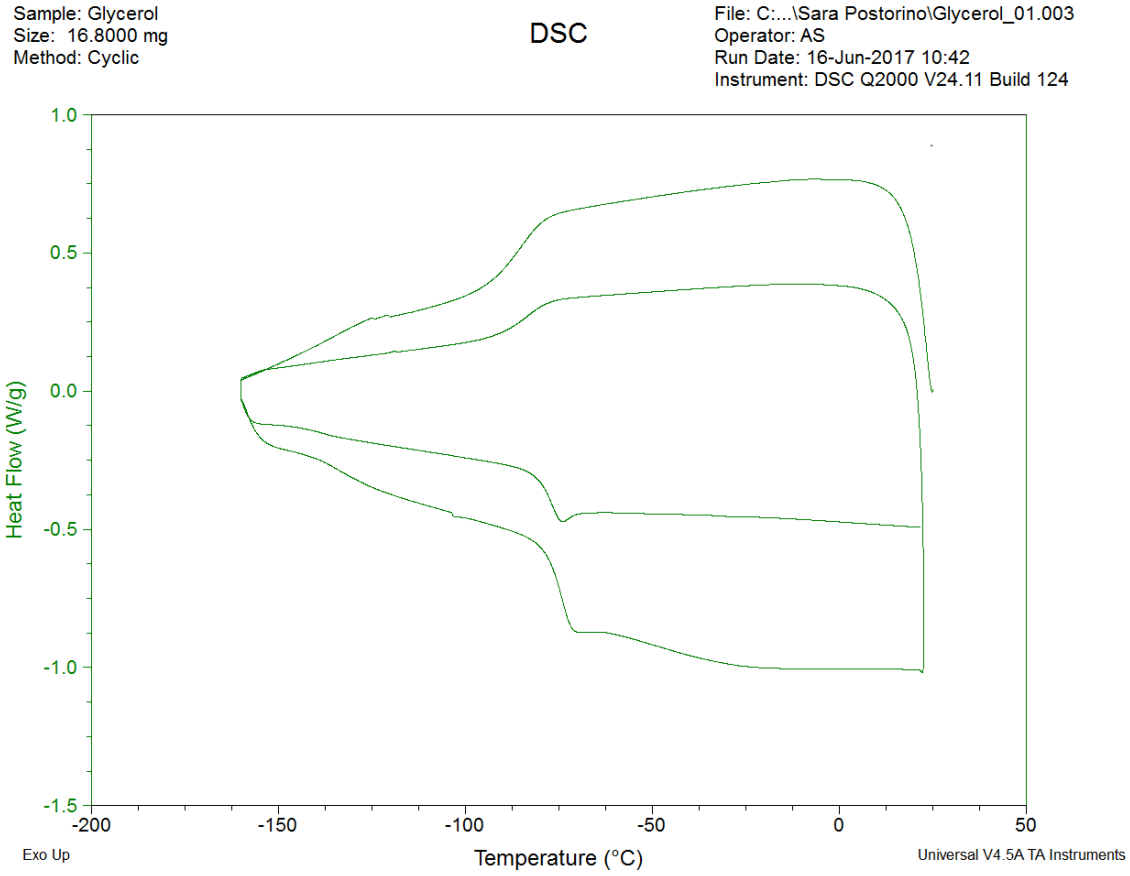


Figure 3.3: Complete Heat flow curve of Glycerol showing different ramp rates: cooling at $20K/min$ and then at $10K/min$; heating at $20K/min$ and $20K/min$

Glycerol is interesting because it shows only the glass transition and the change in specific heat is very big making the transition easy to detect. The viscosity make the manual filling difficult, but this problem can be overcome using a pump.

The analysis of the transition is reported in the following page.

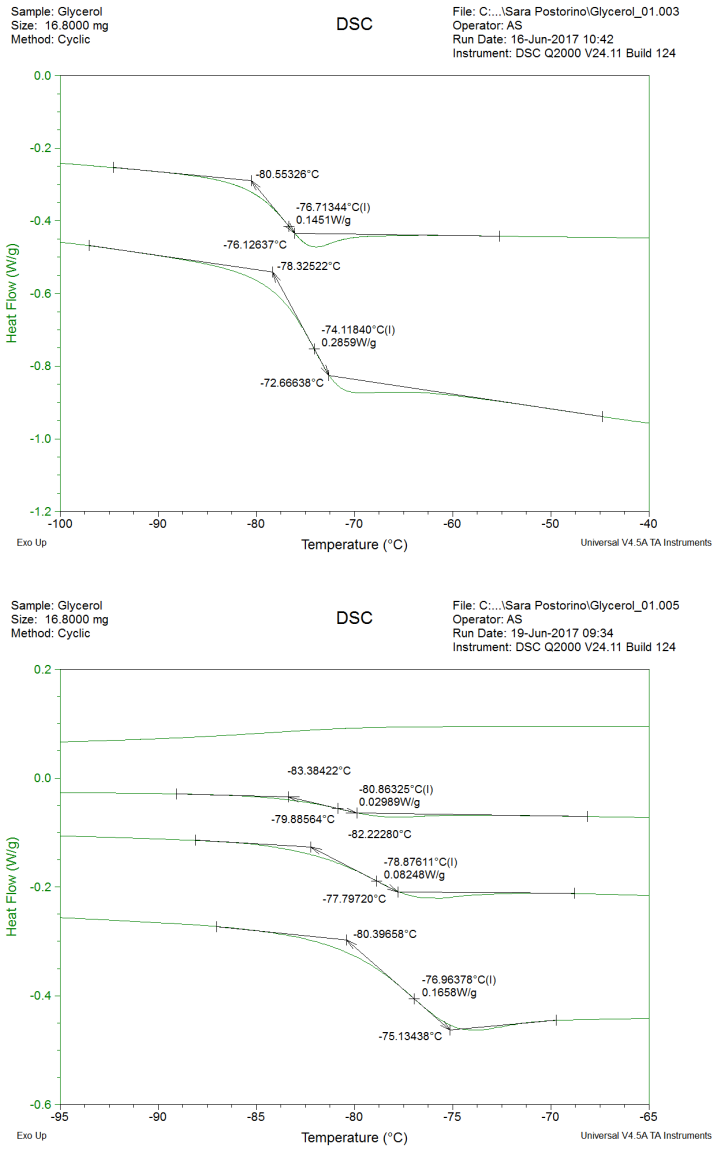


Figure 3.4: Glass transitions at different rates *a*) $10K/min$ and $20K/min$, *b*) $10K/min$,$5K/min$ and $2K/min$

3.3 Buthanol

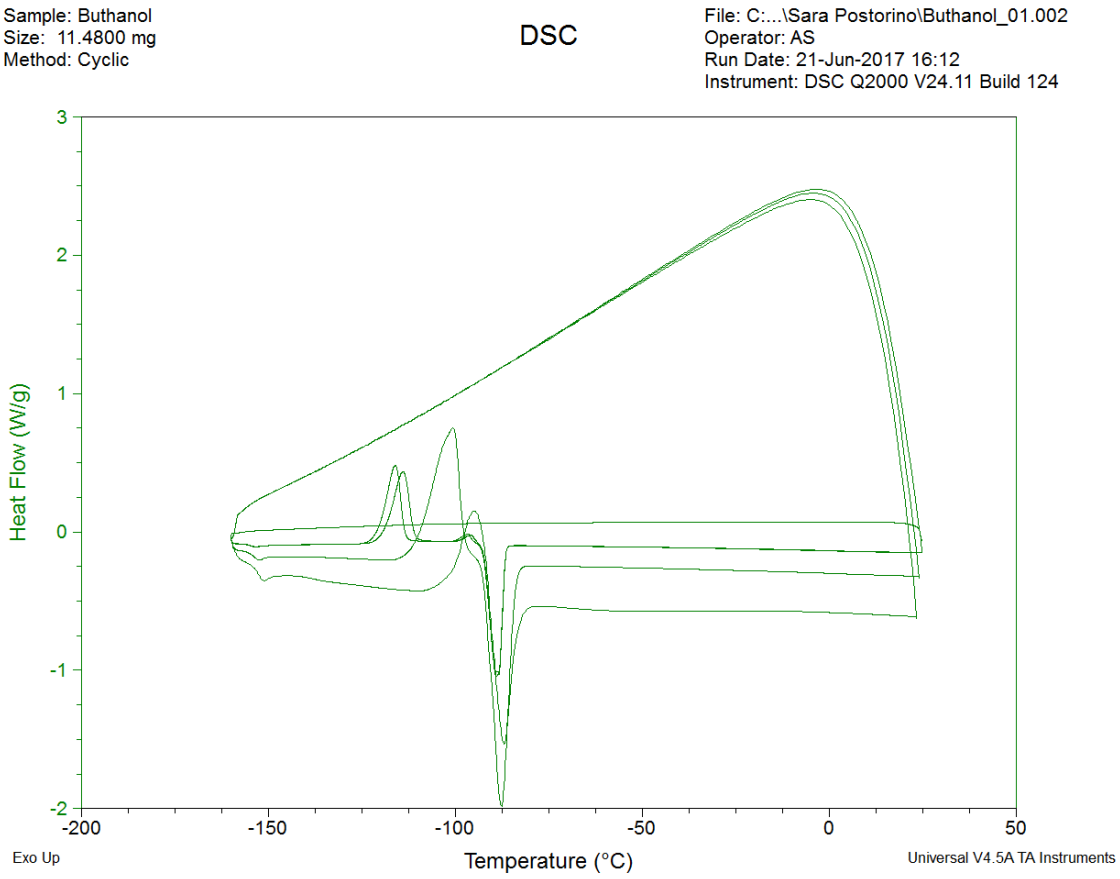


Figure 3.5: Complete Heat flow curve of Buthanol showing different ramp rates: cooling with maximum rate and then at $2K/min$; heating at $10K/min$, $5K/min$ and $2K/min$

At slow rate the three transitions of buthanol are well separated but at higher rates melting and crystallization peaks become really close and this make this glass former not a good candidate.

The analysis of the transitions is reported in the following page.

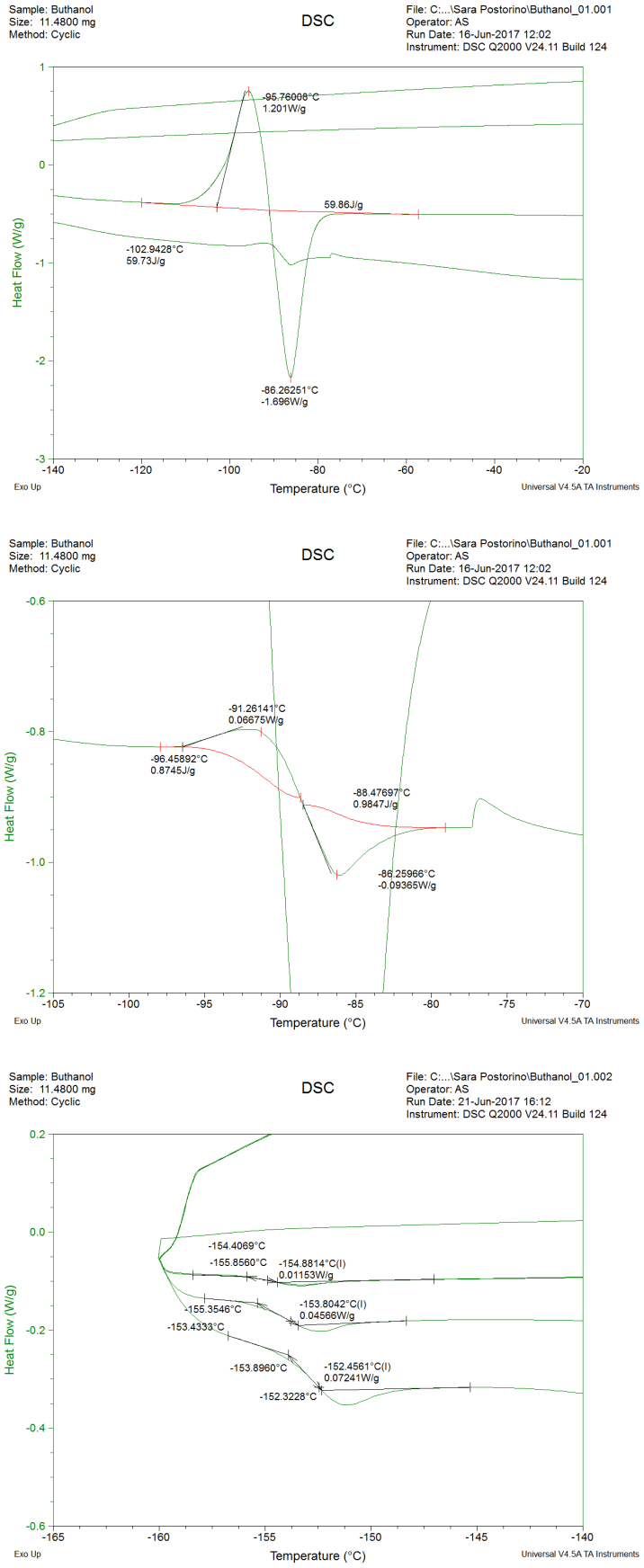


Figure 3.6: a),b) Crystallization and melting peaks at different rates, c) Glass transitions at different rates

3.4 Ethylene Glycol

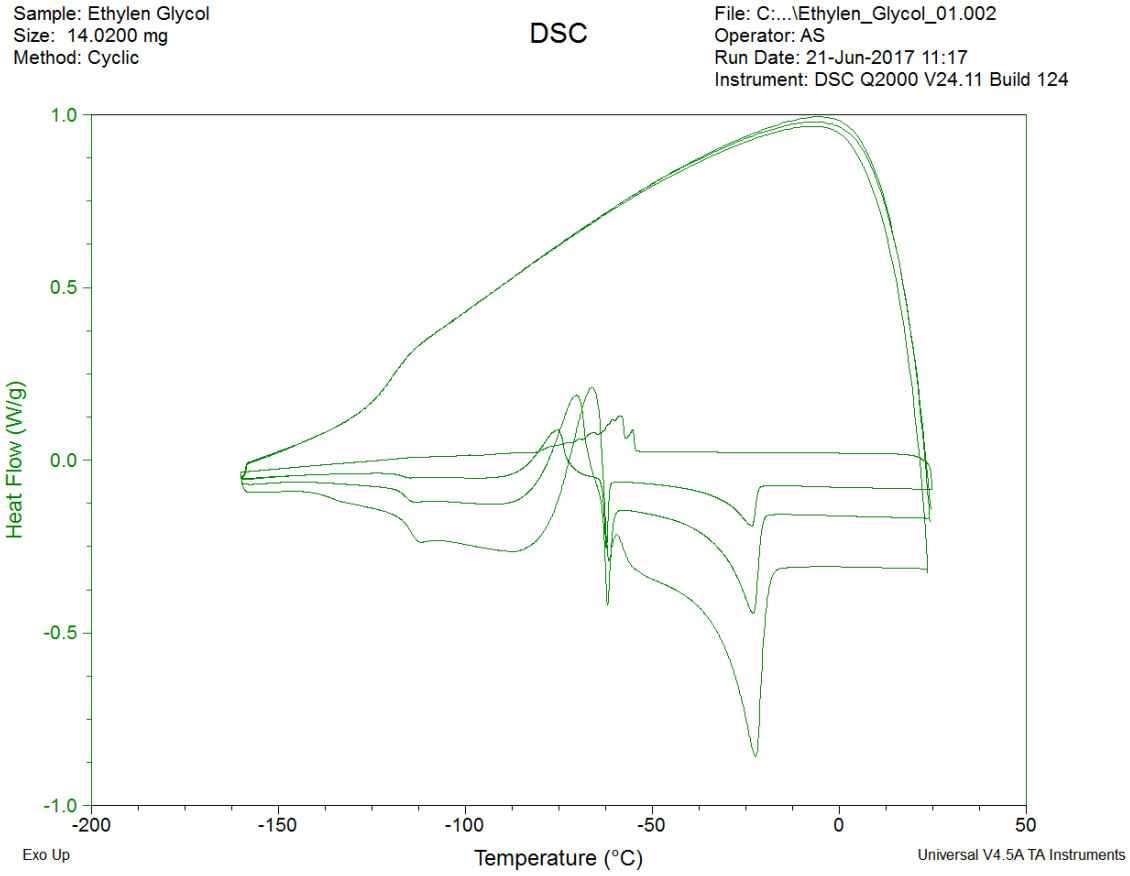


Figure 3.7: Complete Heat flow curve of Ethylen glycol showing different ramp rates: cooling at maximum rate and then at $2K/min$; heating at $10K/min, 5K/min$ and $2K/min$

Ethylen Glycol shows a good glass transition but it becomes very weak incase of slow heating. In addition the other tansitions show a behavior that is not suitable for the calorimetry tests.

The analysis of the transitions is reported in the following page.

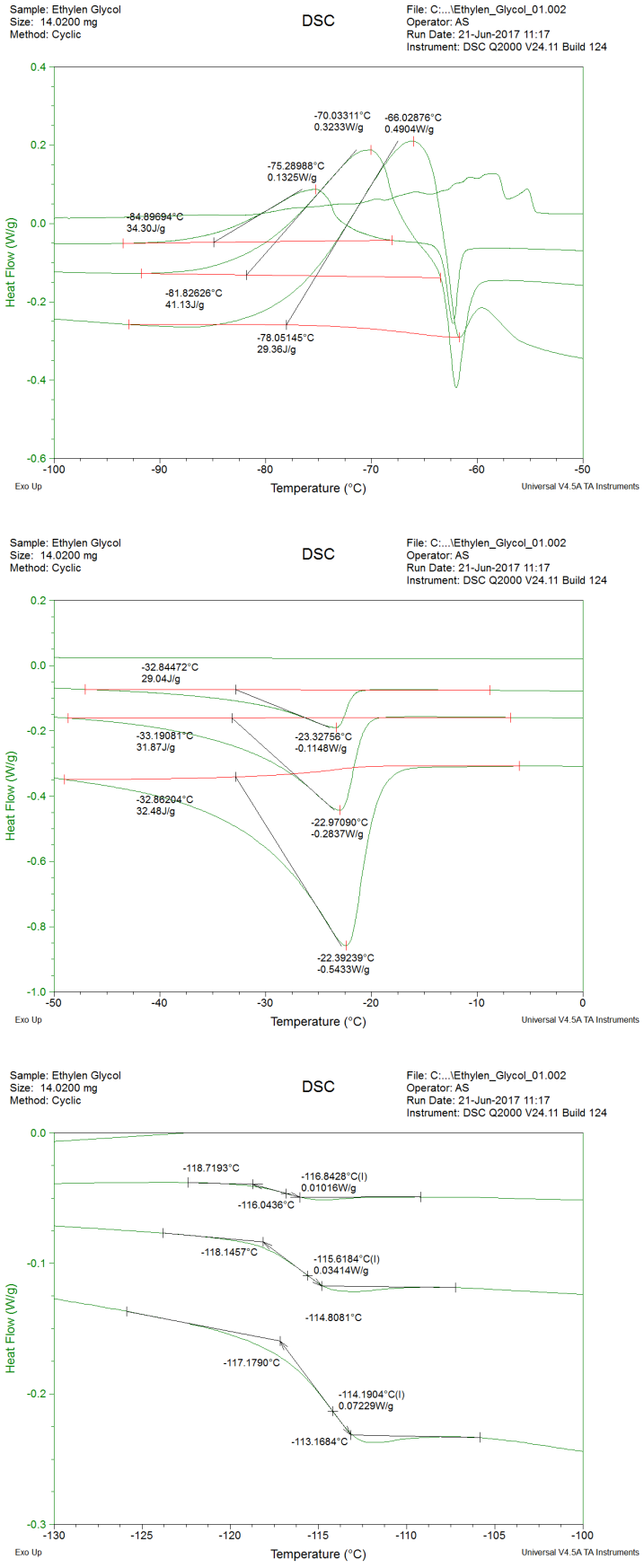


Figure 3.8: a)Crystallization peaks at different rates, b) Melting peaks at different rates, c) Glass transitions at different rates

3.5 Cyclohexanol

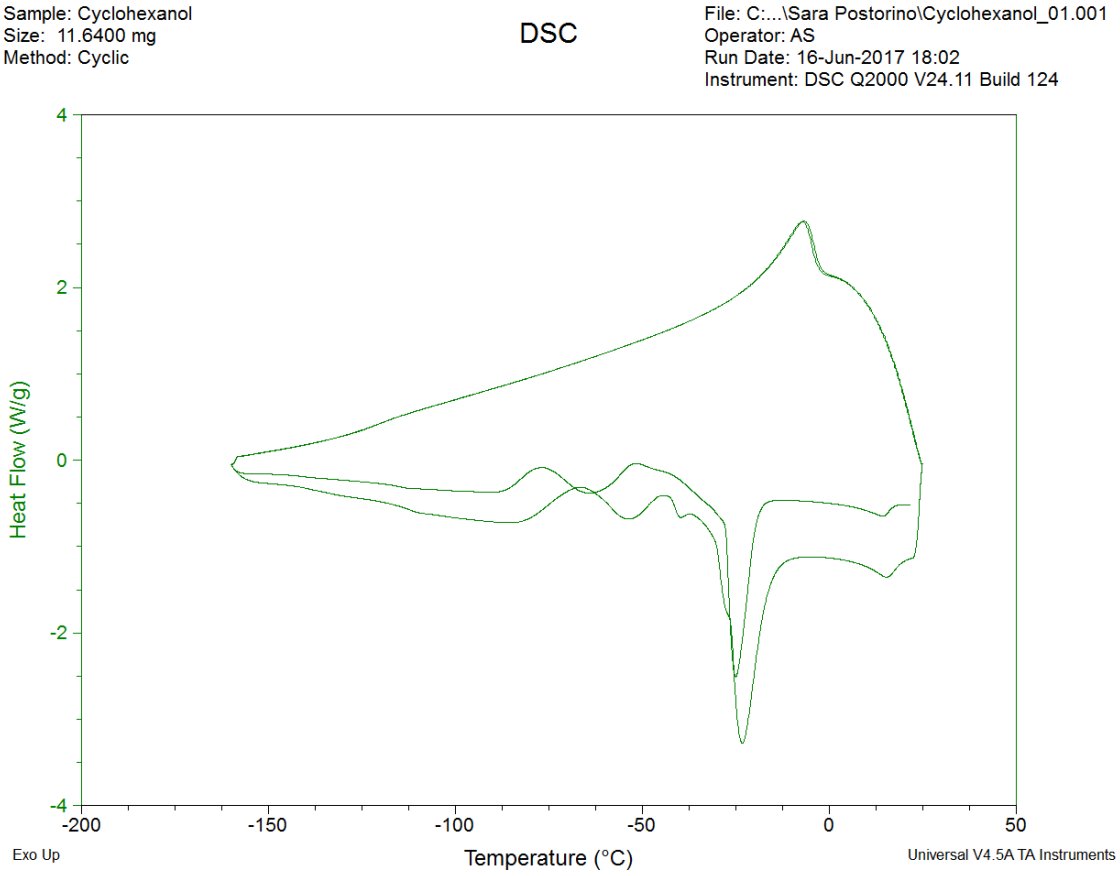


Figure 3.9: Complete Heat flow curve of Toluene showing different ramp rates: cooling at maximum rate; heating at $10K/min$ and $5K/min$

Glass Tansition of cycloheaxanol is very weak and the strange behavior showed by the heat flow curve is due to its plastic crystal phase. For this reason cyclohexanol is not suitable for the tests.

The analysis of the transitions is reported in the following page.

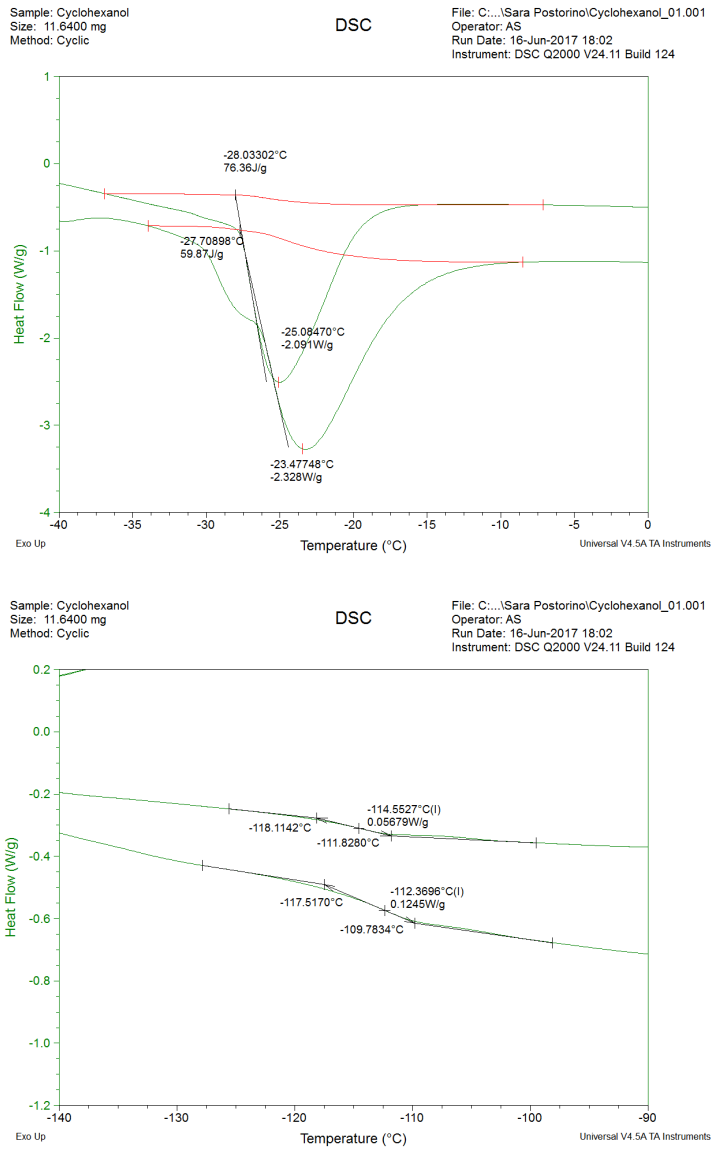


Figure 3.10: a) Melting peaks at different rates, b) Glass transitions at different rates

3.6 Dimethylsulfoxide

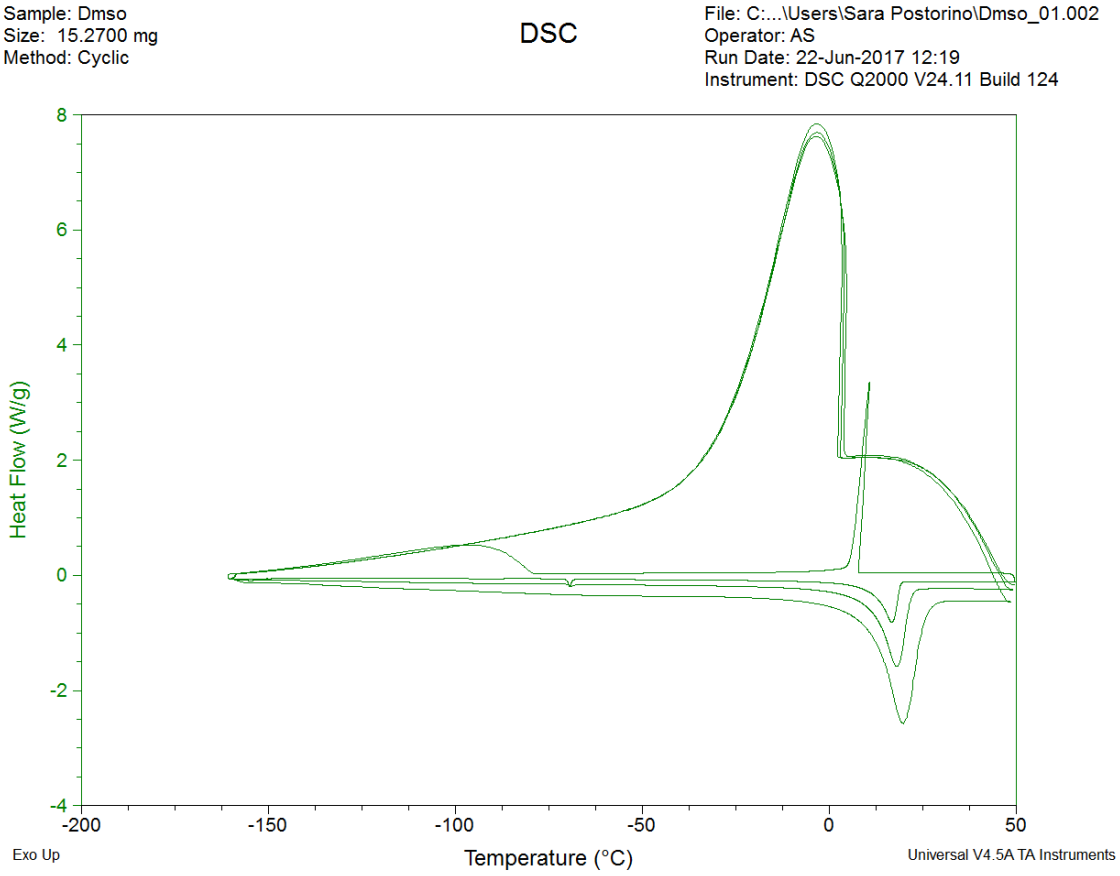


Figure 3.11: Complete Heat flow curve of Dimethylsulfoxide showing different ramp rates: cooling at maximum rate and then at $2K/min$; heating at $10K/min, 5K/min$ and $2K/min$

The behavior of ths glass former is really strange and clearly makes it not suitable for the calorimeter testing. This result can also be caused by the bad quality of the sample because it was not pure dimethylsufoxide but there was a sulfur component.

CHAPTER 4

Linkers

The first important thing to do was the choice of the linkers. They are crucial because they connect the cans to a braket that is in direct contact with the colder CCR walls; $50K$in this case while the calorimeter has to go from $70K$ to $270K$. There are four different linkers, they have been tested in vacuum running with three different rates: $2K/min, 1K/min$ and $0.5K/min$. During the measurements temperatures have been recorded on all the four sensors, the two heaters output and the current given to the heaters. The decision of recoding both current and heating power was made because the power signal presents steps due to the bad resolution and the data is given in percentage rather than in a precise unit, while the current can be measured with more accuracy and then the power can be calculated knowing the resistance of the heaters used. The resistance in our case is of 73.1Ω, but nothing is known about how much it changes its value in temperature. If it will be decided in the future to use the current signal to detect transitions (more accurate), it would be necessary to have a curve of resistance in order to get accurate data for the heating powers. The difference between the two cans for temperature, heating power and current has been calculated, because all of them could be indicator for a termodynamic transition; the other purpose of this measure is also to see which one is the best.
To check the homogeneity of the system and the heat conduction via the linkers also the temperature gradient along the can has been calculated.
To make the measurements the sample used is Toluene $(2.4g)$.

Results confirm that the best insulation can be achieved using the G-10 linkers as expected looking at thermal conductivities shown in the table below:

Linker	Thermal Conductivity $(W/m \cdot K)$
Honeycomb*	**
Aluminium	~ 205
Stainless Steel	~ 16
G-10 resin	~ 0.29

Table 4.1: * The honeycomb Linkers are made of Stainless Steel but the honeycomb pattern should lower thermal conduction. Due to its particular sturcture thermal conductivity for this linker is not available

In none of these measurements there are visible glass transitions but there are melting and crystallization peaks.

4.1 Honeycomb Linkers

4.1.1 All temperatures

It is already clear from these graph that there is a big gradient that will be analysed in the following pages. The ramps are linear and the temperatures of the ends with the heaters matches exactly in all the graphs while while the other two ends have a visible difference. This tells us that the environment is not homogeneous.

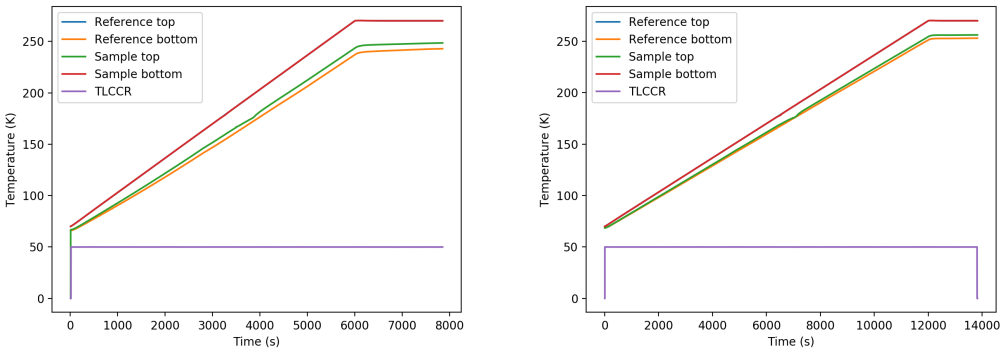


Figure 4.1: Honeycomb Linkers $2K/min$ and $1K/min$

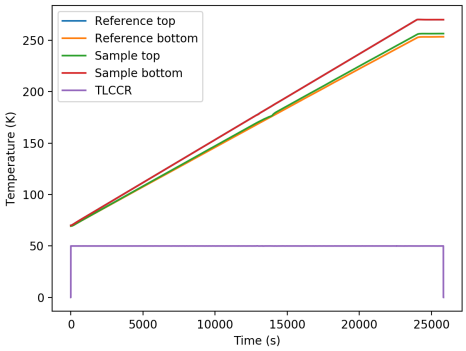


Figure 4.2: Honeycomb Linkers $0.5K/min$

4.1.2 Gradient

From this data emerges that the gradient is bigger if one goes faster (about $30K$) but the minimum we can reach is of about $17K$ on the reference can and $14K$ on the sample can. This high gradient has as consequence a partial transition of the sample and this is confirmed by the presence of peaks in the sample gradient curve. A partial transition can be the cause of not seeing any glass transition.

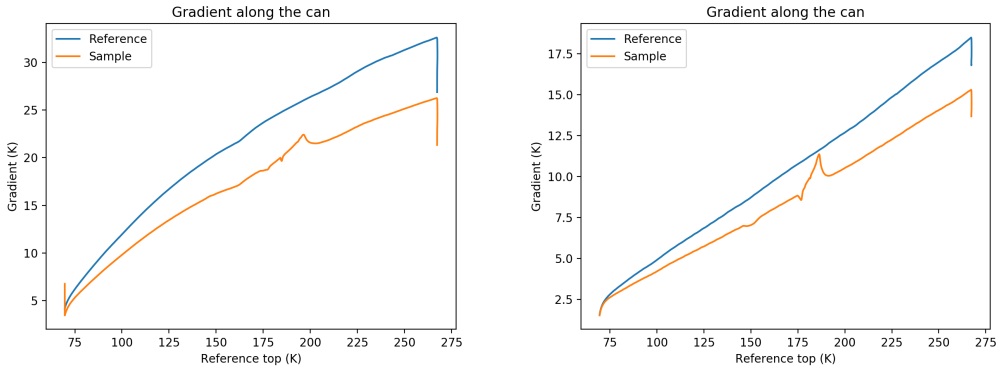


Figure 4.3: Honeycomb Linkers $2K/min$ and $1K/min$

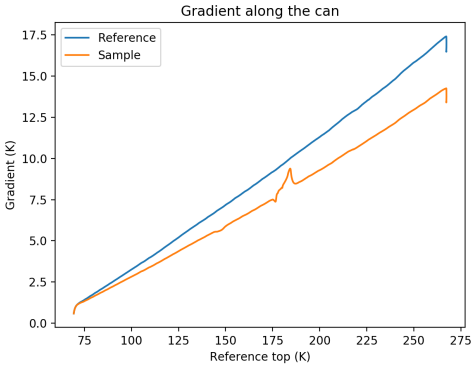


Figure 4.4: Honeycomb Linkers $0.5K/min$

4.1.3 Heaters and currents

The heater output confirms the fact that the environment is not homogeneous because the output on the reference is always higher. Similar conclusions can be made looking at currents.
The steps present in the heater output signal allow us to see only the melting peak while in the current signal, smoother than the other, there is also a small crystallization peak.

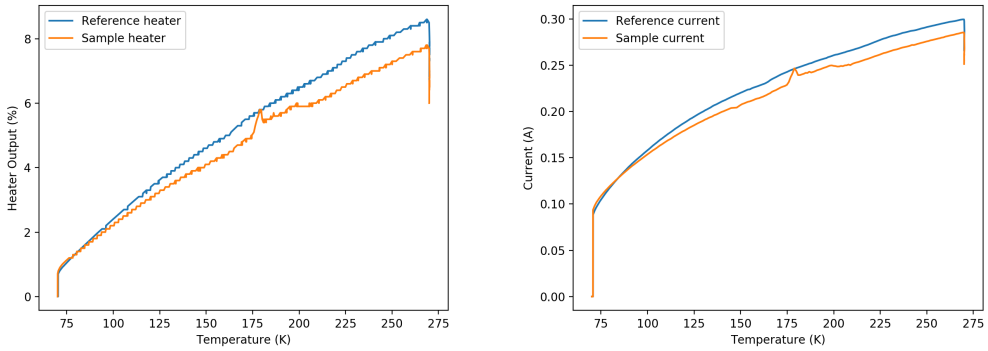


Figure 4.5: Honeycomb Linkers $2K/min$ heaters and currents

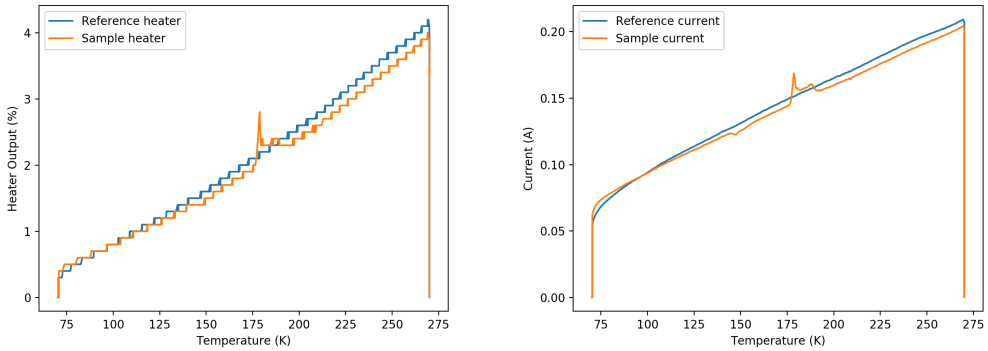


Figure 4.6: Honeycomb Linkers $1K/min$ heaters and currents

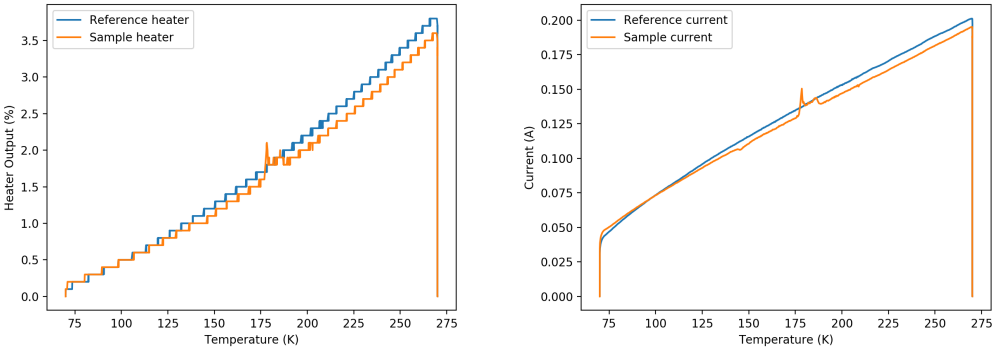
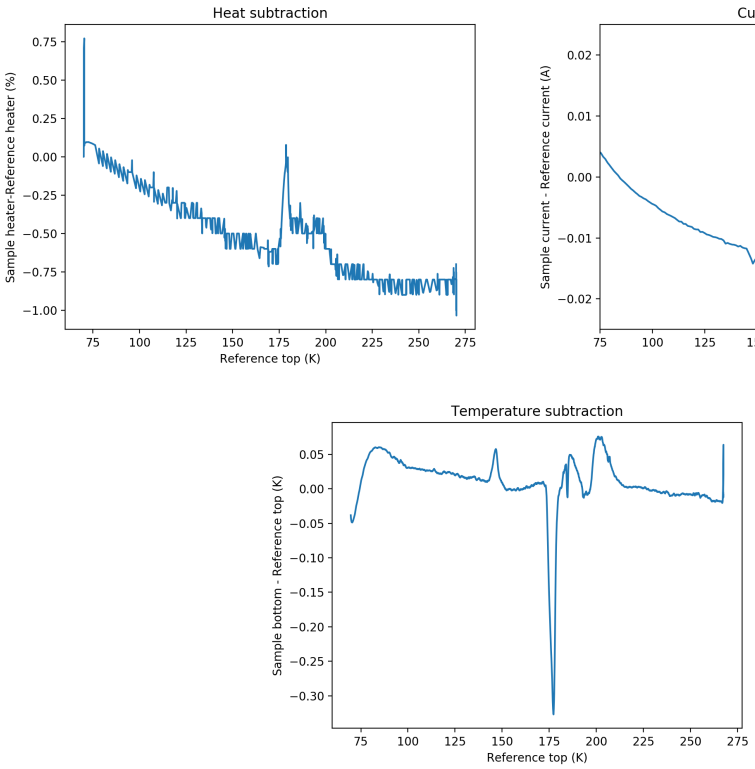


Figure 4.7: Honeycomb Linkers $0.5K/min$ heaters and currents

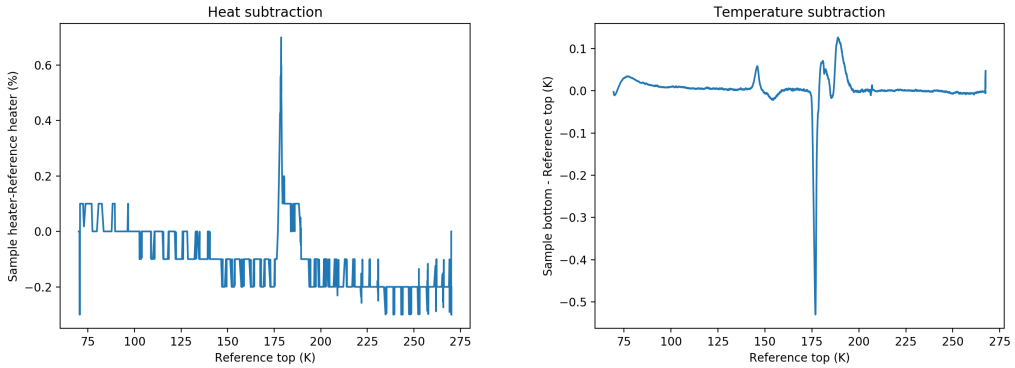
4.1.4 Temperature, current and heating power subtractions

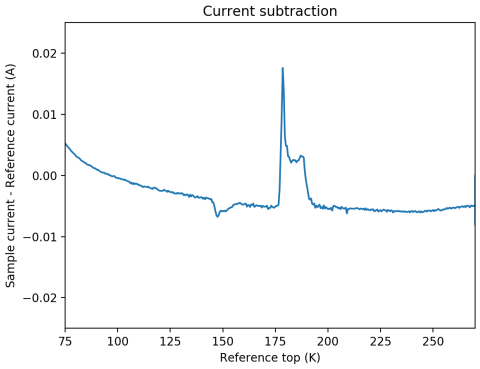
The melting peak is present in every subtraction, while the crystallization peak does not show in the heat subtraction. Temperature always shows the best baseline, but with lower rates also the current presents a good baseline. The heater output seems to become worse as one goes slower, but it is a problem of peak intensity compared to noise.

- $\underline{2K/min}$

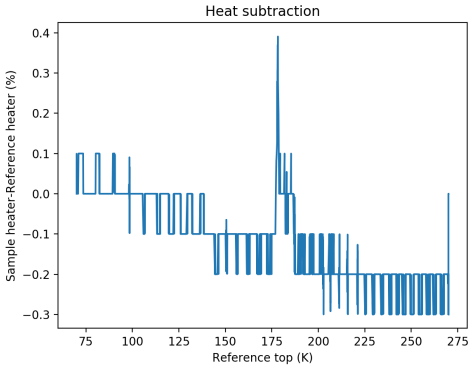


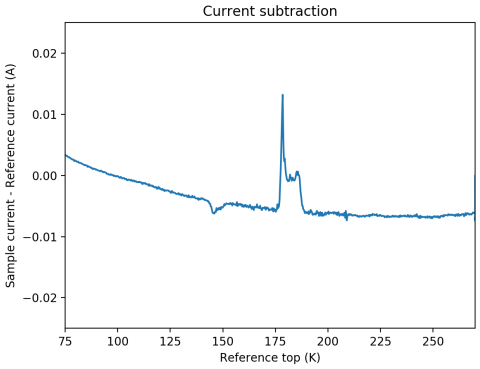
- $\underline{1K/min}$

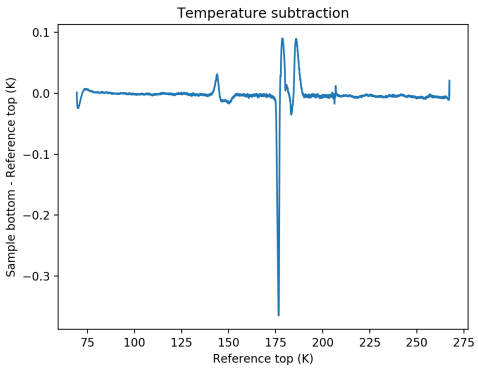




- $\underline{0.5K/min}$







4.2 Aluminium Linkers

4.2.1 All temperatures

With the aluminium linkers there is a bigger difference between the two ends of the can; it may be because of high thermal conductivity of the linkers keeping the end without the heater in contact with the walls of CCR and causing a leak of heat.
The system now, despite the huge gradient is symmetric, there is no big difference between the bottom of the reference and the top of the sample.

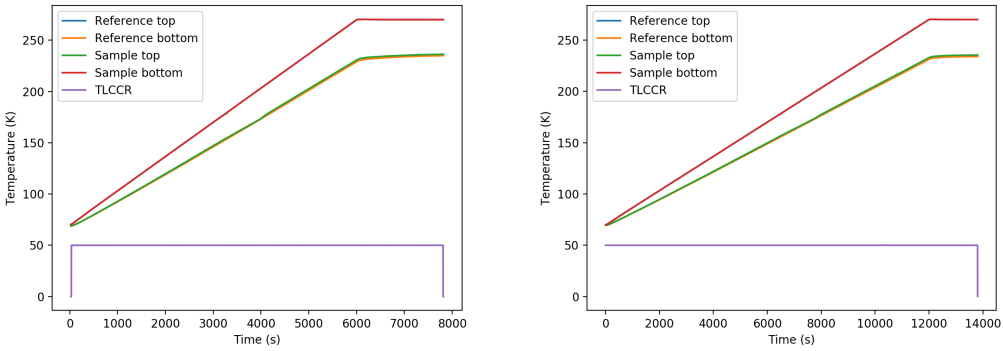


Figure 4.8: Aluminium Linkers $2K/min$ and $1K/min$

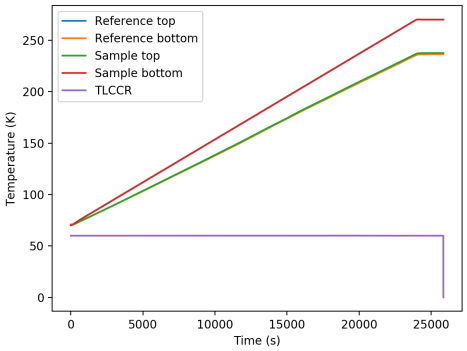


Figure 4.9: Aluminium Linkers $0.5K/min$

4.2.2 Gradient

Using Aluminium the gradient is between $35K$ and $40K$. The two curves are closer due to the fact observed before that there is more symmetry in temperature.

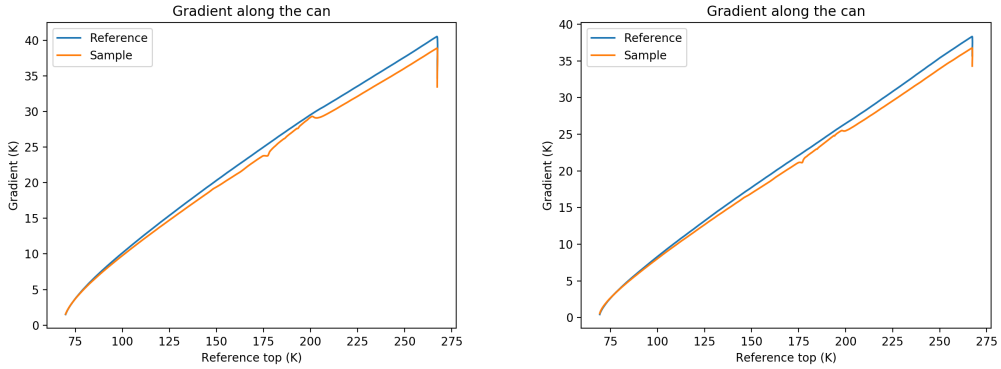


Figure 4.10: Aluminium Linkers $2K/min$ and $1K/min$

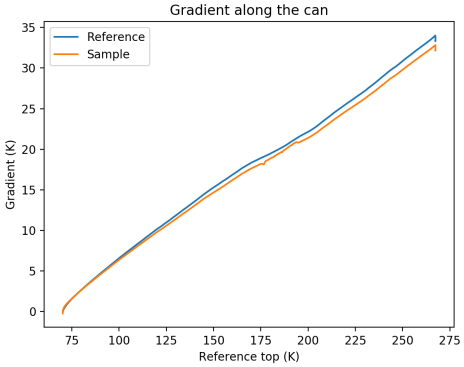


Figure 4.11: Aluminium Linkers $0.5K/min$

4.2.3 Heaters and currents

Also in this case there is a difference between the two output but this time the higher output is on the sample can. This is reasonable because of the extra mass of the sample.
As shown by the following graphs melting and crystallization peaks are visible but smaller than in the previous case.

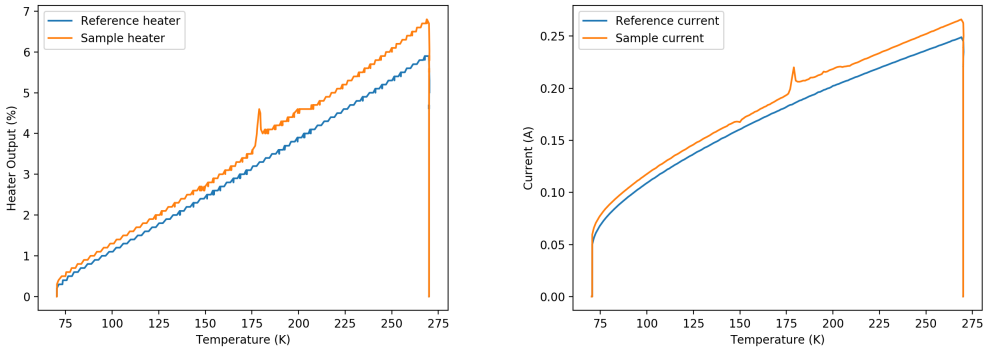


Figure 4.12: Aluminium Linkers $2K/min$ heaters and currents

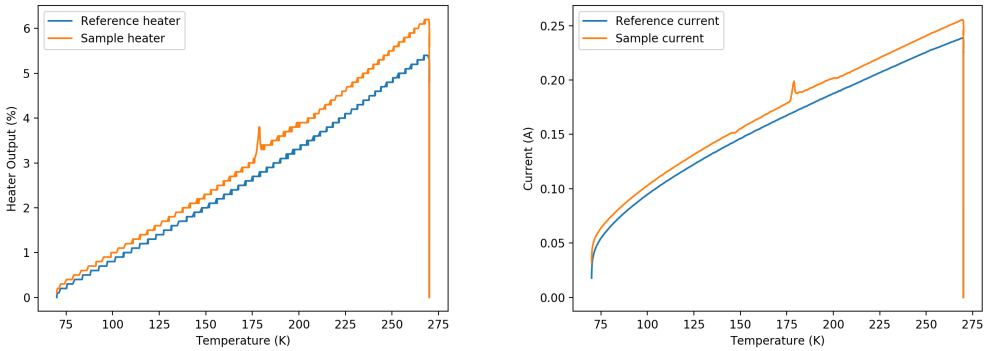


Figure 4.13: Aluminium Linkers $1K/min$ heaters and currents

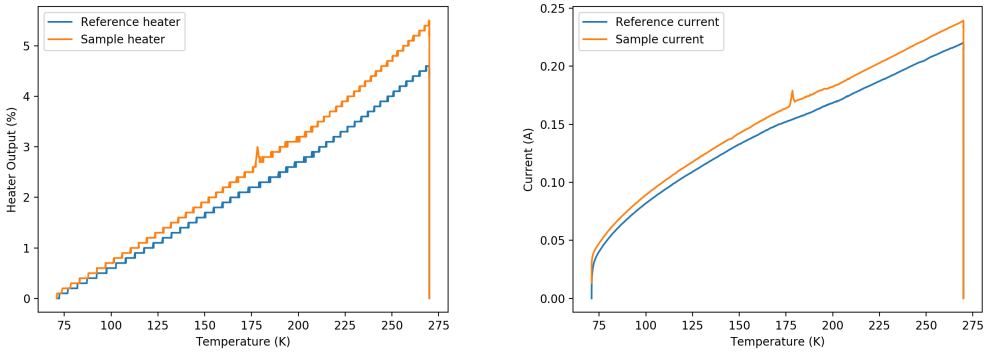


Figure 4.14: Aluminium Linkers $0.5K/min$ heaters and currents

4.2.4 Temperature, current and heating power subtractions

The melting peak is present in every subtraction, while the crystallization peak does not show in the heat subtraction and it is not clearly visible in all the current graphs where it can be confused with a step . The baseline is flat in both current and temperature graphs. There is the same issue encountered before regarding the heater output signal as one goes slower.

- $\underline{2K/min}$

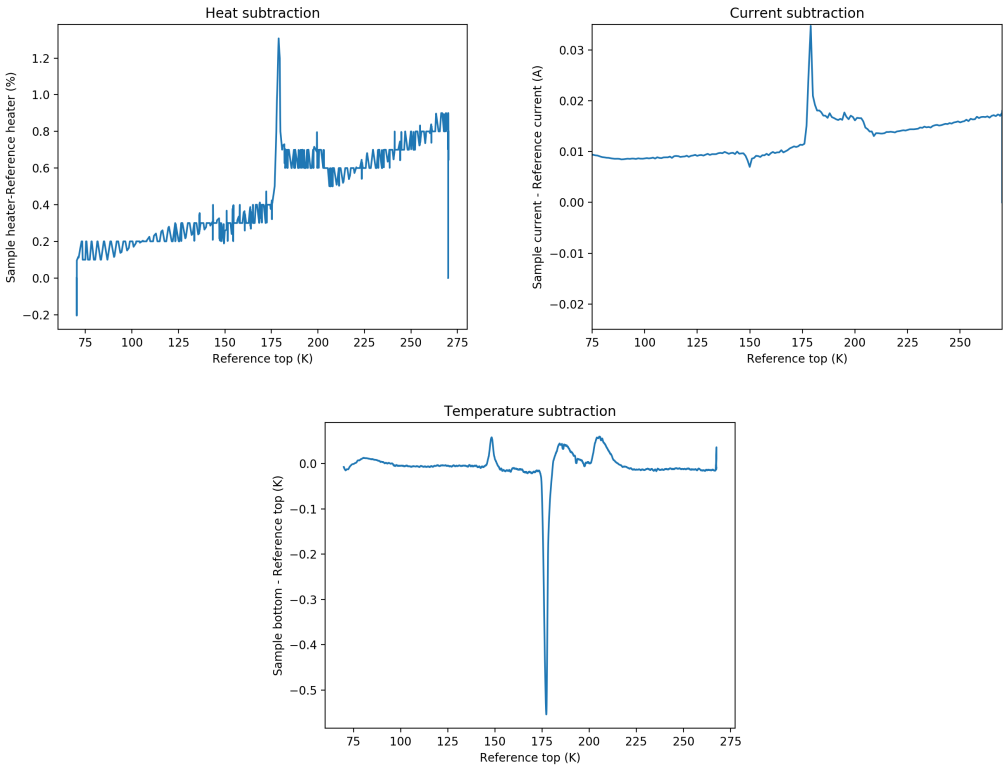
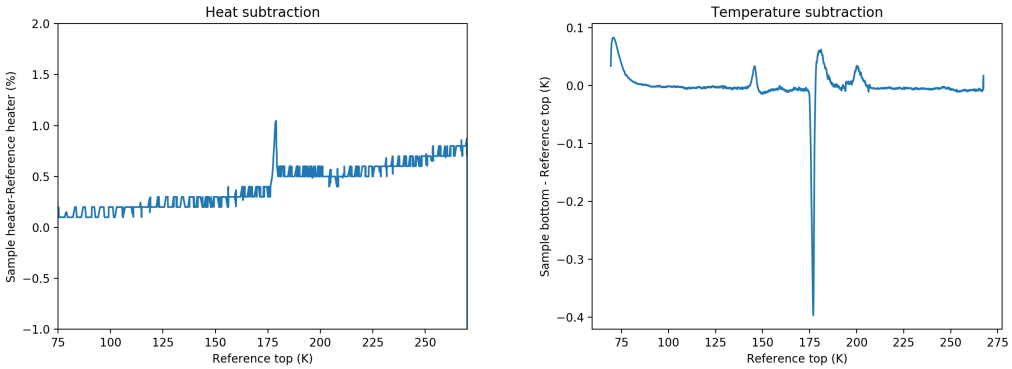
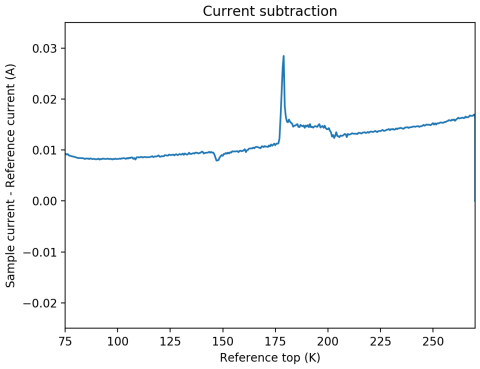


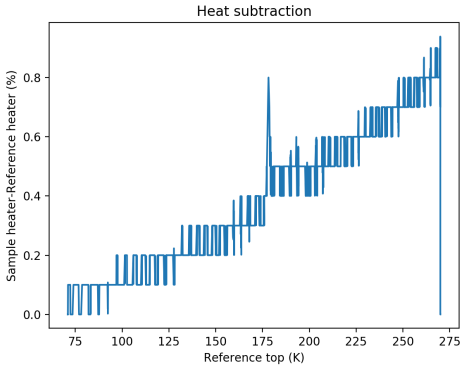
Figure 4.15

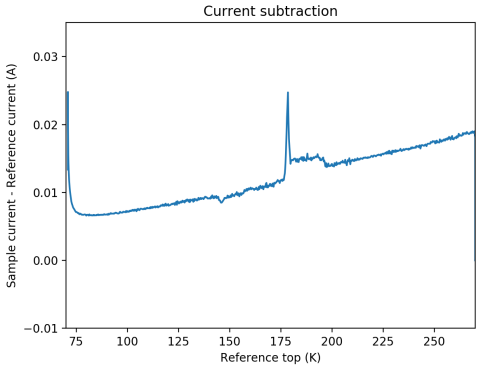
- $\underline{1K/min}$

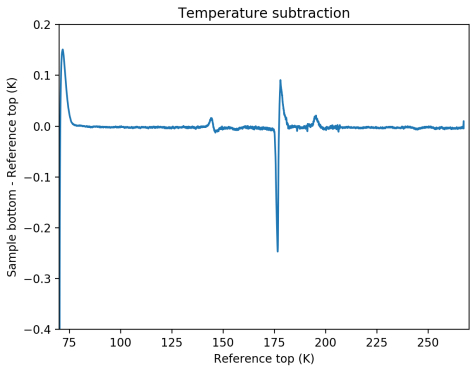



Current subtraction
Sample current - Reference current (A)
Reference top (K)

- $0.5K/min$


Heat subtraction
Sample heater-Reference heater (%)
Reference top (K)


Current subtraction
Sample current - Reference current (A)
Reference top (K)


Temperature subtraction
Sample bottom - Reference top (K)
Reference top (K)

4.3 Stainless Steel Linkers

4.3.1 All temperatures

Using Stainless Steel linkers results look like the ones of the honeycomb case showing that the pattern makes small changes.

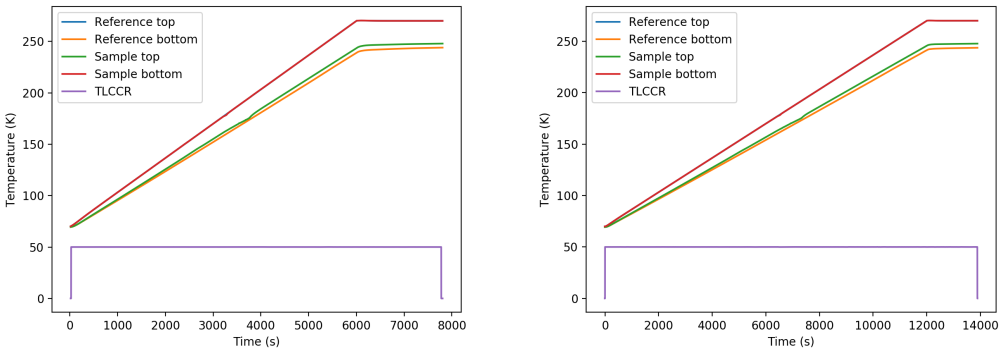


Figure 4.16: Stainless Steel Linkers $2K/min$ and $1K/min$

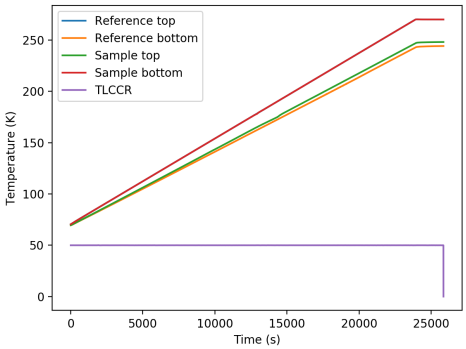


Figure 4.17: Stainless Steel Linkers $0.5K/min$

4.3.2 Gradient

Gradients are comparable with the honeycomb ones but slightly bigger. Their range is between about $28K$ and $31K$ for the reference and in this case there is no possibility to go lower than $20K$ for the sample can. The difference between reference and sample is back now that the linkers are less conductive than aluminium ones. This confirm the hypotesis of a non homogeneous environment.

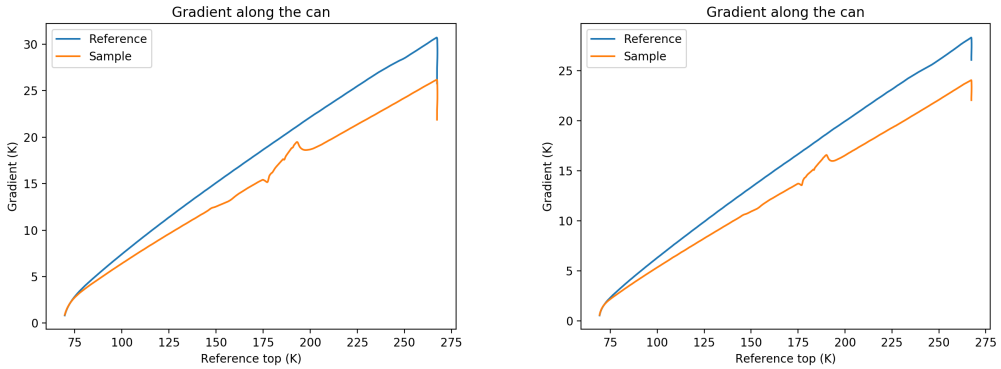


Figure 4.18: Stainless Steel Linkers $2K/min$ and $1K/min$

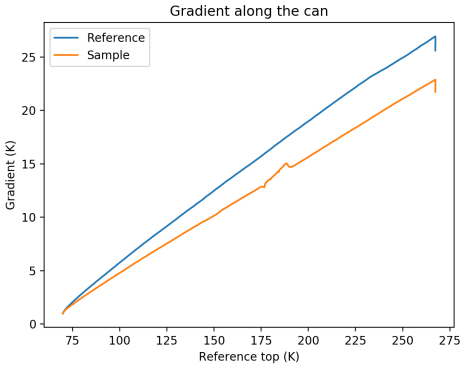


Figure 4.19: Stainless Steel Linkers $0.5K/min$

4.3.3 Heaters and currents

The behavior of current and heater output is the one expexted for the insturment. The two signals are similar and there is only a small difference due to the extra mass, while big differences appear only in correspondence of the transitions.

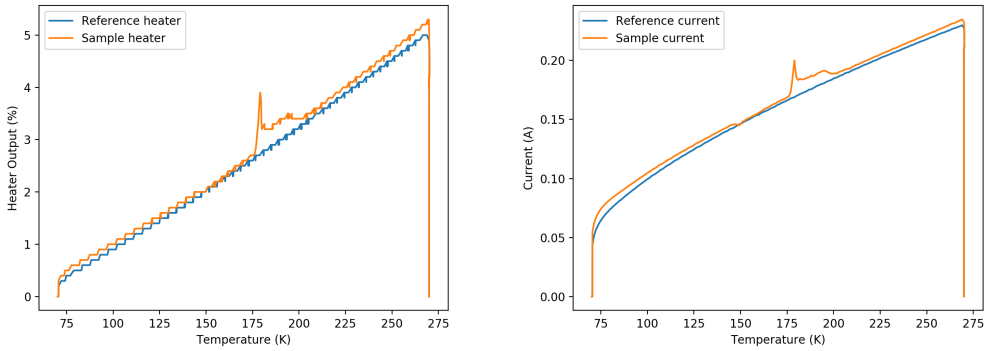


Figure 4.20: Stainless Steel Linkers $2K/min$ heaters and currents

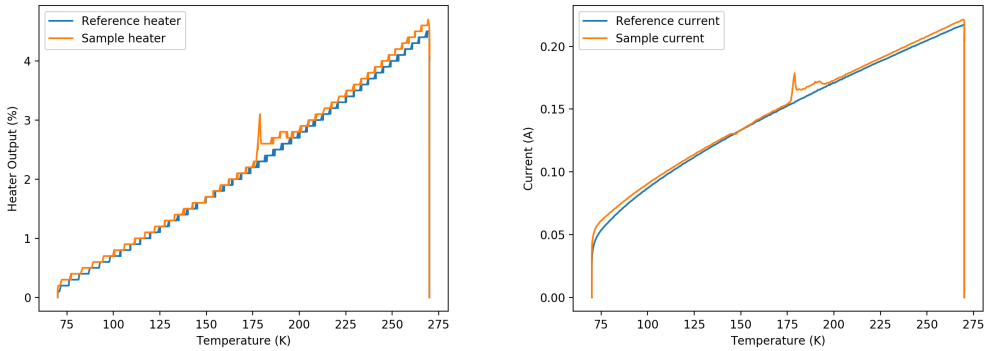


Figure 4.21: Stainless Steel Linkers $1K/min$ heaters and currents

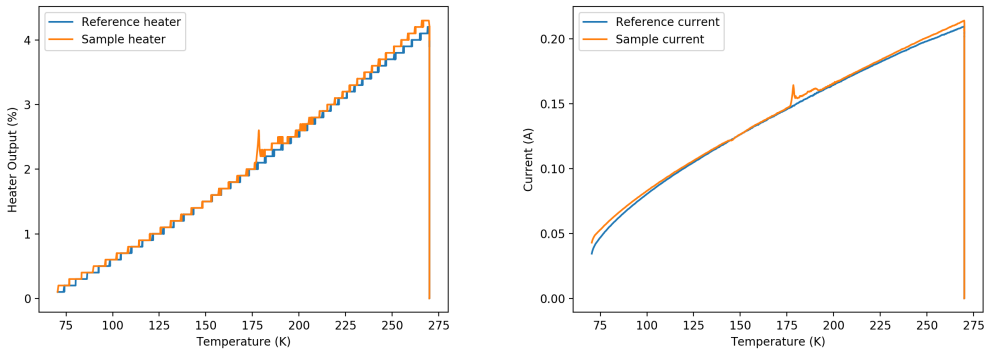


Figure 4.22: Stainless Steel Linkers $0.5K/min$ heaters and currents

4.3.4 Temperature, current and heating power subtractions

With these linkers peaks are smaller so the crystallization peak in the current signal is not very clear and it can be confused with a step like in the aluminium case. On the heater output there are the same problems ecountered before. The best baseline is still in temerature graphs, but the current one is still good enough for thermal analysis.

- $\underline{2K/min}$

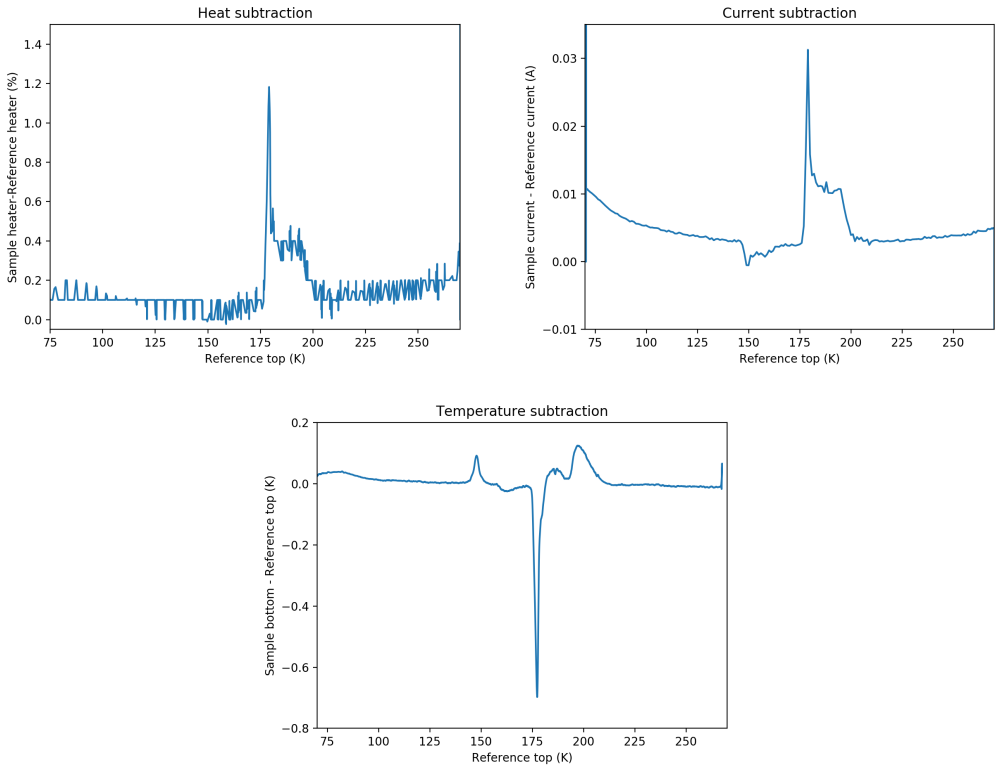
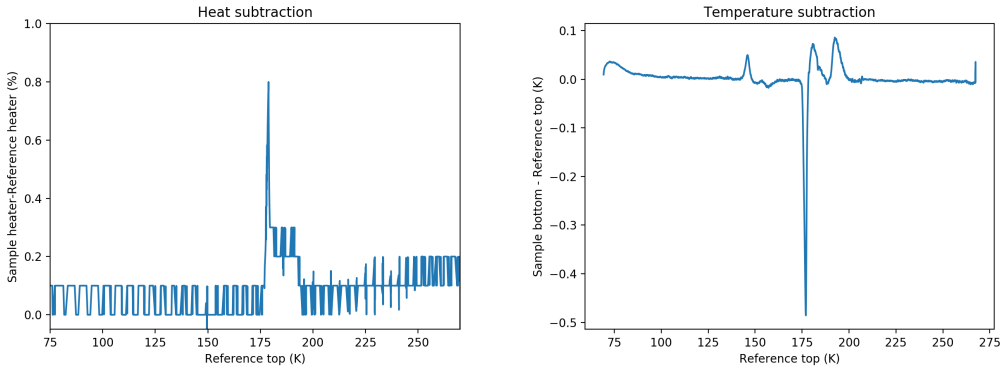
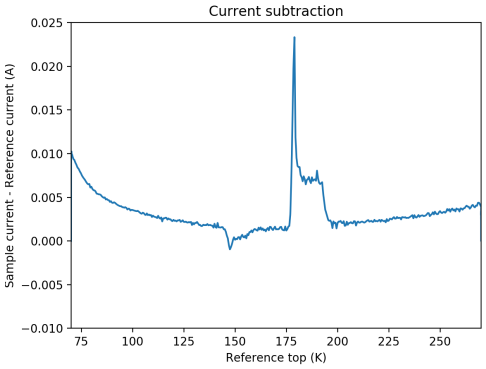


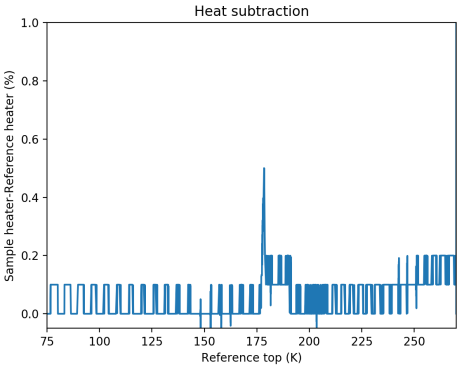
Figure 4.23

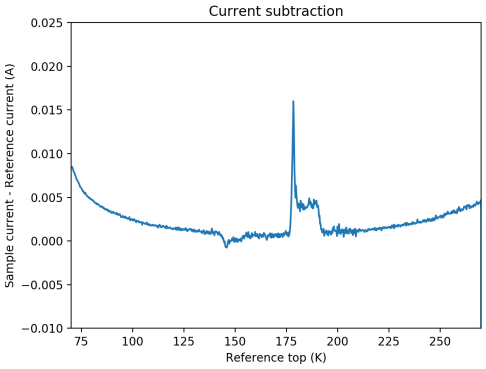
- $\underline{1K/min}$

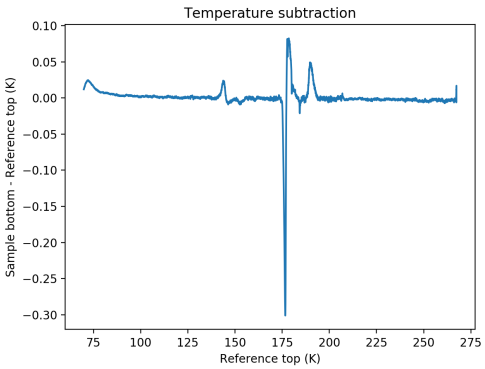




- $\underline{0.5K/min}$







4.4 G-10 Linkers

4.4.1 All temperatures

The G-10 resin gave the best results and it is visible in the graphs below where the lines are a lot closer than in the other cases. This was expected because of its very low conductivity and confirms the idea that one of the causes of the big gradient is the conduction between the cans and the bracket.
A gradient is still visible and this will be analyzed in the next pages.

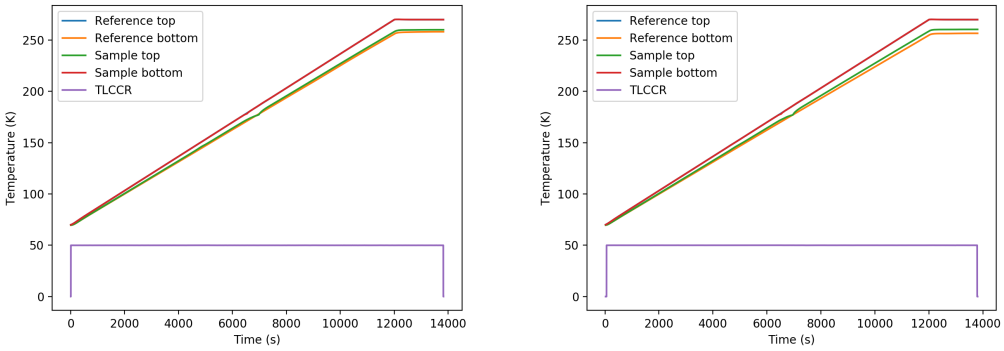


Figure 4.24: G-10 Linkers $2K/min$ and $1K/min$

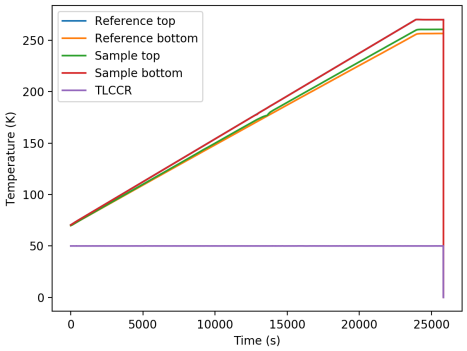


Figure 4.25: G-10 Linkers $0.5K/min$

4.4.2 Gradient

There is a gradient of about $15K$ in this case, the half of the previous ones. As seen before there is a difference between reference and sample. Peaks are present in the sample signal indicating a partial transition inside the can.

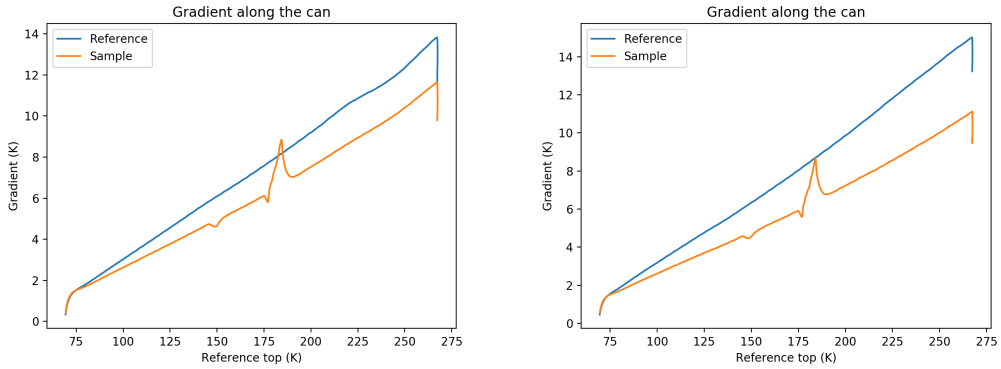


Figure 4.26: G-10 Linkers $2K/min$ and $1K/min$

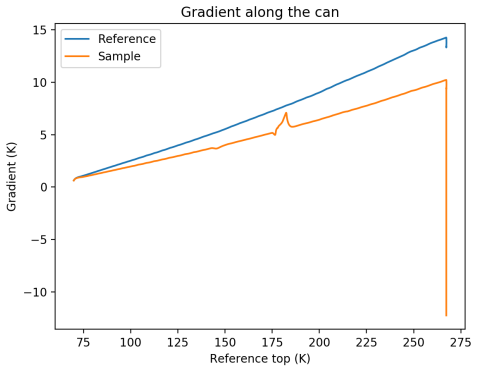


Figure 4.27: G-10 Linkers $0.5K/min$

4.4.3 Heaters and currents

The situation looks like the previous one with good match between the two sources that look similar apart form the transition.

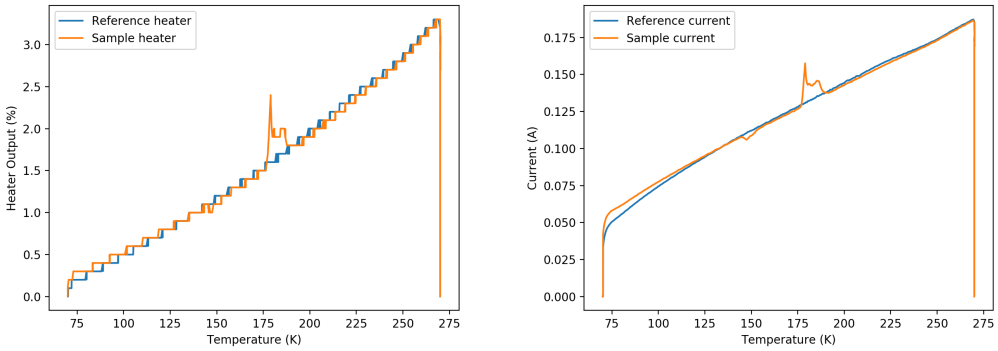


Figure 4.28: G-10 Linkers $2K/min$ heaters and currents

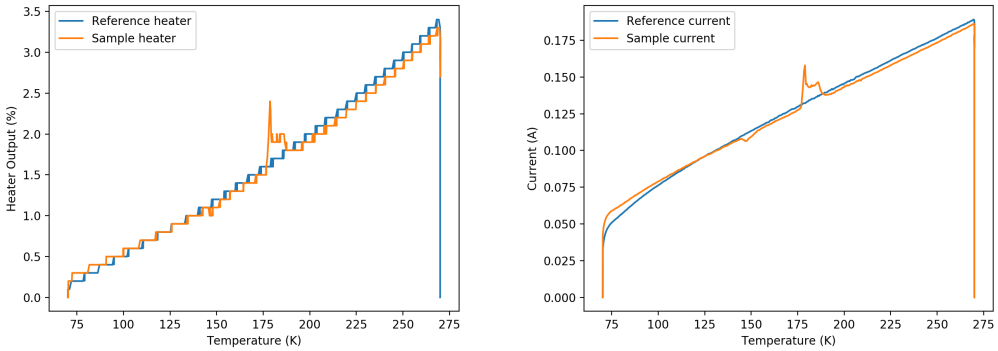


Figure 4.29: G-10 Linkers $1K/min$ heaters and currents

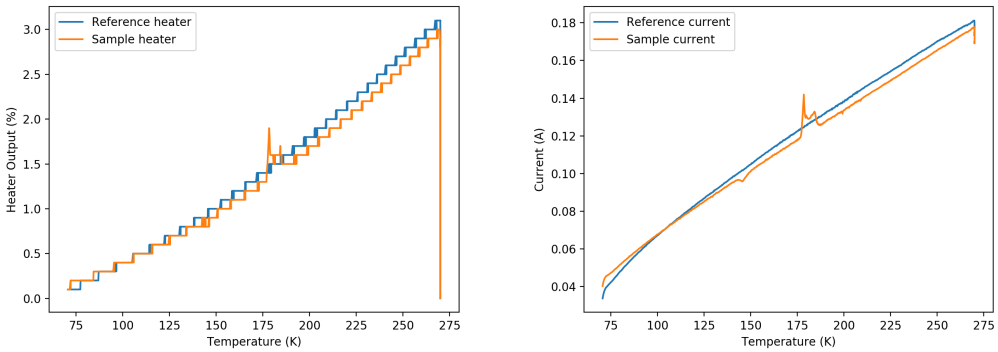
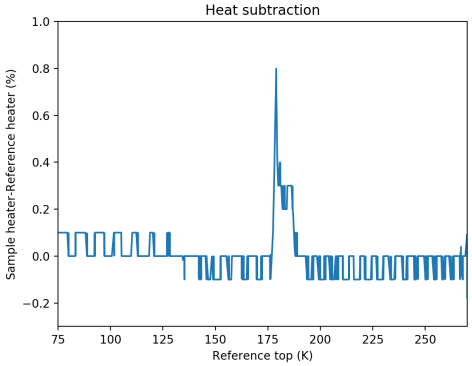


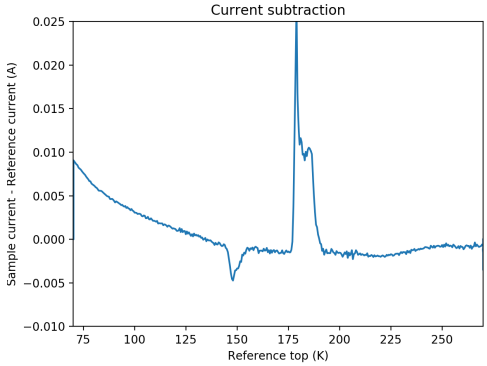
Figure 4.30: G-10 Linkers $0.5K/min$ heaters and currents

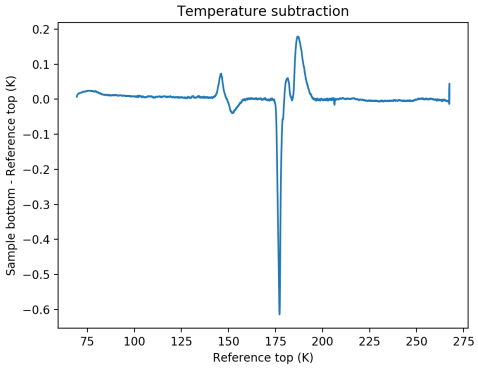
4.4.4 Temperature, current and heating power subtractions

The difference with the other linkers is visible making subtractions: peaks become more intense and more sharp. Crystallization is always clearly visible in current subtraction, but not in the heater output where it is hidden by noise.

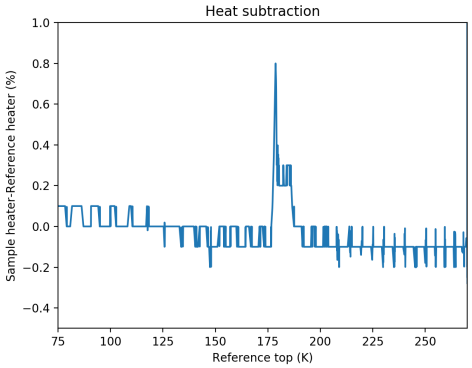
- $\underline{2K/min}$

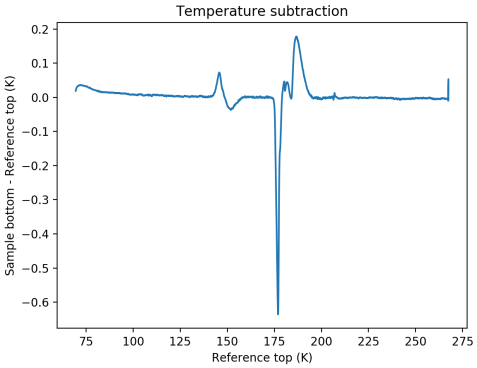


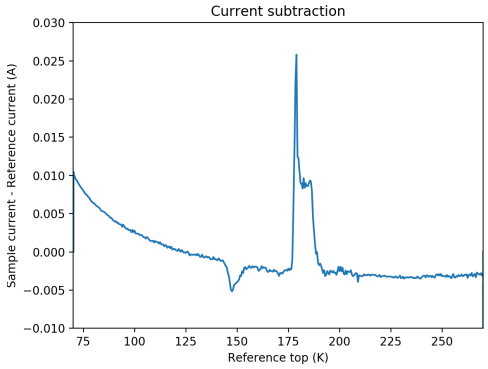




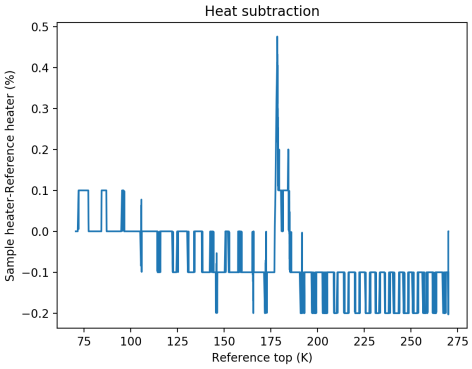
- $\underline{1K/min}$

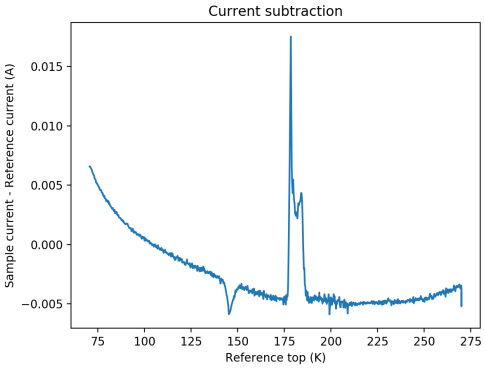


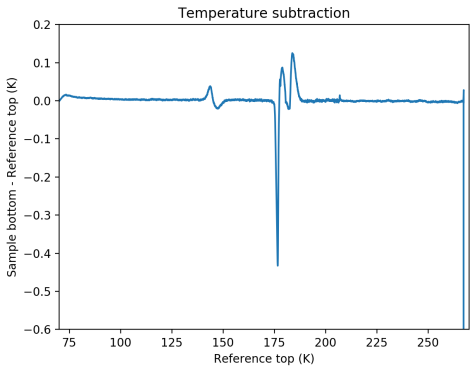



Current subtraction
Sample current - Reference current (A)
Reference top (K)

- $0.5K/min$


Heat subtraction
Sample heater-Reference heater (%)
Reference top (K)


Current subtraction
Sample current - Reference current (A)
Reference top (K)


Temperature subtraction
Sample bottom - Reference top (K)
Reference top (K)

4.5 Comments on results

As expected the best candidate for a futire use is the G-10 resin linker because it gives the lower thermal conduction and this results in a smaller gradient and good melting and crystallization peak.
During these tests it also emerged that there are other issues apart from the thermal conduction in the linkers:

- There is still a very big gradient on the can that was not expected. This can be caused by two factors: firstly the conductivity of the can is not high as expected becauese of the presence of silicon, secondly the cans are not thermally insulated from the CCR although the are in vacuum. The latter is clear from the fact that the cans see the non homogeneity of the environment.
- The environment is not homogeneous and this affects the measurements

In the next chapters there are tests made to confirm these observations and find a possible solution.

CHAPTER 5

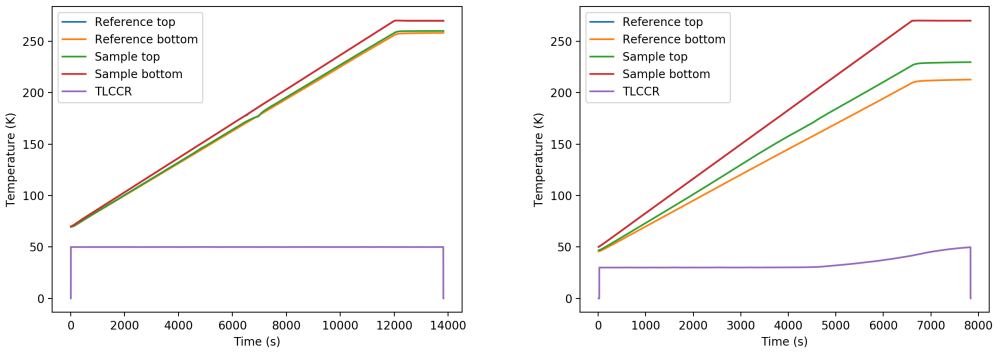
Environment

The environment inside the CCR is not homogeneous; the folowing tests have the aim of finding the best way to perform the experiments only modifing the environment.
All the following measurements are made using G-10 linkers and the same Toluene sample.

5.1 Helium vs Vacuum

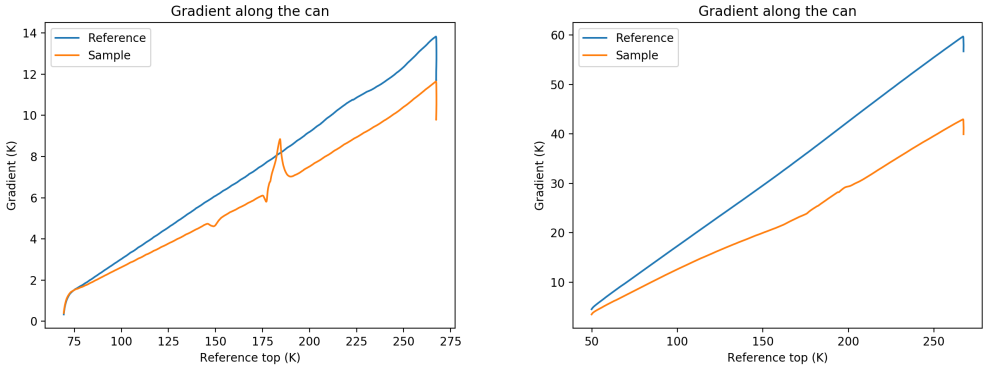
Some of the previous measurements have been repeated in 30*mbar* of helium instead of vacuum to see if this helps with the homogeneity. The results are not encouraging because it seems to be even less homogeneous and the gradients become huge. Because of this also the signals from currents and heaters are worse than in the vacuum case and the peaks almost disapper.
In the next pages there is a comparison between two measurements with a ramp rate of $2K/min$ and CCR fixed at $50K$ performed one in helium and one in vacuum.

Helium is also interesting because it makes measurements faster, easier and may allow to get better glass transitions. Faster because if one cools in helium it takes some hours less then the vacuum cooling and this suits our experiments on glass transitions, because faster cooling gives better results avoiding an almost complete crystallization, that may happen if the sample is cooled very slowly. According to this the helium cooling is useful to get visible glass transitions, but if one does measurements in series that must be in vacuum, the helium valve needs to be closed and opened manually, so doing measurements in helium would make the procedure easier.



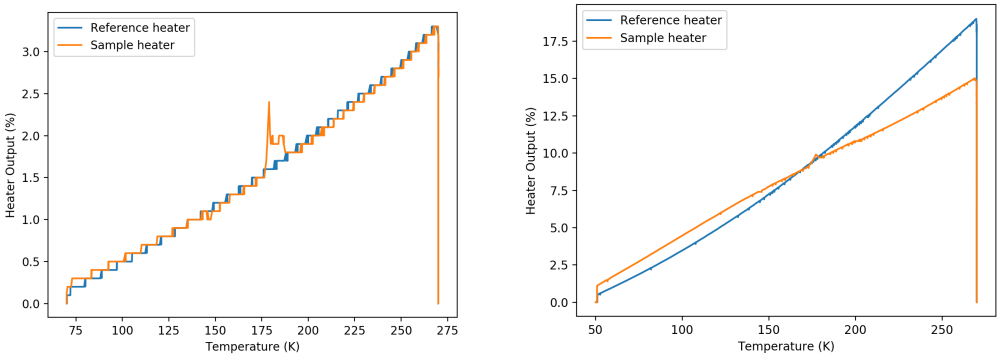
(a) $2K/min$ in Vacuum **(b)** $2K/min$ in Helium

Figure 5.1: All temperatures



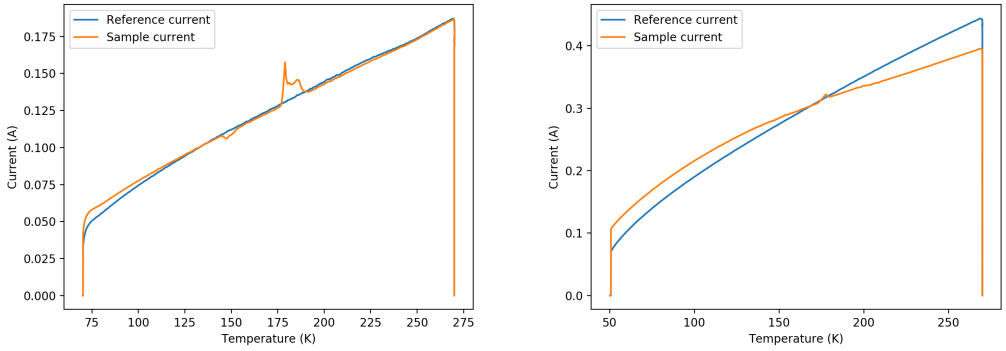
(a) $2K/min$ in Vacuum **(b)** $2K/min$ in Helium

Figure 5.2: Gradient along the can



(a) $2K/min$ in Vacuum **(b)** $2K/min$ in Helium

Figure 5.3: Heat power data

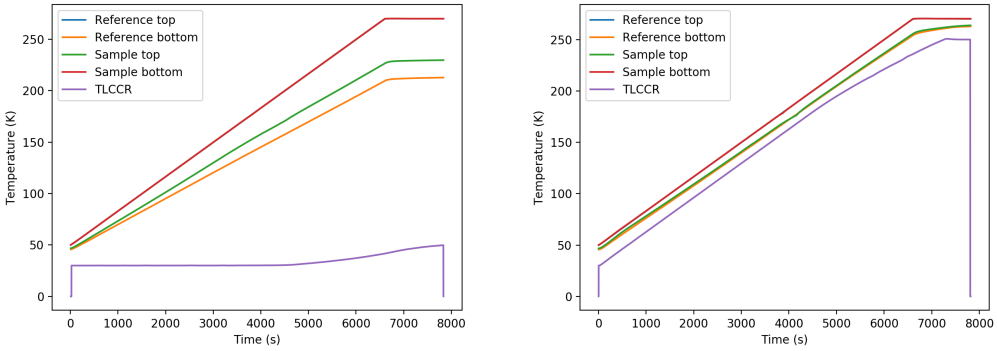


(a) $2K/min$ in Vacuum **(b)** $2K/min$ in Helium

Figure 5.4: Current data

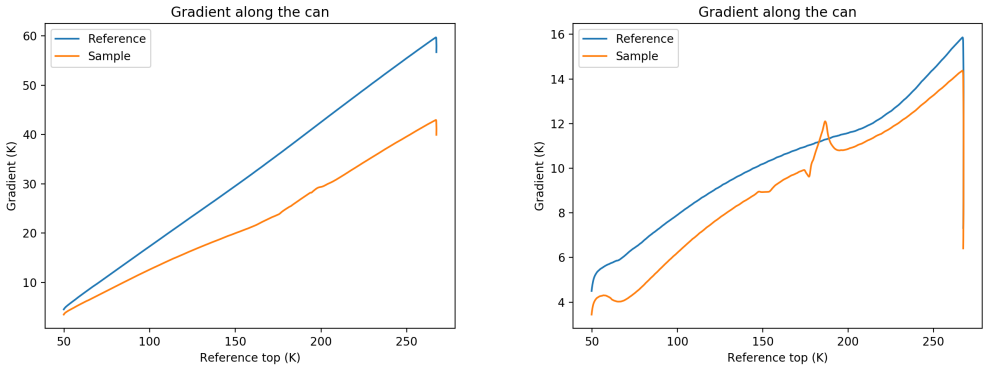
5.2 Ramping CCR vs fixed temperature CCR

In order to get a better gradient and mainly with the aim to find a way to make measurements in helium we tried to ramp the CCR at the same rate of the cans. In this way the bad results obtained with helium is recovered and we get a case similar to the vacuum one. The problem of the inhomogeneity is still not solved. Anyway the next comparison shows that measurements in helium are possible.



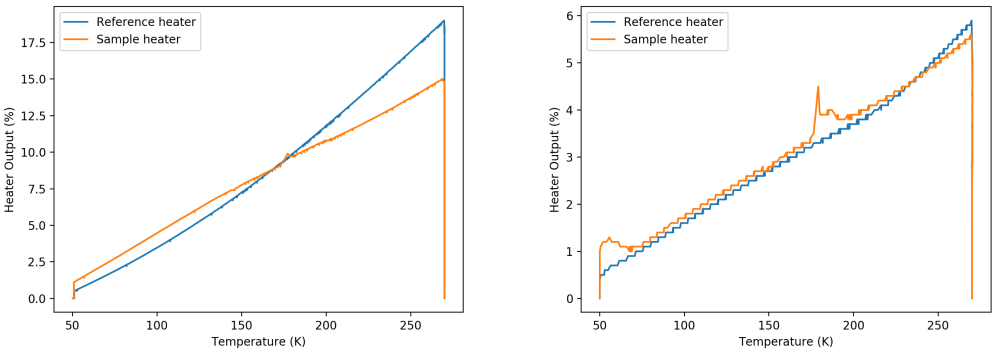
(a) $2K/min$ with fixed CCR **(b)** $2K/min$ with ramping CCR

Figure 5.5: All temperatures



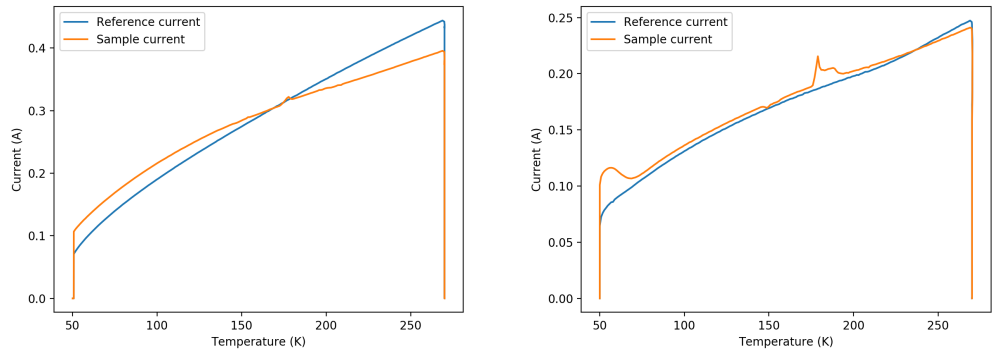
(a) $2K/min$ with fixed CCR **(b)** $2K/min$ with ramping CCR

Figure 5.6: Gradient along the can



(a) $2K/min$ with fixed CCR **(b)** $2K/min$ with ramping CCR

Figure 5.7: Heat power data



(a) $2K/min$ with fixed CCR **(b)** $2K/min$ with ramping CCR

Figure 5.8: Current data

CHAPTER 6

Other tests

The following tests have been partly guided by some measurements done in 2016 and partly are a mean to discover something more about the instrument. All measurements are made using the same Toluen sample of 2.4g.

- In 2016 because of issues with system control experiments have been performed under different conditions. In particular there was a measurement in which the sensors used for heat control have been changed using the one at opposite end of the can so it was opposite to the heater. The results were different than in the case of usual temperature control and it has been decided to repeat the test to see if this can make improvements to the instrument. These measurements are in the Reversed sensors tests section shown in comparison with the corresponding measurement made with the usual temperature control. Measurements are made in vacuum. Results showed that peaks get worse in current and heat signals; a bigger spike appears in the temperature subtraction and also the gradient gets worse.
- The only way to try a proper DTA is to keep the heaters off and ramp the CCR in presence of helium. Good thermal data were not expected because of the non homogeneity of the system and the result is that it is not possible to match the requirements for this tecnique, but we can learn something about conduction trough the linkers that seem to conduct only a very small amount of heat. It emerged that the environment is locally homogeneous around the cans but with different temperatures between them. This may be because of the CCR, but also because of the fact of having a cap on the reference and not on the sample. The results of this measurement are in the Heaters off test section.

6.1 Reversed sensors tests

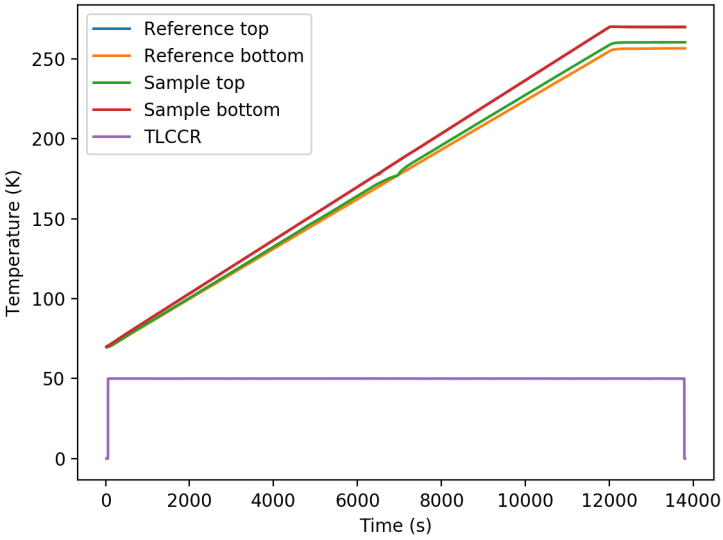


Figure 6.1: $1K/min$ ususal temperature control

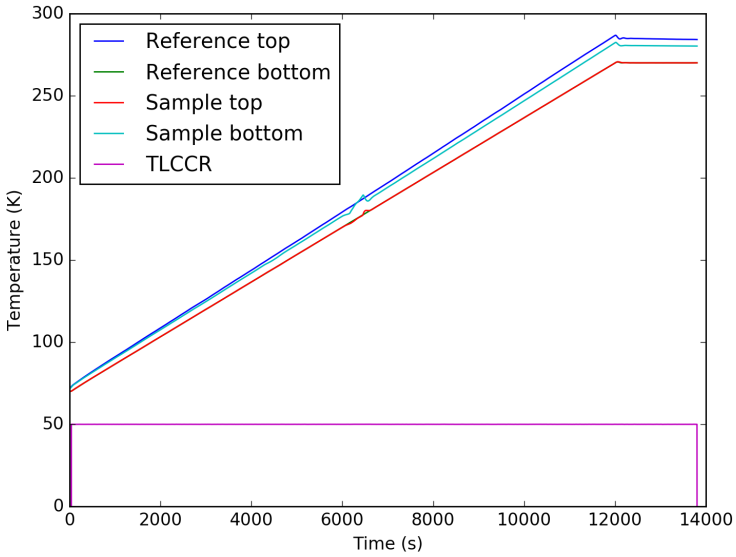


Figure 6.2: $1K/min$ reversed tenperature control

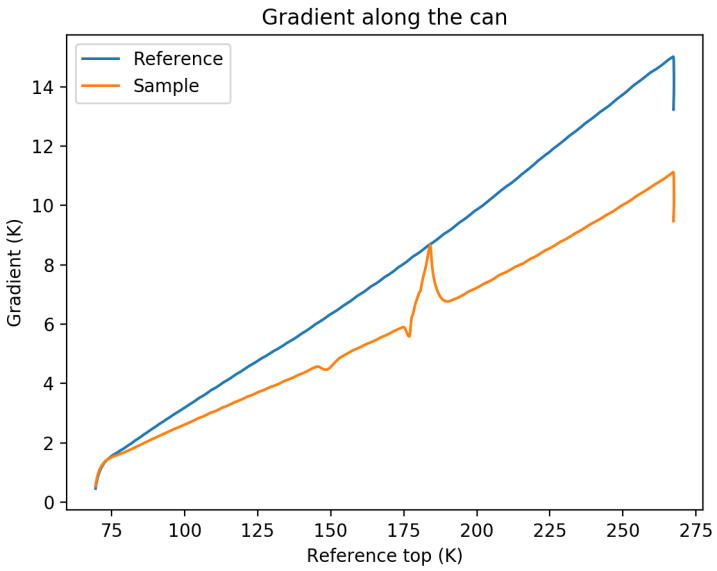


Figure 6.3: $1K/min$ ususal temperature control

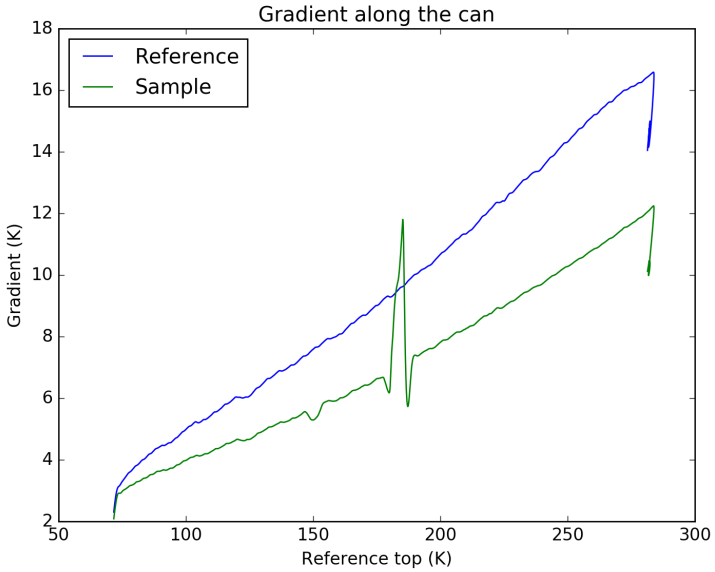


Figure 6.4: $1K/min$ reversed tenperature control

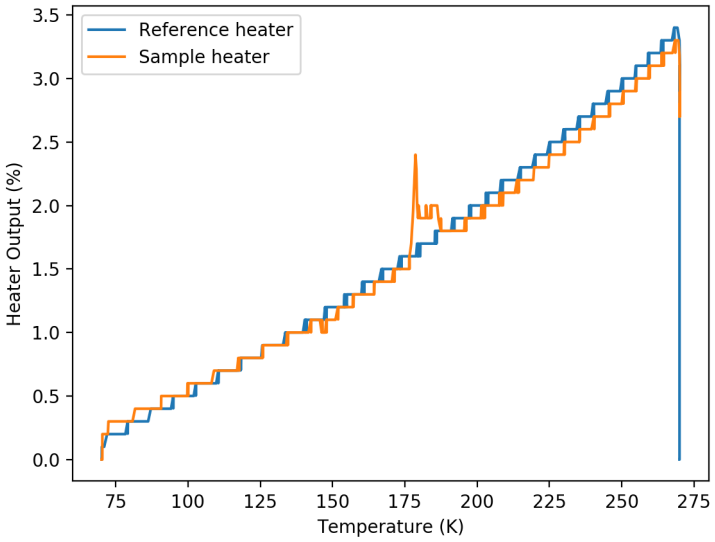


Figure 6.5: $1K/min$ ususal temperature control

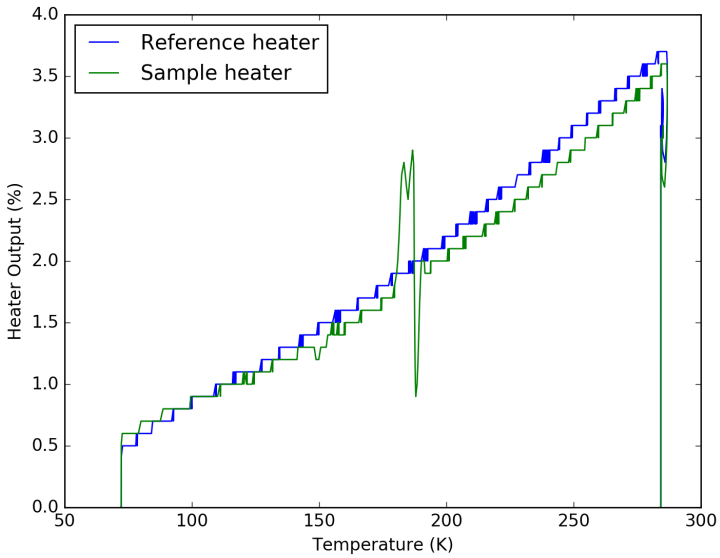


Figure 6.6: $1K/min$ reversed tenperature control

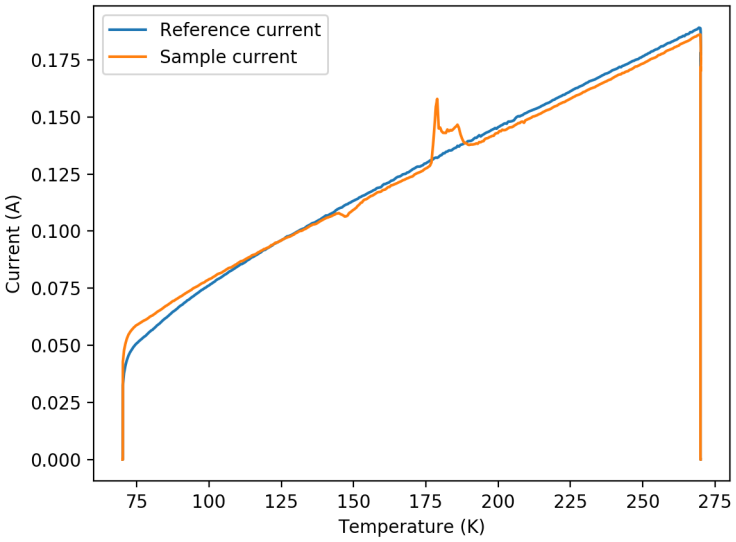


Figure 6.7: $1K/min$ ususal temperature control

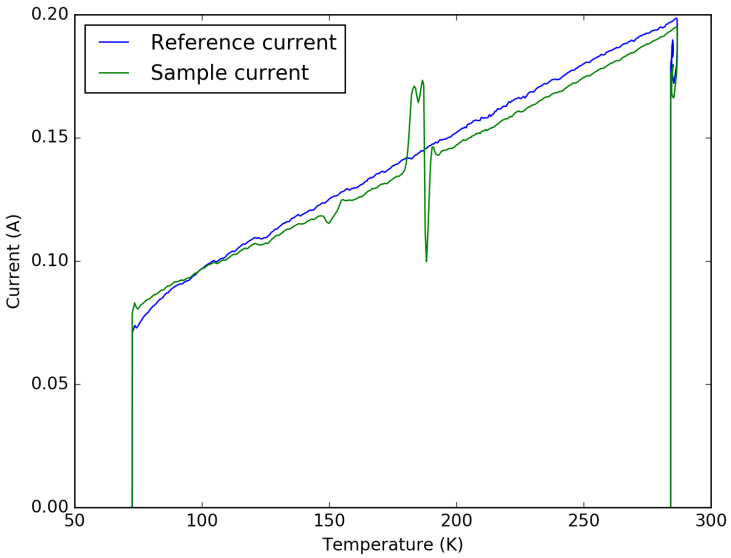


Figure 6.8: $1K/min$ reversed tenperature control

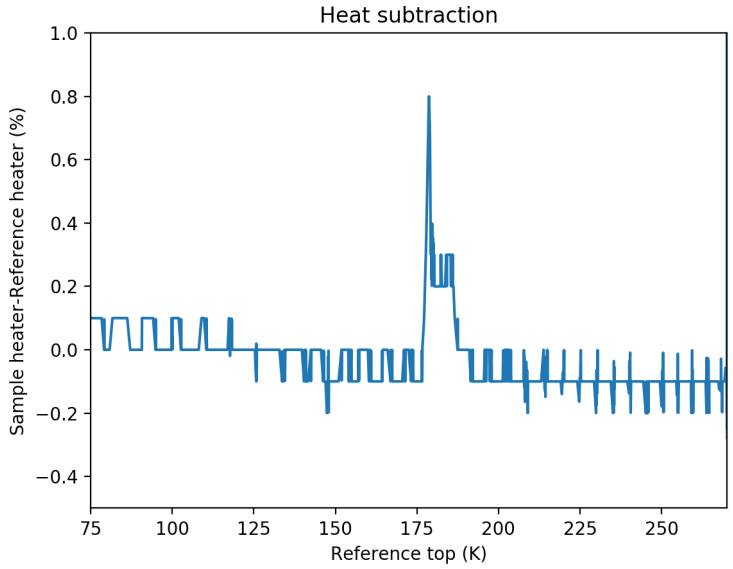


Figure 6.9: $1K/min$ ususal temperature control

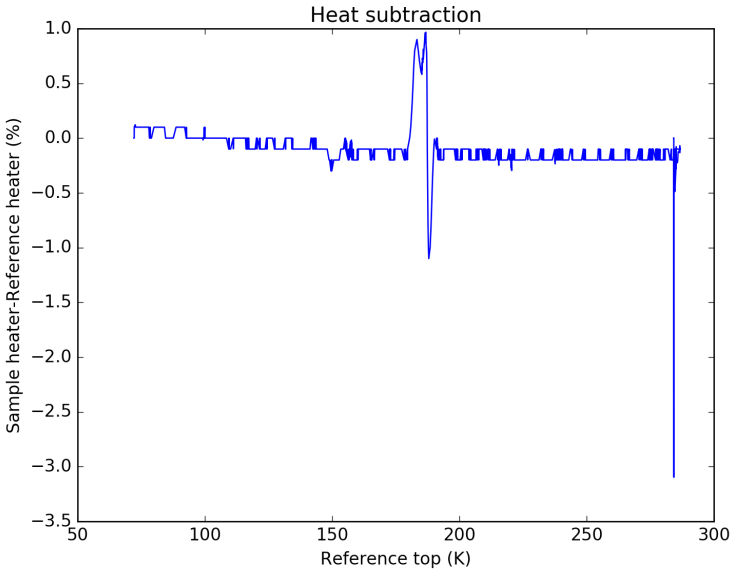


Figure 6.10: $1K/min$ reversed tenperature control

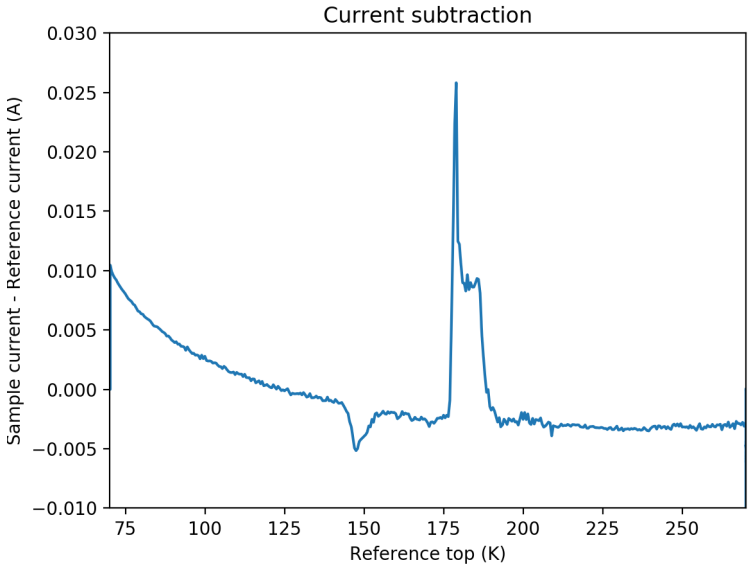


Figure 6.11: $1K/min$ ususal temperature control

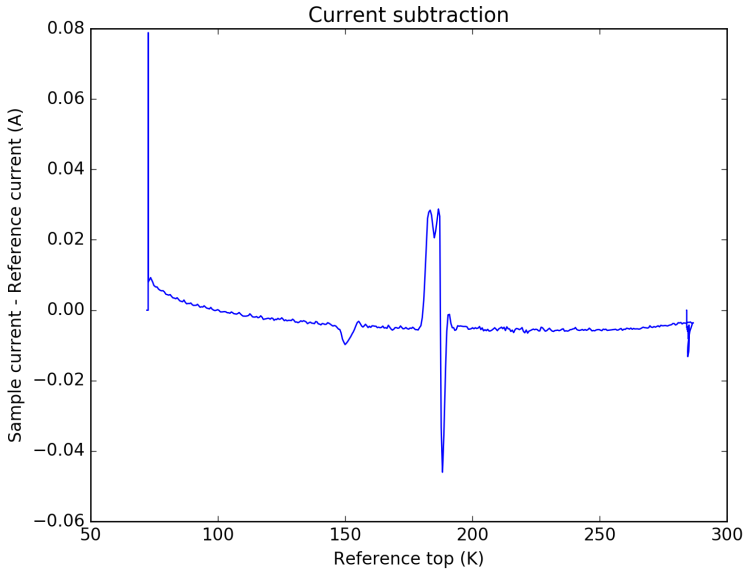


Figure 6.12: $1K/min$ reversed tenperature control

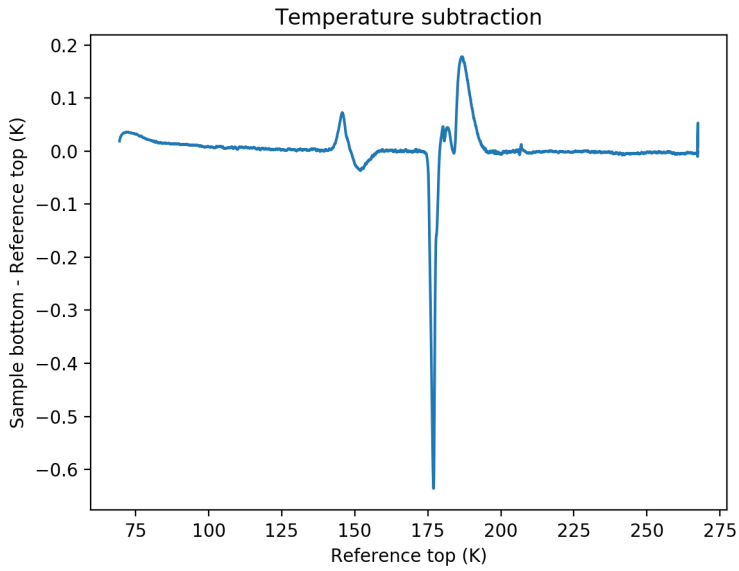


Figure 6.13: $1K/min$ ususal temperature control

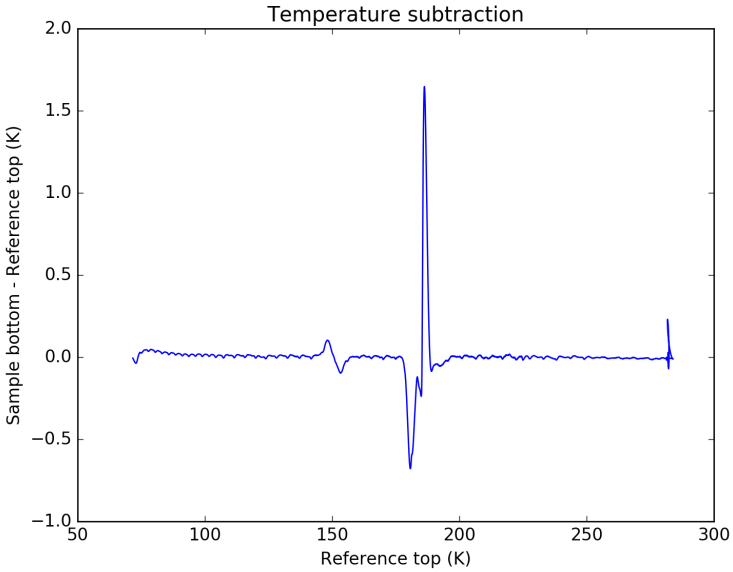
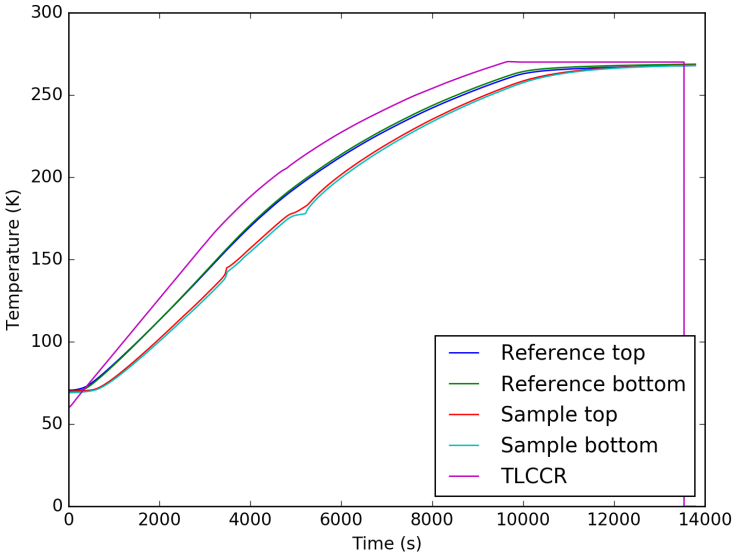


Figure 6.14: $1K/min$ reversed tenperature control

6.2 Heaters off tests



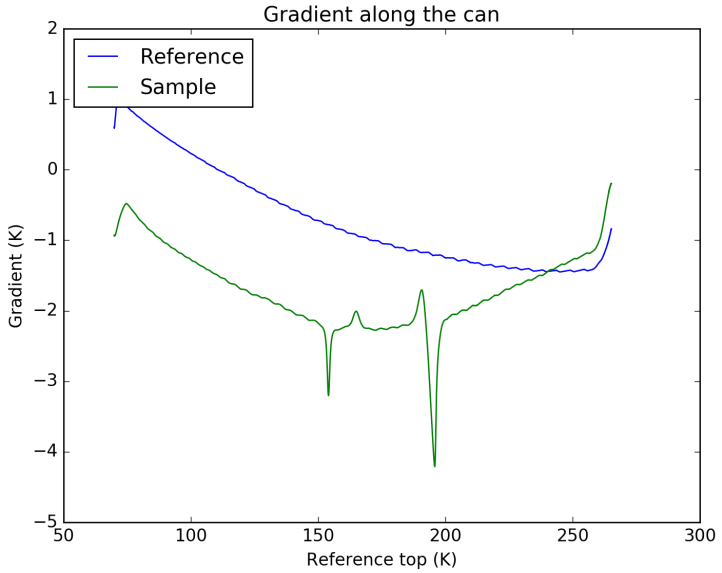


Figure 6.15: $2K/min$ ramping CCR, heaters off

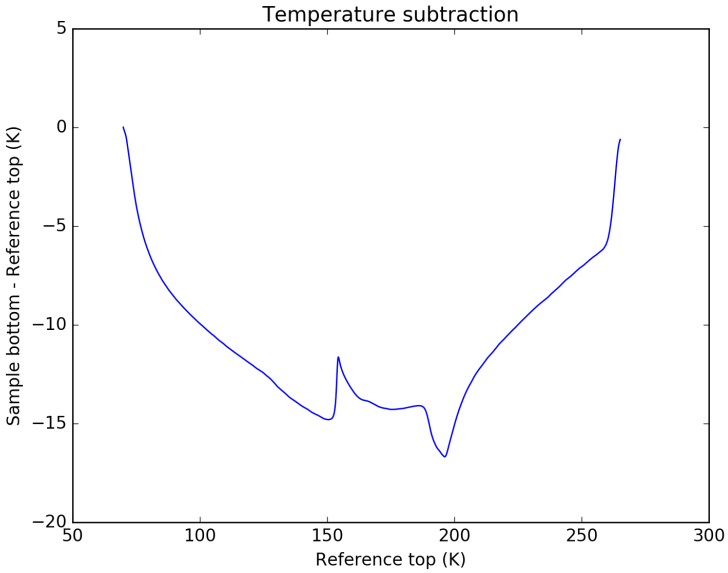


Figure 6.16: $2K/min$ ramping CCR, heaters off

CHAPTER 7

Empty-empty characterization

A modification to the instrument was needed to see glass transiton, but before doing it, a few tests with empty cans have been performed in order to chacterize the system and understand what to change.

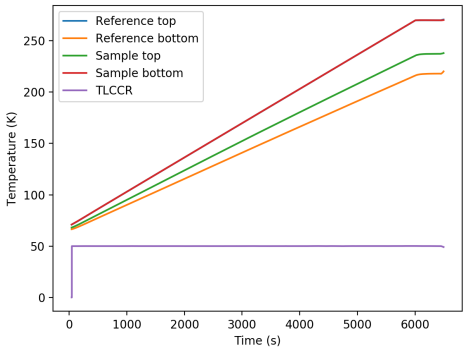
Several tests have been performed:

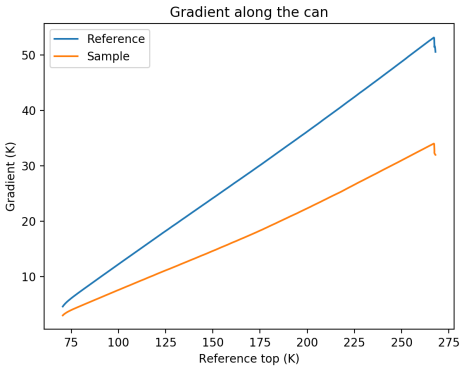
- Measurements at $2K/min$ with fixed CCR at $50K$ in Helium and in Vacuum confirmed that, with this setup, the system has homogeneity problems and that being in helium is the worst condition with the fixed CCR.
- $2K/min$ with ramping CCR in Helium
- Some measurements have been repeated moving the heaters on the end near the linker to see if the gradient is only caused by the thermal conduction to the bracket or also by conductivity of the cans that is lower than expected. These measurements show a slight decrease of the gradient but they confirm the fact that the low conduction along the can is a cause of the temperature gradient.
- The measurement with the heaters turned off has been repeated, but with empty cans, to study the environment and the conduction through the linkers without any extra mass to take into account. The temperature subtraction is also shown in this case. The result is a proof of the impossibility of a proper DTA measurement because the non homogeneity does not allow us to have a good baseline.

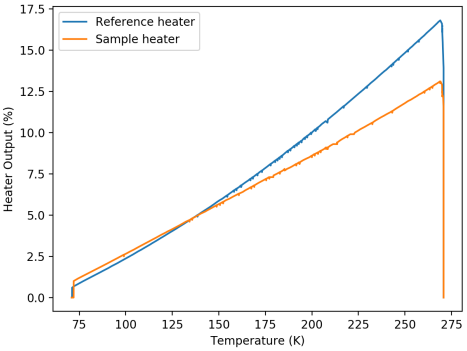
Measurements in general confirm the obsevations made at the end of the previous chapter: environment not homogeneous but quite homogeneous around the single can; there may be a small conduction through the linker, because the top of the sample can is 1 or 2K hotter than the bottom, but the insulation is still quite good. The material used to build the cans is not as conductive as expected.

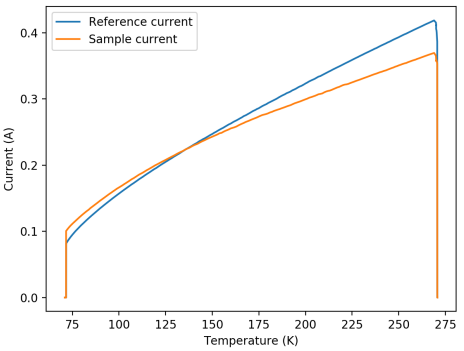
7.1 Empty-empty tests with fixed CCR

- $2K/min$ in Helium

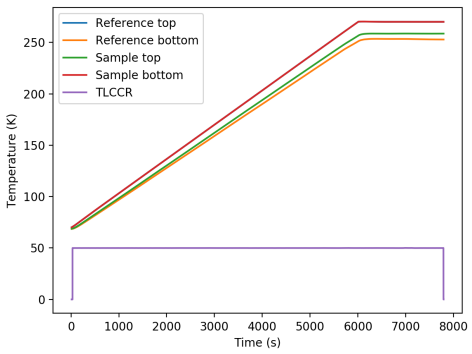


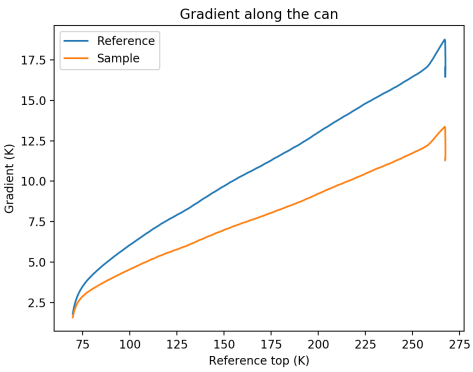


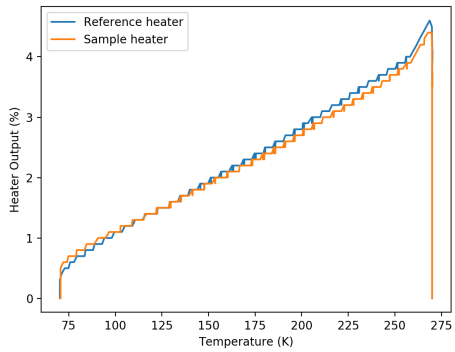


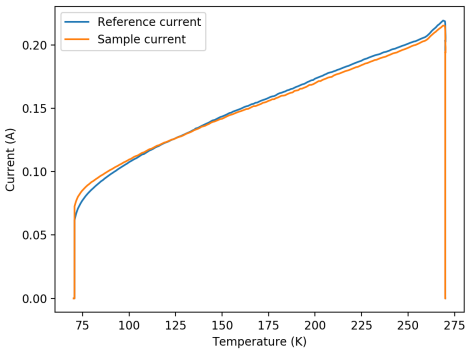


- $2K/min$ in Vacuum

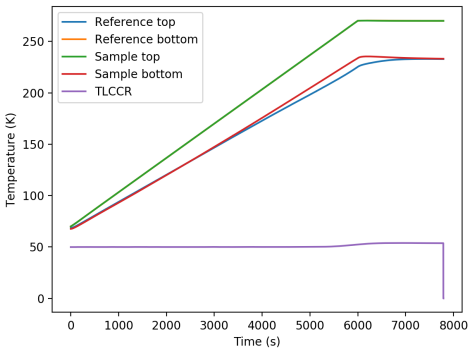


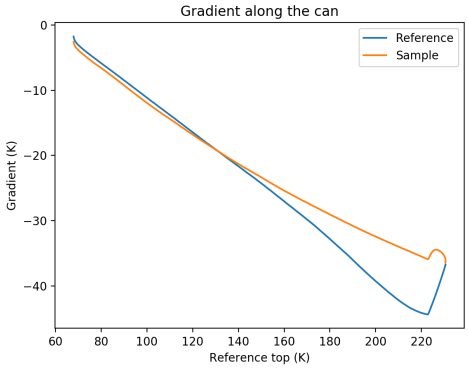


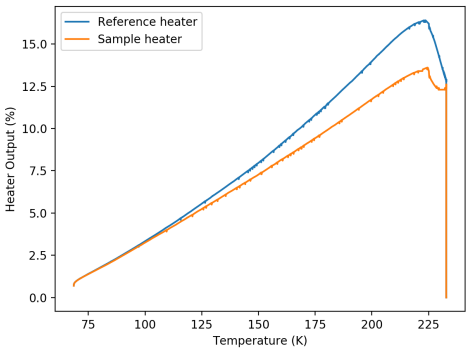


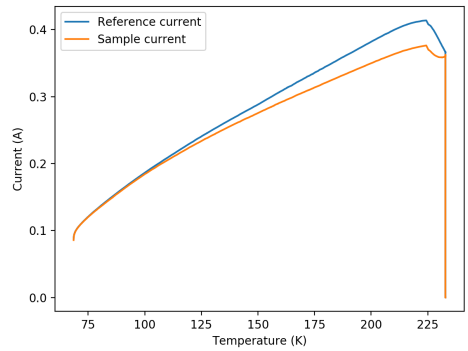


- $2K/min$ in Helium with reversed heaters



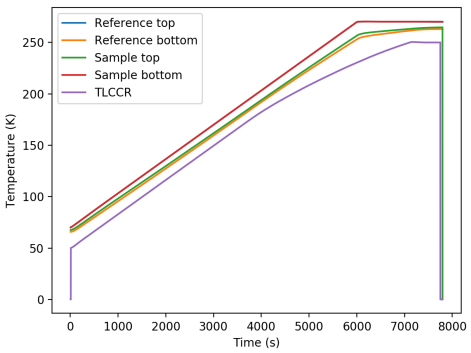


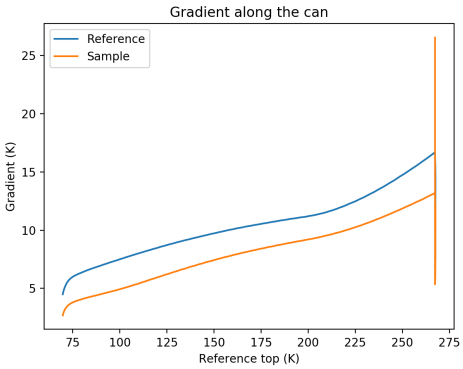


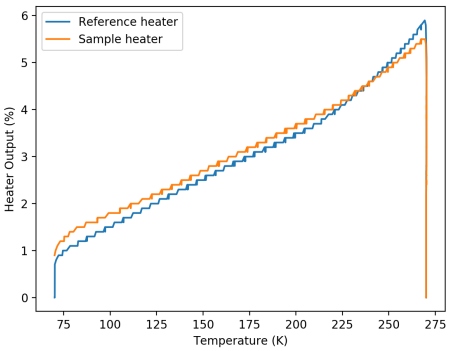


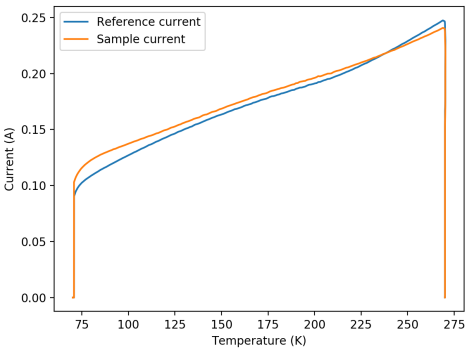
7.2 Empty-empty tests with ramping CCR

- $2K/min$ in Helium

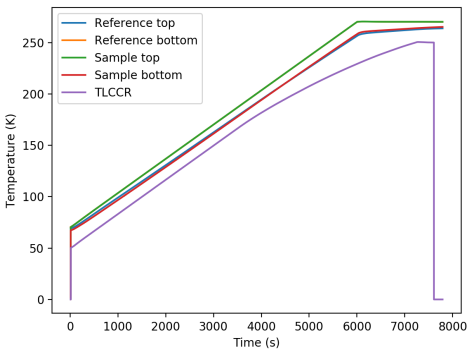


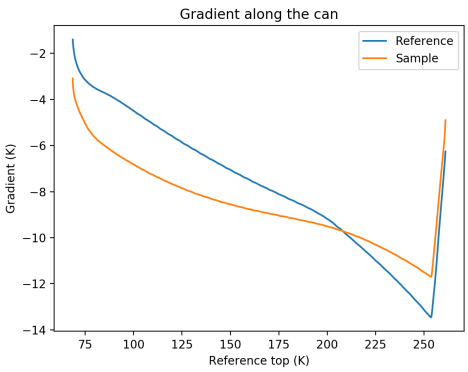


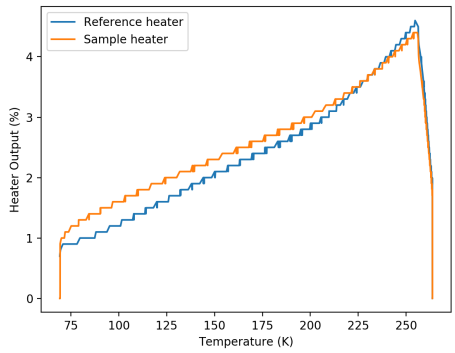


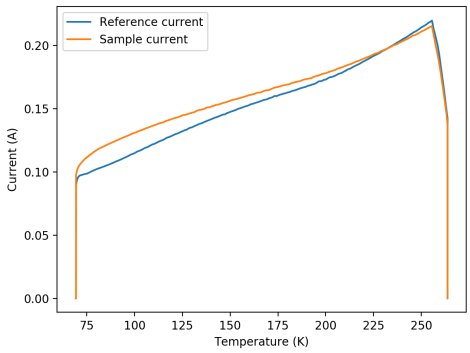


- $2K/min$ in Helium with reversed heaters



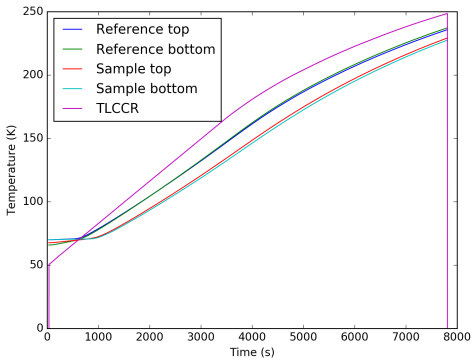


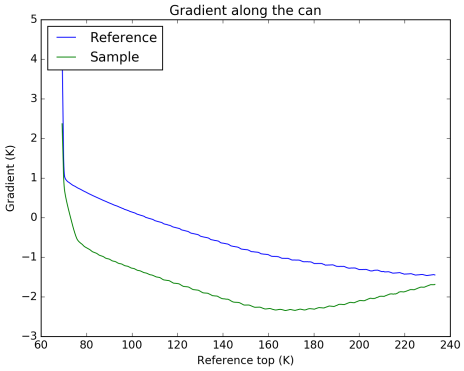


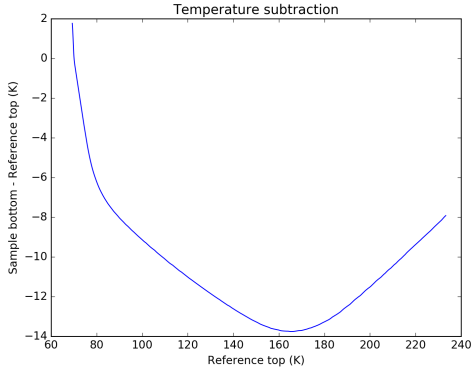


7.3 Empty-emptytests with heaters off

- CCR ramping $2K/min$, in Helium







CHAPTER 8

New Instrument setup with two heaters per can

To solve the temperature gradient problem two heaters have been added, one per can. This cannot guarantee a complete homogeneity of the temprature along the can but may reduce the problem. Every heater is controlled by the sensor in the pocket near it.
First attempts were made with the CCR fixed at $50K$ in helium but setting the medium heating power both cans found problems to reach $270K$. The medium heating power allows to get a better signal from the heater output due to the better resolution. Ramping CCR choice the medium heating power is enough to perform the experiment.

All measurements with this setup showed glass transiton. The samples used are Toluene and Glycerol.

8.1 New measurements showing glass transitions

8.1.1 Toluene

All temperatures

Now all the four temperatures measured on cans match, but emerges from this graphs that the CCR used cannot ramp faster than $1K/min$.

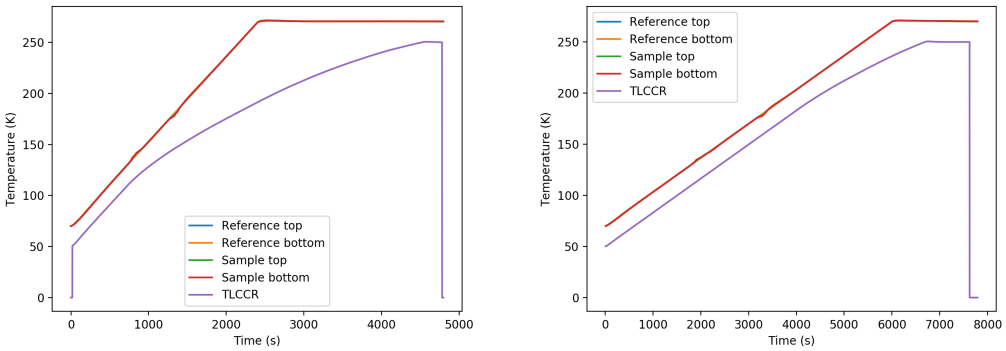


Figure 8.1: $5K/min$ on the left and $2K/min$

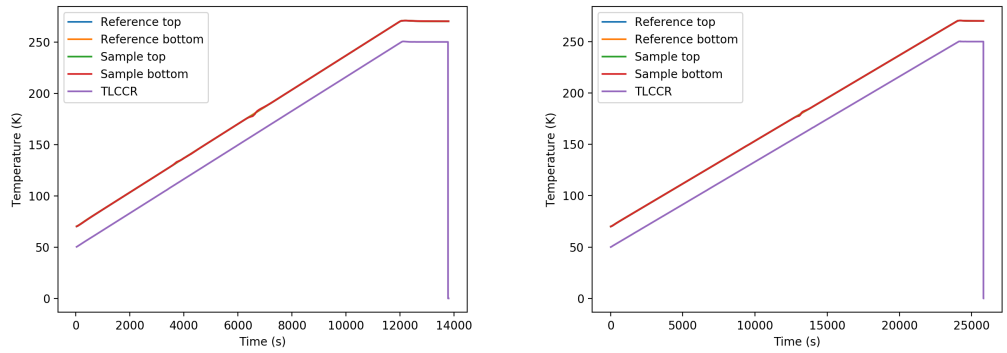


Figure 8.2: $1K/min$ on the left and $0.5K/min$

Gradient

The gradient is close to zero in the reference can while in the sample can it reaches $1K$ in correspondence of the transitions. This is a consequence of a residual inhomogeneity, in particular there is a different response of the temperature control. This is a small effect so it does not affect much data.

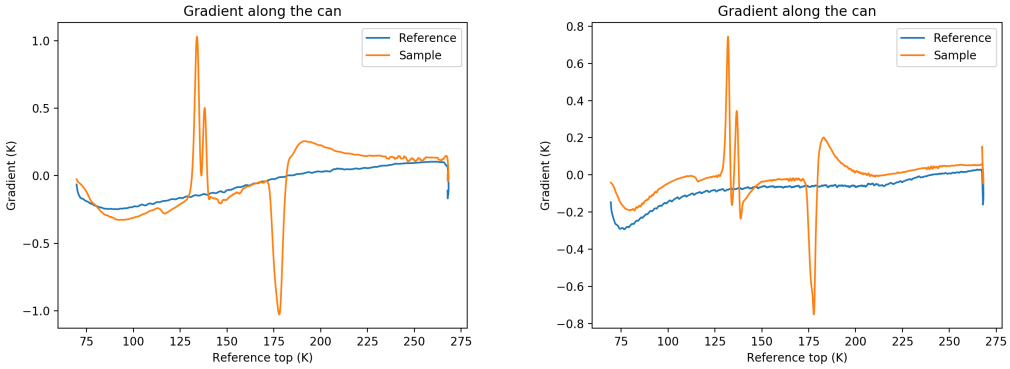


Figure 8.3: $5K/min$ on the left and $2K/min$

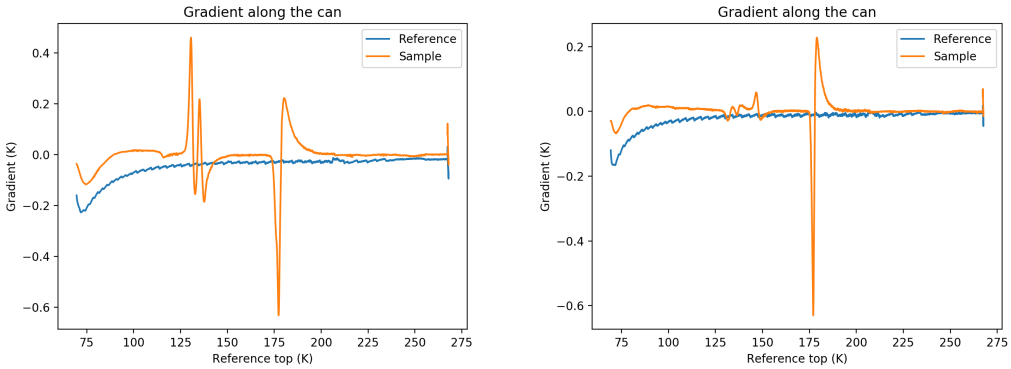


Figure 8.4: $1K/min$ on the left and $0.5K/min$

Heaters and currents

All the transitions are already visible in the sample outputs. The inhomogeneity of the environment causes the differences between the top and the bottom of each can and may also cause the strange behavior of the reference outputs in the $1K/min$ and $0.5K/min$ measurements.

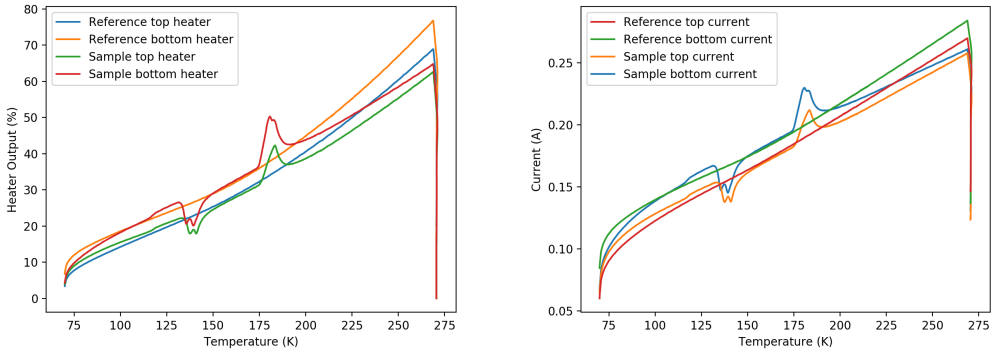


Figure 8.5: $5K/min$ heaters on the left and currents on the right

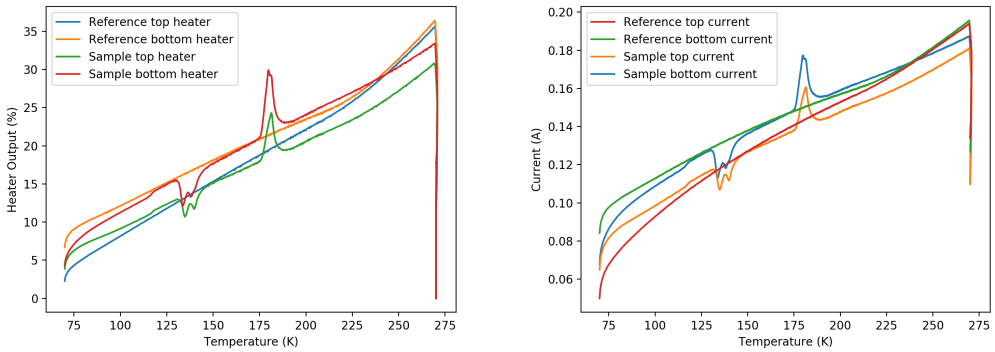


Figure 8.6: $2K/min$ heaters on the left and currents on the right

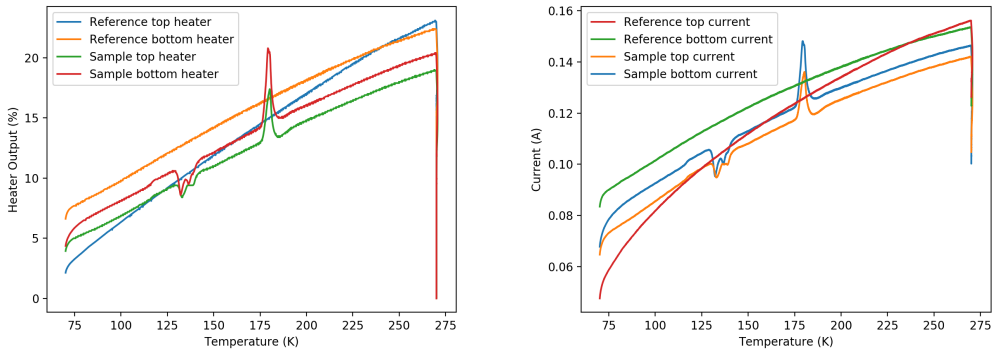


Figure 8.7: $1K/min$ heaters on the left and currents on the right

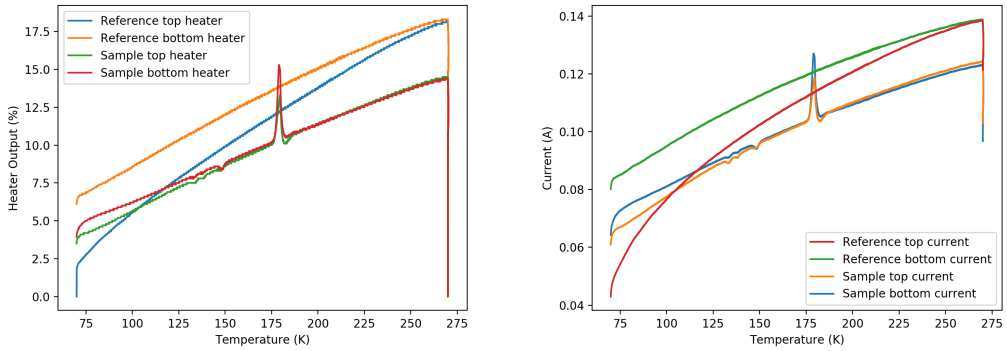
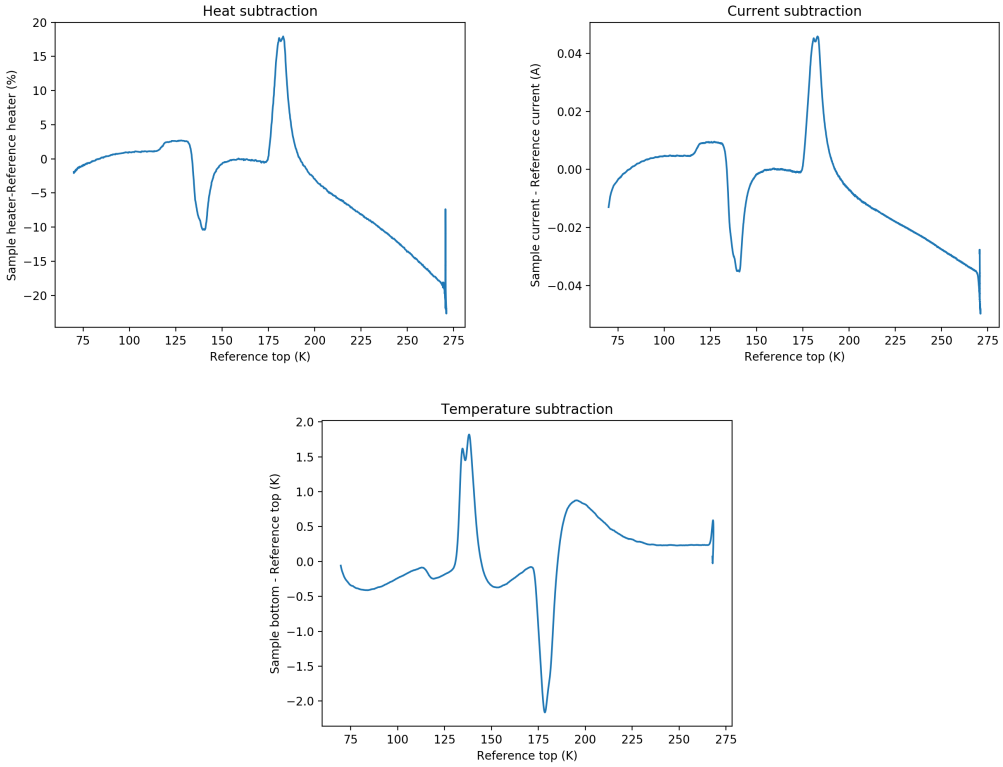


Figure 8.8: $0.5K/min$ heaters on the left and currents on the right

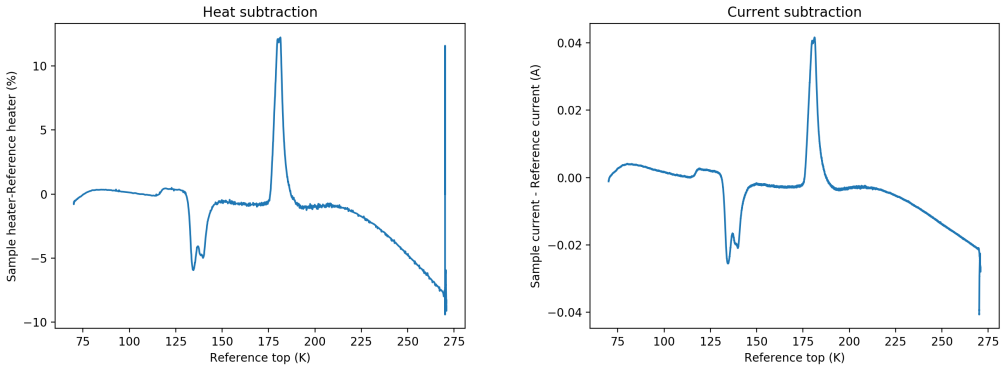
Temperature, current and heating power subtractions

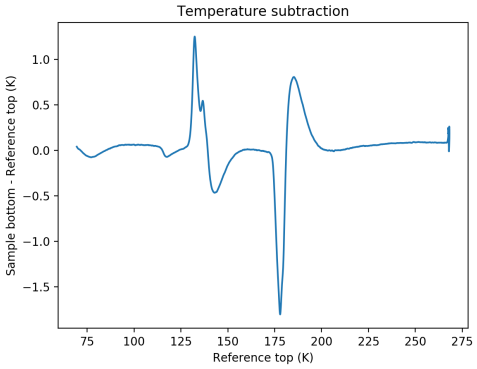
Changing the set up all the transitions expected show but temperature seems a worse indicator for transitions if compared to previous results.

- $\underline{5K/min}$All the transitions are clearly visible but the baseline is not flat as needed. This may be caused by the different behavior at the ends of the cans that becomes more visible going faster, when the system has not much time to equilibrate.

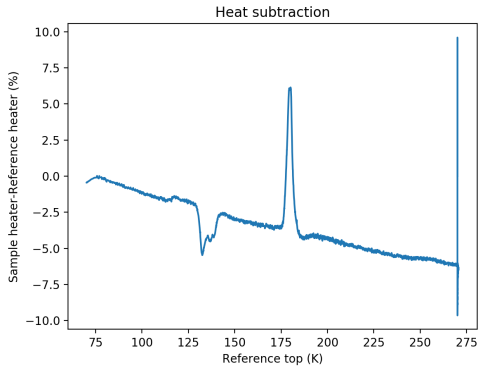


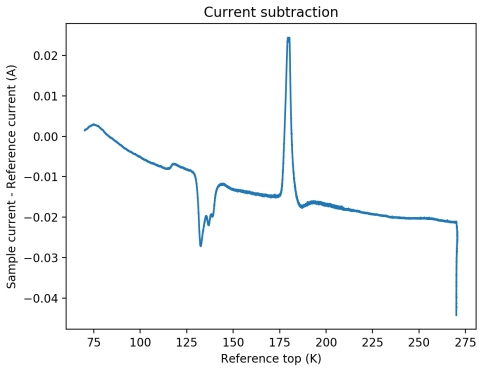
- $\underline{2K/min}$The baseline improves but there are still problems at high temperatures because of the CCR not ramping linearly that makes the reference output be higher.

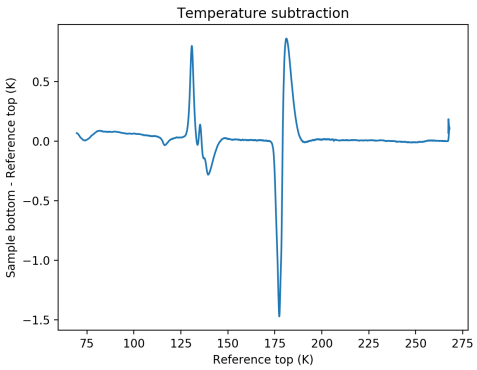




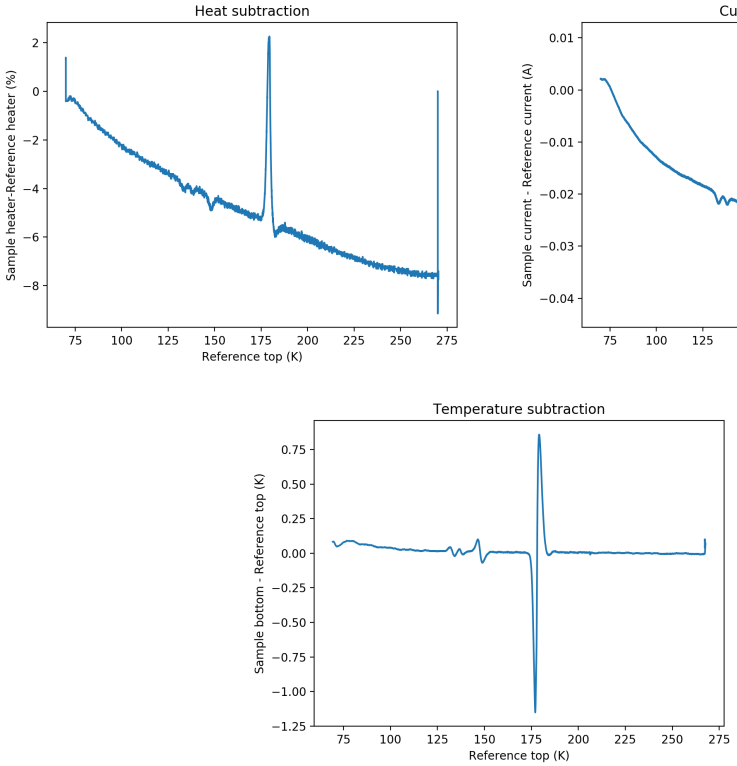
- $\underline{1K/min}$ Using this ramp rate the high temperature problem is recovered and all the transitions are still visible.







- $\underline{0.5K/min}$ In this case the ramp is very slow and this results in smaller steps and peaks making some transitions difficult to be detected.



8.1.2 Glycerol

Changing the sample the only big difference is that the glass transition is still visible at $0.5K/min$, this is because the step in the case of glycerol is a lot bigger than in toluene. Glycerol also shows a strange behaviour and makes difficult to get a good baseline.

All temperatures

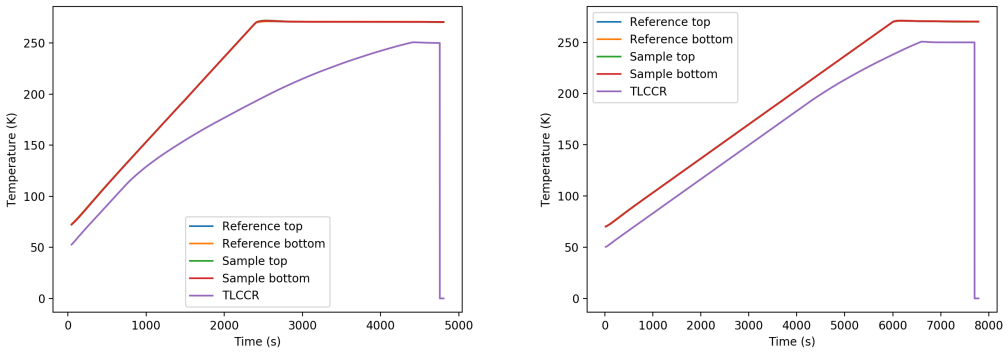


Figure 8.9: $5K/min$ on the left and $2K/min$

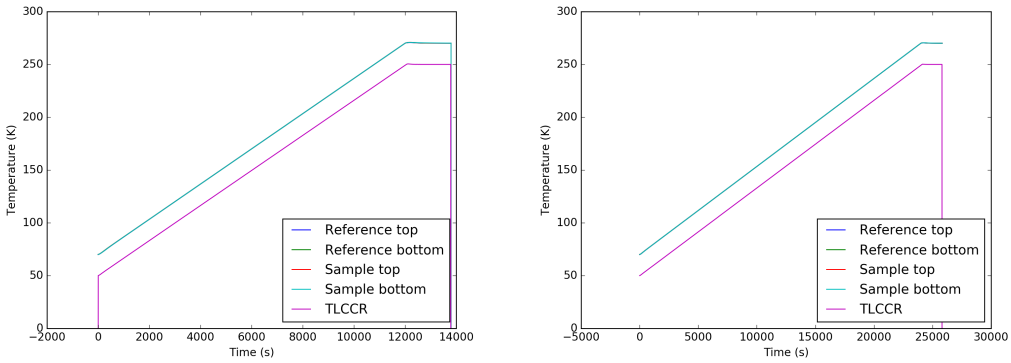


Figure 8.10: $1K/min$ on the left and $0.5K/min$

Gradient

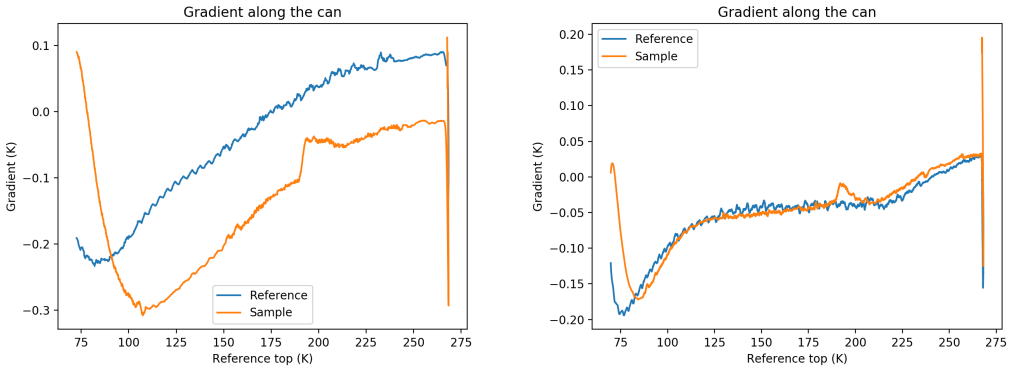


Figure 8.11: $5K/min$ on the left and $2K/min$

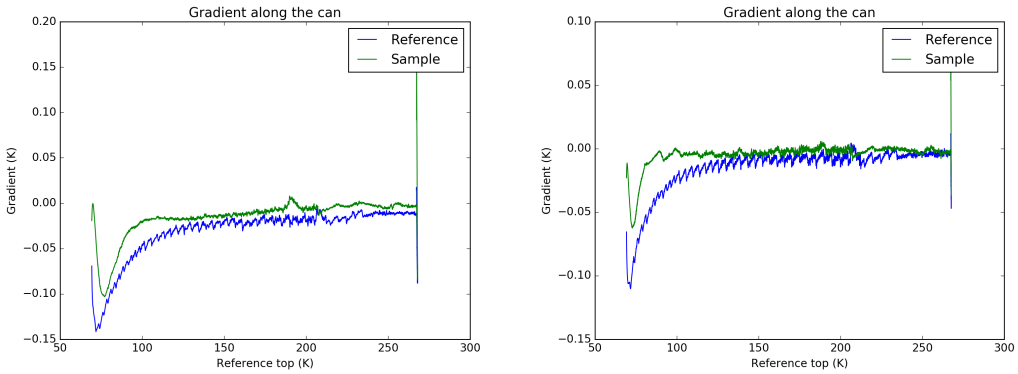


Figure 8.12: $1K/min$ on the left and $0.5K/min$

Heaters and currents

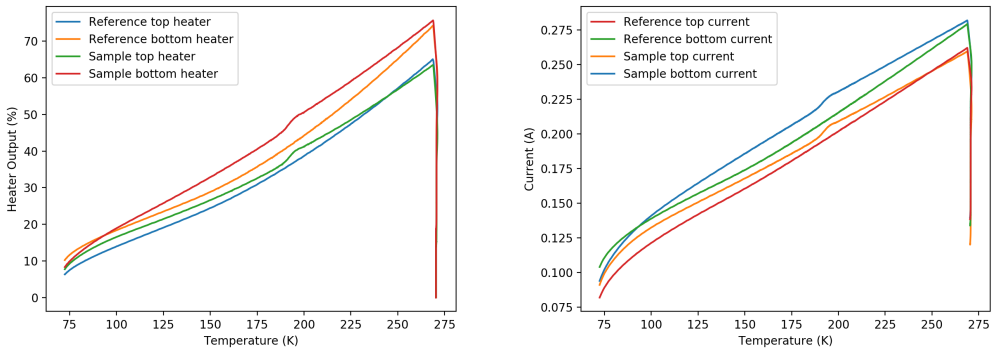


Figure 8.13: $5K/min$ heaters on the left and currents on the right

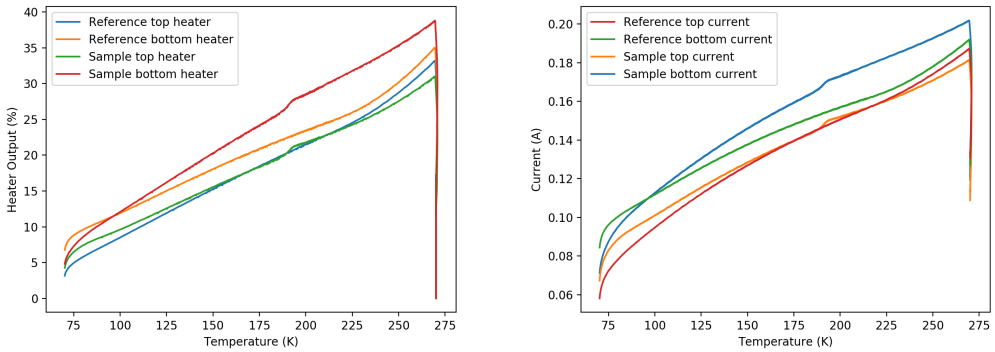


Figure 8.14: $2K/min$ heaters on the left and currents on the right

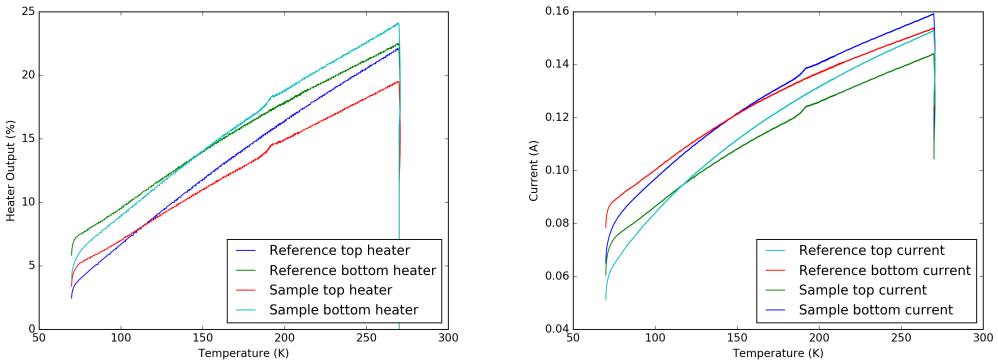


Figure 8.15: $1K/min$ heaters on the left and currents on the right

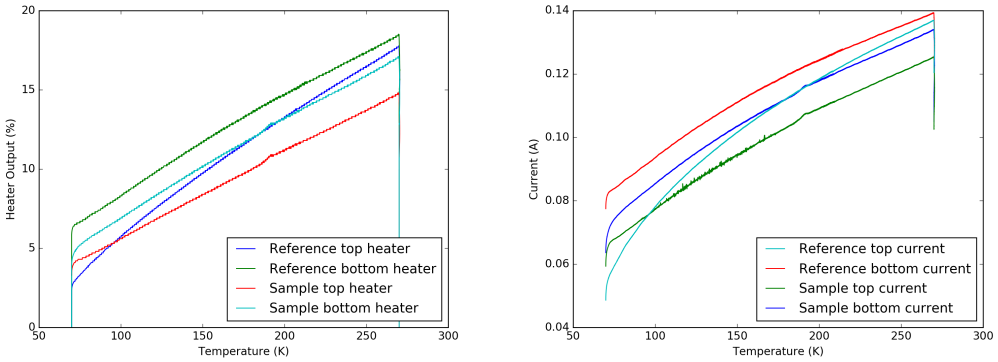
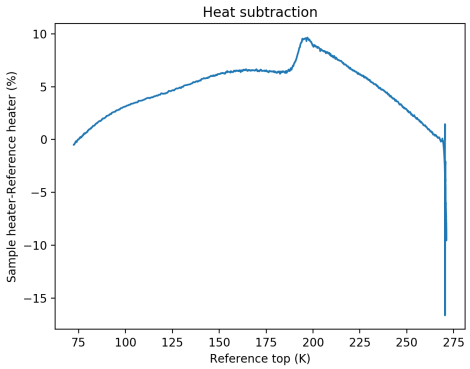
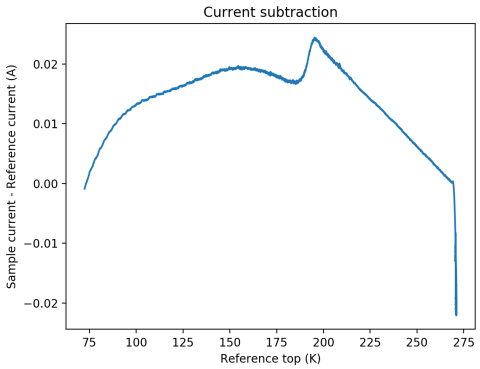


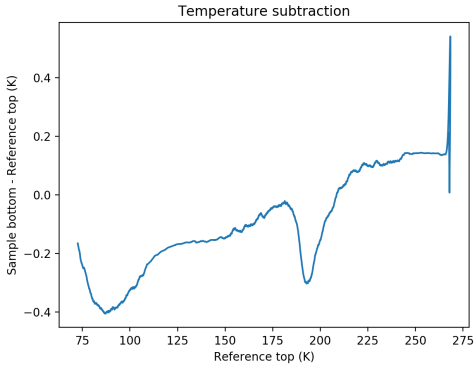
Figure 8.16: $0.5K/min$ heaters on the left and currents on the right

Temperature, current and heating power subtractions

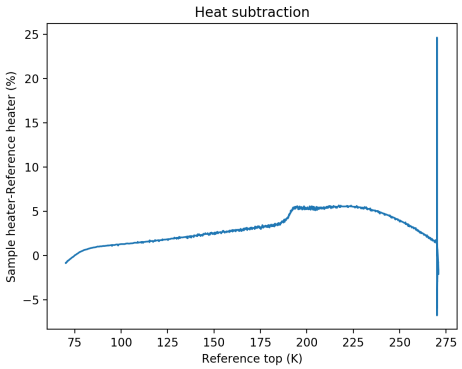
- $5K/min$

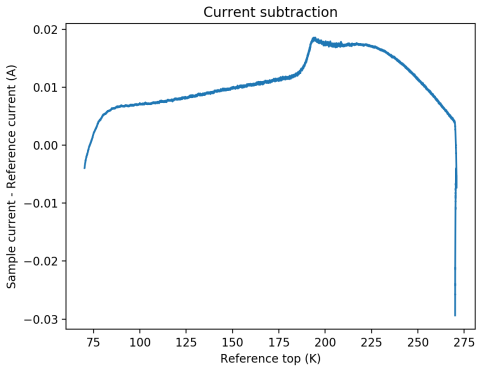


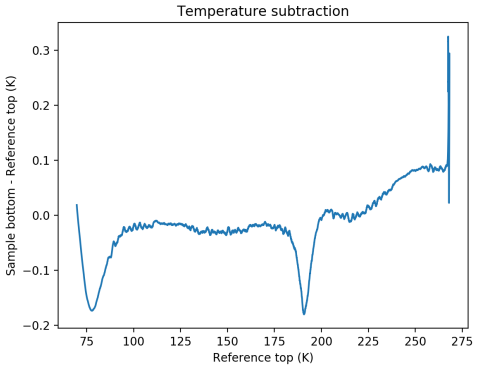




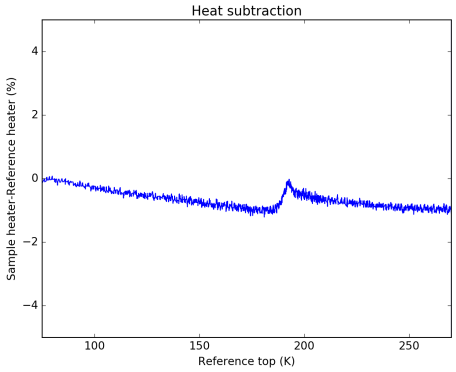
- $2K/min$

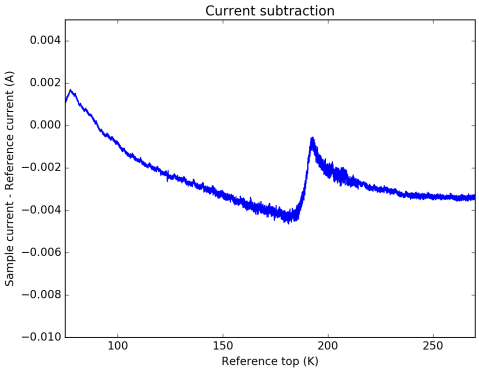


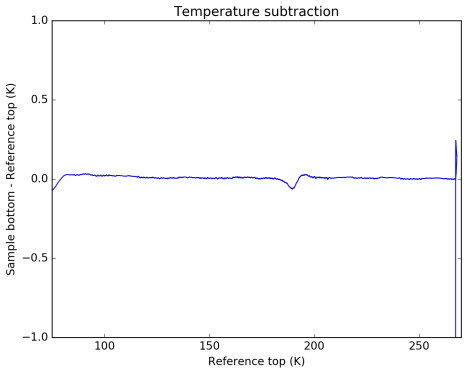




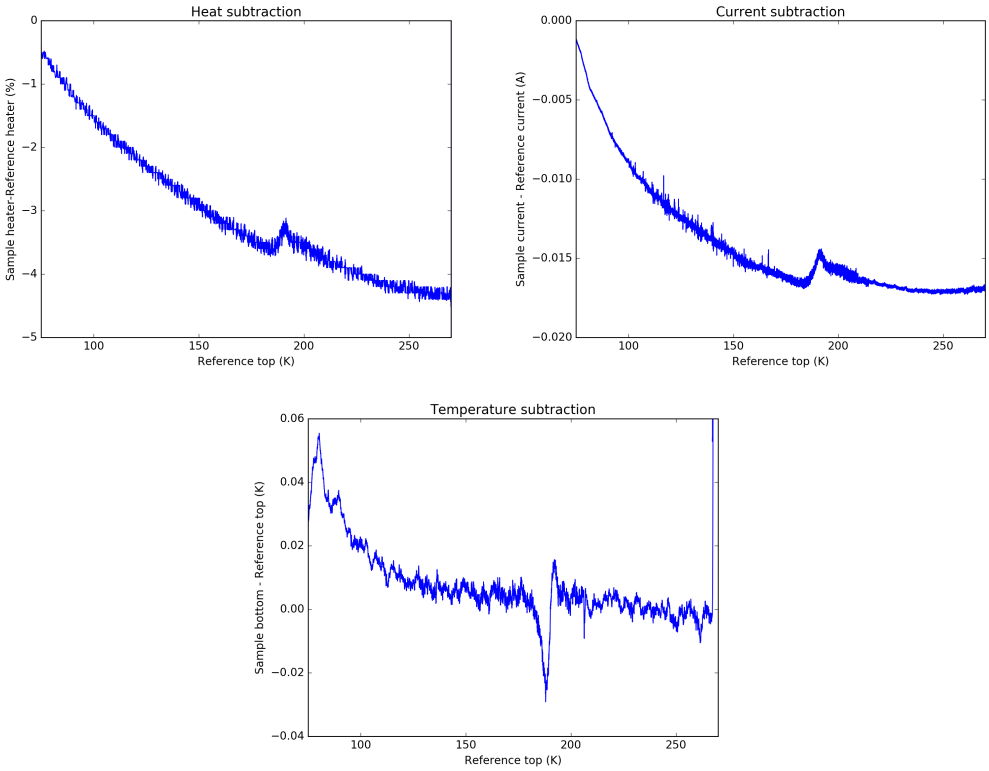
- $\underline{1K/min}$







- $0.5K/min$

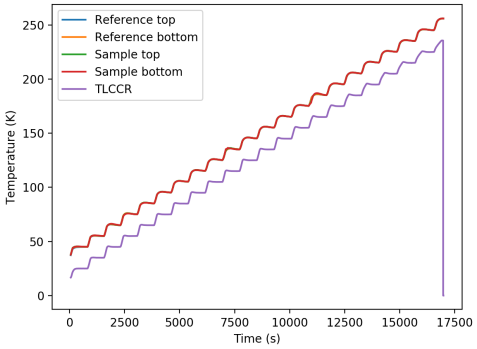

Heat subtraction
Sample heater-Reference heater (%)
Reference top (K)
Current subtraction
Sample current - Reference current (A)
Reference top (K)
Temperature subtraction
Sample bottom - Reference top (K)
Reference top (K)

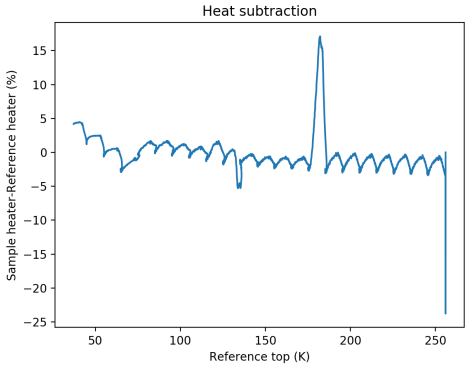
CHAPTER 9

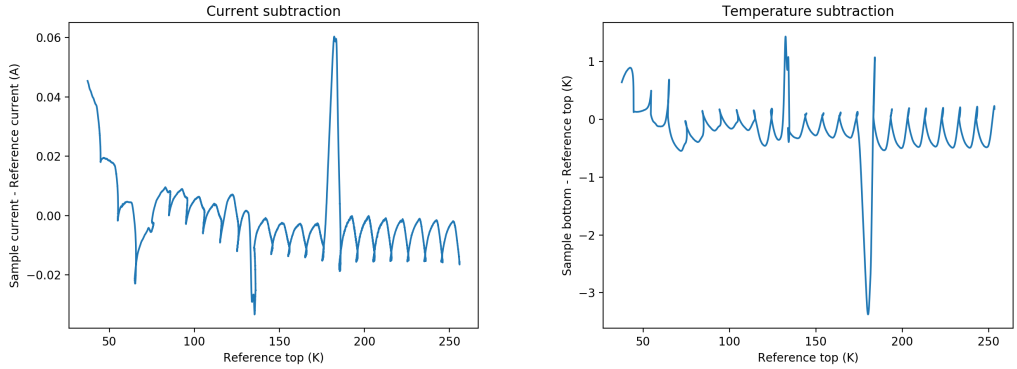
Future developement

9.1 Neutron measurements

Now that the calorimeter works the first challenge is to adapt it to neutron measurements. In order to do this, some tests using a procedure that stays in between of the linear ramps and the usual way in which neutron experiments are performed:during the elastic scan the ramp in temperature is made using "steps" usually every $10K$ and sitting for $10min$ on every step. In the following measurement the steps are substituted by quick ramps $(5K/min)$ that stop every $10K$ for $10min$.
Results highlight the fact that someting needs to be changed or the glass transitions are not visible. The measurements shown below have been made with the Toluene sample used before.







9.2 Limitations and possible improvements

During all these tests some limitations and possible improvements have emerged:

- Temperature sensors give a strict limit that makes difficult to find samples if one looks for D_2O solutions of surfactants or materiarls showing similar transitions and obviously if one looks at polymers. The latter has not ben a problem until because the sample holder allowed us to use only materials liquid at room temperature.
  Anyway by changing the sensors there will be a wider range of measurements possible.
- The cans limit us because of two things. Firstly the sample has to be liquid. A low viscosity to makes it easier to fill, but using a pump it is possible to use also viscous liquids like glycerol. Moreover, thinking about neutron measurements, the thickness of the sample anular region $(0.5mm)$ makes it suitable only for measurement of deuterated materials, while for non deuterated one should use a $0.1mm$ thick anular cavity.
  The can issue is the most difficult to solve because requires a new project for the can.
- A quick fix to the fact that the CCR cannot ramp at rates higher than $1K/min$ is the addition of another heater.

APPENDIX A

List of measurements

A.1 Linkers tests

Run	Sample	Linker	Environment	Rate	CCR	Notes
74923	Toluene	Honeycomb	Vacuum	$2K/min$	CCR 50K	
74925	Toluene	Honeycomb	Vacuum	$1K/min$	CCR 50K	
74927	Toluene	Honeycomb	Vacuum	$0.5K/min$	CCR 50K	
74930	Toluene	Honeycomb	Vacuum	$1K/min$	CCR 1K/min	
74932	Toluene	Honeycomb	Vacuum	$1K/min$	CCR 50K	Reversed sensors
74934	Toluene	Honeycomb	Vacuum	$1K/min$	CCR 1K/min	
74936	Toluene	Honeycomb	Vacuum	$1K/min$	CCR 1K/min	Medium heating power
74938	Toluene	Honeycomb	Vacuum	$1K/min$	CCR 1K/min	Low heating power
74944	Toluene	Aluminium	Vacuum	$1K/min$	CCR 50K	
74945	Toluene	Aluminium	Vacuum	$0.5K/min$	CCR 50K	
74947	Toluene	Aluminium	Vacuum	$2K/min$	CCR 50K	
74949	Toluene	Aluminium	Vacuum	$1K/min$	CCR 50K	Reversed sensors
74951	Toluene	Stainless Steel	Vacuum	$1K/min$	CCR 50K	Reversed sensors
74953	Toluene	Stainless Steel	Vacuum	$2K/min$	CCR 50K	
74955	Toluene	Stainless Steel	Vacuum	$1K/min$	CCR 50K	
74956	Toluene	Stainless Steel	Vacuum	$0.5K/min$	CCR 50K	
74986	Toluene	G-10	Vacuum	$2K/min$	CCR 50K	
74960	Toluene	G-10	Vacuum	$1K/min$	CCR 50K	
74961	Toluene	G-10	Vacuum	$0.5K/min$	CCR 50K	
74965	Toluene	G-10	Vacuum	$1K/min$	CCR 50K	Reversed sensors

A.2 Further tests with G-10 linkers

Run	Sample	Linker	Environment	Rate	CCR	Notes
74975	Empty	G-10	Vacuum	$2K/min$	CCR 50K	
74979	Toluene	G-10	Helium	$2K/min$	CCR $2K/min$	
74982	Toluene	G-10	Helium	$1K/min$	CCR $1K/min$	
74984	Toluene	G-10	Vacuum	$5K/min$	CCR 50K	
74986	Toluene	G-10	Vacuum	$2K/min$	CCR 50K	
74988	Toluene	G-10	Vacuum	$1K/min$	CCR 50K	
74990	Toluene	G-10	Vacuum	$2K/min$	CCR 50K	
74992	Toluene	G-10	Vacuum	$5K/min$	CCR 50K	
74994	Toluene	G-10	Vacuum	$1K/min$	CCR 50K	
74996	Toluene	G-10	Helium	$5K/min$	CCR $5K/min$	
74998	Toluene	G-10	Helium	$2K/min$	CCR $2K/min$	
75000	Toluene	G-10	Helium	$1K/min$	CCR $1K/min$	
75002	Toluene	G-10	Helium	$0.5K/min$	CCR $0.5K/min$	
75004	Toluene	G-10	Helium	$5K/min$	CCR $5K/min$	
75006	Toluene	G-10	Helium	$2K/min$	CCR $2K/min$	
75008	Toluene	G-10	Helium	$1K/min$	CCR $1K/min$	
75010	Toluene	G-10	Helium	$0.5K/min$	CCR $0.5K/min$	
75012	Toluene	G-10	Helium	$2K/min$	CCR $50K$	Helium cooled
75015	Toluene	G-10	Vacuum	$2K/min$	CCR $50K$	Helium cooled

A.3 Empty-Empty characterization

Run	Sample	Linker	Environment	Rate	CCR	Notes
75017	Empty	G-10	Helium	$2K/min$	CCR $50K$	
75019	Empty	G-10	Vacuum	$2K/min$	CCR $50K$	
75025	Empty	G-10	Vacuum	$2K/min$	CCR $2K/min$	
75027	Empty	G-10	Vacuum	heaters off	CCR $2K/min$	
75029	Empty	G-10	Helium	$2K/min$	CCR $2K/min$	
75031	Empty	G-10	Helium	heaters off	CCR $2K/min$	
75033	Empty	G-10	Helium	$2K/min$	CCR $2K/min$	CCR $-10K$
75035	Empty	G-10	Helium	heaters off	CCR $2K/min$	CCR $-10K$

A.4 Reversed heaters (Empty, Toluene, Glycerol)

Run	Sample	Linker	Environment	Rate	CCR	Notes
75037	Empty	G-10	Helium	$2K/min$	CCR $2K/min$	
75039	Empty	G-10	Vacuum	$2K/min$	CCR $2K/min$	
75041	Empty	G-10	Vacuum	$2K/min$	CCR $2K/min$	CCR -10
75058	Empty	G-10	Helium	$2K/min$	CCR $50K$	
75069	Empty	G-10	Helium	$2K/min$	CCR $2K/min$	
75083	Empty	G-10	Helium	heaters off	CCR $2K/min$	
75095	Empty	G-10	Vacuum	heaters off	CCR $2K/min$	
75097	Empty	G-10	Vacuum	$2K/min$	CCR $2K/min$	CCR -10
75099	Toluene	G-10	Helium	$5K/min$	CCR $5K/min$	CCR -10
75101	Toluene	G-10	Helium	$2K/min$	CCR $2K/min$	CCR -10
75103	Toluene	G-10	Helium	heaters off	CCR $2K/min$	
75105	Toluene	G-10	Helium	heaters off	CCR $4K/min$	
75107	Toluene	G-10	Helium	$5K/min$	CCR $5K/min$	CCR -10
75109	Toluene	G-10	Helium	$2K/min$	CCR $2K/min$	CCR -10
75111	Toluene	G-10	Helium	heaters off	CCR $2K/min$	
75113	Toluene	G-10	Helium	heaters off	CCR $4K/min$	
75115	Toluene	G-10	Helium	$5K/min$	CCR $5K/min$	CCR -10
75117	Toluene	G-10	Helium	$2K/min$	CCR $2K/min$	CCR -10
75119	Toluene	G-10	Helium	heaters off	CCR $2K/min$	
75121	Toluene	G-10	Helium	heaters off	CCR $4K/min$	
75123	Toluene	G-10	Vacuum	$2K/min$	CCR $2K/min$	CCR -10
75125	Glycerol	G-10	Helium	$5K/min$	CCR $5K/min$	CCR -10, only $1.4g$ of sample
75127	Glycerol	G-10	Helium	$2K/min$	CCR $2K/min$	CCR -10, only $1.4g$ of sample
75117	Glycerol	G-10	Helium	heaters off	CCR $4K/min$	only $1.4g$ of sample

A.5 Two heaters per can (Toluene, Glycerol, attempts with steps)

Number of run	Sample	Linker	Environment	Rate	CCR	Notes
75131	Toluene	G-10	Helium	$5K/min$	CCR $50K$	
75133	Toluene	G-10	Helium	$2K/min$	CCR $50K$	
75135	Toluene	G-10	Helium	$1K/min$	CCR $50K$	
75137	Toluene	G-10	Helium	$0.5K/min$	CCR $50K$	
75141	Toluene	G-10	Helium	$5K/min$	CCR $50K$	
75143	Toluene	G-10	Helium	$5K/min$	CCR $5K/min$	
75145	Toluene	G-10	Helium	$2K/min$	CCR $2K/min$	
75147	Toluene	G-10	Helium	$1K/min$	CCR $1K/min$	
75149	Toluene	G-10	Helium	$0.5K/min$	CCR $0.5K/min$	
75151	Toluene	G-10	Vacuum	$2K/min$	CCR $2K/min$	Helium cooled
75153	Toluene	G-10	Helium	$1K/min$	CCR $1K/min$	CCR -10
75155	Toluene	G-10	Helium	$1K/min$	CCR $1K/min$	CCR -5
75157	Toluene	G-10	Helium	$2K/min$	CCR $2K/min$	CCR -5
75159	Toluene	G-10	Helium	steps ramp $5k/min$	steps ramp $5k/min$	
75165	Toluene	G-10	Helium	steps ramp $5k/min$	steps ramp $5k/min$	CCR -10
75220	Glycerol	G-10	Helium	$5K/min$	CCR $5K/min$	
75222	Glycerol	G-10	Helium	$2K/min$	CCR $2K/min$	
75228	Glycerol	G-10	Helium	$5K/min$	CCR $5K/min$	
75230	Glycerol	G-10	Helium	$2K/min$	CCR $2K/min$	
75232	Glycerol	G-10	Helium	$1K/min$	CCR $1K/min$	
75234	Glycerol	G-10	Helium	$0.5K/min$	CCR $0.5K/min$	

A.6 Two heaters per can steps separating data collection

The steps are every $10K$ and the waiting time is of $10min$

Number of run	Sample	Linker	Environment	Type of data	Notes
75167					
75169					
75171					
75173					
75175					
75177					
75179					
75181					
75183					
75185					
75187					
75189	Toluene	G-10	Helium	Thermodynamic data (before the step)	CCR -10
75191					
75193					
75195					
75197					
75199					
75201					
75203					
75205					
75207					
75209					
75211					
75213					
75215					
75217					

Number of run	Sample	Linker	Environment	Type of data	Notes
75168					
75170					
75172					
75174					
75176					
75178					
75180					
75182					
75184					
75186					
75188					
75190					
75192					
75194	Toluene	G-10	Helium	Plateau	CCR −10
75196					
75198					
75200					
75202					
75204					
75206					
75208					
75210					
75212					
75214					
75216					
75218					

APPENDIX B

Python scripts

The script for data extractions allows to get data from IRIS and prepares them in a file the can be used with plotting programs. The other script can be used, after the one for data extaction, to post-process data. It does a moving average to get rid of noise and generates all the plots showed in this report. The first one is made to work with Mantid, while the second can be used both with Mantid or alone.
There are two versions of both because of the different setups used.

B.1 Data extraction for one-heaters-per-can setup

```
from mantid import config
from scipy.interpolate import interp1d


instrum = 'IRIS'
runs = []
path = 'C:/Users/mnu25638/Documents/Calor'
config['default.instrument'] = '%s' % instrum

############################################################
def TableOut(t):
    """Writes out tables as tab separated .dat file for
     ↪  other graphing programmes"""
    runname=str(runno)
    if not os.path.exists('%s/%s' % (path, runname)):
        os.makedirs('%s/%s' % (path, runname))

    header = mtd[t].getColumnNames()
    FileOut = open('%s/%s/interp.txt' % (path, runname),
     ↪  'w')
    for column in range(mtd[t].columnCount()):
        title = header[column]
        FileOut.write(str(title))
        FileOut.write('\t')
```

```
    FileOut.write('\n')
    for row in range(mtd[t].rowCount()):
        for column in range(mtd[t].columnCount()):
            value = mtd[t].cell(row, column)
            FileOut.write(str(value))
            FileOut.write('\t')
        FileOut.write('\n')
    FileOut.close()
    print "File written: %s/%s/interp.txt" % (path,
     ↪  runname)
##########################################################


for runno in runs:
    print "Loading data..."
    ws = Load('%s' % runno)
    run = ws.getRun()
    types = ['RT','RB','ST','SB','RH','SH',
     ↪  'CL','CR','TLCCR']
    type_names = {'RT' : 'Ref_top','RB' : 'Ref_bot','ST'
     ↪  : 'Sample_top','SB' : 'sample_bot','RH' :
     ↪  'Ref_heater_output','SH' :
     ↪  'Sample_heater_output','CL' :
     ↪  'CURRENT_LEFT','CR' : 'CURRENT_RIGHT','TLCCR' :
     ↪  'TLCCR'}


    print "Setting interpolation settings from
     ↪  Ref_top..."
    # create array from Ref_top
    RT = ExportTimeSeriesLog(InputWorkspace = 'ws',
     ↪  LogName='Ref_top', IsEventWorkspace=False)
    x = RT.readX(0)
    y = RT.readY(0)
    firstdata = np.array([x,y])
    firsttime = np.array(firstdata[0:1, 0:].flatten(),
     ↪  dtype=float)
    firstval= np.array(firstdata[0:,1:].flatten(),
     ↪  dtype=float)

    # Using time limits to create the adjusted time
     ↪  axis, with same number of points, but linearly
     ↪  spaced
    start = np.amin(firsttime)
    end = np.amax(firsttime)
    points = np.shape(firsttime)[0]*5
    adj_time = np.linspace(start, end, points,
     ↪  endpoint=True)
    #print adj_time
```

```
    print "Setting up table..."
    vallist=[]
    ianterp = CreateEmptyTableWorkspace()
    ianterp.addColumn(type = "float", name = "time",
     ↪  plottype = 1)

    for type in types:
        print "Adding %s column" % type_names[type]
        ianterp.addColumn(type = "float", name = "%s" %
         ↪  type, plottype = 2)
        logdata = ExportTimeSeriesLog(InputWorkspace =
         ↪  'ws', LogName=type_names[type],
         ↪  IsEventWorkspace=False)
        x = logdata.readX(0)
        y = logdata.readY(0)
        data = np.array([x,y])
    #    print np.shape(data)

        print "Interpolating %s data" % type_names[type]
        interp_linear = interp1d(x, y, kind='linear',
         ↪  bounds_error=False, fill_value=0)
        interp_val = interp_linear(adj_time)
        vallist.append(interp_val)
        RenameWorkspace(InputWorkspace = 'logdata',
         ↪  OutputWorkspace=type)
    #    print vallist

    print "Filling table:"
    for pt in range(points):
#        print "row %s" % pt
        newrow =[]
        newrow.append(adj_time[pt])
        for foo in range(len(types)):
            param = vallist[foo][pt]
            newrow.append(param)
        ianterp.addRow(newrow)

    print "Writing output..."

    TableOut('ianterp')

    print "Done!"
```

B.2 Data extraction for two-heaters-per-can setup

```
from mantid import config
from scipy.interpolate import interp1d


instrum = 'IRIS'
```

```
runs = []
path = 'C:/Users/mnu25638/Documents/Calor'
config['default.instrument'] = '%s' % instrum

##############################################################
    """Writes out tables as tab separated .dat file for
     ↪  other graphing programmes"""
    runname=str(runno)
    if not os.path.exists('%s/%s' % (path, runname)):
        os.makedirs('%s/%s' % (path, runname))

    header = mtd[t].getColumnNames()
    FileOut = open('%s/%s/interp.txt' % (path, runname),
     ↪  'w')
    for column in range(mtd[t].columnCount()):
        title = header[column]
        FileOut.write(str(title))
        FileOut.write('\t')
    FileOut.write('\n')
    for row in range(mtd[t].rowCount()):
        for column in range(mtd[t].columnCount()):
            value = mtd[t].cell(row, column)
            FileOut.write(str(value))
            FileOut.write('\t')
        FileOut.write('\n')
    FileOut.close()
    print "File written: %s/%s/interp.txt" % (path,
     ↪  runname)
##############################################################


for runno in runs:
    print "Loading data..."
    ws = Load('%s' % runno)
    run = ws.getRun()
    types = ['RT','RB','ST','SB','RTH','RBH','STH',
     ↪  'SBH', 'CST', 'CSB','CRT','CRB','TLCCR']
    type_names = {'RT' : 'Ref_top','RB' : 'Ref_bot','ST'
     ↪  : 'Sample_top','SB' : 'sample_bot','RTH' :
     ↪  'Ref_heater_top_output','RBH' :
     ↪  'Ref_heater_bot_output','STH' :
     ↪  'Sample_heater_top_output','SBH' :
     ↪  'Sample_heater_bot_output','CST' :
     ↪  'CURRENT_ST','CSB' : 'CURRENT_SB','CRT' :
     ↪  'CURRENT_RT','CRB' : 'CURRENT_RB','TLCCR' :
     ↪  'TLCCR'}


    print "Setting interpolation settings from
     ↪  Ref_top..."
```

```
# create array from Ref_top
RT = ExportTimeSeriesLog(InputWorkspace = 'ws',
 ↪  LogName='Ref_top', IsEventWorkspace=False)
x = RT.readX(0)
y = RT.readY(0)
firstdata = np.array([x,y])
firsttime = np.array(firstdata[0:1, 0:].flatten(),
 ↪  dtype=float)
firstval= np.array(firstdata[0:,1:].flatten(),
 ↪  dtype=float)

# Using time limits to create the adjusted time
 ↪  axis, with same number of points, but linearly
 ↪  spaced
start = np.amin(firsttime)
end = np.amax(firsttime)
points = np.shape(firsttime)[0]*5
adj_time = np.linspace(start, end, points,
 ↪  endpoint=True)
#print adj_time

print "Setting up table..."
vallist=[]
ianterp = CreateEmptyTableWorkspace()
ianterp.addColumn(type = "float", name = "time",
 ↪  plottype = 1)

for type in types:
    print "Adding %s column" % type_names[type]
    ianterp.addColumn(type = "float", name = "%s" %
     ↪  type, plottype = 2)
    logdata = ExportTimeSeriesLog(InputWorkspace =
     ↪  'ws', LogName=type_names[type],
     ↪  IsEventWorkspace=False)
    x = logdata.readX(0)
    y = logdata.readY(0)
    data = np.array([x,y])
#    print np.shape(data)

    print "Interpolating %s data" % type_names[type]
    interp_linear = interp1d(x, y, kind='linear',
     ↪  bounds_error=False, fill_value=0)
    interp_val = interp_linear(adj_time)
    vallist.append(interp_val)
    RenameWorkspace(InputWorkspace = 'logdata',
     ↪  OutputWorkspace=type)
#    print vallist

print "Filling table:"
for pt in range(points):
```

```
#         print "row %s" % pt
        newrow =[]
        newrow.append(adj_time[pt])
        for foo in range(len(types)):
            param = vallist[foo][pt]
            newrow.append(param)
        ianterp.addRow(newrow)

    print "Writing output..."

    TableOut('ianterp')

    print "Done!"
```

B.3 Moving averages and graphs for one-heaters-per-can setup

```
import numpy as np
import math as m
import matplotlib.pyplot as plt
import pylab
import os



###### This code needs the interp files that you can
 ↪  extract by using  Mantidplot and the code
 ↪  TimeSeries.py #########



def extract_columns():

        lines=0

        inputfile=open('%s/%s/interp.txt' % (path,
         ↪  runname),'r')
        time,rt,rb,st,sb,rh,sh,cl,cr,ccr=np.loadtxt(inputfile,dtype='float',
         ↪  delimiter='        ',skiprows=1,
         ↪  usecols=(0,1,2,3,4,5,6,7,8,9),
         ↪  unpack=True,ndmin=0)

        #print time,rt,rb,st,sb,rh,sh,cl,cr,ccr

        return time,rt,rb,st,sb,rh,sh,cl,cr,ccr

def Moving_average(data, subset):
        data_avg=[]
        for i in range(len(data)-subset+1):
                data_avg.append(sum(data[l] for l in
                 ↪  range(i,i+subset-1))/subset)
```

```
        return data_avg


def All_temp_graph():

        plt.xlabel('Time (s)')
        plt.ylabel('Temperature (K)')
        #plt.title('')
        r_t,=plt.plot(time,ref_top)
        r_b,=plt.plot(time,ref_bot)
        s_t,=plt.plot(time,samp_top)
        s_b,=plt.plot(time,samp_bot)
        tlccr,=plt.plot(time,TLCCR)

        plt.legend([r_t,r_b,s_t,s_b,tlccr],['Reference
         ↪  top','Reference bottom', 'Sample top',
         ↪  'Sample bottom','TLCCR'])

        plt.savefig('%s/%s/All_temperatures_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def All_currents():

        plt.xlabel('Temperature (K)')
        plt.ylabel('Current (A)')
        #plt.title('')
        c_l,=plt.plot(ref_top,curr_left)
        c_r,=plt.plot(ref_top,curr_right)
        #plt.axis([75,270,0.08,0.35])
        plt.legend([c_l,c_r],['Reference
         ↪  current','Sample current'])

        plt.savefig('%s/%s/All_currents_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def All_heaters():

        plt.xlabel('Temperature (K)')
        plt.ylabel('Heater Output (%)')
        #plt.title('')
        r_h,=plt.plot(ref_top,ref_heat)
        s_h,=plt.plot(ref_top,samp_heat)
        #plt.axis([75,270,0.08,0.35])
```

```
        plt.legend([r_h,s_h],['Reference heater','Sample
         ↪  heater'])

        plt.savefig('%s/%s/All_heaters_%s.png'
         ↪  %(path2,runname,runname),
         ↪  dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()


def Temp_subtraction():

        temp_sub=[]
        for i in
         ↪  range(min(len(samp_bot_avg),len(ref_top_avg))):
                temp_sub.append(samp_bot_avg[i] -
                 ↪  ref_top_avg[i])

        plt.xlabel('Reference top (K)')
        plt.ylabel('Sample bottom - Reference top (K)')
        plt.title('Temperature subtraction')

        t_sub,=plt.plot(ref_top_avg,temp_sub)
        #plt.axis([70,270,-0.6,0.2])


        plt.savefig('%s/%s/Temp_sub_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def Curr_subtraction():

        curr_sub=[]
        for i in
         ↪  range(min(len(curr_left),len(curr_right))):
                curr_sub.append(curr_right[i] -
                 ↪  curr_left[i])

        plt.xlabel('Reference top (K)')
        plt.ylabel('Sample current - Reference current
         ↪  (A)')
        plt.title('Current subtraction')

        c_sub,=plt.plot(ref_top,curr_sub)
        #plt.axis([70,270,-0.075,0.05])


        plt.savefig('%s/%s/Curr_sub_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
```

```
        plt.close()
        #plt.show()

def Heat_subtraction():

        heat_sub=[]
        for i in
         ↪  range(min(len(samp_heat),len(ref_heat))):
                heat_sub.append(samp_heat[i] -
                 ↪  ref_heat[i])

        plt.xlabel('Reference top (K)')
        plt.ylabel('Sample heater-Reference heater (%)')
        plt.title('Heat subtraction')

        h_sub,=plt.plot(ref_top,heat_sub)
        #plt.axis([75,270,-0.5,1])


        plt.savefig('%s/%s/Heat_sub_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()


def Gradient():

        grad_s=[]
        grad_r=[]
        for i in range(len(samp_bot_avg)):
                grad_s.append(samp_bot_avg[i] -
                 ↪  samp_top_avg[i])
                grad_r.append(ref_top_avg[i] -
                 ↪  ref_bot_avg[i])


        plt.xlabel('Reference top (K)')
        plt.ylabel('Gradient (K)')
        plt.title('Gradient along the can')

        r_g,=plt.plot(ref_top_avg,grad_r)
        s_g,=plt.plot(ref_top_avg,grad_s)
        #plt.axis([75,270,0.08,0.35])
        plt.legend([r_g,s_g],['Reference','Sample'])


        plt.savefig('%s/%s/Gradient_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()
```

```
runnames=[]
path= 'C:/Users/mnu25638/Documents/Calor'
path2='C:/Users/mnu25638/Desktop/Graph'
runfile= open('%s/run.txt' % (path),'r')
runnames=runfile.read().splitlines()
print runnames
for line in range(len(runnames)):
        runname=runnames[line]

        inputfile=open('%s/%s/interp.txt' % (path,
         ↪  runname),'r')
        print '\nFile correctly opened, run', runname
        print'Extracting data of run',runname

        time,ref_top,ref_bot,samp_top,samp_bot,ref_heat,samp_heat,curr_left,curr_r

        print'All columns extracted'
        ###### Moving Averages on Temperatures #######

        print 'Averaging Temperature'

        ref_top_avg=Moving_average(ref_top, 100)
        ref_bot_avg=Moving_average(ref_bot, 100)

        samp_top_avg=Moving_average(samp_top, 100)
        samp_bot_avg=Moving_average(samp_bot, 100)
        print 'Reference top, Reference bottom,Sample
         ↪  top, Sample bottom have been averaged'
        if not os.path.exists('%s/%s' % (path2,
         ↪  runname)):
                os.makedirs('%s/%s' % (path2, runname))

###### Generating Graphs ########

        print 'Generating the graphs'

        All_temp_graph()
        All_currents()
        All_heaters()
        Temp_subtraction()
        Curr_subtraction()
        Heat_subtraction()
        Gradient()



        inputfile.close()
runfile.close()
```

B.4 Moving averages and graphs for two-heaters-per-can setup

```
import numpy as np
import math as m
import matplotlib.pyplot as plt
import pylab
import os



###### This code needs the interp files that you can
 ↪  extract by using  Mantidplot and the code
 ↪  TimeSeries.py #########



def extract_columns():

        lines=0

        inputfile=open('%s/%s/interp.txt' % (path,
         ↪  runname),'r')
        time,rt,rb,st,sb,rth,rbh,sth,sbh,cst,csb,crt,crb,ccr=
         ↪  np.loadtxt(inputfile, dtype='float',
         ↪  delimiter='        ', skiprows=30,
         ↪  usecols=(0,1,2,3,4,5,6,7,8,9,10,11,12,13),
         ↪  unpack=True, ndmin=0)

        #print time,rt,rb,st,sb,rh,sh,cl,cr,ccr

        return
         ↪  time,rt,rb,st,sb,rth,rbh,sth,sbh,cst,csb,crt,crb,ccr

def Moving_average(data, subset):
        data_avg=[]
        for i in range(len(data)-subset+1):
                data_avg.append(sum(data[l] for l in
                 ↪  range(i,i+subset-1))/subset)

        return data_avg



def All_temp_graph():

        plt.xlabel('Time (s)')
        plt.ylabel('Temperature (K)')
        #plt.title('')
        r_t,=plt.plot(time,ref_top)
        r_b,=plt.plot(time,ref_bot)
```

```
        s_t,=plt.plot(time,samp_top)
        s_b,=plt.plot(time,samp_bot)
        tlccr,=plt.plot(time,TLCCR)

        plt.legend([r_t,r_b,s_t,s_b,tlccr],['Reference
         ↪  top','Reference bottom', 'Sample top',
         ↪  'Sample bottom','TLCCR'],loc='lower right')

        plt.savefig('%s/%s/All_temperatures_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def All_currents():

        plt.xlabel('Temperature (K)')
        plt.ylabel('Current (A)')
        #plt.title('')
        c_sb,=plt.plot(ref_top,curr_sb)
        c_st,=plt.plot(ref_top,curr_st)
        c_rb,=plt.plot(ref_top,curr_rb)
        c_rt,=plt.plot(ref_top,curr_rt)
                #plt.axis([75,270,0.08,0.35])
        plt.legend([c_rt,c_rb,c_st,c_sb],['Reference top
         ↪  current','Reference bottom current','Sample
         ↪  top current','Sample bottom
         ↪  current'],loc='lower right')

        plt.savefig('%s/%s/All_currents_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def Ref_currents():

        plt.xlabel('Temperature (K)')
        plt.ylabel('Current (A)')
        #plt.title('')
        c_rb,=plt.plot(ref_top,curr_rb)
        c_rt,=plt.plot(ref_top,curr_rt)
                #plt.axis([75,270,0.08,0.35])
        plt.legend([c_rt,c_rb],['Reference top
         ↪  current','Reference bottom current'])

        plt.savefig('%s/%s/Ref_currents_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def Sample_currents():
```

```
        plt.xlabel('Temperature (K)')
        plt.ylabel('Current (A)')
        #plt.title('')
        c_sb,=plt.plot(ref_top,curr_sb)
        c_st,=plt.plot(ref_top,curr_st)
        #plt.axis([75,270,0.08,0.35])
        plt.legend([c_st,c_sb],['Sample top
         ↪  current','Sample bottom current'])

        plt.savefig('%s/%s/Sample_currents_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def Currents_sum():

        plt.xlabel('Temperature (K)')
        plt.ylabel('Current (A)')
        #plt.title('')
        c_s,=plt.plot(ref_top,curr_sb+curr_st)
        c_r,=plt.plot(ref_top,curr_rb+curr_rt)
        #plt.axis([75,270,0.08,0.35])
        plt.legend([c_r,c_s],['Reference
         ↪  currents','Sample currents'])

        plt.savefig('%s/%s/Currents_sum_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()


def All_heaters():

        plt.xlabel('Temperature (K)')
        plt.ylabel('Heater Output (%)')
        #plt.title('')
        r_h_t,=plt.plot(ref_top,ref_top_heat)
        r_h_b,=plt.plot(ref_top,ref_bot_heat)
        s_h_t,=plt.plot(ref_top,samp_top_heat)
        s_h_b,=plt.plot(ref_top,samp_bot_heat)
        #plt.axis([75,270,0.08,0.35])
        plt.legend([r_h_t,r_h_b,s_h_t,s_h_b],['Reference
         ↪  top heater','Reference bottom
         ↪  heater','Sample top heater', 'Sample bottom
         ↪  heater'],loc='lower right')

        plt.savefig('%s/%s/All_heaters_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
```

```
        #plt.show()

def Ref_heaters():

        plt.xlabel('Temperature (K)')
        plt.ylabel('Heater Output (%)')
        #plt.title('')
        r_h_t,=plt.plot(ref_top,ref_top_heat)
        r_h_b,=plt.plot(ref_top,ref_bot_heat)

        #plt.axis([75,270,0.08,0.35])
        plt.legend([r_h_t,r_h_b],['Reference top
         ↪  heater','Reference bottom heater'])

        plt.savefig('%s/%s/Ref_heaters_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def Sample_heaters():

        plt.xlabel('Temperature (K)')
        plt.ylabel('Heater Output (%)')
        #plt.title('')
        s_h_t,=plt.plot(ref_top,samp_top_heat)
        s_h_b,=plt.plot(ref_top,samp_bot_heat)
        #plt.axis([75,270,0.08,0.35])
        plt.legend([s_h_t,s_h_b],['Sample top heater',
         ↪  'Sample bottom heater'])

        plt.savefig('%s/%s/Samp_heaters_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def Heaters_sum():

        plt.xlabel('Temperature (K)')
        plt.ylabel('Heater Output (%)')
        #plt.title('')
        r_h,=plt.plot(ref_top,ref_top_heat+ref_bot_heat)
        s_h,=plt.plot(ref_top,samp_top_heat+samp_bot_heat)

        #plt.axis([75,270,0.08,0.35])
        plt.legend([r_h,s_h],['Reference
         ↪  heaters','Sample heaters'])

        plt.savefig('%s/%s/Heaters_sum_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
```

```
        #plt.show()

def Temp_subtraction():

        temp_sub=[]
        for i in
         ↪  range(min(len(samp_bot_avg),len(ref_top_avg))):
                temp_sub.append((samp_bot_avg[i]) -
                 ↪  (ref_top_avg[i]))

        plt.xlabel('Reference top (K)')
        plt.ylabel('Sample bottom - Reference top (K)')
        plt.title('Temperature subtraction')

        t_sub,=plt.plot(ref_top_avg,temp_sub)
        plt.axis([75,270,-0.04,0.06])


        plt.savefig('%s/%s/Temp_sub_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def Curr_subtraction():

        curr_sub=[]
        for i in range(min(len(curr_sb),len(curr_rb))):
                curr_sub.append((curr_sb[i]+curr_st[i])
                 ↪  - (curr_rt[i]+curr_rb[i]))

        plt.xlabel('Reference top (K)')
        plt.ylabel('Sample current - Reference current
         ↪  (A)')
        plt.title('Current subtraction')

        c_sub,=plt.plot(ref_top,curr_sub)
        plt.axis([75,270,-0.02,0.0])


        plt.savefig('%s/%s/Curr_sub_%s.png'
         ↪  %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()

def Heat_subtraction():

        heat_sub=[]
        for i in
         ↪  range(min(len(samp_top_heat),len(ref_top_heat))):
```

```
                heat_sub.append((samp_top_heat[i]+samp_bot_heat[i])
                 ↪ - (ref_bot_heat[i]+ref_top_heat[i]))

        plt.xlabel('Reference top (K)')
        plt.ylabel('Sample heater-Reference heater (%)')
        plt.title('Heat subtraction')

        h_sub,=plt.plot(ref_top,heat_sub)
        plt.axis([75,270,-5,0])


        plt.savefig('%s/%s/Heat_sub_%s.png'
         ↪ %(path2,runname,runname),dpi=200,figsize=(8,5))
        plt.close()
        #plt.show()


def Gradient():

        grad_s=[]
        grad_r=[]
        for i in range(len(samp_bot_avg)):
                grad_s.append(samp_bot_avg[i] -
                 ↪ samp_top_avg[i])
                grad_r.append(ref_top_avg[i] -
                 ↪ ref_bot_avg[i])


        plt.xlabel('Reference top (K)')
        plt.ylabel('Gradient (K)')
        plt.title('Gradient along the can')

        r_g,=plt.plot(ref_top_avg,grad_r)
        s_g,=plt.plot(ref_top_avg,grad_s)
        #plt.axis([75,270,0.08,0.35])
        plt.legend([r_g,s_g],['Reference','Sample'])


        plt.savefig('%s/%s/Gradient_%s.png'
         ↪ %(path2,runname,runname),dpi=200,figsize=(8,5))

        #plt.show()
    #plt.close()

runnames=[]
path= 'C:/Users/Utente/Desktop'
path2='C:/Users/Utente/Desktop'
runfile= open('%s/run.txt' % (path),'r')
runnames=runfile.read().splitlines()
print runnames
```

```
for line in range(len(runnames)):
        runname=runnames[line]

        inputfile=open('%s/%s/interp.txt' % (path,
         ↪  runname),'r')
        print '\nFile correctly opened, run', runname
        print'Extracting data of run',runname

        time,ref_top,ref_bot,samp_top,samp_bot,ref_top_heat,ref_bot_heat,sam

        print'All columns extracted'
        ###### Moving Averages on Temperatures #######

        print 'Averaging Temperature'

        ref_top_avg=Moving_average(ref_top, 100)
        ref_bot_avg=Moving_average(ref_bot, 100)

        samp_top_avg=Moving_average(samp_top, 100)
        samp_bot_avg=Moving_average(samp_bot, 100)
        print 'Reference top, Reference bottom,Sample
         ↪  top, Sample bottom have been averaged'
        if not os.path.exists('%s/%s' % (path2,
         ↪  runname)):
                os.makedirs('%s/%s' % (path2, runname))


###### Generating Graphs ########

        print 'Generating the graphs'

        All_temp_graph()
        All_currents()
        Ref_currents()
        Sample_currents()
        Currents_sum()
        All_heaters()
        Ref_heaters()
        Sample_heaters()
        Heaters_sum()
        Temp_subtraction()
        Curr_subtraction()
        Heat_subtraction()
        Gradient()



        inputfile.close()
runfile.close()
```