TECHNISCHE UNIVERSITÄT WIEN

European Research Council
Established by the European Commission

DIPLOMARBEIT

# Cuprates: Electrical Resistivity under Uniaxial Pressure

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Technische Physik**

eingereicht von

**Benjamin Klebel**
Matrikelnummer 01125855

ausgeführt am Institut für Festkörperphysik
der Fakultät für Physik der Technischen Universität Wien

Betreuung
Betreuer: Associate Prof. Dr. Neven Barišić
Mitwirkung: Projektass. Wojciech Tabiś PhD
Projektass. Dr.rer.nat. Naveen Chogondahalli

Wien, Thursday 16th May, 2019

(Unterschrift Verfasser) (Unterschrift Betreuer)

# Abstract

The discovery of superconductivity in 1911 and the subsequent development of technologies allowing practical applications brought important scientific and technological advancements. In this regard, the ultimate goal is to find a superconductor which is easy to use at room-temperature. High temperature superconductivity (maximum $T_c$ $\sim$133 K) as found in copper oxides ("cuprates") was a big step in this direction. However, despite great efforts to understand cuprates, no theory which captures all defining characteristics of these high-transition-temperature (high-$T_c$) superconductors has found wide acceptance in the scientific community. A deeper understanding of those materials, ultimately leading to a microscopic theory of high temperature superconductivity could substantially quicken technological advances, which would hopefully bring benefit to society.

Recent experimental observations sparked a phenomenological model of superconductivity in cuprates. According to this model, one charge carrier being localised per unit cell mediates the pairing interaction required for superconductivity. As the localization of the one carrier emerges in a spatially inhomogeneous manner, and the Fermi surface evolution exhibits strongly anisotropic behaviour, the experimental observations and the suggested model imply an importance of certain details of the crystal structure. In particular, there appears to be a strong coupling between charge and lattice degrees of freedom. Therefore, the process which leads to pairing likely depends on the crystal structure symmetry (and related local distortions). More insight into the response of the electronic system on structural distortions in cuprates is therefore essential for an extension of the phenomenological model to a microscopic description of cuprate superconductivity. In this regard, the work conducted within the scope of this thesis addresses one of the most urgent questions of solid state physics.

Cuprates are comprised of a variety of different compounds with complex structural differences. The effect of structure could be studied by a comparison between compounds. However, those variations are not systematic. Therefore, we applied uniaxial pressure to tune the crystal symmetry in a clean and continuous way, independently of other parameters. We have studied $Nd_{2-x}Ce_xCuO_4$ – NCCO.

The main challenge was the development of a uniaxial pressure cell. Within the framework of this thesis, the author contributed to the development of spring- and gas-driven cells. In particular the gas-driven cell is a major experimental achievement: It enables *in-situ* changes of the pressure and a temperature-independent measurement of the applied force. Furthermore, a cryostat was restored and software tools for automatic measurement control were developed, enabling measurements of electrical resistivity over extended ranges of temperature and magnetic field. The spring-driven cell was integrated into this setup, allowing measurements of the electrical resistivity under uniaxial pressure down to liquid helium temperatures.

With these setups, we measured the electrical resistivity at room temperature as a function of uniaxial pressure. Furthermore, we measured the electrical resistivity under uniaxial pressure as a function of temperature in the vicinity of the transition temperature $T_c$. The resistivity measured perpendicular to the direction of the compressive force is found to decrease at room temperature upon applying uniaxial pressure. Moreover, $T_c$ was observed to decrease with increased uniaxial pressure.

The developed experimental technique is expected to enable further results and development of complementary techniques that will help to advance the understanding of cuprate superconductors and possibly other materials, in the future.

# Acknowledgements

I want to thank my supervisor Prof. Neven Barišić for his continued support, starting with my work in his lab as a Bachelor's student, up to now, at the end of my master programme. Many late night discussions with him, about physics and the world in general, broadened my horizon on quite a few levels, for which I am very grateful. Furthermore, I want to thank my co-supervisor Wojciech Tabiś for his advice and guidance in all matters associated with experiments, as well as his supervision of this manuscript, both of them being of paramount importance for the successful finalisation of this thesis. Additionally, I want to thank fellow members of the research group Izabela Biało, Naveen Chogondahalli and Kristijan Velebit for their support, and many useful discussions.

Studying physics at university can be a tiring process – I deem it therefore all the more important for successful students to encounter not only competent but also friendly and supportive lecturers and tutors. Thomas Schäfer and Prof. Alessandro Toschi have been a bright light in a grey landscape for extended amounts of time during my studies. For me, their supportive way of educating students made quite a difference. Additional notable mentions in this regard must also include Prof. Herbert Balasin, Andreas Ipp, Prof. Karl Svozil and Prof. Ewa Weinmüller.

Research at a university is only possible with a large number of support staff, ranging from institute support staff (for example machine workshop, secretaries, facility support staff) via the cleaning staff to the building infrastructure department GUT (including the gatekeepers for late-night visits to the university). Their contribution might often go unnoticed, however it is invaluable to me, thus I want to thank them here.

Life and studying would be miserable without friends – therefore, I want first and foremost to thank Christian Heissenberger for the time spent together, outside of and within university. However, he was not the only contributor, so I want to mention some friends without whom TU Wien would truly have been a dreary place: Laurids Brandl, Patrick Chalupa, Matthias Ebert, David Erkinger, Thomas Fabian, Matthias Kraihammer, Peter Lindner, Lukas Mennel, Axel Polaczek, Philip Renz, Vinzenz Stummer, Susanne Wagner, Clemens Watzenböck, and Raphaela Wutte. Although life without the university is sometimes severely overrated, living and friendship which is unrelated to physics does help to stay in good (mental) health. Therefore I want to thank Anna Hampl and Katharina Huszar, among others.

Almost last, but very much not least important to mention is my family, who have been vital for my existence, and generally speaking a very important place of support for whatever I might be pursuing. It is a great pleasure to be part of this family, and to draw energy and support from this resource. Most of all I want to thank my brother Thomas Klebel, my father Wilhelm Klebel and my mother Elisabeth Klebel.

Alas, I am unable to express in words my profound gratitude for the many years of friendship and the relationship with my partner, Theresia Knobloch. As such I will not try to define the indefinable lest I diminish the meaning she has brought to my life and as a consequence my work.

# Contents

# List of Abbreviations

| Abbreviation | |
|---|---|
| AF | antiferromagnetism / antiferromagnetic |
| ARPES | angle resolved photoemission spectroscopy |
| BCS | Barden-Cooper-Schrieffer |
| CDW | Charge density wave |
| DAC | diamond anvil cell |
| DFT | density functional theory |
| DMFT | dynamical mean field theory |
| FL | Fermi liquid |
| FSR | Fermi surface reconstruction |
| FWHM | full width at half maximum |
| GPIB | general purpose interface bus (IEEE-488) |
| GUI | graphical user interface |
| HTSC | high temperature superconductor |
| ILM | intelligent level meter |
| ITC | intelligent temperature controller |
| LHC | large hadron collider |
| LR-AF | long-range antiferromagnetism |
| MIR | Mott-Ioffe-Regel limit |
| MRI | magnetic resonance imaging |
| NMR | nuclear magnetic resonance |
| PID | proportional-integral-derivative |
| PD | phase diagram |
| PG | pseudogap |
| PPMS | physical property measurement system |
| QCP | quantum critical point |
| QO | quantum oscillations |
| RIXS | resonant inelastic X-ray scattering |
| RXS | resonant x-ray scattering |
| SC | superconductivity / superconductor |
| SM | strange metal |
| SQUID | superconducting quantum interference device |
| STM | scanning tunnelling microscopy |
| STS | scanning tunnelling spectroscopy |
| UP | uniaxial pressure |
| VI | LabView VI virtual instrument |
| VTI | variable temperature insert |
| XRD | X-ray powder diffraction |
| Bi2212 | $Bi_2Sr_2CaCu_2O_8$ |
| Hg1201 | $HgBa_2CuO_{4+\delta}$ |
| NCCO | $Nd_{2-x}Ce_xCuO_4$ |
| YBCO | $YBa_2Cu_3O_{7-\delta}$ |

Table 0.1: Abbreviations

# List of Figures

# List of Tables

# 1 General Introduction to Superconductivity

Superconductivity was discovered in the laboratories of K. Onnes in 1911 (*1*), following his accomplishment of liquefying helium in 1908 (Nobel Prize for liquefaction in 1913). The dc resistivity of mercury vanished upon decreasing temperature below 4.15 K ($T_c$ of Hg). Hitherto the scientific community was divided by the question of how the resistance in metals would behave at very low temperatures. At absolute zero, it was believed that either all electrons would be strongly bound, turning metals into insulators, or nothing of significance would occur near zero Kelvin, with a resistivity continuing smoothly to either zero, or a finite (low) value.

The Meisssner-Ochsenfeld effect (perfect diamagnetism[1]) was discovered in 1933 (*2*), more than twenty years later. This discovery laid out the grounds to describe superconductivity not as "perfect conductivity", but a new state of matter, distinct from everything that had been known before. A phenomenological description was given by V. L. Ginzburg and L. Landau in 1950 (*3*), by using the theory of second order phase transitions, which was published by L. Landau in 1937 (*4*).

**BCS Theory** Though the discovery of superconductivity and its phenomenological description were separated by almost 40 years, the "BCS"-theory (*5*, *6*), a microscopic theory of superconductivity, was proposed shortly afterwards (1957) by J. Bardeen, L. Cooper and J. R. Schrieffer (Nobel Prize for theoretical description in 1972).

The main ingredient of the BCS theory is the Cooper-pair (after L. Cooper): Due to a small, but net attractive (and in BCS time retarded) force, pairs of electrons overcome the Coulomb repulsion below $T_c$. Those pairs condense to a coherent macroscopic quantum state, where individual particles can not be distinguished from one another. The single coherent quantum state entails a coherent motion of Cooper pairs through the lattice, without any dissipation.

This pairing is responsible for the important feature of superconductors: An energy gap of $2\Delta$ in the density of states, which is centred at the Fermi-level. Dissipation free transport in superconductors can be described by this gap, as scattering processes (below $T_c$) exhibit excitation energies which are smaller than the gap – therefore, they are prohibited.

The discovery of the isotope effect (*7*, *8*) in 1950 provided strong evidence of the importance of electron-phonon coupling: Higher $T_c$'s can be found in materials with lighter isotopes, compared to lower $T_c$'s in heavier materials (*7*). As it became evident that the mass of atoms of a material is an important factor influencing the value of $T_c$, it was deduced that phonons must also be important, since the phonon frequencies depend on the atomic mass. Thus the electron-phonon coupling was found to be the mediator (pairing "glue") of the attractive force between individual electrons in conventional BCS theory.

**High Temperature Superconductors ("HTSCs")** Using BCS theory, N. Ashcroft predicted superconductivity at high temperature for metallic hydrogen already in 1968 (*9*). Recently, following this prediction, experiments performed under extremely high pressures exceeding 150 GPa were conducted and a $T_c$ of 203 K was found in $H_2S$[2] (*10*, *11*).

[1] A diamagnetic material, when exposed to a magnetic field, will itself generate a field, pointing in the opposite direction. The flux density of the generated field depends on the diamagnetic material. In the case of superconductors this relation is 1 : 1: the field is completely (perfectly) counteracted.

[2] $H_2S$ is the original compound – under pressure it transforms into $H_3S$, which in turn becomes superconducting.

The case of cuprates seems to be different. Hitherto, certain material properties were regarded as prohibiting superconductivity: These included magnetism, insulating behaviour at elevated temperatures, oxygen and rare-earth elements as part of the stoichiometry, a low density of electronic states close to the Fermi surface and a low symmetry of the crystal lattice. However, G. Bednorz and K. A. Müller discovered superconductivity in the ceramic material $La_{5-x}Ba_xCu_5O_{5(3-y)}$, 1986 (*12*). In contrast to the mentioned assumptions, the HTSCs with the highest transition temperatures exhibit strong anisotropy (layered structures), insulating behaviour at room temperature and peculiar magnetic properties. The trilayer material $HgBa_2Ca_2Cu_3O_{8+\delta}$ exhibits the highest $T_c$ found so far ($\sim 133\,K$) at ambient pressure.

**Applications and Room Temperature Superconductivity** Many practical applications of superconductivity already exist: One example is commercially available superconducting electromagnets generating magnetic fields in the range of 7–14 T (32 T record) for NMR[3] (MRI[4]) broadly used in physics/chemistry, medicine, and particle accelerators such as the LHC[5] in Switzerland, as well as magnetometers in laboratory equipment.

The ability of a material to carry large current density is jet another property where superconductors greatly outperform other materials. One of the most important limiting factors in applications is the critical current of superconducting materials: Superconductivity is hindered by strong magnetic fields as well as high electrical currents (suppressed at the compound dependent "critical field" and "critical current"). It is not always easy to achieve high critical currents in complex superconductors which exhibit a high $T_c$. Materials with low critical currents (material becomes non-superconducting at low current densities) are therefore impractical for technical applications.

In addition, the practical use of superconducting materials is limited by the need for cryogenics: Most of them are being cooled with liquid helium, even though superconductors with $T_c$'s above 77 K (the boiling point of liquid nitrogen) are available.[6] Cooling to very low temperatures with liquid helium, also enhances the critical current density, as it scales inversely with temperature. It is quite likely that a material which is in its superconducting state at room temperature, would still be cooled, to increase the critical current. However, this is a secondary problem.

To discover superconductivity at room temperature in a material which is easily forged into wires and/or thin films, potentially with a high critical current, would be a major breakthrough. All applications based on such a material would however depend on the specifics of the material itself, which are quite unpredictable as of today. However, in contrast to the twenties of the last century, we are not orders of magnitude away from a $T_c$ at room temperature, as nowadays only a factor of $\sim 2$ is missing. Still, even just to understand how we arrived at this factor poses a great challenge, and the way beyond remains unclear.

In order to help pave this way for such technological advances, it is of utmost importance to decipher the mysteries of cuprate HTSCs.

[3] Nuclear Magnetic Resonance
[4] Magnetic Resonance Imaging
[5] Large Hadron Collider
[6] Even though superconductors with $T_c$'s above 77 K are available for technological use, difficulties related to material properties (they can not be made into wires, only thin films are available) most often weigh heavier than the advantage of liquid nitrogen cooling versus liquid helium cooling.

# 2 Cuprates – A Working Model Description

*Cuprates have been under close scrutiny by the scientific community, since their discovery in 1986. This should not only be attributed to the high $T_c$, but also to the variety of astonishing effects found in this material group. Many of the effects described in this chapter have been subsequently found in other classes of materials (Iron-based pnictides (13, 14), organic materials (15–17), as well as graphene and others (18–20)), however, it is generally agreed, that a theory which can describe cuprates accurately will be a very promising starting point for other materials as well, and vice versa. The following chapter describes the main difficulties the scientific community faces in attempting to understand superconductivity in the cuprates. One possible path to a coherent description of the cuprates will be discussed in detail.*

**Crystal Structure** The principal building block of the cuprates is their $CuO_2$ sheet, laying parallel with the crystallographic *ab*-plane. The $CuO_2$ sheets are physically separated by compound specific "charge-reservoir" layers. The composition of the charge reservoir layers determines the "family" a certain compound belongs to. This entails distinct crystal structures (tetragonal, orthorhombic, ...). The most prominent cases are presented in fig. 2.1.

One important material parameter is doping. Doping is most often achieved by introducing interstitial oxygen into the charge-reservoir layers. Another possibility is that atoms of the

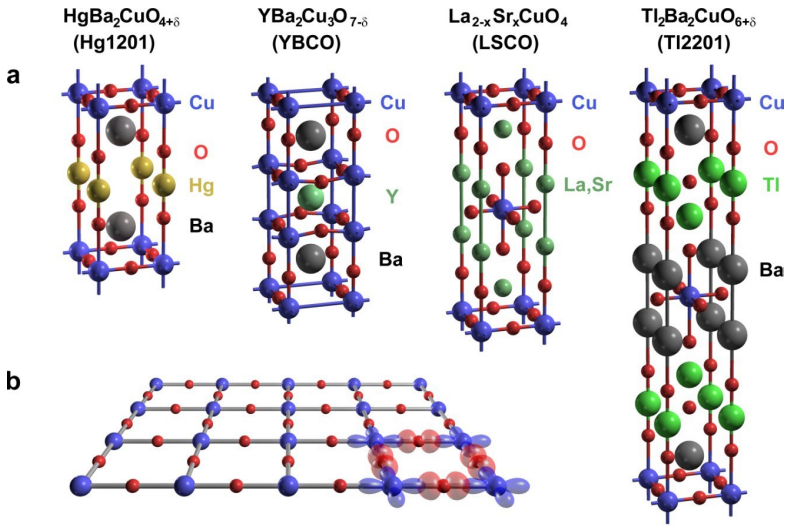



Figure 2.1: (a) Crystal structure of selected families of cuprates. Hg1201 and Tl2201 are hole-doped by interstitial oxygen (in $HgBa_2CuO_{4+\delta}$ in the Hg layer). In LSCO, $La^{3+}$ is exchanged with $Sr^{2+}$ – therefore $p$ (the number of doped holes) equals $x$. In YBCO the oxygen atoms are introduced into the crystal lattice forming Cu-O chains.
The $CuO_2$ planes are the main building block of all cuprates. Depending on the number of $CuO_2$ layers within the primitive unit cell, cuprates might be classified as "single-layer" materials, (one $CuO_2$ plane per unit-cell) and "multilayer" materials (multiple $CuO_2$ layers which are separated by "spacer" atoms – e.g. yttrium in YBCO). (b) The $CuO_2$ layer, with the highlighted electronic orbitals of Cu $3d_{x^2-y^2}$ (blue) and O $2p_\sigma$ ($2p_x$ and $2p_y$ – red). Extracted from (*21*) with kind permission from PNAS.

lattice are substituted, introducing either holes or electrons to the system. It is important to distinguish in-plane substitution, out-of-plane substitution and interstitial doping – all three types have different impacts on the crystal lattice. In-plane substitution signifies a replacement of atoms from the $CuO_2$ layers (*22*). In materials doped through out-of-plane substitution, atoms of the charge-reservoir layers are replaced, which introduces point disorder further away from the *ab*-plane. Doping is called interstitial if no atomic substitution occurs, but additional atoms are introduced between the translationally fixed atomic lattice sites, as is the case with oxygen. As interstitial doping alters the lattice structure comparably far away from the *ab*-plane, it introduces much less disorder directly to them than for example in-plane substitution.

Depending on the atoms being introduced by substitution, both electron-doped and hole-doped compounds can be synthesised. Interstitial doping with oxygen provides two holes in the $CuO_2$ layers per introduced oxygen atom. However, in light of recent findings, it appears that superconductivity in electron doped cuprates can be attributed to the same mechanisms as in hole-doped cuprates (*23*). There, hole pockets, which appear in the same doping range as superconductivity, are very likely responsible for the emerging superconductivity. Thus, we will limit the discussion to the hole-doped cuprates.

## 2.1 The Phase diagram

### 2.1.1 Parent Compound

For physical properties of the cuprates, the relevant electronic orbitals are i) the $3d_{x^2-y^2}$ orbital from copper, and ii) the $2p_x$ and $2p_y$ orbitals of the oxygen atoms.[1] The "parent compounds" (undoped cuprates) are at half filling: there exists an odd number of electrons (1 hole) per $CuO_2$ site. According to band theory, this should be a good conductor (metal). Nevertheless, due to the strong Coulomb interaction between electrons, the cost for two holes occupying the same copper $3d_{x^2-y^2}$ orbital is high. Therefore, one hole is frozen at every $CuO_2$ site. This is expressed in a splitting of the corresponding band[2] in a lower and upper Hubbard band (LHB and UHB in fig. 2.2), which are separated by a gap $U_d \sim 7{-}10\,\mathrm{eV}$. This picture alone (fig. 2.2a) would be best described as a Mott-insulator (*25*). However, $U_d$ is not the smallest energy gap in the undoped cuprates: As observed in x-ray absorption spectroscopy (and shown in fig. 2.2b), the smallest energy gap ($\Delta \sim 2{-}4\,\mathrm{eV}$) for charge excitations in cuprates is actually between the upper Hubbard band, and the oxygen 2p band (*26–28*) – thus, the parent compounds are charge-transfer insulators (*29*).

Due to this Hubbard band splitting, one charge carrier per primitive unit cell is localised by the strong electron correlation.

Since the spin of those localised carriers (holes) is still a dynamical degree of freedom, it causes a magnetic interaction between Cu sites within the $CuO_2$ layers (*ab*-plane). Through virtual hopping and due to the Pauli exclusion principle, a static antiferromagnetic (Néel) order is formed, where the magnetic moments on Cu sites point in alternating directions (*31*, *32*).

[1] Note that in the parent compound, the bands from oxygen 2p orbitals are doubly occupied (filled), and thus do not contribute to electronic transport mechanisms (*24*).

[2] Spectral weight is transferred away from the band at the Fermi level, split into two bands above and below, between which charge excitations can occur.


Figure 2.2: Schematic display of the difference between a Mott-Hubbard insulator (a), where the smallest optical gap ($U_d$) separates upper and lower Hubbard bands (UHB and LHB), and a charge-transfer insulator (b), where the smallest gap ($\Delta$) separates one Hubbard band from a band originating in a distinct electronic orbital. The corresponding orbitals in the CuO-planes of cuprates are the copper $3d_{x^2-y^2}$ (UHB and LHB) and the oxygen 2p (O 2p). This depiction is purely schematic: the exact, doping dependent, relative positions and shapes of the bands are hitherto unknown. Extracted from (*30*).

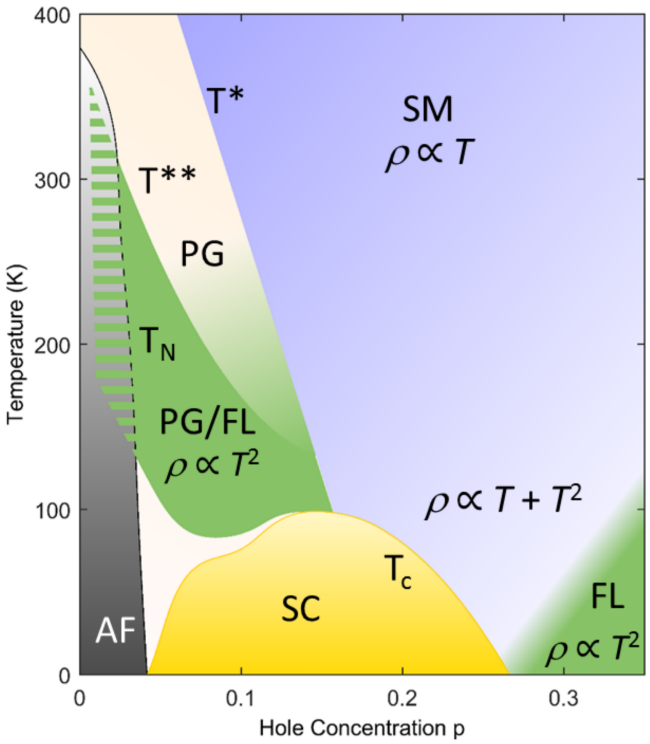



Figure 2.3: Generic phase diagram of the cuprates – the various phases (discussed in the main text) include the anti-ferromagnetic insulating phase (AF), superconductivity (SC), Fermi liquid at high doping levels (FL), strange metal (SM) and pseudogap (PG) at intermediate doping levels. Adapted from (*33*), with kind permission of the authors.

### 2.1.2 Superconducting Phase

Doping the $CuO_2$ planes will initially introduce carriers (holes) to the oxygen 2p orbitals. Upon increasing doping, away from half filling, cuprates become conductors, the long range antiferromagnetic order is destroyed and superconductivity emerges at a doping level $p_{min} \sim 0.05$. With further doping, the transition temperature rises until $p_{opt} \sim 0.16$. This doping level, where the materials exhibit their highest $T_c$, is called "optimal doping", while doping levels below $p_{opt}$ are referred to as "underdoped". All doping levels above $p_{opt}$ are called "overdoped". Superconductivity vanishes at approximately $p_{max} \sim 0.3$.

While the materials may seem to be very distinct from each other in their composition and structure, the effect of doping seems to be similar in all of the cuprates.[3] A good example is the general shape of the superconducting dome, which is ubiquitous in the cuprates.

It is possible that paired electrons/holes are the charge-carriers of superconductivity in the cuprates since one flux quantum corresponds to $h/2e$, the same as in BCS (*34*). However, the exact mechanism of pairing is not understood, though some hints are available: There are indications that the pairing process occurs in the $CuO_2$ planes, and it is established that the corresponding superconducting pairing gap exhibits $d$-wave symmetry, in contrast to an $s$-wave symmetry for conventional BCS superconductors (*35*). "$d$-wave" symmetry refers to the wave-function of the Cooper-pairs, which includes an orbital part. This part can be expanded in spherical harmonics, which leads to an angular momentum quantum number $l = 0, 1, 2, 3, ...$, denoted as $s$-/$p$-/$d$-/$f$-wave, ..., in accordance with naming conventions for atomic orbitals. The $d$-wave symmetry of the energy gap was first established in 1993 (*36*), through magnetic flux modulation measurements on YBCO dc-SQUIDs – for an extensive review refer to ref. (*35*).

The orbital state of the superconducting wave function has serious implications for the momentum dependence of the superconducting gap: Where the wave function changes sign (in momentum space) there is a node in the pairing amplitude, which entails a vanishing superconducting gap. In the scientific community, the terms "antinode" and "node" derived from the d-wave symmetry, are used independently from descriptions of the superconducting pairing scenarios for the directions along the $CuO_2$ bonds, and diagonally to them, respectively[4] (see fig. 2.4).

The superconducting energy gap can be directly measured with angle resolved photoemission (ARPES) measurements (*37*).

### 2.1.3 Fermi Liquid at High Doping

In the heavily overdoped regime, a conventional Fermi liquid (FL) behaviour of the cuprates has been firmly established: The electrical resistivity shows a pronounced quadratic temperature dependence, a strong indicator for a FL (*38*). Also, a large, hole-like Fermi surface was detected by ARPES and quantum oscillation measurements (*39*, *40*). This is in good agreement with the band structure, calculated by local-density approximation techniques (*41*).

[3] Note that not all doping levels can be reached in all compounds, due to the intrinsic structural behaviour of certain compounds. For example, highly overdoped samples of YBCO can not be synthesized. Oxygen doping enters into oxygen chains and this limits the maximum reachable doping of YBCO. Without involving additional layers or substitution, the doping can not be increased since all chain-sites are occupied. Hitherto non-superconducting Hg1201 has not been reported to have been synthesised.

[4] $(\pi, 0)/(0, \pi)$ and $(\pi, \pi)$ directions in momentum space.

### 2.1.4 "Strange Metal"

At intermediate doping, the cuprates exhibit a regime which was named "bad" or "strange" metal, as depicted in fig. 2.3. One distinction from normal metals stems from the unusual behaviour of the resistivity's temperature dependence:

In the simplest metals, the temperature dependence of the resistivity is usually discussed in terms of defects and phonons (Bloch-Grüneisen law), while a contribution from electron-electron interaction is negligibly small. Defects contribute a constant (in temperature) term, phonons a term which is linear at $T > \Omega_D$, where $\Omega_D$ is the Debye-temperature ($\rho_{phonons} \sim T$). The phonon contribution vanishes at low temperatures, receding with $T^5$, leading to a non-zero residual resistivity in materials which do not exhibit superconductivity. At an elevated temperature, the phonon contribution (scattering rate due to electron-phonon coupling) saturates, when the mean free path of the charge-carriers $l$ approaches the dimensions of the lattice parameters $a$ (known as the Mott-Ioffe-Regel limit (*42*)). At this point, an increased phonon amplitude/density (due to thermal excitation) does not elevate the scattering probability for charge-carriers any longer.

Although a significant contribution from electron-electron interactions (deviation from $\rho \sim T$ behaviour) could be expected in the vicinity of a charge-transfer insulating regime, the cuprates exhibit a $\rho \sim T$ behaviour over a wide range of temperatures and doping: No saturation of the resistivity was observed to the highest measured temperatures ($\sim 1100\,\mathrm{K}$) (*43*, *44*). Remarkably, in some optimally doped samples, the resistivity does not deviate from the linear temperature dependence at low temperatures either, even close to the superconducting state. However, heavily overdoped samples exhibit Fermi-liquid transport characteristics ($\rho \sim T^2$) at low temperatures.

If indeed the resistivity is considered as solely dependent on the scattering rate ($\rho \sim 1/\tau$) the picture of quasiparticles within the Fermi liquid theory seems to break down. This breakdown is based on the assumption of a linear dependence between the resistivity, the scattering rate, and the temperature ($1/\tau \sim \rho \sim T$) from evaluations of electronic transport measurements (*45*). When staying within this framework, the mentioned features call for completely new physical descriptions. Notably, this strange metal regime evolves smoothly into the Fermi liquid behaviour seen in heavily overdoped samples.

### 2.1.5 Pseudogap (PG)

Through various probes (mainly STM/S – scanning tunnelling microscopy/spectroscopy and ARPES), a partial gap is measured in the cuprates. The "partial gap" denotes an energy gap, which is not isotropic in momentum space: The gap opens at the antinodes of the Fermi surface. This *"pseudo"-gap*, clearly observed by ARPES below a characteristic temperature $T^*$, seems to evolve smoothly into the superconducting gap below $T_c$, as measured by STS (*46*). The remaining parts of the Fermi surface in the PG regime, documented by ARPES, are called Fermi-arcs (*47*, *48*).

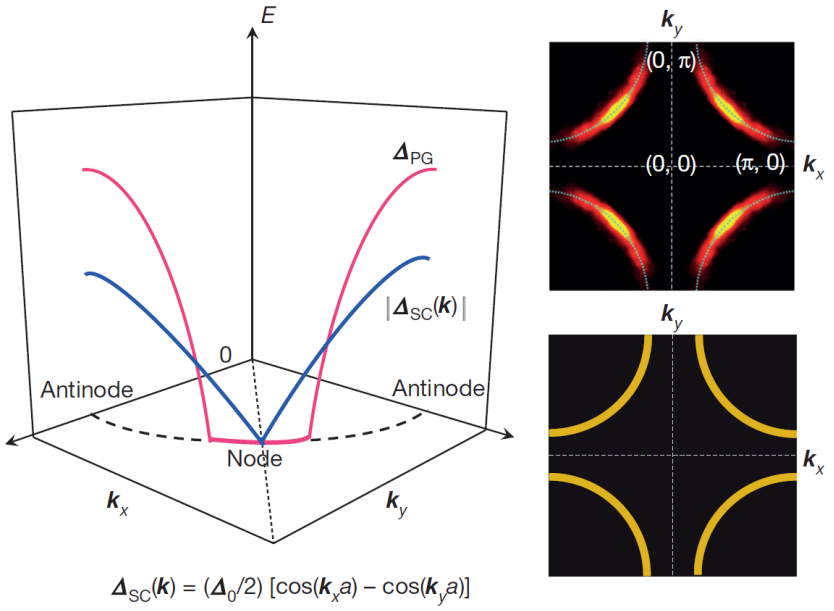



Figure 2.4: The topology of the Fermi surface in the cuprates. The large Fermi surface predicted by band structure calculations (lower right panel) compared to the Fermi surface as observed in ARPES measurement in the underdoped regime (upper right panel). On the left, the different angular dependencies for the superconducting gap and the pseudogap are shown in momentum space – the corresponding function for of the momentum dependence of the superconducting gap in its *d*-wave symmetry is displayed too. Extracted from (*49*) with permission.

**Importance of the Pseudogap** It seems that at the heart of the problem of accurately describing the cuprates lie the strong electron-electron interactions which are also responsible for the charge-transfer- (/Mott-) insulating character of the parent compound. Mott-insulators at exactly half-filling are fairly well understood and can be described by dynamical mean field theory (DMFT) (*50*). Upon strong doping (introducing enough additional charge carriers), the metallic characteristics predicted by band theory calculations (e.g. density functional theory – DFT) are re-induced in materials which are Mott-insulators at half-filling (*51*). This state of matter is understood well too.

The difficult conundrum is the intermediate doping range, and how the crossover/transition between these regimes occurs. This problem is further exacerbated in the cuprates, by the existence of the 2p oxygen bands in close vicinity to the 3d copper bands, causing the charge-transfer insulating behaviour (*29*). The momentum dependent nature of the gap in this intermediate doping range is particularly surprising. A complete solution, which solves the many-body problem of electron-electron interactions non-perturbatively on all length scales, is therefore the actual ultimate goal for *ab-initio* calculations[5] (*52*, *53*).

However, it might well be that such a complete solution is not required to understand the microscopic origin of high-temperature superconductivity. From a point of view which is deeply rooted in the quest to further progress, we currently deem solutions to the high-temperature superconductivity question to be more important than a complete description of many-body electron-electron interactions.

[5] *ab-initio* stands for "from first principles", without any empirical parameters.

**Charge Density Wave (CDW)** The pseudogap hosts a number of broken symmetry states, such as CDW-order, spin density wave (SDW) order, and nematic order (*54–56*).

> *A [...] density wave is a state of matter that breaks lattice translational symmetry [...]. In such a state any density-like observable, [...] will be modulated with a periodicity different to that of the lattice. One such quantity is charge density, which gives its name to this state of matter.* (*30*)

A ubiquitous CDW order has been established in the cuprates through many experimental probes[6] (*57–75*) – for an exhaustive review, see (*76*). This charge order appears in underdoped samples, in a region centred around $p \sim 1/8$, where it seems to be the strongest. Around the same doping level, $T_c$ deviates from its parabolic doping dependence, being reduced around the "1/8 anomaly" (*58*). The doping dependence of this reduction is similar in all studied cuprates, however the strength of it is compound dependent, being correlated with the strength of the CDW signal[7] (stronger CDW order entails stronger suppression of $T_c$). This points to the CDW as a competing order to superconductivity in the cuprates (*68*, *74*, *75*, *77*).

CDW order causes a reconstruction of the Fermi surface: Quantum oscillations (QO) are observed in $YBa_2Cu_3O_{7-\delta}$ (YBCO) and $HgBa_2CuO_{4+\delta}$ (Hg1201), revealing a Fermi surface of small electron pockets in high magnetic fields (∼20–80 T) around the same doping levels (*78*, *79*).

[6] scanning tunneling microscopy (STM), nuclear magnetic resonance (NMR) Resonant X-ray scattering (RXS), resonant inelastic X-ray scattering (RIXS), X-ray powder diffraction (XRD), neutron scattering (only for stripe order)

[7] Intensity of the Bragg-peak satellite, measured as the area under the peak (*77*).

## 2.2 Overview of Theories of Superconductivity in Cuprates

A lack of theories describing different observed effects is not the main problem of the scientific community investigating high-temperature superconductors, like cuprates. Indeed there are plenty. As already discussed above, cuprates are highly complex materials, with a rich variety of observed phenomenons and anomalies. As a result, many different models which each capture some of the observed phenomena have been put forth. However, none of them have become widely accepted as being a complete description of the behaviour of cuprate superconductors. Regarding acceptance, the phenomenological model which lies at the foundation of this thesis (described in section 2.3) is not an exception. Furthermore, the nature of the pseudogap and of other observed anomalous effects appears to be closely related to the emergence of superconductivity in cuprates, therefore a large body of research and models is focused on resolving these problems, not necessarily explicitly including a possible pairing mechanism, though superconductivity is clearly the underlying motivation in most models.

Therefore, a brief overview of selected theoretical models is described here, loosely following refs. (*49*, *80*). Following the overview, the phenomenological model which motivates the work of this thesis is described in section 2.3.

- Theories which base superconductivity on an anomalous order found in the cuprates were proposed for almost every anomalous order found to date. This includes spin and charge density wave order (*54*), "nematic" order (*55*, *56*), d-density wave order (*81*), and loop-current order (*82*, *83*).

- Claims of a theoretical description via a purely Hartree-Fock mechanism were published in 2013 (*84*).

- The possibility of superconducting pairing mediated by antiferromagnetic spin-fluctuations (possibly distinct from spin density wave order) was investigated by numerous theoretically focused research groups. This approach is related to the close proximity of superconductivity to an antiferromagnetically ordered regime in the undoped parent compound of the cuprates, as shown in fig. 2.3 (*85*, *86*).

- Some argue that quantum critical fluctuations from quantum critical points (QCP) (*87*) mediate the pairing interaction in cuprate superconductors. This is largely based on an extrapolation of the characteristic temperature of the pseudogap ($T^*$) to zero temperature at $p = p_c$, assuming a quantum critical point to be at the end of the "ordered phase" (*45*, *88*). The QCP is proposed to lie well within the superconducting phase ("hidden" by it), and is distinct from the critical doping levels $p_{min}$ and $p_{max}$ between which superconductivity is observed.

- A number of theories attribute the pseudogap to pre-formed superconducting pairs which lack phase coherence at elevated temperatures, and condensate to a coherent, bulk superconducting phase at lower temperatures (*89*). Very recently a pre-pairing regime of superconducting fluctuations has been firmly established (*90–96*), and it appears only in the vicinity of $T_c$. In this regard, it is worth mentioning, that the pseudogap has been thoroughly demonstrated to be markedly different from the superconducting gap (*97–107*).

The above listed theoretical approaches are not necessarily mutually exclusive: Many models which combine a variety of the described concepts (*55*) exist. Neither has yet found universal acceptance in the scientific community.

## 2.3 Ideas followed in this Thesis

*The phenomenological model proposed in refs. (33, 108–110) is described in this section. It captures recently established universal features of the cuprates. Of these features, the temperature and doping dependence of the electrical resistivity is most prominent. The model describes the phenomenon of one charge carrier per $CuO_2$ unit being localised at low doping and low temperature, which has strong implications for the description of the phase diagram. Experiments conducted and presented in this thesis are motivated by this model and designed to test its validity.*

### 2.3.1 Localisation of "The One"

Essential effects observed in the cuprates can be understood based on two simple notions: a) The number of itinerant Fermi liquid charge carriers in the cuprates depends "linearly" on temperature, non-linearly on doping and b) (structural) inhomogeneity plays an important role in the localisation described below.

At low doping, and low temperature, Hall coefficient measurements show the number of itinerant charge carriers to be $p$ (*111*). However at high doping (low temperature), in the Fermi liquid regime, this number of itinerant charge carriers is found to be $1+p$ (*112*), by Hall coefficient and quantum oscillation experiments (*49*). This entails an evolution of the charge carrier density from one regime to the other, in agreement with the pseudogap notion.

In fig. 2.5, a very simple possibility for this crossover is shown (as proposed in ref. (*109*)). In the model, two subsystems of charge carriers (holes) exist: One is itinerant, the other is localised (at low doping/temperature). The doping $p$ directly adds only to the itinerant subsystem. The localised subsystem contains exactly one hole per $CuO_2$ unit below $T^{**}$. This hole is separated from the Fermi level by a doping dependent localisation gap $\Delta$, which is spatially inhomogeneous. The origin of this localisation is attributed to the strong electronic correlations, which produce charge-transfer insulating behaviour in the parent compound of the cuprates, the exact nature of which is still under investigation (*113*, *114*).

Across this localisation gap, charge carriers are thermally excited, as in a standard semiconductor, giving rise to the temperature dependence of the itinerant charge carrier density. Figure 2.6 shows the resulting dependency of localisation on doping and temperature, as calculated from experimental data in ref. (*109*).

### 2.3.2 Resistivity in the Normal State

Generally speaking, electrical resistivity of a material depends on the electronic scattering rate, the effective mass, and the charge carrier density. Contributions from distinct mechanisms which influence those parameters, are known for a wide range of materials. In the case of cuprates, the charge carrier density and effective mass are most often assumed to be independent of temperature (*49*), despite the presence of the pseudogap. Thus, the strange behaviour of the resistivity is attributed to a linearly temperature dependent scattering rate from low to high temperatures, entailing the need for completely new theories of charge transport in solids.

ARPES measurements clearly identify the depletion of the density of states in near the antinodal direction of the Fermi surface in the pseudogap, a strong indicator for changes in the charge carrier density. This is completely neglected in the assumed linearly temperature dependent scattering rate.

If one considers the charge carrier density as possibly mutable, an entirely novel approach should be employed to determine the scattering rate. The charge carrier density can be measured through the Hall resistivity – ref. (*33*) showed it to depend linearly on the temperature, above

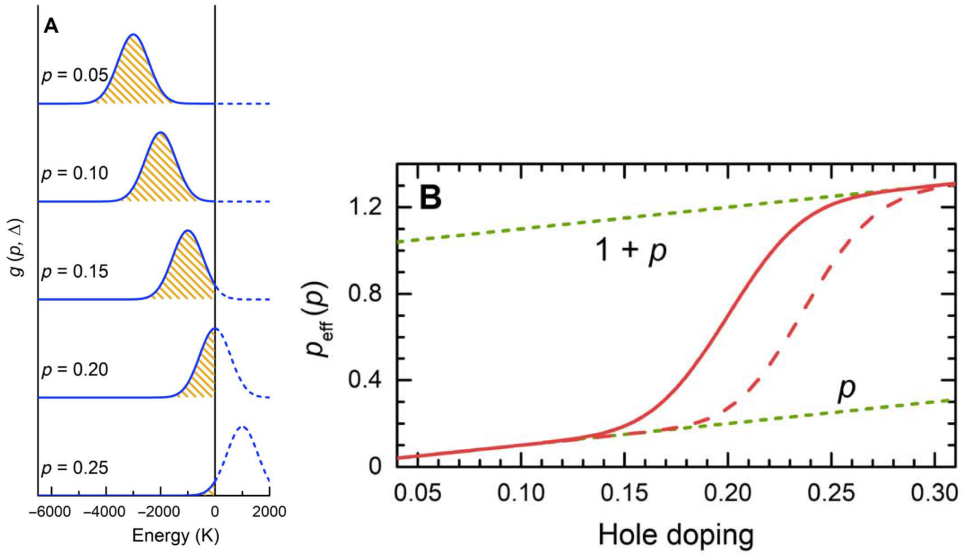



Figure 2.5: The simple Gaussian gap distribution function at different doping levels (left) and the effective hole doping at $T = 0$ (right) as calculated in ref. (*109*). The effective doping shows a smooth crossover from $p$ to $1 + p$ charge carriers – the dashed red line indicates the skewed Gaussian gap distribution function used to compare with LSCO data, as displayed in fig. 2.6. There, the density of itinerant charge carriers starts to deviate close to optimal doping. Adapted from (*109*) with permission.

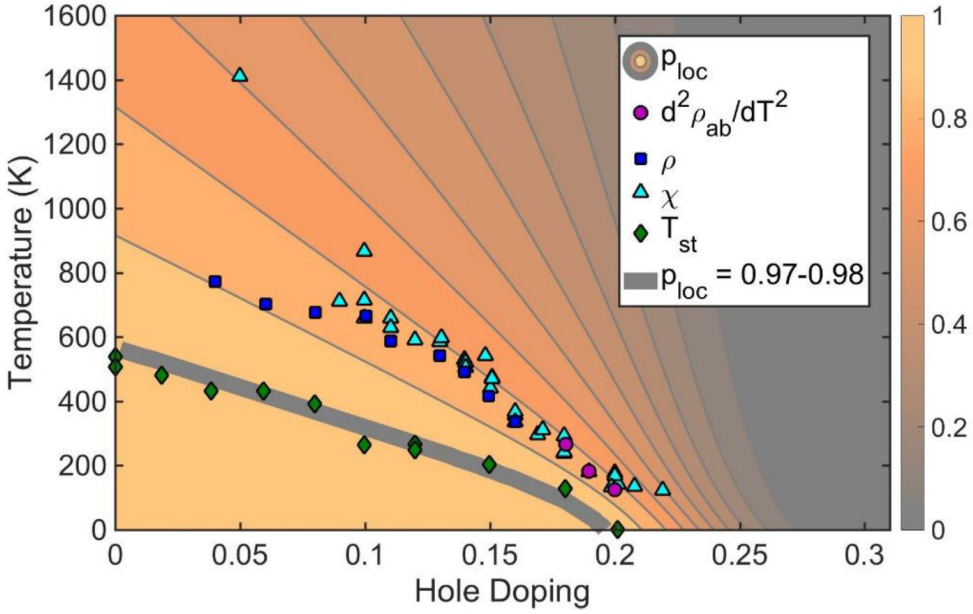



Figure 2.6: The fraction of localised holes calculated in ref. (*109*) (compare the figure above) compared to the highest measured temperatures $T^*$ for LSCO (all data but those labelled $T_{st}$). Iso-lines of the contour plot denote increments of 10% of localisation. $T_{st}$ marks the structural tetragonal-to-orthorhombic transition of LSCO – remarkably, it coincides with an iso-line of effectively full localisation of one hole per unit cell ($\sim 98\,\%$), indicating a coupling between charge and lattice degrees of freedom. Extracted from (*109*) (supplementary materials) with permission.

the temperature $T^{**}$ – below $T^{**}$, it is approximately constant (see fig. 2.5). This is a signature of the gradual localisation of one carrier per unit cell (with temperature), which is completed below $T^{**}$. The scattering rate, retrieved by taking into account the contribution of the modified charge carrier density to the resistivity, is shown experimentally to be universal (doping and compound independent) and quadratic in temperature (*33*).

Therefore, the linear dependence of the resistivity on temperature is determined by the ratio of the scattering rate and the changing carrier density. The quadratic temperature dependence of the scattering rate is a reminiscence of the Fermi liquid behaviour found on the overdoped side. However, this behaviour in resistivity is masked by a linear contribution from the thermal excitation of charge carriers. The Mott-Ioffe-Regel (MIR) limit ((*42*) – described above) is not crossed in this description, as the scattering rate is the same as on the overdoped side, where this limit can be defined rather simply.

### 2.3.3 Pseudogap

**Evolution from the Parent Compound** Within the scope of the model depicted above, the momentum dependence of the pseudogap is not touched for now. However, the pseudogap phenomenon is simply interpreted as a signature of the gradual (de-)localisation process, in which the antinodal states appear to be predominantly affected, while nodal states remain untouched.

The charge-transfer gap is identified as the localisation gap of one carrier per unit cell. With increased doping, the Cu 3d Hubbard bands are altered in a non-trivial way, reducing the gap magnitude, and causing the gradual de-localisation of this carrier. This process might be further complicated by the hybridization of oxygen 2p orbitals with (initially) the lower Hubbard band of the Cu 3d orbitals, slowly "releasing" the localised one (see fig. 2.6).

**Emergent CDW** Within the proposed model, the observed CDW order is to be understood as an emergent effect, which competes against superconductivity. The measured signals of CDW order are strongest where the superconductivity (marked by the magnitude of $T_c$) is weakest. At low temperatures and high magnetic fields ($\sim$80 T), the CDW reconstructs the Fermi surface to small electron pockets – however, the state which emerges upon suppressing superconductivity with magnetic field is not seen as being the ground state of the material, from which superconductivity itself arises. Therefore, the CDW is unrelated to the origin of superconductivity or the pseudogap, both of which extend to much higher temperatures.

### 2.3.4 Modelling spatial Inhomogeneity across Samples – Percolation

In bismuth based cuprates, STM measurements (*115*, *116*) showed a strongly spatial inhomogeneous distribution of the superconducting gap (nanoscale regions with distinct $T_c$'s, as inferred from the measured gap). Additionally, various experimental probes detected signatures of pre-formed Cooper pairs above $T_c$ (superconducting fluctuations up to $\sim$ 10 K above $T_c$) (*90–96*). In ref. (*108*), percolation theory is applied (system of nanoscale regions with distinct $T_c$), to describe how bulk superconductivity emerges from the initially disconnected superconducting patches. Signatures of the percolative effect are found in measurements of the torque magnetometry, electrical resistivity, Seebeck coefficient, specific heat, and the tomographic density of states. Additionally, it is shown that *Ginzburg-Landau superconducting fluctuations are considerably weaker than inhomogeneity effects* (*108*).

### 2.3.5 Superconductivity: Result of Interplay of the two electronic Subsystems

The phenomenological model proposed in ref. (*109*) links the superfluid density with both the density of itinerant charge carriers, and the density of localised holes (the two distinct subsystems of the electron dynamics, described in section 2.3.1): While the itinerant charge carriers are considered as the ingredients of Cooper pairs, the localised holes provide pairing interaction through bosonic excitations. In the model, a local but time-retarded attractive interaction (the microscopic origin of which is not explained within the scope of the phenomenological model) mediates the pairing and leads to superconductivity. In the underdoped cuprates, the decreasing number of itinerant charge carriers lowers the probability of local interactions occurring within a certain time scale and thus decreases the superfluid density. The decrease of $T_c$ in overdoped cuprates stems from the diminishing number of localised holes, with which the pairing interaction is associated. This is expressed in the parabolic doping dependence of $T_c$, which was found to be universal in the cuprates early on (*117*).

Further support for the proposed model can be deduced from the above mentioned STM measurements (*115*, *116*): In overdoped samples, a surprising anti-correlation between gap maps (below $T_c$) and conductance maps (above $T_c$) was documented. In locations where there is high conductance in the normal phase the superconducting gap is smaller than in locations where there is low conductance. This is understandable in the light of the proposed model: On the overdoped side, the superfluid density (and therefore the superconducting gap magnitude in the spatial inhomogeneous distribution) is mainly reduced by the absence of localised holes, which in turn augments normal phase conductivity.

In ref. (*109*), the intrinsic gap inhomogeneity is not linked to the inhomogeneous doping, since the model parameters which best explain the experimental data from the different compounds are very similar, although mercury-, bismuth- and lanthanum-based cuprates exhibit strong differences in the disorder introduced by doping. Instead, mechanical strain and inhomogeneous local lattice distortions of electronic origin are suggested to be possible sources for the inhomogeneity in the system. Therefore, one important question is how charge couples to the crystal lattice in cuprates, in particular, to determine the interplay between the localised holes and the crystal structure.

### 2.3.6 Pairing Interaction

As the pairing glue is associated with the localised carrier, the dynamics of the latter are of paramount importance for a microscopic theory describing superconductivity in cuprates. Various scenarios for a pairing excitation involving the dynamics of the localised carrier may be envisioned. Most of them are of excitonic nature (*15*, *118*).

One prospective candidate is a dynamic fluctuation of the carrier within one unit cell, entailing a time-dependent probability density function, which is smeared out over multiple electron orbitals (of distinct atoms, like copper and oxygen) in the unit cell. Another possibility would be processes which involve the electron orbitals of only one atom in the unit cell (most probably copper), with transitions between different orbital states ($3d_{x^2-y^2}$ to $3d_{xy}$ for example) (*119*).

Our goal is to narrow down the set of possible scenarios, by placing additional constrictions through experimental probes, thus testing the assumptions of the various models. In particular, if strong coupling between charge and lattice degrees of freedom exists, externally applied distortions of the lattice should be reflected in the electronic dynamics. Therefore, we focus on the impact of extrinsic lattice distortions on electronic transport properties, including superconductivity.

The $T_c$ of a material seems to be generally reduced by lowering the overall structural symmetry, for example in orthorhombic crystal structures compared to tetragonal structures. In this respect, local lattice distortions (for example buckling and breathing of the $CuO_2$ planes (*120–122*) and corresponding modifications of the Cu-O bonds) could play an important role (*123–125*). A breaking of tetragonal symmetry (in corresponding compounds) is expected to lead to a reduced $T_c$. How the Cu-O bond configuration affects an optimised interaction strength, and therefore affects $T_c$, remains an open question.

The big variety of compound families and their differing structures give an opportunity to study such differences. However, these differences in structure are not systematic, and in some compounds, doping is difficult to tune (and to measure precisely). Therefore, in the scope of this thesis, we systematically and extrinsically tune the lattice symmetry by applying uniaxial pressure. The idea is to measure the impact of changes of the crystal structure and correlate it with the changes in electronic transport properties. In the following chapters, we show electrical resistivity measurements under uniaxial pressure at room-temperature, as well as the influence of uniaxial pressure on the $T_c$ of the cuprate $Nd_{2-x}Ce_xCuO_4$ (NCCO). We measured the electrical resistivity both within the *ab*-plane (crystallographic [100]-*a* and [010]-*b* directions), and perpendicular to the *ab*-plane (along the crystallographic [001]-*c* direction), subsequently referred to as "*ab*-plane resistivity" and "*c*-axis resistivity", respectively.

# 3 Experimental Details

*In this chapter, the used experimental methods will be discussed. This includes the preparation of the samples, the control of the measuring conditions and the basic principles of the conducted measurements.*

## 3.1 Experimental Setups

### 3.1.1 Hydrostatic Pressure

Hydrostatic pressure is a commonly used technique: most material research under pressure is conducted under isotropic/hydrostatic pressure. A good example is the diamond anvil cells (DAC) (*126*). In a DACs, samples are immersed in a fluid, which is contained in a gasket and compressed between two diamonds, applying hydrostatic pressure on the samples. Hydrostatic pressure is purely isotropic – crystalline materials (compared to amorphous materials or fluids) are not purely isotropic,[1] even though they exhibit a range of symmetries (most prominently translational symmetry). Therefore, when hydrostatic pressure is applied, a complex, compound and sample dependent pattern of uniaxial strains/stresses develops inside of the material.

Crucial parts in setting up DAC transport experiments include i) the continuous alignment of the diamonds, ii) a proper routing of electrical connections to samples inside the fluid, iii) the homogeneous build-up of pressure on the diamonds, which are prone to shatter under inhomogeneous pressure distributions, and iv) ensuring that the pressure on samples remains purely hydrostatic.[2]

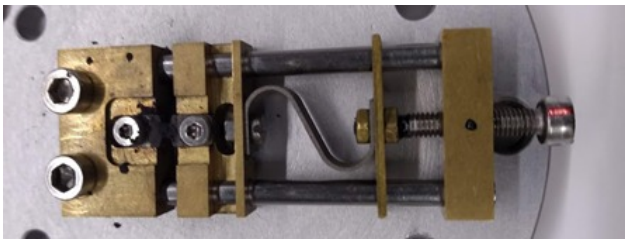
### 3.1.2 Uniaxial Pressure

The application of uniaxial pressure aims at inducing stress along a single crystallographic axis inside of a material (in contrast to the stress patterns developing under hydrostatic pressures described above). Uniaxial and hydrostatic pressure techniques are complementary and both should be applied. Within the framework of this work, the development of hydrostatic pressure cell (DAC) was continued and augmented. However, for the questions to be answered within the scope of this thesis, uniaxial pressure is better suited than hydrostatic pressure. Therefore, the focus of this thesis is on experiments under uniaxial pressure.

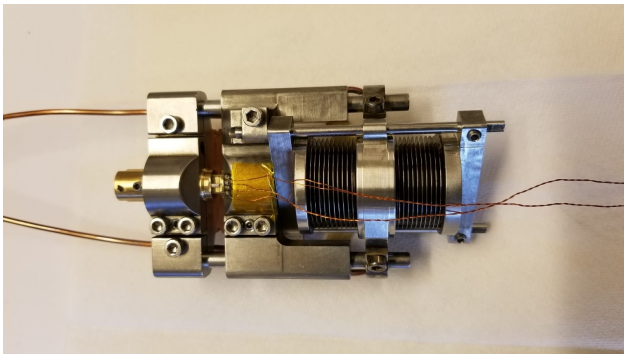
There are similarities and contrasts in the experimental setups of hydrostatic and uniaxial pressure:a) The alignment of pressurised materials is very important in the presented designs. Specifically, alignment of the diamonds in DACs is transferred to an alignment of samples in uniaxial pressure. In order to restrict lateral movement and lateral stress on samples, it is therefore essential to restrict the lateral movement of the parts which deliver load onto samples. This is performed to ensure purely uniaxial pressure on samples. b) The routing of electrical

[1] Isotropy and anisotropy in condensed matter is usually discussed above the level of crystal formation. While cubic materials (like body centred cubic iron) are considered isotropic, tetragonal or orthorhombic materials are considered anisotropic to varying degrees. "Pure isotropy", as used in the subsection about hydrostatic pressure, denotes isotropy as found in amorphous materials, or liquid/gaseous phases.

[2] If a sample touches the gasket or the diamonds, the pressure can no longer be hydrostatically isotropic.



(a)



(b)

Figure 3.1: (a) Spring-driven and (b) gas-driven uniaxial pressure cell. The basic design principles of the cells share similarities, but are fundamentally different: Two outer parts are connected through rails, along which a moving part can be pushed. The pressure is applied through either a spring, or gas-pressurised bellows.

connections to samples is of negligible concern in uniaxial pressure environments: In the presented designs, a sample is suspended between two points, glued to clamps or opposing parallel flat surfaces (see fig. 3.3b).

Two pressure cells were constructed and used in the resistivity measurements: 1. a spring-driven pressure cell for temperature dependent measurements and 2. a gas-driven cell for room-temperature measurements. Both cells are shown in fig. 3.1. The spring-driven cell operates on an S-shaped titanium spring, which is elastically deformed as force is applied. Titanium is used as the springs' material to allow for high forces within the elastic regime of the spring. The spring deformation was measured simultaneously with the applied force using a calliper and a scale for calibration purposes. The result of the calibration is presented in fig. 3.2. A linear fit to the experimental data gives a force of $47 \pm 3\,\mathrm{N\,mm^{-1}}$ upon a contracting deformation of the spring.

One end of a sample is clamped to the stationary part of the pressure cell, and the second end of the sample is clamped to the mobile part, pressed by the spring. Additionally, epoxy glue[3] is used to strengthen this connection between sample and clamps. The maximum acceptable force applied with the current spring amounts to 50 N, but a spring with a higher constant can be used. Higher pressures can be achieved with the gas-driven cell: A gas pressure of 1 bar relates to ∼15.4 N of force being applied to the sample. Currently the upper limit of the supplied gas pressure is ∼10 bar.

The gas-driven cell is based on two metal bellows, which are pressurized with helium gas. A sample is mounted on a sample cartridge, which is inserted into the cell (see fig. 3.3). The sample is fixed to the cartridge with epoxy glue. A pressure difference between the two bellows moves the centre part of the cell (right side in fig. 3.1b) applying force on one end of the cartridge while the other part of the cartridge is fixed.

The cartridge is designed to facilitate the mounting process of a sample. In order to avoid sample damage during transporting or inserting the cartridge to the cell, the cartridge was designed to avoid relative lateral and rotatory movements of the parts holding the two ends of the sample. The detailed drawings are attached as section B.1.

[3] Loctite Stycast 2850FT with the catalyst 24LV.

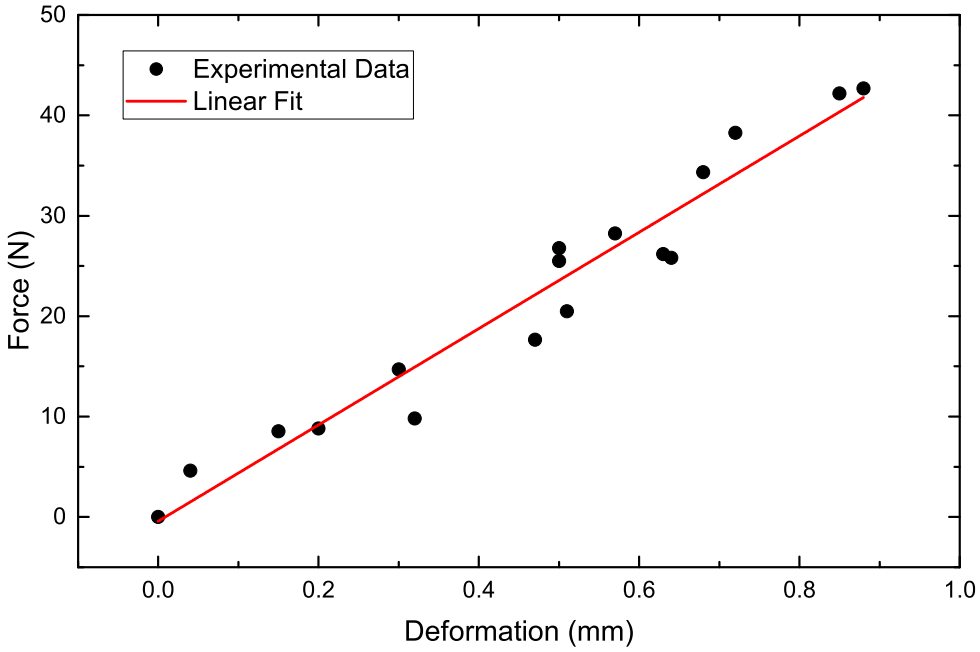
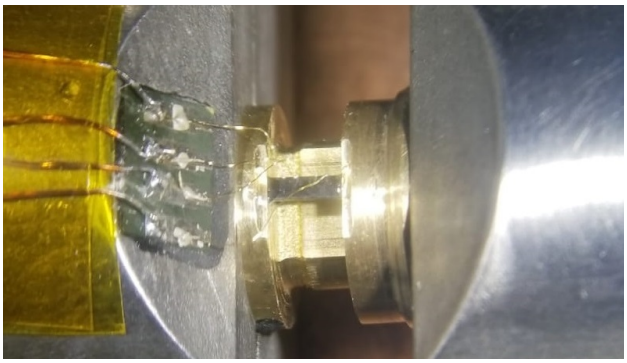
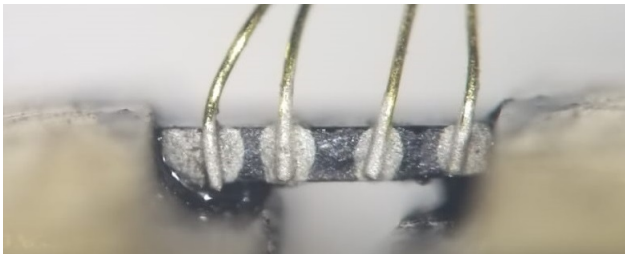



Figure 3.2: The spring calibration. From the linear fit, the slope equals to $47 \pm 3\,\mathrm{N\,mm^{-1}}$, with an intercept of $-0.4 \pm 1.5\,\mathrm{N}$. The non-zero intercept and its uncertainty bear importance only when cycling the pressure through zero, as only the deformation of the spring is being measured, not its full length. The measurements of the deformation of the spring, as well as of the geometry of the sample, have a strong impact on the uncertainty of the estimated pressure: The achieved pressure depends on the sample cross-section.



(a)



(b)

Figure 3.3: Details of the sample mounting on the sample cartridge. In (a), the configuration of wires and sample on clamp and cartridge is shown. One possibility of how to glue a sample to the cartridge is shown in (b) – in this case the sample sits on the ledge in the cartridge, and is additionally secured by glue.

The pressure in the sample is derived as

$$p = F/A_{\text{sample}}$$

where $p$ is the pressure, $F$ is the applied force and $A$ is the cross-section of a sample. A table of the corresponding cross-sections of the samples can be found in chapter 4 (table 3.2). Applying maximal force to samples with cross sections of around 1 mm x 0.3 mm resulted in maximal pressures of $90 \pm 35$ MPa in the spring-driven cell, and $400 \pm 20$ MPa in the gas-driven cell.

### 3.1.3 Cryostat and Temperature Control

**Cryostat System** Resistivity measurements in the temperature range between 300 K and 1.4 K were performed in an Oxford Spectromag splitcoil cryostat, at TU Wien. The system is a helium-gas-flow cryostat, cooling is done by placing a probe into the "variable temperature insert" (VTI), where a continuous stream of cool He gas is maintained. The VTI is shielded from the room-temperature environment by a vacuum chamber, the He tank, and a nitrogen jacket. Both the nitrogen jacket and helium tank are inside the vacuum chamber.

A sample is loaded into the VTI inside a dedicated insert, constructed within the framework of this thesis. The electrical connections and cernox temperature sensors (see section 3.2.1) are connected to external devices via 24-pin Fischer-connectors.

**Sample Holder** The sample holder has been designed and developed to serve a multiple purposes. The main features include:

- A sample space well thermalised with the cryostat. It can host a number of samples for which electronic transport measurements can be performed simultaneously.
- Multiple wires which can electrically connect samples with external instruments.
- Space and connections for additional sample holders and experimental setups, for example a pressure cell.

Samples can be glued directly to the sample holder, however, in order to attach the pressure cell, two threaded holes were used in conjunction with a spacer, holding the cell 2 mm away from the sample stage to allow room for wiring..

The signal leads providing the connection between external instruments and the sample space are thermalised at the top part of the sample holder. All wires are wound around a copper block. This thermalisation allows the sample space to be cooled to the lowest possible temperatures of the cryostat (1.4 K). More details are shown in figs. 3.4 and 3.5.

**Thermalisation of a Sample in the Pressure Cell** In order to maintain the best possible thermalisation of the sample and the pressure cell, the sample space was filled with He as an exchange gas at atmospheric pressure. This is important, because the thermal conductivity through solids in the setup is particularly limited at low temperatures: A small cross-section of the connection between the brass shield and copper sample holder severely restricts the heat exchange between the He flow in the VTI, and the copper sample holder. Moreover, the experimental requirements of the spring-driven pressure cell considerably restricts heat exchange between the stationary and mobile parts. As the sample is clamped between those stationary and mobile parts, an additional heat conduction channel had to be established.

By using exchange gas, both of the above mentioned effects can be mitigated: Heat can be transferred to the sample space through the whole surface of the shield, and the sample is in direct contact with the heat exchange medium within the sample space.

For experiments performed in exchange gas, it is desirable to retain a high pressure (at least ambient pressure) in the space around the sample: A reduced gas pressure would lead to reduced thermal conduction through the gas. In order to maintain the constant exchange gas pressure in the sample space, the sample holder was enclosed in a brass shield. This shield was connected to the probe in a vacuum tight manner.

Such a vacuum seal for low temperatures is generally made using a gasket made of a soft metal, such as indium. An indium wire of 2mm was pressed by screwing the shield onto a brass part on the probe. In order to manufacture indium wires of this thickness, a custom-built indium press was used.

### 3.1.4 Temperature Control and Data Acquisition

All the software controlling the cryostat and the data acquisition system was developed within the framework of this thesis. The programs were written in python.

For the cryostat and temperature dependent resistivity measurements, a complete graphical user interface (GUI) was developed, aimed at providing users with a robust tool to conduct measurements. The GUI (`PyQt` Framework) was built with a Quantum Design physical property measurement system (PPMS) in mind, resulting in a user-friendly, reliable and uninterrupted operation.

For measurements on the gas-driven pressure cell (resistivity and pressure), comparably short scripts were written, which enable concurrent control of pressure and resistivity measurements, by means of a Lock-In Amplifier (Stanford SR830).

For a more exhaustive description of the Cryostat-GUI, refer to section C.

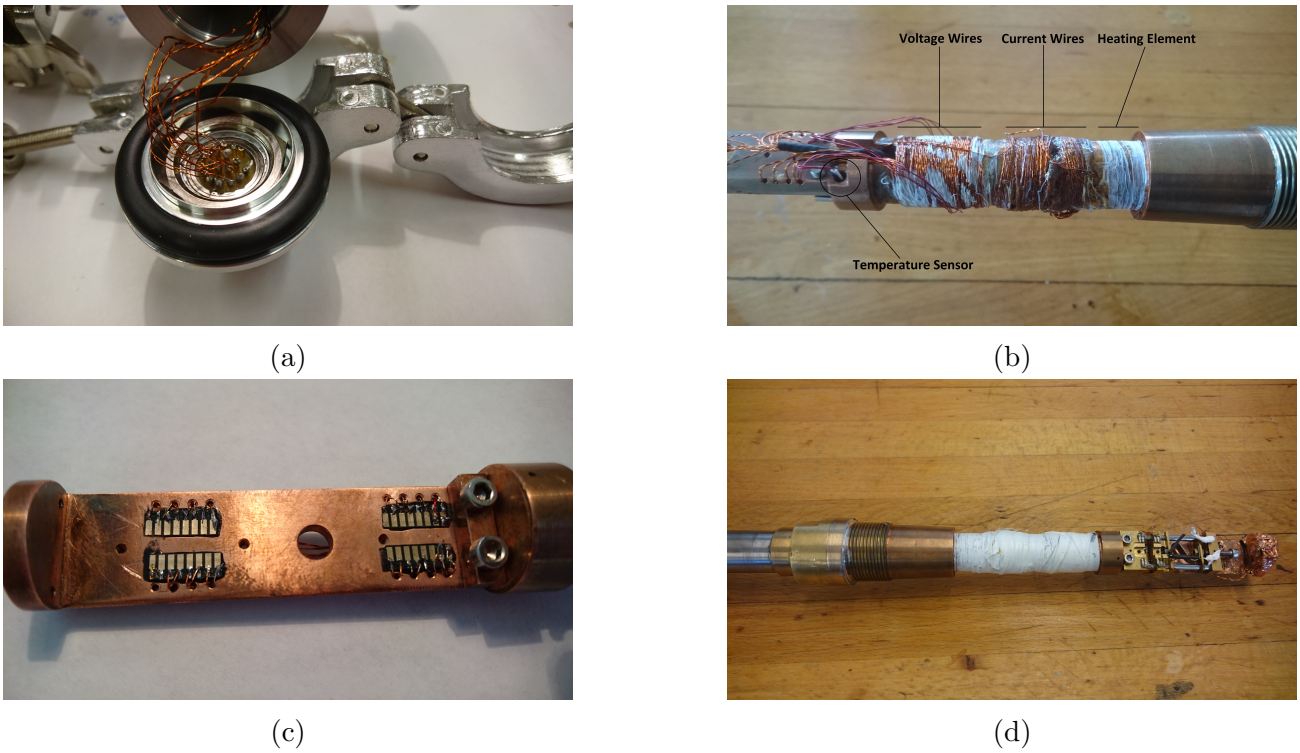



Figure 3.4: Wiring of the probe. In (a), a Fisher connector at the top of the probe rod is shown. The same wires as visible in (a) are coiled around the sample holder for thermalisation (b), and soldered to contact pads on the sample holder, shown in (c). The complete sample holder, with the spring-driven pressure cell fitted, is shown in (d).

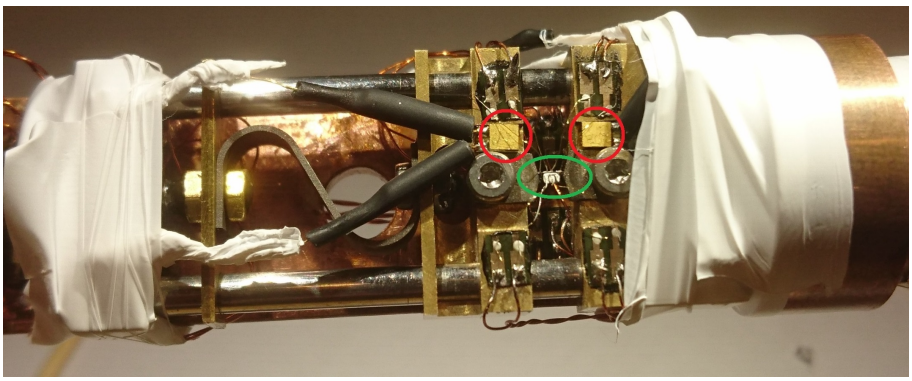


Figure 3.5: The spring-driven pressure cell, mounted on the sample holder. Highlighted in green is the sample, mounted between the clamps. The two temperature sensors are highlighted in red. Teflon tape (white stripes) was used to secure all wires, to prevent damaging them when applying/removing the vacuum shield. See section 3.1.2 for more details.

## 3.2 Experimental Principles

### 3.2.1 Temperature Measurement

The temperature inside the sample environment was measured using Cernox® sensors from LakeShore Cryotronics Inc.®. These sensors are thin film resistors of zirconium oxynitride deposited on sapphire, the resistivity of which has a characteristic temperature dependence. This resistivity is measured by a LakeShore temperature controller (by way of the "quasi"-four-point technique described below), and converted into temperature in Kelvin, based on previously obtained calibration data. Such calibration data of two sensors (calibration performed within this project) is presented in fig. 3.6. The sensitivity is greater at low temperatures, since the temperature dependence of the resistivity reaches up to $10 \times 10^5\,\Omega\,\mathrm{K}^{-1}$ below 2 K, and goes down to $10\,\Omega\,\mathrm{K}^{-1}$ at 300 K (Cernox® CX-1050, see fig. 3.6).

**Arrangement of Sensors** In order to achieve the highest accuracy and precision in a temperature measurement of a sample, temperature sensors and samples should be in thermal equilibrium with one large heat sink (the sensors preferably being embedded directly into the heat sink). The design of the spring-driven pressure cell (serving as a sample holder) did not allow for a sensor placement inside the pressure cell. In order to achieve the most accurate temperature measurement in this case, three sensors were distributed within the sample holder and pressure cell: One "CX-AA" cylindrical sensor, embedded in a solid copper block in the middle (lengthwise) of the sample holder, and two "CX-SD" sensors on the pressure cell, in close vicinity to the sample (see fig. 3.5).

**Hysteresis** With this configuration of sensors, a small hysteresis of the resistivity is observed around the superconducting transition temperature. This effect arises in measurements performed with temperature sweeps, due to a temperature gradient between sample and sensor. The gradient, in turn, is caused by the imperfect thermalisation of the system, and essentially depends on ramping rate. A higher ramping rate entails a stronger hysteresis. Such a hysteresis can be avoided by waiting for a stabilisation at each specific temperature, allowing all parts to reach a thermal equilibrium. Such a stabilisation scheme could not be implemented due to time constraints. However, if a temperature sweep is done slowly enough (0.01 Kmin), the difference between heating up and cooling down can be reduced to $<0.03$ K. The measured hysteresis is accounted for in the data analysis process, see sections 4.1.1 and A.

### 3.2.2 Resistivity Measurements

**Four-Point Measurement** In order to avoid contributions from contact-resistances in the measured resistivity, the four-point measurement scheme was implemented. The experimental setup requires four electrical contacts attached to the sample: While current is being supplied by two "outer" contacts (see fig. 3.7), the drop in electrical potential is measured between two "inner" contacts. As long as the resistance of the sample is small compared to the internal resistance of the voltmeter, only a negligible part of the current will flow through the voltmeter, which would otherwise alter the measured value. In this way, a high accuracy can be obtained in the resistivity measurement, as the contribution from lead and contact resistances is negligible.

This measuring technique also allows the testing of the quality of the cuprate samples, due to their layered nature. If the current in a sample is not homogeneous, the measured resistivity may behave very unexpectedly, and in the case of the *ab*-plane resistance it may even display negative values. This effect can stem from cracks in the crystal, or clusters of inhomogeneities in the crystal lattice. An example of such a measurement is shown in the lower panel of fig. 3.8.

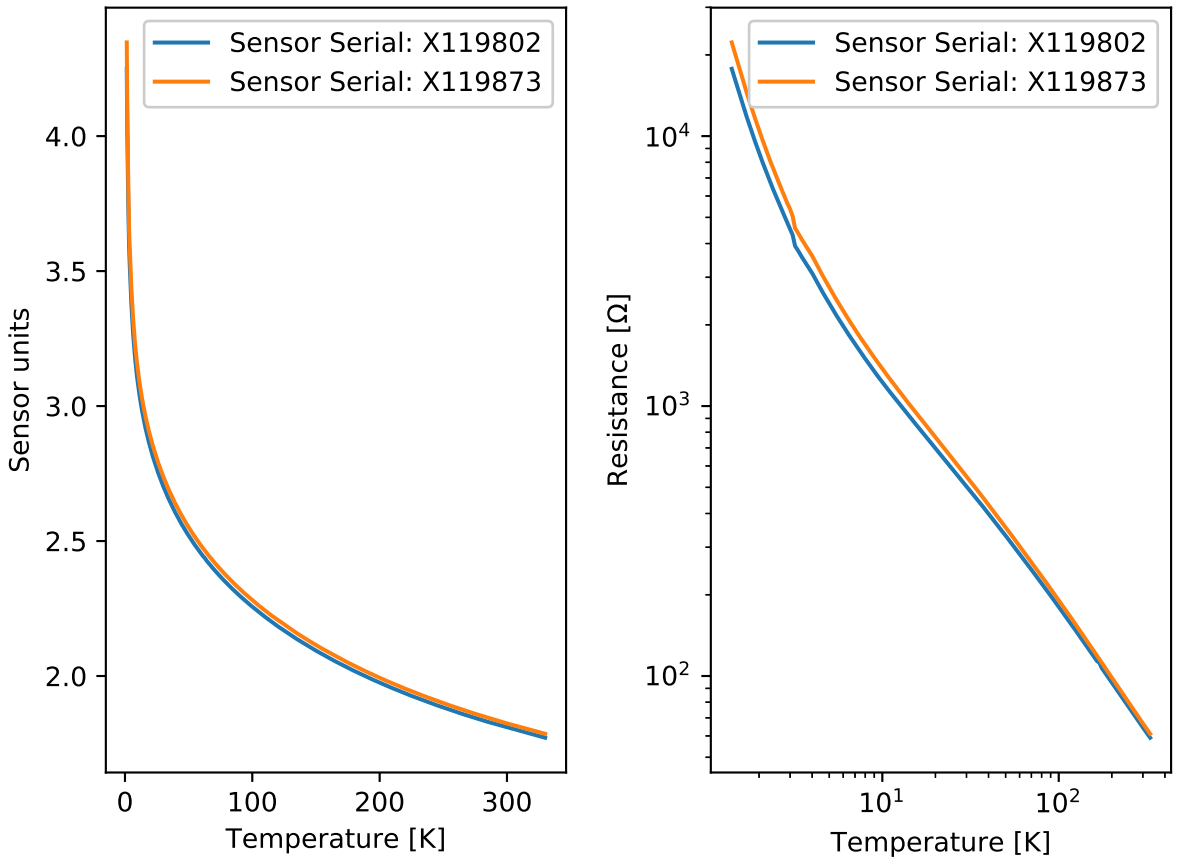



Figure 3.6: Sensor units (left) and Resistance (right) of the two Cernox® sensors which were calibrated against a third one, and subsequently used for temperature measurements. The small kink between 3 K and 4 K originates from difficulties in the temperature control around the boiling point of helium. As no measurements were performed around these temperatures, this issue was ignored.

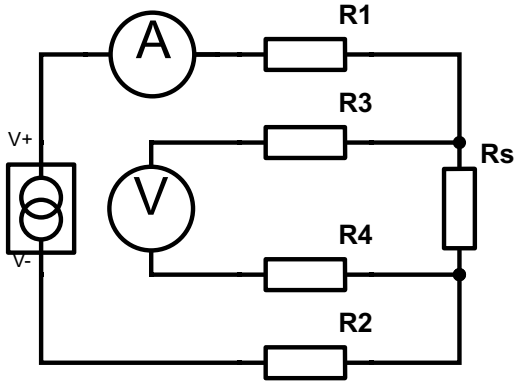



Figure 3.7: Schematics of the four point resistance measurement. Rs denotes the sample resistance, R1-4 stand for the lead and contact resistances of all wires involved.

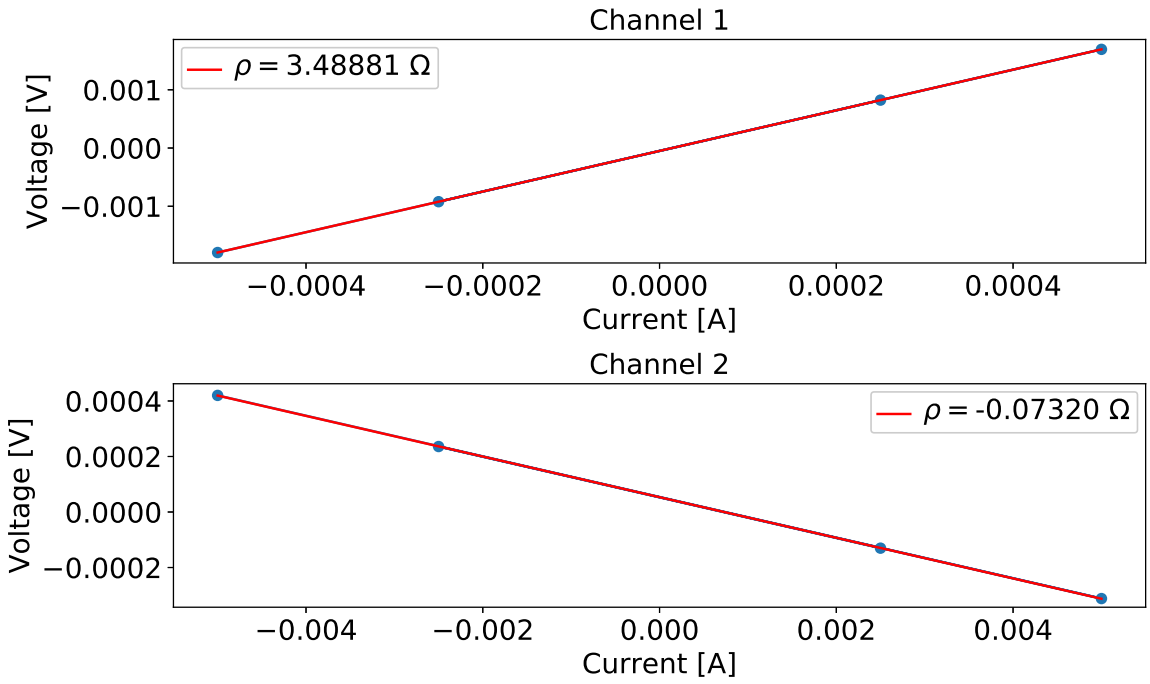



Figure 3.8: The I(V) resistance measurement in two different channels (top and bottom). While channel 1 measured a reasonable resistance value (3.4888 Ω), in channel 2, a seemingly negative resistance of −0.0732 Ω was registered. In both channels, the sample shows ohmic behaviour, however the scientific value and physical meaning of negative resistances is limited.

Negative resistance is physically meaningless in regard to the true value of resistance of a sample – in such a case the samples were not used in the experiments.

**I(V) Measurement – DC Resistivity** In order to assess whether the resistance is measured correctly, the measurements were conducted with a) both current polarities and b) with more than one value of applied current. Therefore, a single resistivity "measurement point" is derived from at least four values: The applied excitation current was 100 % and 50 % of a given value, with both polarities. The (at least) four measured voltages where fitted with a linear function, the slope of which corresponds to the resistance of the sample (see fig. 3.8). In order to assess the proper magnitude for the excitation currents, such an IV-curve was measured with a finer resolution and to higher currents than were eventually used in the measurements.

In conjunction with the above described temperature control system (see section 3.1.3), measuring devices were integrated into a general control software (see section C). While currents for the IV-linear measurements were applied with Keithley 6221 current sources, the corresponding voltages were measured with Keithley 2182A nanovoltmeters.

**AC Resistivity** Resistivity measurements performed on the gas-driven pressure cell were conducted in an alternating current configuration utilising a Stanford SR830 Lock-In Amplifier.

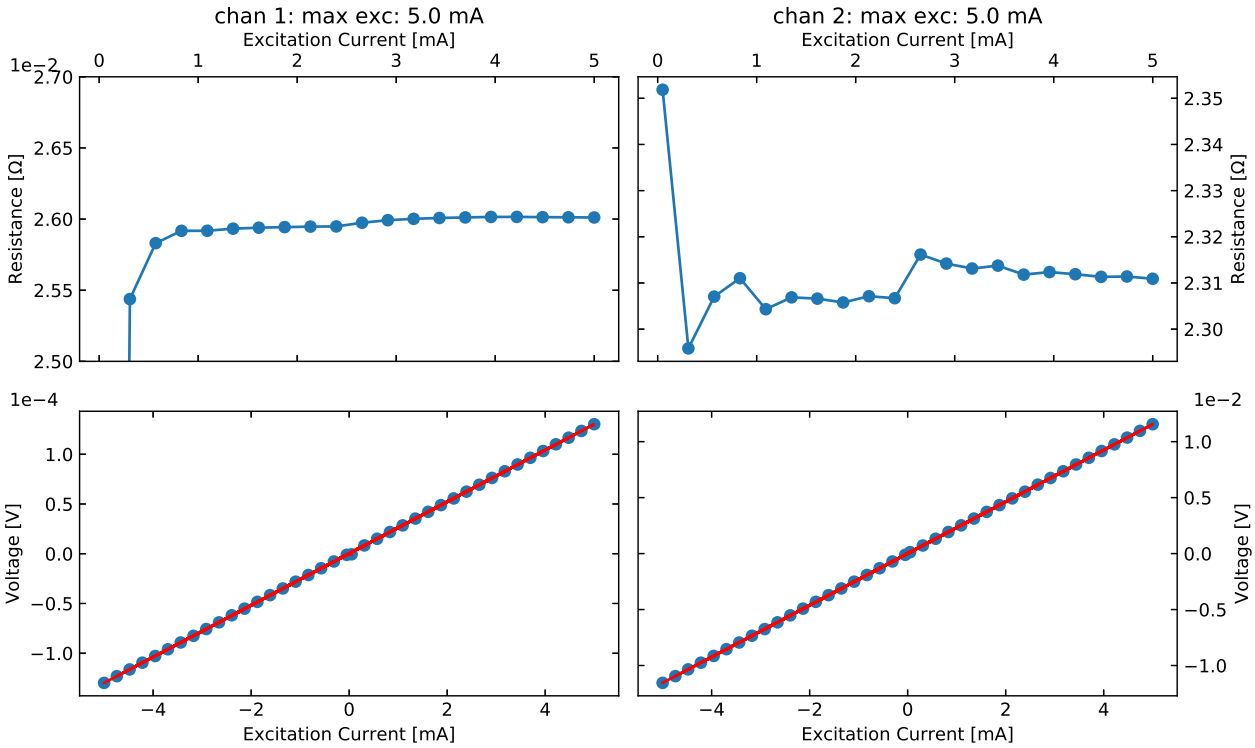



Figure 3.9: Representative resistance measurements and corresponding I(V)-curves. In the lower panels, the dependence of measured voltage on applied excitation current is shown. The upper panels demonstrate the correspondingly calculated resistances.

## 3.3 Sources of Systematic Errors

### 3.3.1 Current and Voltage Wires

In order to reduce mutual influence between excitation and measuring wiring, the corresponding leads were separated:

The contact pads in the sample space were divided into two rows, one for excitation and one for measuring wiring. The wires were wound separately around the sample holders (thermalisation) upper part, placed well apart from each other. In the probe rod, current and sensing wires reside within individual shrink tubes. Additionally, the two wire systems run in individual (doubly shielded) cables from the probe to the instruments, where the connections are physically separated. All wiring from the connection panel (where the cables and instruments are joined) to the soldering pads (and in some cases to sensors/the sample) is made of twisted pairs of wires.

### 3.3.2 Undesired Heating of Samples - IV Curve

In the normal phase of a superconducting sample, the flow of current through such a sample is a dissipative process. As this dissipation generates heat, current through a highly resistive sample might locally increase its temperature. This change in temperature would not be detected by the temperature sensors, and could lead to discrepancies in the measured resistivity. To avoid this heating effect, a minimal current is used in the experiments, however its value should provide for a sufficiently accurate measurement.

In order to establish suitable values for excitation currents, voltage is measured as a function of the excitation current, so called I(V) curve measurements. In fig. 3.9, I(V) curves representative for the conducted measurements are shown.

## 3.4 Sample Preparation

Although there is a number of methods to prepare single-crystal samples for electronic transport measurements, only the technique used within the framework of this thesis is discussed.

In order to achieve electrical contacts with low intrinsic resistance (typically a few Ohms), a clean surface must be prepared in advance. Although various techniques to obtain such clean surfaces exist, cleaving and polishing are the most widely used in preparing the electrical contacts on single crystals of cuprates. In the case of NCCO, polishing provides a sufficiently clean surface for preparing such contacts.

### 3.4.1 Sample Geometry

In order to achieve a high accuracy in resistivity measurements, the geometry of a sample is important. Ideally, a sample should be a long rectangular cuboid, with infinitesimally small contacts being placed almost at the ends of the longest dimension. The smaller the contacts are, the higher the accuracy that can be achieved in estimating the geometric factor. Furthermore, over the length of the sample, its cross-section should not change. For measurements of the *ab*-plane resistivity, the thickness of the sample (extent of the sample along the *c*-axis) should therefore be constant along the length of the sample. Moreover, in uniaxial pressure measurements, the sides of a sample (where pressure is applied) should be flat and parallel.

Additional constraints on the sample geometry arise from measurements under uniaxial pressure: For the gas-driven cell, ideally, two perfectly parallel and flat surfaces should be prepared,

perpendicular to the crystallographic axis along which force is applied. If this is impossible, the connection between sample and anvils can be rectified with glue.[4] However, this is only applicable at comparably low pressures, as a sample, which can withstand higher compression forces than the commercial glue can withstand lateral forces, might escape the pressure cell. In the spring-driven cell, the samples are glued and clamped, therefore the restrictions on the geometry arising from resistivity measurements suffice in this case.

The geometries mentioned for resistivity measurements were achieved by polishing with fine sandpaper. The sides of the samples were cut to be approximately parallel, and epoxy glue was used to attach them to the pressure cells.

### 3.4.2 Attaching Wires

The attachment of the wires is a very critical task in the contacting procedure. Gold wires with diameters of 10−50 µm are attached directly on the crystal surface, using silver epoxy.[5] The surface tension of the silver epoxy plays an important role in the placement of the contacts: If it is too high, the silver paste will not properly adhere to the sample, if it is too low, contacts are prone to flow together.

In the case of measurements under uniaxial pressure, requirements due to the fixation of samples to a pressure cell further exacerbate this procedure.

### 3.4.3 Curing Contacts

For the placement of contacts, Dupont 6838 epoxy was used which needs to be cured at high temperatures (400 °C) to properly electronically connect a wire with the bulk of the material. However, curing contacts for too long time intervals will degrade them, increasing the contact resistance. Nevertheless, if a sample needs to be contacted on multiple sides, it is not possible to cure all contacts at once.

In order to harden individual sets of contacts, samples were repeatedly cured at lower temperatures (180 °C) for short time spans. After all contacts were attached on one sample, it was cured at a higher temperature (400 °C) again. In this fashion, low contact resistances of 0.5−3 Ω were achieved. Nevertheless, in some cases the contacts exhibit a higher resistance of up to 30 Ω.

[4] Even if such parallel surfaces can be established, a sample should be held in place by glue which surrounds the boundary between sample and anvils.

[5] Dupont 6838 conducting epoxy.

## 3.5 Samples ($Nd_{2-x}Ce_xCuO_4$ – NCCO):

The compound which was used to test our experimental setup, and to run initial measurements was the archetypal electron-doped cuprate $Nd_{2-x}Ce_xCuO_4$ (NCCO). The sample preparation process for this material is comparatively easy – future measurements will be performed on other materials (for example $HgBa_2CuO_{4+\delta}$).[6] The basic structure of NCCO is similar to that of hole-doped cuprates: It exhibits $CuO_2$ layers (*ab*-plane) which are separated by charge-reservoir layers ($Re_2O_2^{2+}$, Re being a rare-earth such as La, Ce, Y). However, in contrast to the hole-doped cuprates which exhibit a "perovskite" T-structure with Cu-O octahedra or pyramids, NCCO exhibits a "fluorite" T′-structure without apical oxygen atoms (located above and below the $Cu^{2+}$ ions). (*127*, *128*)

In the doping-temperature phase diagram of NCCO (displayed in fig. 3.10), there is a regime of long-range antiferromagnetic order (LR-AF) which starts in the parent compound and extends to $x \sim 0.12$, a much higher doping level than in hole-doped cuprates (*129–131*). At high doping levels, above $x = 0.175$, a large hole-like Fermi surface is detected, similar to hole-doped cuprates (*132–135*). In intermediate doping levels, there are two separate regimes at elevated temperatures: Above $x \sim 0.14$, purely dynamic antiferromagnetism prevails at high temperatures (*131*, *136*, *137*), while bulk superconductivity is found at low temperatures (below 26 K). Between $x \sim 0.12$ and $x \sim 0.14$, there is a mixed phase with static, short ranged antiferromagnetism, and superconducting fluctuations at low temperatures (*138–141*). Interestingly, at $x \sim 0.145$ a sign change in the Hall coefficient is observed (from negative to positive) – the Fermi surface of the electron-doped material starts to be dominated by hole-like charge carriers (*128*, *142*, *143*).

The electrical resistivity of four samples[7] was measured under uniaxial pressure – the details of the samples are shown in table 3.1 and table 3.2. The doping levels are estimated at $x = 0.11$ (EUG27) and $x = 0.13$ (ICP13) with only the latter being superconducting. "EUG27" and "ICP13" each denote separate crystals, from which multiple samples were prepared. Those are further distinguished by specification of the pressure cell ("He" and "s" for gas-driven and spring-driven pressure cells, respectively), and a sequence number.

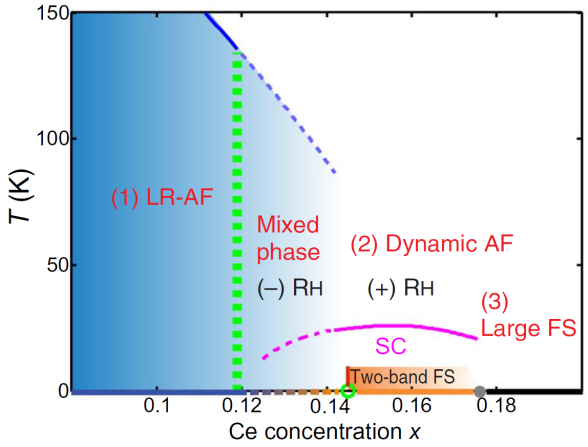
The temperature dependence of the resistivity of one sample from the crystal EUG27 ($x = 0.11$) was measured down to 1.4 K (not shown), without evidence of superconductivity. The application of uniaxial pressure did not evoke superconductivity (or alter the temperature dependence of the resistivity greatly) in this sample, therefore the temperature dependence of the resistivity from this crystal was not analysed within the scope of this thesis.

[6] $HgBa_2CuO_{4+\delta}$ (Hg1201) is considerably more difficult to work with, especially the sample preparation procedure of Hg1201 is a demanding task. NCCO is better suited for tests and initial measurements.

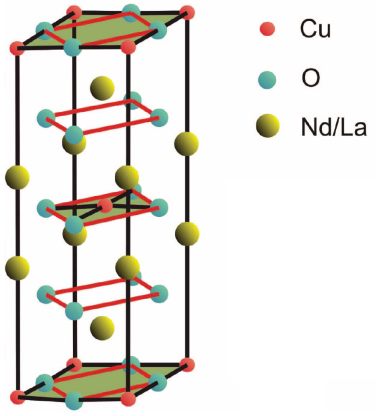
[7] Many more samples were prepared.

| Sample name | doping x | $T_c$ | pressure cell type | two-point contact resistance |
|---|---|---|---|---|
| EUG27_He2 | 0.11 | – | gas-driven | 96–99 Ω |
| EUG27_He4 | 0.11 | – | gas-driven | 8–9 Ω |
| ICP13_He2 | 0.13 | 25.3 K | gas-driven | 27–57 Ω |
| ICP13_s2 | 0.13 | 25.3 K | spring-driven | 2–23 Ω |

Table 3.1: List of the measured samples. The contact resistance is measured between two contacts which belong to one single channel (excitation or voltage measurement) – the presented range corresponds to lowest and highest thus measured values. Samples from the crystal EUG27 do not exhibit superconductivity.




(a)

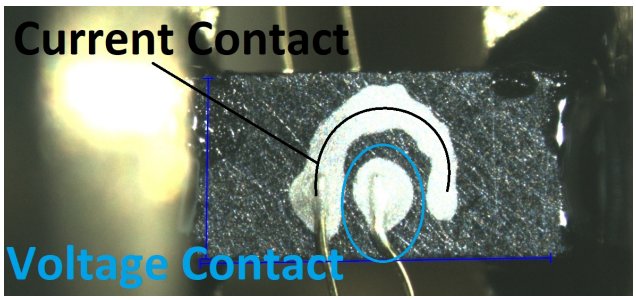



(b)

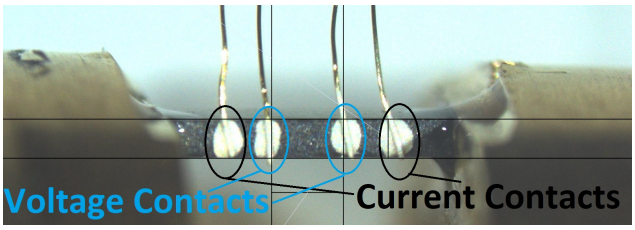
Figure 3.10: Phase diagram (a) and crystal structure (b) of NCCO. (−) $R_H$ and (+) $R_H$ denote regimes with negative/positive Hall coefficient. The phase diagram is classified into four regimes: (1) long ranged, static antiferromagnetism; (2) short ranged, dynamic antiferromagnetism with superconductivity at low temperatures; a mixed state between (1) and (2); (3) a regime with a large Fermi surface (ref. (*23*) and refs. therein). Adapted from refs. (*23*, *128*) with permission.

| sample | resistivity configuration | channel length | resistivity cross-section | pressure cross-section |
|---|---|---|---|---|
| EUG27_He2 | *c*-axis | 0.35 mm | 1.95 x 0.88 mm | 0.88 x 0.35 mm |
| EUG27_He4 | *ab*-plane | 0.72 mm | 0.88 x 0.752 mm | 0.88 x 0.392 mm |
| ICP13_He2 | *c*-axis | 0.256 mm | 1.72 x 1.264 mm | 1.264 x 0.256 mm |
| ICP13_s2 | *c*-axis | 0.20 mm | 3.37 x 1.336 mm | 1.336 x 0.205 mm |
| | *ab*-plane | 0.39 mm | 0.205 x 1.336 mm | |

Table 3.2: Sample geometries of the measured samples. "resistivity configuration" here denotes the geometry in which resistivity was measured, "channel length" denotes the corresponding distance between voltage contacts.




(a) EUG27_He2




(b) EUG27_He4

Figure 3.11: Two of the samples described in table 3.2. (a) and (b) show samples and attached contacts for the *c*-axis and *ab*-plane resistivity measurements, respectively. Both are mounted on the gas-driven pressure cell, mounted in two different possible ways: In (a), the sample is located on top of the ledge of the cartridge, and glued with stycast, while in (b), the sample was directly glued to the face of the anvils. Additionally, in (b) the sample was glued using conventional epoxy, which was sufficiently strong for the investigated pressure range.

# 4 Results

*The response of the resistivity of $Nd_{2-x}Ce_xCuO_4$ to uniaxial pressure was measured in two configurations: a) the dependence of $T_c$ on applied pressure (dc), and b) at room temperature with small steps of pressure (ac). In this chapter, the results of these measurements are presented and discussed.*

## 4.1 Resistivity under Uniaxial Pressure

### 4.1.1 Data Analysis Process for the Temperature Dependent Resistivity Measurements

The main goal of the conducted measurements is the determination of the superconducting transition temperature $T_c$ as a function of uniaxial pressure. In fig. 4.1, an example dataset of the temperature dependent resistivity measurement is shown. The data analysis of the temperature dependent resistivity measurements is explained step by step on the dataset for the intermediate pressure level p1 = $76 \pm 25$ MPa, collected for the sample ICP13_s2 with $x = 0.13$, $T_c = 25.3$ K.

$T_c$ is defined as the midpoint of the transition, derived as the peak centre of the first derivative of the resistivity (see fig. 4.2). The transition to the superconducting state is reasonably sharp (about 0.75 K wide), with a tail at lower temperatures (3 K), which can be attributed to inhomogeneity in the sample. Here, inhomogeneity means that different parts of the sample exhibit slightly different $T_c$'s, thus becoming superconducting at different temperatures.

In the following analysis the low-T tail will be neglected, focussing on the sharp part of the transition (close to the onset in this case), as the change of $T_c$ with applied pressure is much more visible there. To this end, only the data up to 9 K above and below $T_c$ were used in the analysis ($25.3 \pm 9.0$ K). These data were normalised to the value at 26 K for better comparability. As discussed in section 3.2.1, a small temperature hysteresis was observed in the resistivity data. Therefore, the data retrieved from measuring during heating and cooling cycles were separately submitted to cubic spline interpolations. The interpolated data was subsequently averaged, resulting in fig. 4.2a and fig. 4.2c (for details see fig. 4.1 and section A).

In order to extract more quantitative information about the behaviour of the transition and its pressure dependence, the first derivative of the resistivity was fitted with two Gaussian peaks (see fig. 4.3). The position of the two Gaussian peaks is related to $T_c$ by forming the weighted mean of the peak centres. Thus the aforementioned definition of $T_c$ as midpoint of the transition is satisfied. The area under the curve of the respective fitted peak is used as weight.

### 4.1.2 Pressure Dependence of the Transition

In fig. 4.2 and table 4.1, the observed variation of $T_c$ with applied uniaxial pressure is demonstrated.

A shift to lower temperatures was found in the $T_c$ of both the planar resistivity (within the *ab*-plane), and the resistivity channel perpendicular to it (along the crystallographic *c*-axis). $T_c$ decreases by 0.22 K (*ab*-plane) and 0.14 K (*c*-axis), from 0 MPa to $90 \pm 35$ MPa. The magnitude of the shift of $T_c$ decreased with increasing pressure. This corresponds to a rate of $\Delta T_c/\Delta p = -2.5\,\mathrm{K\,GPa^{-1}}$ for the *ab*-plane resistivity and $\Delta T_c/\Delta p = -1.5\,\mathrm{K\,GPa^{-1}}$ for the *c*-axis resistivity.

Furthermore, the transition broadens with increased pressure. This is observable through i) the shift of the centres of the fitted peaks (moving apart by ∼0.05 K), while ii) the sum of the full widths of their half maximum (FWHM) increases (by 0.05 K and 0.11 K in the resistivity within the *ab*-plane and along the *c*-axis respectively), see fig. 4.4 and table 4.1.

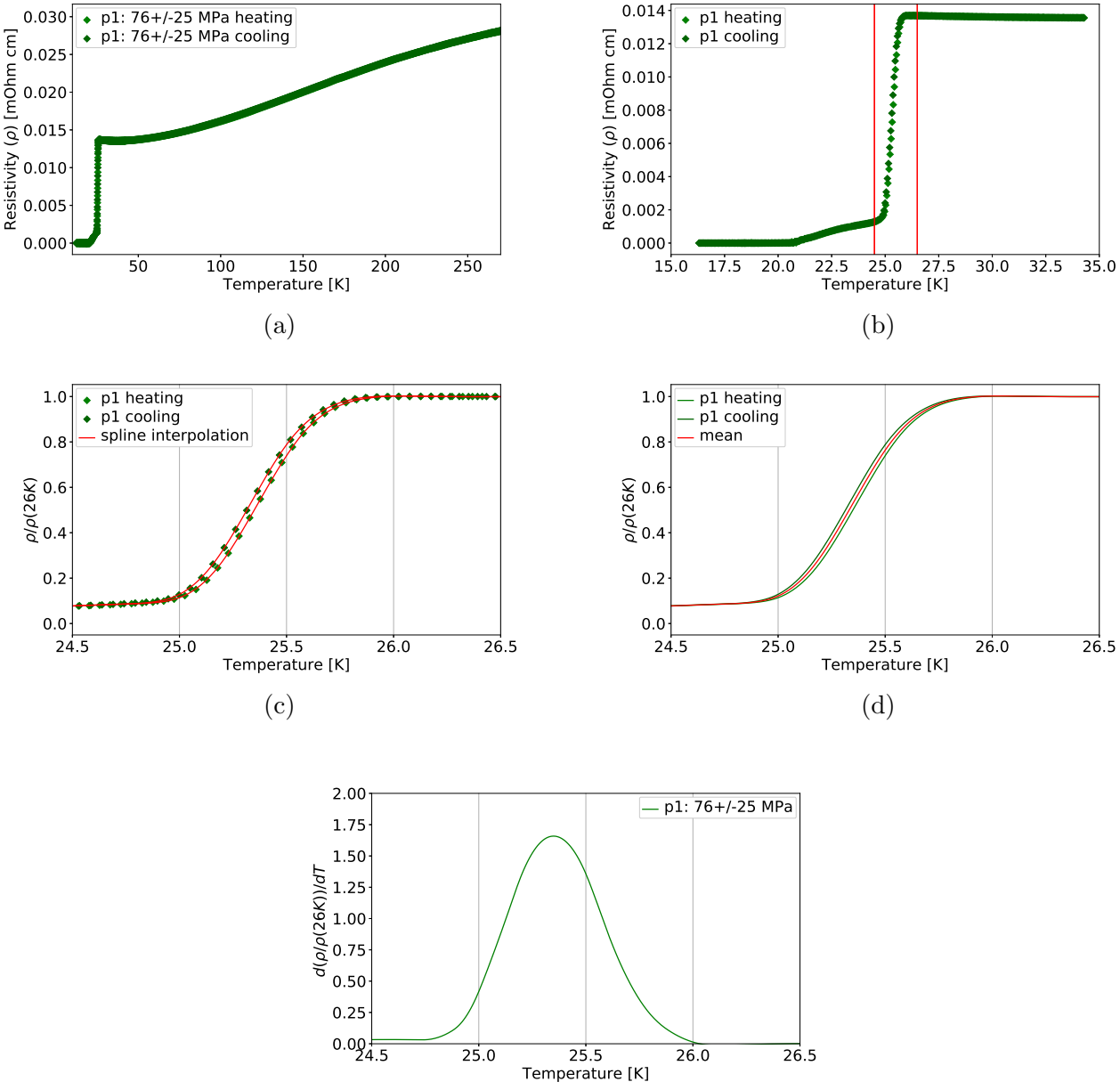



Figure 4.1: Data analysis procedure for representative *c*-axis resistivity measurements performed on sample ICP13_s2, at pressure $p1 = 76 \pm 25$ MPa. In (a) the resistivity is shown as a function of temperature between 4–270 K; (b) shows the investigated temperature range around $T_c$, the region indicated in red is shown in subsequent figures; (c) resistivity data normalised to its value at 26 K; in (d), the average resistivity value of the data collected on heating and cooling is presented. The error in both resistivity and temperature measurement is estimated to be smaller than the symbol size. See section A for data and corresponding analysis of all measured resistivity configurations under all investigated pressures.

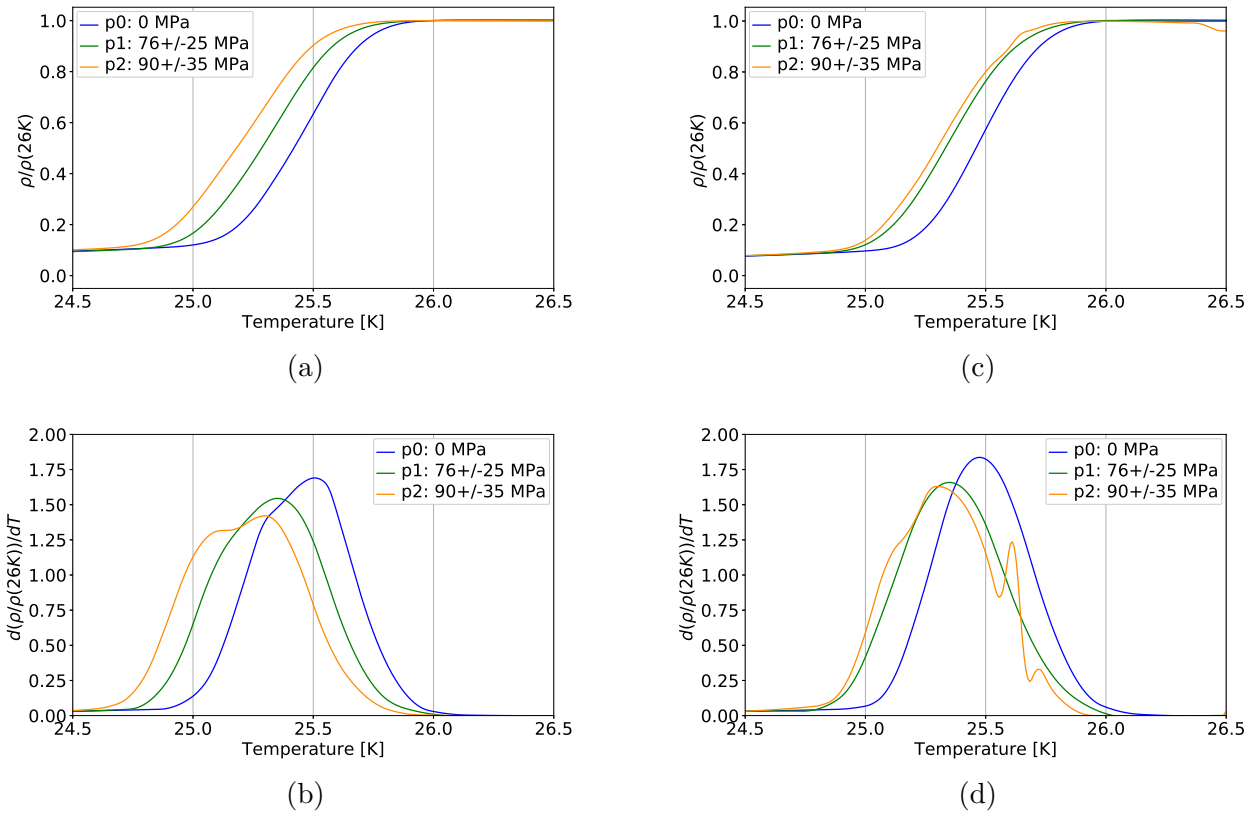
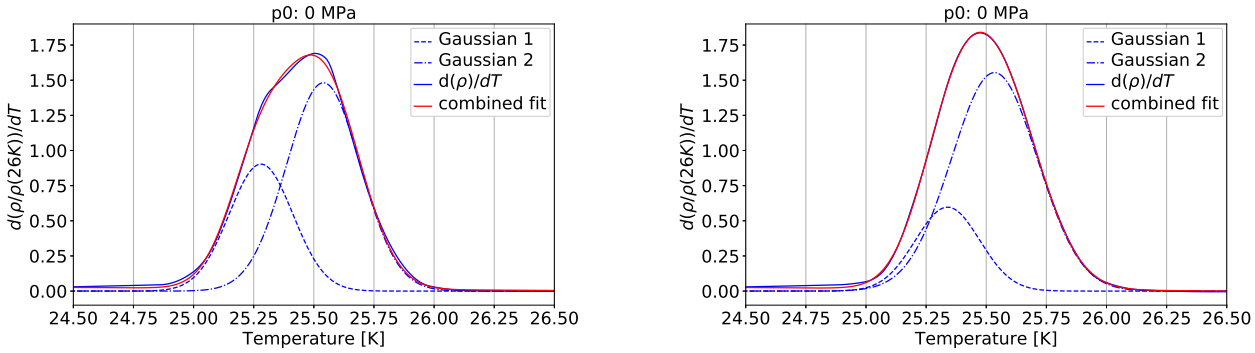


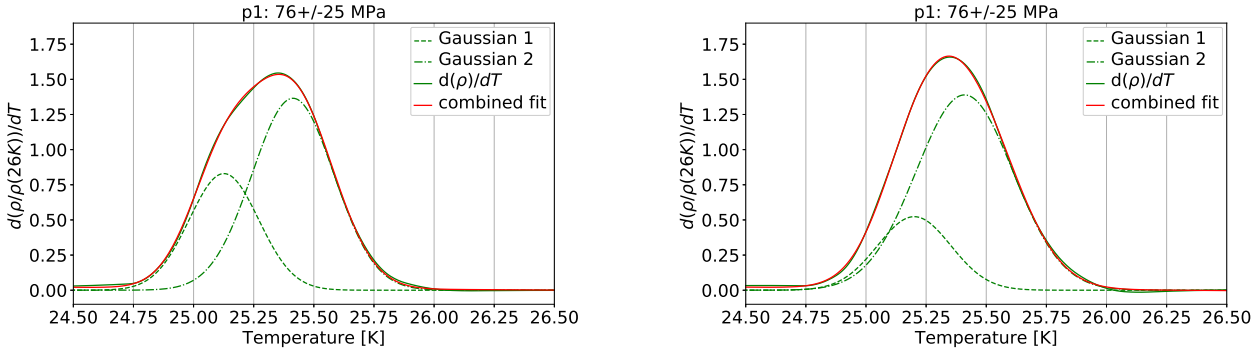

Figure 4.2: Resistivity and its first derivative near the transition in sample ICP13_s2, in the *ab*-plane resistivity (left: a and b) and the *c*-axis resistivity (right: c and d), respectively. p0 relates to ambient pressure, p1 and p2 are at $76 \pm 25\,\mathrm{MPa}$ and $90 \pm 35\,\mathrm{MPa}$ respectively. The resistivity is normalized to its value at 26 K.




(a)



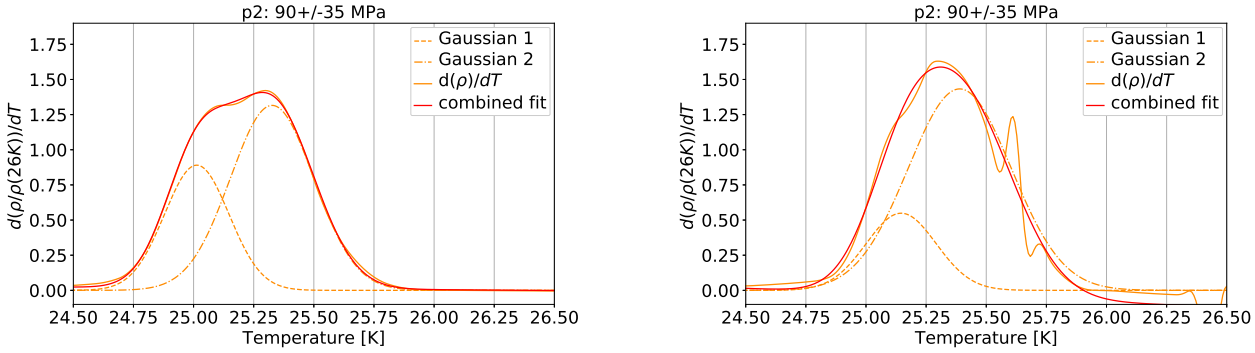

(d)




(b)




(e)

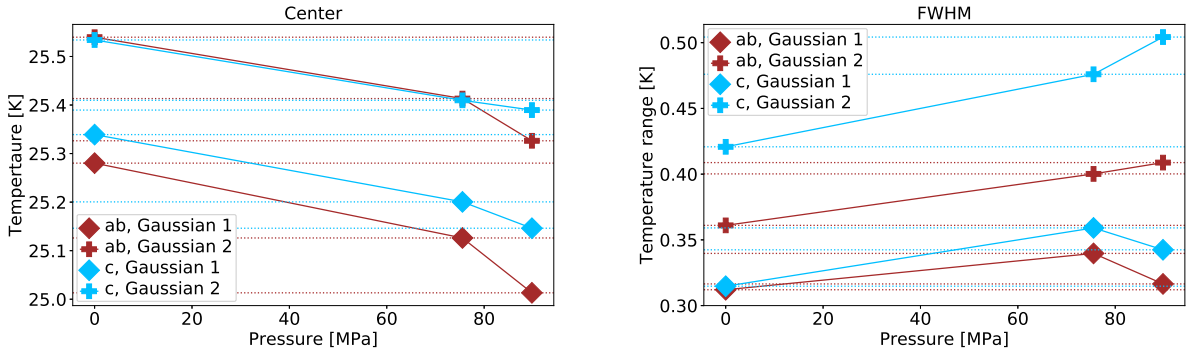



(c)




(f)

Figure 4.3: The first derivative of the resistivity (ICP_s2), around the superconducting transition. On the left (a-c) the planar resistivity is shown, on the right (d-f) the resistivity along the $c$-axis.




(a) (b)

Figure 4.4: Evolution of fitted Gaussians as a function of pressure. The data were obtained from the fit of two Gaussian peaks to $d(\rho/\rho(26K))/dT$ for the sample ICP13_s2. (a) Peak centre positions of Gaussian 1 and Gaussian 2 refer to the two functions fitted additively to the transition in the respective measurement. The decrease of $T_c$ with increased pressure is evident. (b) The corresponding full widths at half maximum of the fitted peaks. The increase of the sum from both fitted peaks is less evident, see table 4.1 for details. Full lines are guides to the eye: straight connections between measured values. Dotted lines are horizontal droplines which facilitate observing the evolution of the respective values.

| Value | Formula | p0 | p1 | p2 | p2−p0 |
|---|---|---|---|---|---|
| | | *ab*-plane resistivity | | | |
| Centre separation | $G_2 - G_1$ | 0.2594 K | 0.2871 K | 0.3132 K | 0.0538 K |
| Centre mean | $(G_1 + G_2)/2$ | 25.4100 K | 25.2697 K | 25.1698 K | −0.2402 K |
| $T_c$: Weighted mean | $\frac{(G_1 \times w_1 + G_2 \times w_2)}{(w_1 + w_2)}$ | 25.4585 K | 25.3258 K | 25.2360 K | −0.2225 K |
| FWHM sum | $FWHM_1 + FWHM_2$ | 0.6731 K | 0.7399 K | 0.7253 K | 0.0520 K |
| | | *c*-axis resistivity | | | |
| Centre separation | $G_2 - G_1$ | 0.1949 K | 0.2092 K | 0.2434 K | 0.0458 K |
| Centre mean | $(G_1 + G_2)/2$ | 25.4365 K | 25.3051 K | 25.2679 K | −0.1686 K |
| $T_c$: Weighted mean | $\frac{(G_1 \times w_1 + G_2 \times w_2)}{(w_1 + w_2)}$ | 25.4995 K | 25.3728 K | 25.3531 K | −0.1464 K |
| FWHM sum | $FWHM_1 + FWHM_2$ | 0.7355 K | 0.8351 K | 0.8467 K | 0.1111 K |

Table 4.1: Specifics of the Gaussian peaks fitted to the superconducting transition. $G_1$ and $G_2$ refer to the centres of the respectively fitted Gaussian peak, see figs. 4.3 and 4.4. The "Centre separation" $(G_2 - G_1)$ is therefore the distance between the peak centres and the "Centre mean" is the average temperature of the two peak centres. The weighted centre mean is an average of the temperatures of the two peak centres, weighted with the area under the respective curve of the peak ($w_1$ and $w_2$). This relates to $T_c$, described in the main text. p0, p1 and p2 relate to ambient pressure, $76 \pm 25$ MPa and $90 \pm 35$ MPa, respectively. p2−p0 refers to the difference between the respective values at p2 and p0.

### 4.1.3 Room-Temperature Pressure Dependence

The *ab*-plane and *c*-axis resistivity were measured as a function of pressure within the *ab*-plane at room temperature. For these measurements the gas-driven pressure cell was used in conjunction with a LockIn-Amplifier. Pressure was increased gradually, by cycling with increasing values of the maximal pressure.

In the measured samples, the *c*-axis resistivity was observed to depend on the applied uniaxial pressure (linear decrease with pressure). The data displayed in figs. 4.5 and 4.6 represent an average of 50 measurements. In fig. 4.5a, from the average, every $20^{\text{th}}$ is shown for clarity. The standard deviation of the average is shown as error bars in fig. 4.6b. The data in figs. 4.5a and 4.5b have similar uncertainty of the pressure as in fig. 4.6 – for uncertainty in the *ab*-plane resistivity refer to section 4.1.4.

In fig. 4.6a, the horizontal error bars also include the uncertainty of the sample geometry estimation, which is systematic and thus contributes a constant factor to the overall uncertainty of the applied pressure.

In contrast, the resistivity within the *ab*-plane was not observed to be systematically affected by uniaxial pressure in the investigated pressure range (up to ∼250 MPa) – for details, see section 4.1.4.

**Hysteresis** The pressure dependency of the *c*-axis resistivity shows hysteresis: The difference between measurements during increasing and decreasing pressure amounts to a difference corresponding to 60 and 130 MPa in sample EUG27_He2 and ICP13_He2, respectively. The slope of the resistivity response to applied uniaxial pressure is different for the two measured samples. The same is true for the slopes of the resistivity response to applied uniaxial pressure in the loading and unloading process in the measurement of the sample ICP13_He2: In fig. 4.6b the slopes are indicated by straight lines. Corresponding values of the rate of change of the resistivity are listed in table 4.2.

**Conditioning Effect** There are small conditioning effects observable in all samples: The resistivity in ambient pressure does not fully revert to its original value after a cycle of applied pressure. This effect is systematic within the measurement cycles for each individual sample – the direction of the deviation remains the same for every individual sample. However, those changes were not systematic across the measured samples, as there is a general trend to increased resistivity in EUG27_He2, but trends of decreased resistivity in both EUG27_He4 and ICP13_He2. Moreover, the observed conditioning effects are small compared to the measured values and the observed hysteresis.

| sample | process | colour indication | $\Delta\rho/\Delta p$ [% GPa$^{-1}$] | $\Delta\rho/\Delta p$ [Ω cm GPa$^{-1}$] |
|---|---|---|---|---|
| EUG27_He2 | loading | # cyan | -3.35 | -0.73 |
| | unloading | # red | -3.07 | -0.65 |
| ICP13_He2 | loading | # black | -5.1 | -0.53 |
| | unloading | # yellow | -4.5 | -0.45 |

Table 4.2: Rates of change of the resistivity dependent on pressure. Loading and unloading refer to the process of applying (loading) and releasing pressure (unloading) on a sample. Values are approximate, extracted from manually placed straight lines in fig. 4.6b. The colour indication refers to the same straight lines drawn in fig. 4.6b.

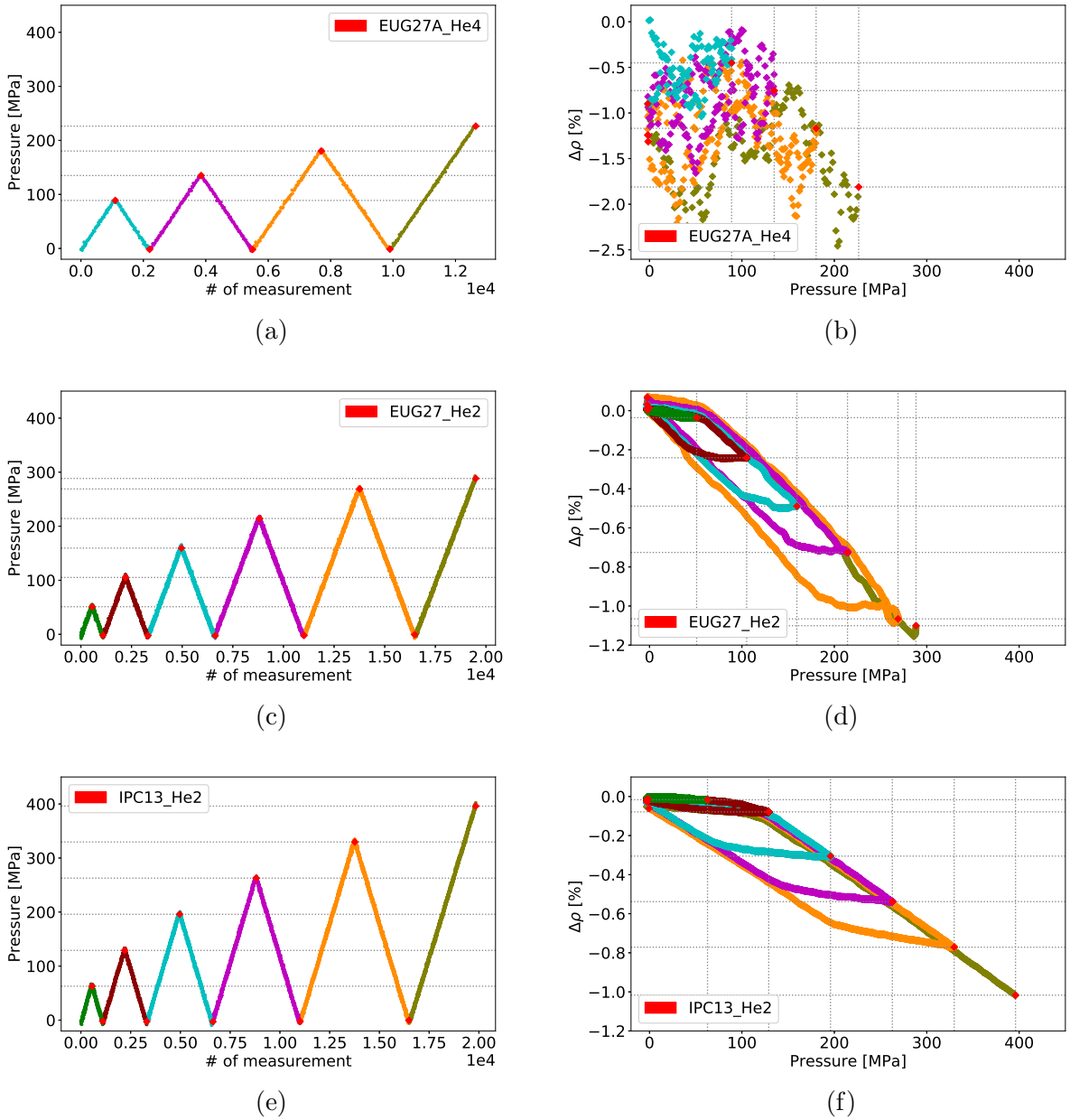
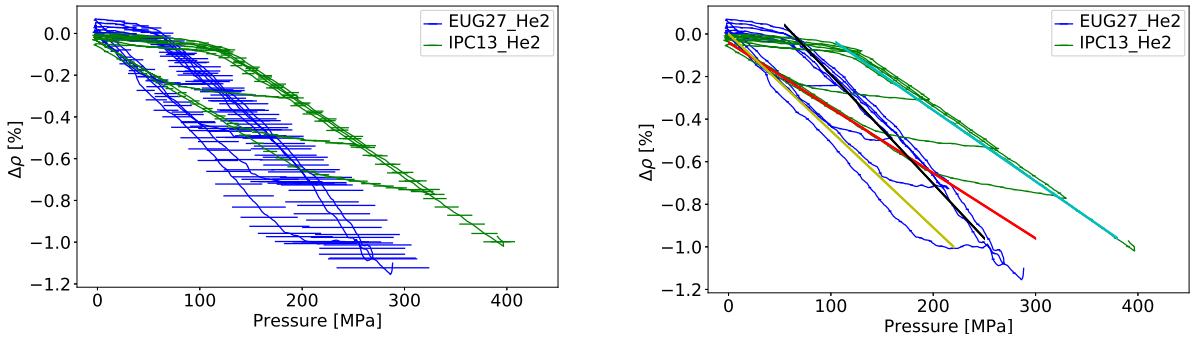
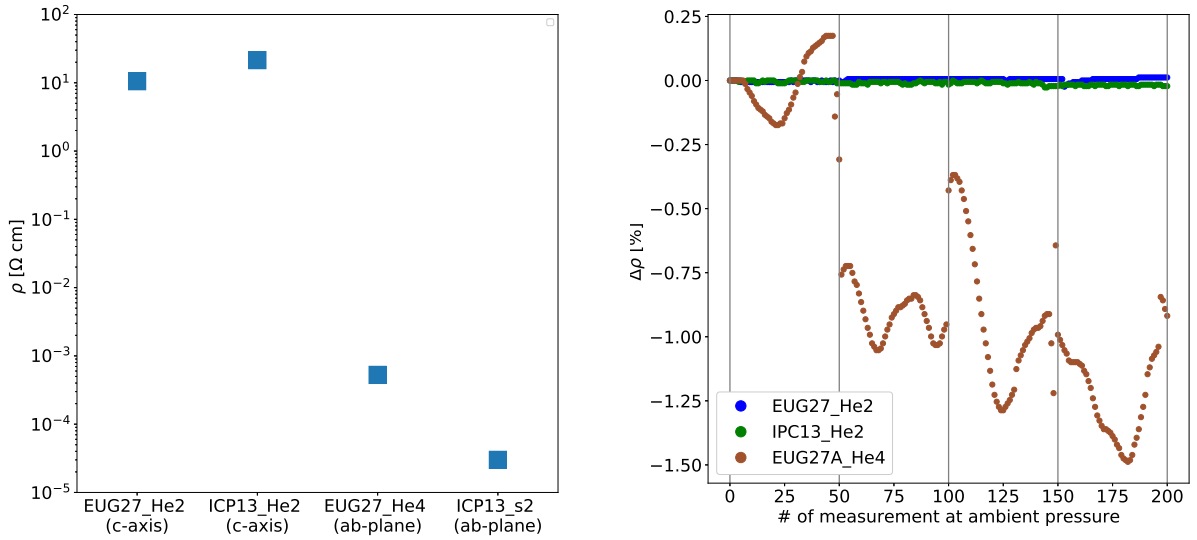



Figure 4.5: Pressure cycles and measured resistivity response at room-temperature. In (a, c, e) the applied pressure is presented. (b, d, f) The dependence of resistivity on applied pressure in percent (in sample EUG27_He4, EUG27_He2, and ICP13_He2 – *ab*-plane, *c*-axis and *c*-axis, respectively), is presented as a function of the corresponding pressure in MPa. The colour codes are the same in plots of the same row. Grey lines and red diamonds are guides to the eye, marking the spots of highest and lowest pressure in a cycle. In (a) and (b), only every $20^{\text{th}}$ point is shown.




(a) (b)

Figure 4.6: The change of resistivity (along $c$-axis) in percent, plotted against the corresponding pressure in MPa. In (a), the error bars include all contributions to the uncertainty, according to the estimated error in measuring the sample geometry. Within the limit of the uncertainty, the data from two samples with different doping levels ($x = 0.11$ and $x = 0.13$) do not overlap. In (b), the error bars include only the standard deviation of the average of 50 measurements for clarity. Straight lines are guides to the eye, placed manually. Corresponding values are listed in table 4.2.




(a) (b)

Figure 4.7: Direct comparisons of observed $ab$-plane and $c$-axis resistivity behaviour. In (a), the absolute values of resistivity (average of all measured points from the respective samples) are shown on a logarithmic scale, demonstrating the difference of about 4 orders of magnitude. As a consequence, presented in (b), the noise level for relative changes in the resistivity (at ambient pressure) is much greater in the $ab$-plane resistivity. The grey lines separate intervals at ambient pressure – one cycle of applying and releasing pressure is thus represented by one grey line.

### 4.1.4 Systematic Errors in the Room-Temperature Pressure dependent Resistivity

**AC Resistivity Precision** In fig. 4.7b, intervals of 50 measurements (not averaged) of the resistivity at ambient pressure are shown, separated with grey lines. The data from the first interval are taken before any pressure is applied. For relative changes in the resistivity at ambient pressure, the noise level in the *ab*-plane resistivity measurement (EUG27_He4) is much higher than in the *c*-axis resistivity measurement. This is due to the *ab*-plane resistivity being much smaller than the *c*-axis resistivity (shown in fig. 4.7a) as well as the limited precision of the resistivity measurement in the ac configuration with the Lock-In amplifier.

For comparison, the average resistivity in the normal state of sample ICP13_b2 (measured in the DC resistivity configuration) is shown in fig. 4.7a too. The *ab*-plane resistivity of sample ICP13_b2 is one order of magnitude lower than the *ab*-plane resistivity of sample EUG27_He4. Therefore, the measurement of the *ab*-plane resistivity of the sample EUG27_He4 represents the noise level of the AC resistivity measurement configuration.

A response of the *ab*-plane resistivity to uniaxial pressure (if it exists) must be smaller than the noise level which is present in the measurement – we can therefore neither confirm nor disprove the existence of such a response. In future experiments, more sensitive methods should be used. This could be achieved either by decreasing the noise level in the ac resistivity measurement configuration with the LockIn-amplifier, or by switching to the dc resistivity measurement setup used for temperature dependent measurements, where a sufficient measurement precision was clearly demonstrated. Due to time constraints (including the availability of the gas-driven pressure cell) it was not possible to implement the proposed measures within the scope of this work.

**Absolute Value of the Pressure in the Gas Driven Pressure Cell** In contrast to diamond anvil pressure cells (DACs), where spectral lines of ruby are used to measure the pressure directly (*126*, *144*), a technique for a concurrent, direct measurement of applied uniaxial pressure was not yet available in the setup used in this work. We want to stress that this thesis is embedded in a wider scope of ongoing research – future designs will rectify shortcomings of the designs presented here.

In the scope of this work, the pressure was estimated based on an estimation of the applied force and a samples' cross-section. The uncertainty in measuring the geometry of samples has a strong impact on the uncertainty of the estimated pressure. This uncertainty contribution will likely stay part of measurements with uniaxial pressure cells in the foreseeable future, regardless of the design of the specific cell. However, this contribution to the uncertainty is constant in pressure ranges where the cross-section of a sample does not change significantly.

In the case of the spring-driven cell, measuring the deformation of the titanium spring introduces an additional source of uncertainty. In the presented configuration, the error arising from these uncertainties is estimated between 5 MPa and 40 MPa, depending on the applied pressure (shown in fig. 4.6).

Additionally, moving parts of the pressure cell (regardless of its type) might take part of the load which is intended for the sample. This effect is not unique to uniaxial pressure cells: In DACs, a plot of achieved pressure against applied external load always shows significant hysteresis (see fig. 4.8). According to the review in ref. (*144*), this is due to changing stress patterns in the used gasket:

> *When the force is reduced, the initial pressure drop is relatively slow. This is because the stress pattern in the gasket must be modified until all of the gasket moves radially*

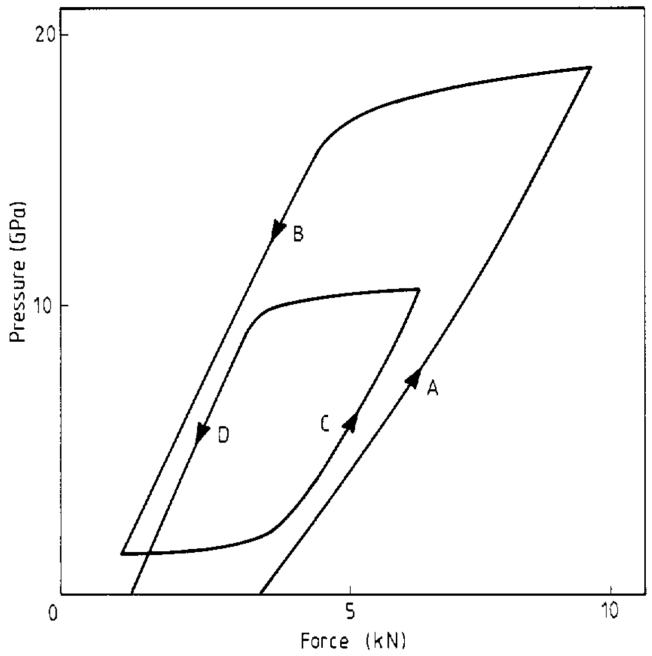



Figure 4.8: A load-pressure diagram typical for DACs when using metal gaskets. The pressure is increased (A), decreased (B), increased again to a lower pressure (C) and reduced to zero (D). Extracted from (*144*).

> *outwards. When this new stress pattern is established, the rate of decrease of pressure with force increases.* (*144*)

Characteristics of the hysteresis are as follows: *i*) The hysteresis is not symmetric in respect to the applied force: The hysteresis loop is biased towards positive values of applied force. *ii*) The slopes of pressure upon loading and unloading the cell differ: In the process of unloading, there is a clear linear behaviour (which is visible in B and D of fig. 4.8) once the reaction of stress patterns in the gasket is completed. However, the loading process appears to be non-linear, and the slope of the dependence of pressure on force differs from the slope when unloading, except for the highest forces applied ($\sim 10\,\mathrm{kN}$, relating to $\sim 20\,\mathrm{GPa}$ in this case).

The effect of such modifications is expected to be negligible in the spring-driven cell, as the achieved pressure is relatively small. In the gas-driven cell however, we expect this effect to become apparent: Between the bellows (where the applied force is introduced) and the sample are clamped connections, and the sample cartridge. We expect these parts (especially the cartridge) to exhibit a similar hysteresis of change in stress patterns, in the range of the applied external loads.

This includes an offset: A certain amount of force needs to be applied to the system before pressure starts to act on the sample. Additionally, the effect which generates the hysteresis contributes to the slope of any measured data (dependent on pressure). The magnitude of these contributions can only be assessed by calibration measurements. Any rate of change dependent on the applied pressure which is derived from uniaxial pressure measurements without a calibration will contain this systematic error.

Therefore, a calibration measurement should be performed, with exactly the same environment as for measurements of samples. Ideally, a design change of the pressure cells which allows *in-situ* pressure measurements should be considered. Due to constraints in time and resources, it was not possible to take such measures within the scope of this work.

## 4.2 Discussion

Measurements under uniaxial pressure are a relatively novel approach to tuning the physical properties of solids. It has been used in the study of magnetic ordering (*145–147*) and thermal expansion and specific heat (*148*). However, the resistivity under uniaxial pressure has not been studied extensively in the cuprates (*149–151*). Because of a lack of literature, the results presented in this thesis can be confirmed only partially for now. More measurements are currently being conducted. The results will be discussed in terms of expectations based on the phenomenological model described in section 2.3.

**Comparison with Literature** Resistivity data of NCCO under uniaxial pressure was hitherto unavailable. However, data of the electrical resistivity of NCCO at ambient pressure can be found (*23*, *135*, *152*). The *ab*-plane resistivity of NCCO at 300 K can be estimated at $3 \times 10^{-4}\,\Omega\,\mathrm{cm}$ and $2 \times 10^{-4}\,\Omega\,\mathrm{cm}$ for doping levels $x = 0.11$ and $x = 0.13$. The corresponding *c*-axis resistivity is $11\,\Omega\,\mathrm{cm}$ and $6\,\Omega\,\mathrm{cm}$, respectively (*152*).

The *ab*-plane resistivity of the measured samples shown in fig. 4.7a amounts to $5.00 \pm 2.96 \times 10^{-4}\,\Omega\,\mathrm{cm}$, and $3.0 \pm 0.1 \times 10^{-5}\,\Omega\,\mathrm{cm}$ (EUG27_He4 and ICP13_s2 at $x = 0.11$ and $x = 0.13$). The *c*-axis resistivity amounts to $10.5 \pm 2.6\,\Omega\,\mathrm{cm}$ and $21.5 \pm 4.4\,\Omega\,\mathrm{cm}$ (EUG27_He2 and ICP13_He2 at $x = 0.11$ and $x = 0.13$). While the *c*-axis resistivity of our sample at $x = 0.11$ is similar to the literature value, the sample at $x = 0.13$ exhibits a higher resistivity than what was observed in ref. (*152*). Moreover, the *c*-axis resistivity at the doping level $x = 0.13$ is higher than at $x = 0.11$, in contrast to the reported behaviour. This could possibly be due to crystal degradation affecting the inclusion layers in the sample ICP13_He2 prior to the experiments presented here.

### 4.2.1 Influence of Uniaxial Pressure on Superconductivity

Temperature dependent resistivity measurements under uniaxial pressure were performed on a superconducting NCCO sample with a doping of $x = 0.13$ ($T_c = 25.3\,\mathrm{K}$). The measurements revealed a monotonic decrease in the critical temperature with the rate $\Delta T_c/\Delta p = -2.5\,\mathrm{K\,GPa^{-1}}$ in the case of the *ab*-plane resistivity and $\Delta T_c/\Delta p = -1.5\,\mathrm{K\,GPa^{-1}}$ for the *c*-axis resistivity. In addition to the shift of $T_c$ towards lower values, the transition becomes broader by about $0.5\,\mathrm{K\,GPa^{-1}}$. Although the exact mechanism of the observed effect requires further study, this behaviour is consistent with the model described in section 2.3. In the discussed model, localised carriers mediate the attractive force between itinerant holes, binding them into Cooper pairs. The distribution of the localized hole is expected to play an important role for the microscopic mechanism of pairing. In the scope of the proposed model, we expect a higher symmetry to lead to higher transition temperatures, and consequently, the breaking of the lattice symmetry should lead to a decrease in $T_c$. We assume that the way the localized hole accommodates within the $CuO_2$ unit is the key part of understanding how it exactly mediates the pairing.

The results of the variation in transition temperature with uniaxial pressure presented here therefore support the proposed theoretical model. However, in order to ascertain the intrinsic properties of the cuprates in regard to the response of $T_c$ under uniaxial pressure, more measurements, including on samples of different compounds (both with and without tetragonal symmetry) should be conducted.

### 4.2.2 Impact of Uniaxial Pressure on Electrical Resistivity at Room Temperature

Due to technical challenges, the uniaxial pressure dependent study of the *ab*-plane and *c*-axis resistivity was performed at room-temperature in $Nd_{2-x}Ce_xCuO_4$ ($x = 0.11$ and $x = 0.13$, as listed in tables 3.1 and 3.2). Although no clear pressure dependence was observed in the *ab*-plane resistivity, the absolute value of the *c*-axis resistivity decreased upon applying uniaxial pressure.

**Hysteresis** The hysteresis of the resistivity response to applied uniaxial pressure bears a striking similarity with the one shown in fig. 4.8, which is observed in diamond anvil pressure cells. The hysteresis in DACs stems from stress patterns which form in the gasket, before there can be a linear dependence between applied load and pressure.

We therefore attribute the observed hysteresis to a similar effect of changing stress patterns, in this case in the clamp system and the sample cartridge. The different slopes of the observed decrease in resistivity (in different samples) could be possibly attributed to distinct reactions to applied pressure in samples of different doping levels. It is important to separate such a contribution from contributions due to a change in the measurement configuration: The sample ICP_He2 was measured with a new, improved cartridge.

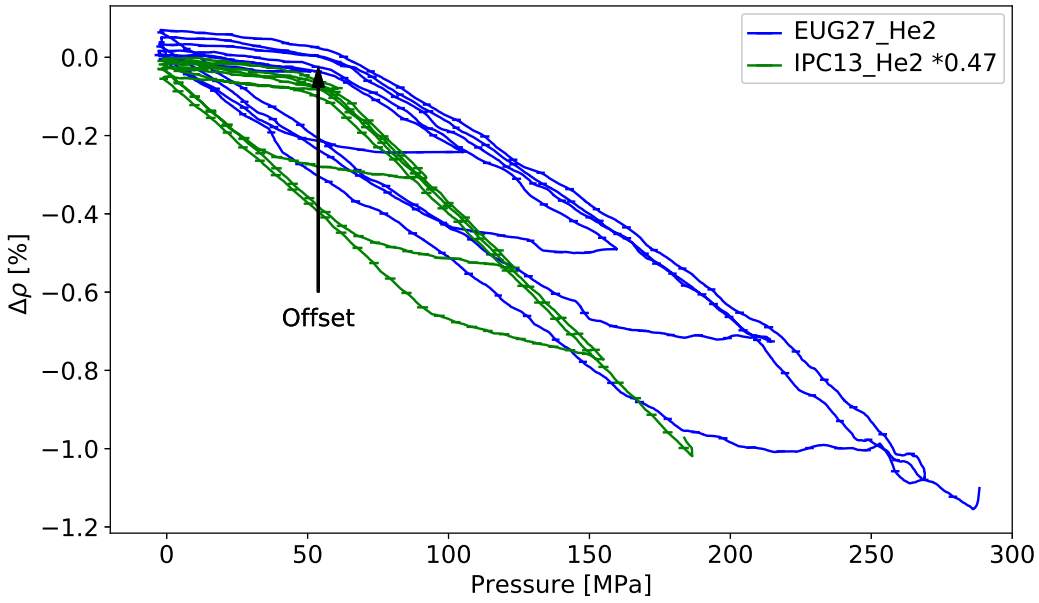
Figure 4.9 demonstrates that two separate scaling factors are necessary to match either *i*) the offset in the hysteresis or *ii*) the slope $\Delta\rho/\Delta p$ [$\%\,\mathrm{GPa}^{-1}$]. The same is true for resistivity data plotted as $\Delta\rho/\Delta p$ [$\Omega\,\mathrm{cm}\,\mathrm{GPa}^{-1}$]. This indicates that the difference in behaviour between the two samples is not entirely due to the different measuring configurations: One single factor matching both offsets and slopes would have pointed to an equal response of the resistivity to uniaxial pressure. Therefore it can be concluded that $\Delta\rho/\Delta p$ differs between different doping levels – however, $\Delta\rho/\Delta p$ is negative in both investigated cases ($x = 0.11$ and $x = 0.13$).

A systematic error contribution from the transmission of pressure from its origin to the sample (in our case clamp system and cartridge, which are between bellows and sample) will only reduce the measured $\Delta\rho/\Delta p$ compared to the true $\Delta\rho/\Delta p$. If the hysteresis is attributed to this systematic error, 3.07 and $4.5\,\%\,\mathrm{GPa}^{-1}$ (absolute values) can be seen as lower bounds to the physical values of $\Delta\rho/\Delta p$ in samples with doping levels $x = 0.11$ and $x = 0.13$, respectively.

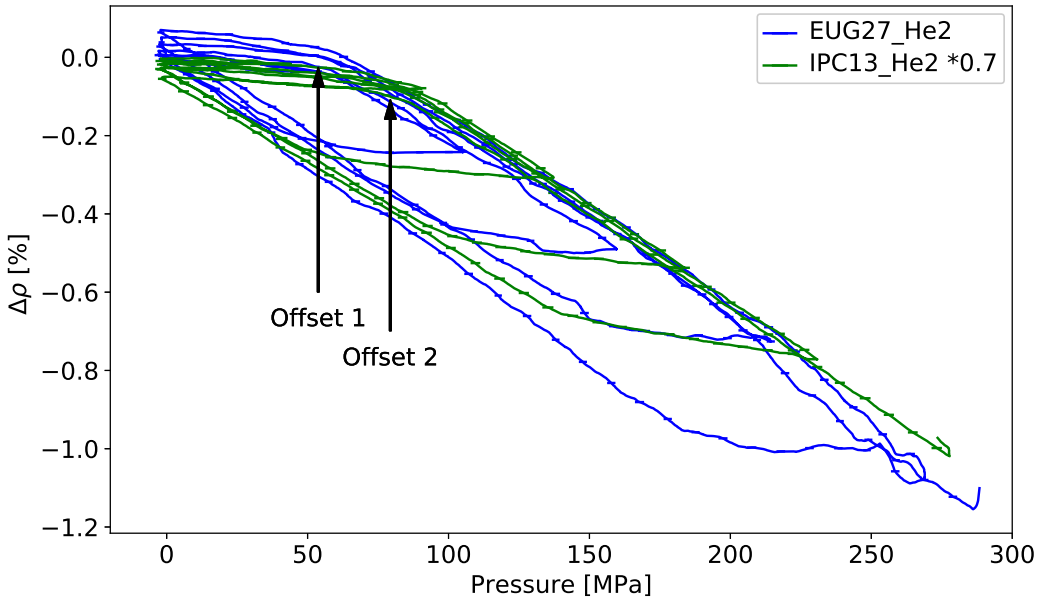
In order to ascertain actual pressures on the samples in future experiments, we advise a careful calibration of the cell and each individual sample cartridge. Furthermore, changes in the design of the sample cartridge, which allow *in-situ* pressure measurements, will be developed.

**Possibility of Plastic Deformation** Ref. (*114*) observed plastic behaviour of LSCO samples under uniaxial pressure above 40 MPa. We could not discern any signature related to plastic deformations in the resistivity measurements of our NCCO samples. However, without data on the relation between stress (pressure) and strain (deformation) no decisive conclusion can be drawn. Corresponding measurements are planned, but were not within the scope of this thesis.

Nevertheless, there are strong indications against plastic deformation: One, is the negative sign of $\Delta\rho/\Delta p$, because plastic deformations and the corresponding structural change in samples are generally expected to yield an increased resistivity, in contrast to the observed behaviour. Another indication is the reversibility of the resistivity: Pressure can be cycled multiple times without systematic lasting impact on the electrical conductivity.




(a)




(b)

Figure 4.9: Rescaled data for qualitative investigations. In (a) and (b), the pressure data of the sample ICP_He2 is scaled with factors 0.47 and 0.7, respectively. The factors were chosen so that in (a) the offsets of the hysteresis are matched, and that in (b) the rates of change of resistivity with applied pressure ($\Delta\rho/\Delta p$ [% $\mathrm{GPa}^{-1}$]) are matched. The arrows indicate the first significant change of the resistivity upon applying pressure.

**Possible Causes of Resistivity Response to Uniaxial Pressure** Various possible explanations for the decreased $c$-axis resistivity as a response to uniaxial pressure exist. Any such explanation should start with a discussion about the mechanism of electrical conductivity along the $c$-axis in cuprates. Presumably, charge carriers from the metallic $CuO_2$ layers tunnel through the insulating charge reservoir layers.

If we consider this tunnelling contribution to the $c$-axis resistivity as dominant, we can approximate the problem with the classic quantum mechanical tunnelling through an energy barrier. The transmission probability for a particle with the energy $E$ to tunnel through an energy barrier of potential $V(x)$ is (*153*):

$$P_T(E) = e^{-2\int_{x_1}^{x_2} \mathrm{d}x \sqrt{\frac{2m}{\hbar^2}[V(x)-E]}}$$

Here, $\hbar$ is the reduced Planck's constant, $x$ denotes the distance in the direction of the particles motion and $m$ is the mass of the moving particle.

From this we can deduce that the number of particles tunnelling through an energy barrier can be increased by variation of specific parameters:

a) Increasing the number of particles available for the tunnelling process. This is in fact independent of the specifics of the tunnelling process.

b) Increasing the energy of the tunnelling particle: $E$.

c) Reducing the energy level of the barrier: $V(x)$.

d) Shortening the barrier (i.e. reducing the distance which a particle has to tunnel through): $x_1 - x_2$.

In our case (i.e. tunnelling of charge carriers through the insulating charge reservoir layers) these parameter variations can be mapped onto:

a) Increasing the density of states close to the Fermi level. This will increase the number of particles which are most likely to participate in the tunnelling process – the ones with the highest energy, $E_F$.

b) Increasing the Fermi level. This will add occupied states at increased energies, above the original highest energy. Therefore, the energy of the particle with the highest probability to tunnel through the barrier is increased.

c) Reducing the energy gap between the Fermi level and the electronic band of the charge reservoir layers which is closest to the Fermi level (i.e. increasing the conductivity of the charge reservoir layers). This directly relates to the maximum of $V(x)$.

d) Reducing the distance between electronic orbitals[1] from adjoining $CuO_2$ layers (i.e. increasing the overlap between those orbitals). This directly relates to $x_1 - x_2$.

From the presented promising results, no definitive conclusion can be drawn as to which of these contributions might be relevant or dominant for the response of the $c$-axis resistivity to uniaxial pressure. All but the last (d) of the mentioned parameter variations would imply a direct dependence of the electronic system on lattice distortions. This would include significant changes to the band structure of the material. Regarding d): An enhanced orbital overlap between $CuO_2$ layers due to applied uniaxial pressure could be understood on the basis of lattice

[1] Considering electronic orbitals which contribute to the conductivity.

distortions alone, which locally reduce the distance between atoms (and thus the orbitals) of different $CuO_2$ layers.

Without additional information regarding the response of NCCO to uniaxial pressure, it is difficult to differentiate between those contributions to the observed effect. Additional measurements under uniaxial pressure, focussing on changes in the mentioned parameters, might bring more insight into those questions.

# 5 Summary and Outlook

The aim of this thesis was to provide reliable evidence for constraints on theories of high temperature superconductivity, by establishing dependencies of physical properties of cuprates on uniaxial pressure. As a primary step, our goal was to develop the uniaxial pressure setups and test them by monitoring the electrical resistivity in cuprates. More specifically, we investigated the dependence of electrical resistivity and $T_c$ of the cuprate $Nd_{2-x}Ce_xCuO_4$ on uniaxial pressure.

To achieve this goal, two different designs of uniaxial pressure cells were developed (spring-driven and gas-driven). The spring-driven cell was deployed in a measurement system with an Oxford Spectromag cryostat for temperature (and magnetic field) dependent resistivity measurements under uniaxial pressure. With this setup, an anti-correlation between $T_c$ and uniaxial pressure was revealed in the cuprate NCCO: As the uniaxial pressure is increased a decrease in $T_c$ is observed. The developed gas-driven pressure cell allows *in-situ* changes of the pressure. For now, it was used to measure the *ab*-plane and *c*-axis resistivity as a function of uniaxial pressure at room-temperature. This measurement revealed a decrease in *c*-axis resistivity upon applying uniaxial pressure within the *ab*-plane.

These results allowed the model described in section 2.3 to be tested: The main conclusion is that the suggested model is supported by the results. The superconducting state is observed to be impaired by uniaxial pressure within the *ab*-plane. Furthermore, the electrical resistivity in the normal state is influenced by global lattice distortions introduced by uniaxial pressure. These effects emerge without apparent lasting changes to the crystal structure (i.e. damage to samples, plastic deformation).

The gas-driven uniaxial pressure cell is planned to be used in conjunction with other measurement techniques too: First and foremost are temperature dependent measurements (like resistivity), which will require a miniaturisation of the pressure cell, to fit into regular cryostats. Additionally planned improvements to the pressure cell include: 1. A detailed calibration of the pressure in respect to the applied gas pressure, and 2. an *in-situ* measurement of the pressure.

A possible next measurement technique which could be approached straightforwardly is the magnetoresistance. Hall coefficient measurements will be distinctly more difficult, as the required contact configuration involves contacts on all four sides (in a plane) of a sample which is not possible in the current design of the gas-driven cell.[1] However, various X-ray measurements are being carried out already. For example, the gas-driven cell was used for RIXS measurements at BESSY (*154*, *155*)). The design of the gas-driven pressure cell opens numerous possibilities to perform surface sensitive measurements (for example ARPES or STM), which are impossible in certain other designs of pressure cells (for example when using piezo stacks due to the required application of high voltages to drive the piezos (*156*)).

[1] Additionally, the sample sizes required for existing designs which allow contacts on all four sides of a plane-shaped sample are not in reach of current synthesis processes for the materials of interest ($HgBa_2CuO_{4+\delta}$).

# Bibliography


1. H. K. Onnes, *Communications from the Physical Laboratory of the University of Leiden* **120b** (1911).
2. W. Meissner, R. Ochsenfeld, German, *Naturwissenschaften* **21**, 787–788, DOI 10.1007/BF01504252 (1933).
3. V. L. Ginzburg, L. D. Landau, *Zh. Eksp. Teor. Fiz.* **20** (1950).
4. L. D. Landau, L. P. Pitaevskii, E. M. Lifshitz, *Statistical physics*, English, Includes index (Oxford [Eng.] : Butterworth- Heinemann, 3rd ed., rev. and enl. / by E.M. Lifshitz and L.P. Pitaevskii, 1980).
5. J. Bardeen, L. N. Cooper, J. R. Schrieffer, en, *Physical Review* **108**, 1175–1204, DOI 10.1103/PhysRev.108.1175 (Dec. 1957).
6. J. Bardeen, L. N. Cooper, J. R. Schrieffer, en, *Physical Review* **106**, 162–164, DOI 10.1103/PhysRev.106.162 (Apr. 1957).
7. E. Maxwell, *Phys. Rev.* **78**, 477–477, DOI 10.1103/PhysRev.78.477 (1950).
8. C. A. Reynolds, B. Serin, W. H. Wright, L. B. Nesbitt, *Phys. Rev.* **78**, 487–487, DOI 10.1103/PhysRev.78.487 (May 1950).
9. N. W. Ashcroft, *Phys. Rev. Lett.* **21**, 1748–1749, DOI 10.1103/PhysRevLett.21.1748 (Dec. 1968).
10. A. P. Drozdov, M. I. Eremets, I. A. Troyan, *ArXiv e-prints* (Dec. 2014).
11. A. P. Drozdov, M. I. Eremets, I. A. Troyan, V. Ksenofontov, S. I. Shylin, *Nature* **525**, Letter, 73–76 (Sept. 2015).
12. J. G. Bednorz, K. A. Müller, *Zeitschrift für Physik B Condensed Matter* **64**, 189–193, DOI 10.1007/BF01303701 (June 1986).
13. Y. Kamihara, T. Watanabe, M. Hirano, H. Hosono, *J. Am. Chem. Soc.* **130**, 3296–3297, DOI 10.1021/ja800073m (Mar. 2008).
14. N. Barišić *et al.*, en, *Physical Review B* **82**, 054518, DOI 10.1103/PhysRevB.82.054518 (Aug. 2010).
15. W. A. Little, *Phys. Rev.* **134**, A1416–A1424, DOI 10.1103/PhysRev.134.A1416 (June 1964).
16. D. Jérome, A. Mazaud, M. Ribault, K. Bechgaard, en, *J. Phyique Lett.* **41**, 95–98, DOI 10.1051/jphyslet:0198000410409500 (Feb. 1980).
17. M.-S. Nam *et al.*, en, *Scientific Reports* **3**, 3390, DOI 10.1038/srep03390 (Dec. 2013).
18. Y. Cao *et al.*, en, *Nature* **556**, 43–50, DOI 10.1038/nature26160 (Apr. 2018).
19. F.-C. Hsu *et al.*, en, *PNAS* **105**, 14262–14264, DOI 10.1073/pnas.0807325105 (Sept. 2008).
20. T. Yajima *et al.*, *J. Phys. Soc. Jpn.* **81**, 103706, DOI 10.1143/JPSJ.81.103706 (Sept. 2012).

21. N. Barišić *et al.*, en, *Proceedings of the National Academy of Sciences* **110**, 12235–12240, DOI 10.1073/pnas.1301989110 (July 2013).

22. Y. Q. Zhang, J. F. Qu, B. M. Wang, X. Q. Xiang, X. G. Li, *Journal of Applied Physics* **102**, 063910, DOI 10.1063/1.2783988 (Sept. 2007).

23. Y. Li *et al.*, *Sci Adv* **5**, eaap7349, DOI 10.1126/sciadv.aap7349 (Feb. 2019).

24. P. A. Lee, N. Nagaosa, X.-G. Wen, *Rev. Mod. Phys.* **78**, 17–85, DOI 10.1103/RevModPhys.78.17 (Jan. 2006).

25. N. F. Mott, *Proceedings of the Physical Society. Section A* **62**, 416–422, DOI 10.1088/0370-1298/62/7/303 (July 1949).

26. A. Fujimori, E. Takayama-Muromachi, Y. Uchida, B. Okai, *Phys. Rev. B* **35**, 8814–8817, DOI 10.1103/PhysRevB.35.8814 (June 1987).

27. T. Takahashi *et al.*, *Phys. Rev. B* **36**, 5686–5689, DOI 10.1103/PhysRevB.36.5686 (Oct. 1987).

28. A. Bianconi *et al.*, *Solid State Communications* **63**, 1135–1139, DOI 10.1016/0038-1098(87)91063-5 (Sept. 1987).

29. J. Zaanen, G. A. Sawatzky, J. W. Allen, *Phys. Rev. Lett.* **55**, 418–421, DOI 10.1103/PhysRevLett.55.418 (July 1985).

30. S. Edkins, *Visualising the Charge and Cooper-Pair Density Waves in Cuprates*, en (Springer International Publishing, 2017).

31. D. Vaknin *et al.*, *Phys. Rev. Lett.* **58**, 2802–2805, DOI 10.1103/PhysRevLett.58.2802 (June 1987).

32. S. Chakravarty, B. I. Halperin, D. R. Nelson, *Phys. Rev. Lett.* **60**, 1057–1060, DOI 10.1103/PhysRevLett.60.1057 (Mar. 1988).

33. N. Barišić *et al.*, *arXiv e-prints*, arXiv:1507.07885 (July 2015).

34. C. E. Gough *et al.*, *Nature* **326**, 855–855, DOI 10.1038/326855a0 (Apr. 1987).

35. C. C. Tsuei, J. R. Kirtley, *Rev. Mod. Phys.* **72**, 969–1016, DOI 10.1103/RevModPhys.72.969 (Oct. 2000).

36. D. A. Wollman, D. J. Van Harlingen, W. C. Lee, D. M. Ginsberg, A. J. Leggett, *Phys. Rev. Lett.* **71**, 2134–2137, DOI 10.1103/PhysRevLett.71.2134 (Sept. 1993).

37. A. Damascelli, Z. Hussain, Z.-X. Shen, *Rev. Mod. Phys.* **75**, 473–541, DOI 10.1103/RevModPhys.75.473 (Apr. 2003).

38. S. Nakamae *et al.*, en, *Physical Review B* **68**, R100502, DOI 10.1103/PhysRevB.68.100502 (Sept. 2003).

39. M. Platé *et al.*, en, *Physical Review Letters* **95**, DOI 10.1103/PhysRevLett.95.077001 (Aug. 2005).

40. B. Vignolle *et al.*, *Nature* **455**, 952–955, DOI 10.1038/nature07323 (Oct. 2008).

41. C. O. Rodriguez, *Phys. Rev. B* **49**, 1200–1204, DOI 10.1103/PhysRevB.49.1200 (Jan. 1994).

42. N. E. Hussey, K. Takenaka, H. Takagi, *Philosophical Magazine* **84**, 2847–2864, DOI 10.1080/14786430410001716944 (Sept. 2004).

43. M. Gurvitch, A. T. Fiory, *Phys. Rev. Lett.* **59**, 1337–1340, DOI 10.1103/PhysRevLett.59.1337 (Sept. 1987).

*44.* S. Martin, A. T. Fiory, R. M. Fleming, L. F. Schneemeyer, J. V. Waszczak, *Phys. Rev. B* **41**, 846–849, DOI 10.1103/PhysRevB.41.846 (Jan. 1990).

*45.* A. Legros *et al.*, En, *Nature Physics* **15**, 142, DOI 10.1038/s41567-018-0334-2 (Feb. 2019).

*46.* C. Renner, B. Revaz, J.-Y. Genoud, K. Kadowaki, Ø. Fischer, *Phys. Rev. Lett.* **80**, 149–152, DOI 10.1103/PhysRevLett.80.149 (Jan. 1998).

*47.* M. R. Norman *et al.*, *Nature* **392**, 157 (Mar. 1998).

*48.* M. R. Norman, H. Ding, *Phys. Rev. B* **57**, R11089–R11092, DOI 10.1103/PhysRevB.57.R11089 (May 1998).

*49.* B. Keimer, S. A. Kivelson, M. R. Norman, S. Uchida, J. Zaanen, *Nature* **518**, 179–186, DOI 10.1038/nature14165 (Feb. 2015).

*50.* K. Held, *Advances in Physics* **56**, 829–926, DOI 10.1080/00018730701619647 (2007).

*51.* K. Lejaeghere *et al.*, en, *Science* **351**, aad3000, DOI 10.1126/science.aad3000 (Mar. 2016).

*52.* G. Rohringer, PhD thesis, TU Vienna, 2014.

*53.* G. Rohringer *et al.*, *Rev. Mod. Phys.* **90**, 025003, DOI 10.1103/RevModPhys.90.025003 (May 2018).

*54.* J. M. Tranquada, *AIP Conference Proceedings* **1550**, 114–187, DOI 10.1063/1.4818402 (Aug. 2013).

*55.* E. Fradkin, S. A. Kivelson, J. M. Tranquada, *Rev. Mod. Phys.* **87**, 457–482, DOI 10.1103/RevModPhys.87.457 (May 2015).

*56.* V. Oganesyan, S. A. Kivelson, E. Fradkin, *Phys. Rev. B* **64**, 195109, DOI 10.1103/PhysRevB.64.195109 (Oct. 2001).

*57.* J. M. Tranquada, B. J. Sternlieb, J. D. Axe, Y. Nakamura, S. Uchida, En, *Nature* **375**, 561, DOI 10.1038/375561a0 (June 1995).

*58.* A. R. Moodenbaugh, Y. Xu, M. Suenaga, T. J. Folkerts, R. N. Shelton, *Phys. Rev. B* **38**, 4596–4600, DOI 10.1103/PhysRevB.38.4596 (Sept. 1988).

*59.* W. Tabis *et al.*, *Nature Communications* **5**, 5875, DOI 10.1038/ncomms6875 (Dec. 2014).

*60.* W. A. Atkinson, A. P. Kampf, S. Bulut, en, *New J. Phys.* **17**, 013025, DOI 10.1088/1367-2630/17/1/013025 (Jan. 2015).

*61.* P. Abbamonte *et al.*, en, *Nature Physics* **1**, 155–158, DOI 10.1038/nphys178 (Dec. 2005).

*62.* C. Howald, H. Eisaki, N. Kaneko, M. Greven, A. Kapitulnik, *Phys. Rev. B* **67**, 014533, DOI 10.1103/PhysRevB.67.014533 (Jan. 2003).

*63.* W. D. Wise *et al.*, en, *Nature Physics* **4**, 696–699, DOI 10.1038/nphys1021 (Sept. 2008).

*64.* J. E. Hoffman *et al.*, en, *Science* **295**, 466–469, DOI 10.1126/science.1066974 (Jan. 2002).

*65.* T. Hanaguri *et al.*, en, *Nature* **430**, 1001–1005, DOI 10.1038/nature02861 (Aug. 2004).

*66.* G. Ghiringhelli *et al.*, en, *Science* **337**, 821–825, DOI 10.1126/science.1223532 (Aug. 2012).

*67.* E. Blackburn *et al.*, *Phys. Rev. Lett.* **110**, 137004, DOI 10.1103/PhysRevLett.110.137004 (Mar. 2013).

*68.* J. Chang *et al.*, en, *Nature Physics* **8**, 871–876, DOI 10.1038/nphys2456 (Dec. 2012).

69. S. E. Sebastian, N. Harrison, G. G. Lonzarich, en, *Rep. Prog. Phys.* **75**, 102501, DOI 10.1088/0034-4885/75/10/102501 (Sept. 2012).

70. S. Gerber *et al.*, en, *Science* **350**, 949–952, DOI 10.1126/science.aac6257 (Nov. 2015).

71. J. Chang *et al.*, en, *Nature Communications* **7**, 11494, DOI 10.1038/ncomms11494 (May 2016).

72. H. Jang *et al.*, en, *PNAS* **113**, 14645–14650, DOI 10.1073/pnas.1612849113 (Dec. 2016).

73. T. Wu *et al.*, en, *Nature* **477**, 191–194, DOI 10.1038/nature10345 (Sept. 2011).

74. B. J. Ramshaw *et al.*, en, *Science* **348**, 317–320, DOI 10.1126/science.aaa4990 (Apr. 2015).

75. M. Vojta, *Advances in Physics* **58**, 699–820, DOI 10.1080/00018730903122242 (Nov. 2009).

76. R. Comin, A. Damascelli, *Annu. Rev. Condens. Matter Phys.* **7**, 369–405, DOI 10.1146/annurev-conmatphys-031115-011401 (Mar. 2016).

77. W. Tabis *et al.*, en, *Physical Review B* **96**, 134510, DOI 10.1103/PhysRevB.96.134510 (Oct. 2017).

78. N. Doiron-Leyraud *et al.*, en, *Nature* **447**, 565–568, DOI 10.1038/nature05872 (May 2007).

79. N. Barišić *et al.*, *Nature Physics* **9**, 761–764, DOI 10.1038/nphys2792 (Nov. 2013).

80. I. M. Vishik, *What are the latest theories in high temperature superconductivity? - Quora*.

81. S. Chakravarty, R. B. Laughlin, D. K. Morr, C. Nayak, *Phys. Rev. B* **63**, 094503, DOI 10.1103/PhysRevB.63.094503 (Jan. 2001).

82. Y. He, C. M. Varma, *Phys. Rev. Lett.* **106**, 147001, DOI 10.1103/PhysRevLett.106.147001 (Apr. 2011).

83. C M Varma, *Journal of Physics: Condensed Matter* **26**, 505701 (2014).

84. R. B. Laughlin, en, *Physical Review B* **89**, 035134, DOI 10.1103/PhysRevB.89.035134 (Jan. 2014).

85. O. Gunnarsson *et al.*, *Phys. Rev. Lett.* **114**, 236402, DOI 10.1103/PhysRevLett.114.236402 (June 2015).

86. A. Toschi, P. Barone, M. Capone, C. Castellani, en, *New J. Phys.* **7**, 7–7, DOI 10.1088/1367-2630/7/1/007 (Jan. 2005).

87. S. Sachdev, B. Keimer, en, *Physics Today* **64**, 29, DOI 10.1063/1.3554314 (Feb. 2011).

88. B. Michon *et al.*, En, *Nature* **567**, 218, DOI 10.1038/s41586-019-0932-x (Mar. 2019).

89. V. J. Emery, S. A. Kivelson, En, *Nature* **374**, 434, DOI 10.1038/374434a0 (Mar. 1995).

90. J. Corson, R. Mallozzi, J. Orenstein, J. N. Eckstein, I. Bozovic, En, *Nature* **398**, 221, DOI 10.1038/18402 (Mar. 1999).

91. M. S. Grbić *et al.*, *Phys. Rev. B* **80**, 094511, DOI 10.1103/PhysRevB.80.094511 (Sept. 2009).

92. L. S. Bilbro *et al.*, en, *Nature Physics* **7**, 298–302, DOI 10.1038/nphys1912 (Apr. 2011).

93. J. Orenstein, J. Corson, S. Oh, J. N. Eckstein, en, *Annalen der Physik* **15**, 596–605, DOI 10.1002/andp.200510202 (2006).

*94.* M. S. Grbić *et al.*, *Phys. Rev. B* **83**, 144508, DOI 10.1103/PhysRevB.83.144508 (Apr. 2011).

*95.* G. Yu *et al.*, *arXiv:1710.10957 [cond-mat]*, arXiv: 1710.10957 (Oct. 2017).

*96.* I. Kokanović, D. J. Hills, M. L. Sutherland, R. Liang, J. R. Cooper, *Phys. Rev. B* **88**, 060505, DOI 10.1103/PhysRevB.88.060505 (Aug. 2013).

*97.* V. Balédent *et al.*, *Phys. Rev. B* **83**, 104504, DOI 10.1103/PhysRevB.83.104504 (Mar. 2011).

*98.* J. Xia *et al.*, *Phys. Rev. Lett.* **100**, 127002, DOI 10.1103/PhysRevLett.100.127002 (Mar. 2008).

*99.* G. Coslovich *et al.*, *Phys. Rev. Lett.* **110**, 107003, DOI 10.1103/PhysRevLett.110.107003 (Mar. 2013).

*100.* R. Daou *et al.*, en, *Nature* **463**, 519–522, DOI 10.1038/nature08716 (Jan. 2010).

*101.* W. S. Lee *et al.*, *Nature* **450**, 81–84, DOI 10.1038/nature06219 (Nov. 2007).

*102.* T. Kondo, R. Khasanov, T. Takeuchi, J. Schmalian, A. Kaminski, en, *Nature* **457**, 296–300, DOI 10.1038/nature07644 (Jan. 2009).

*103.* M. Hashimoto *et al.*, en, *Nature Physics* **6**, 414–418, DOI 10.1038/nphys1632 (June 2010).

*104.* K. Tanaka *et al.*, en, *Science* **314**, 1910–1913, DOI 10.1126/science.1133411 (Dec. 2006).

*105.* M. Le Tacon *et al.*, *Nature Physics* **2**, 537 (July 2006).

*106.* T. Yoshida *et al.*, *Phys. Rev. Lett.* **103**, 037004, DOI 10.1103/PhysRevLett.103.037004 (July 2009).

*107.* T. Kondo, T. Takeuchi, A. Kaminski, S. Tsuda, S. Shin, *Phys. Rev. Lett.* **98**, 267004, DOI 10.1103/PhysRevLett.98.267004 (June 2007).

*108.* D. Pelc *et al.*, *Nature Communications* **9**, 4327, DOI 10.1038/s41467-018-06707-y (Oct. 2018).

*109.* D. Pelc, P. Popčević, M. Požek, M. Greven, N. Barišić, *Sci Adv* **5**, eaau4538, DOI 10.1126/sciadv.aau4538 (Jan. 2019).

*110.* P. Popčević *et al.*, *npj Quantum Materials* **3**, 42, DOI 10.1038/s41535-018-0115-2 (Sept. 2018).

*111.* Y. Ando, Y. Kurita, S. Komiya, S. Ono, K. Segawa, en, *Physical Review Letters* **92**, 197001, DOI 10.1103/PhysRevLett.92.197001 (May 2004).

*112.* A. P. Mackenzie, S. R. Julian, D. C. Sinclair, C. T. Lin, en, *Physical Review B* **53**, 5848–5855, DOI 10.1103/PhysRevB.53.5848 (Mar. 1996).

*113.* D. Pelc *et al.*, *arXiv e-prints*, arXiv:1902.00529 (Feb. 2019).

*114.* D. Pelc, Z. Anderson, B. Yu, C. Leighton, M. Greven, *arXiv:1808.05763 [cond-mat]*, arXiv: 1808.05763 (Aug. 2018).

*115.* K. K. Gomes *et al.*, *Nature* **447**, 569–572, DOI 10.1038/nature05881 (May 2007).

*116.* A. N. Pasupathy *et al.*, *Science* **320**, 196, DOI 10.1126/science.1154700 (Apr. 2008).

*117.* M. R. Presland, J. L. Tallon, R. G. Buckley, R. S. Liu, N. E. Flower, *Physica C: Superconductivity* **176**, 95–105, DOI 10.1016/0921-4534(91)90700-9 (May 1991).

*118.* W. A. Little, en, *J Supercond Nov Magn* **29**, 3–8, DOI 10.1007/s10948-015-3261-9 (Jan. 2016).

*119.* W. A. Little *et al.*, *Physica C: Superconductivity and its Applications*, Proceedings of the 8th International Conference on Materials and Mechanisms of Superconductivity and High Temperature Superconductors **460-462**, 40–43, DOI 10.1016/j.physc.2007.03.031 (Sept. 2007).

*120.* B. Büchner, M. Breuer, A. Freimuth, A. P. Kampf, *Phys. Rev. Lett.* **73**, 1841–1844, DOI 10.1103/PhysRevLett.73.1841 (Sept. 1994).

*121.* N. L. Saini, A. Bianconi, H. Oyanagi, *J. Phys. Soc. Jpn.* **70**, 2092–2097, DOI 10.1143/JPSJ.70.2092 (July 2001).

*122.* F. Carbone, N. Gedik, J. Lorenzana, A. H. Zewail, *Real-Time Observation of Cuprates Structural Dynamics by Ultrafast Electron Crystallography*, en, Research article, 2010, DOI 10.1155/2010/958618.

*123.* H. Eisaki *et al.*, *Phys. Rev. B* **69**, 064512, DOI 10.1103/PhysRevB.69.064512 (Feb. 2004).

*124.* K. Fujita, T. Noda, K. M. Kojima, H. Eisaki, S. Uchida, *Phys. Rev. Lett.* **95**, 097006, DOI 10.1103/PhysRevLett.95.097006 (Aug. 2005).

*125.* N. Poccia *et al.*, en, *PNAS* **109**, 15685–15690, DOI 10.1073/pnas.1208492109 (Sept. 2012).

*126.* D. J. Dunstan, I. L. Spain, *Journal of Physics E: Scientific Instruments* **22**, 913–923, DOI 10.1088/0022-3735/22/11/004 (Nov. 1989).

*127.* Y. Tokura, H. Takagi, S. Uchida, En, *Nature* **337**, 345, DOI 10.1038/337345a0 (Jan. 1989).

*128.* T. Helm, Dissertation, Technische Universität München, 2013.

*129.* P. K. Mang, O. P. Vajk, A. Arvanitaki, J. W. Lynn, M. Greven, *Phys. Rev. Lett.* **93**, 027002, DOI 10.1103/PhysRevLett.93.027002 (July 2004).

*130.* M. A. Kastner, R. J. Birgeneau, G. Shirane, Y. Endoh, *Rev. Mod. Phys.* **70**, 897–928, DOI 10.1103/RevModPhys.70.897 (July 1998).

*131.* E. M. Motoyama *et al.*, en, *Nature* **445**, 186–189, DOI 10.1038/nature05437 (Jan. 2007).

*132.* N. P. Armitage *et al.*, *Phys. Rev. Lett.* **88**, 257001, DOI 10.1103/PhysRevLett.88.257001 (June 2002).

*133.* H. Matsui *et al.*, *Phys. Rev. B* **75**, 224514, DOI 10.1103/PhysRevB.75.224514 (June 2007).

*134.* D. Song *et al.*, *Phys. Rev. Lett.* **118**, 137001, DOI 10.1103/PhysRevLett.118.137001 (Mar. 2017).

*135.* Y. Onose, Y. Taguchi, K. Ishizaka, Y. Tokura, *Phys. Rev. Lett.* **87**, 217001, DOI 10.1103/PhysRevLett.87.217001 (Oct. 2001).

*136.* G. M. Luke *et al.*, *Phys. Rev. B* **42**, 7981–7988, DOI 10.1103/PhysRevB.42.7981 (Nov. 1990).

*137.* Y. Li, en, PhD thesis, University of Minnesota, May 2017.

*138.* M. Abe, K. Kumagai, S. Awaji, T. Fujita, *Physica C: Superconductivity* **160**, 8–16, DOI 10.1016/0921-4534(89)90445-0 (Aug. 1989).

*139.* W. Henggeler *et al.*, en, *EPL* **29**, 233–238, DOI 10.1209/0295-5075/29/3/008 (Jan. 1995).

*140.* S. J. L. Billinge, T. Egami, *Phys. Rev. B* **47**, 14386–14406, DOI 10.1103/PhysRevB.47.14386 (June 1993).

*141.* P. Lightfoot *et al.*, *Physica C: Superconductivity* **168**, 627–636, DOI 10.1016/0921-4534(90)90087-U (July 1990).

*142.* T. Helm *et al.*, *Phys. Rev. B* **92**, 094501, DOI 10.1103/PhysRevB.92.094501 (Sept. 2015).

*143.* W. Jiang *et al.*, *Phys. Rev. Lett.* **73**, 1291–1294, DOI 10.1103/PhysRevLett.73.1291 (Aug. 1994).

*144.* I. L. Spain, D. J. Dunstan, *Journal of Physics E: Scientific Instruments* **22**, 923–933, DOI 10.1088/0022-3735/22/11/005 (Nov. 1989).

*145.* D. W. Tam *et al.*, *Phys. Rev. B* **95**, 060505, DOI 10.1103/PhysRevB.95.060505 (Feb. 2017).

*146.* A. Chacon *et al.*, *Phys. Rev. Lett.* **115**, 267202, DOI 10.1103/PhysRevLett.115.267202 (Dec. 2015).

*147.* J. Park, H. Sakai, A. P. Mackenzie, C. W. Hicks, *Phys. Rev. B* **98**, 024426, DOI 10.1103/PhysRevB.98.024426 (July 2018).

*148.* M. He *et al.*, en, *J. Phys.: Condens. Matter* **30**, 385702, DOI 10.1088/1361-648X/aada1e (Sept. 2018).

*149.* H. Taniguchi *et al.*, *Phys. Rev. B* **88**, 205111, DOI 10.1103/PhysRevB.88.205111 (Nov. 2013).

*150.* M. E. Barber, A. S. Gibbs, Y. Maeno, A. P. Mackenzie, C. W. Hicks, *Phys. Rev. Lett.* **120**, 076602, DOI 10.1103/PhysRevLett.120.076602 (Feb. 2018).

*151.* H.-H. Kim *et al.*, en, *Science* **362**, 1040–1044, DOI 10.1126/science.aat4708 (Nov. 2018).

*152.* Y. Onose, Y. Taguchi, K. Ishizaka, Y. Tokura, en, *Phys. Rev. B* **69**, 024504, DOI 10.1103/PhysRevB.69.024504 (Jan. 2004).

*153.* F. Schwabl, *Quantenmechanik (QM I)*, de (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007), DOI 10.1007/978-3-540-73675-2.

*154.* U. Mueller *et al.*, en, *Eur. Phys. J. Plus* **130**, 141, DOI 10.1140/epjp/i2015-15141-2 (July 2015).

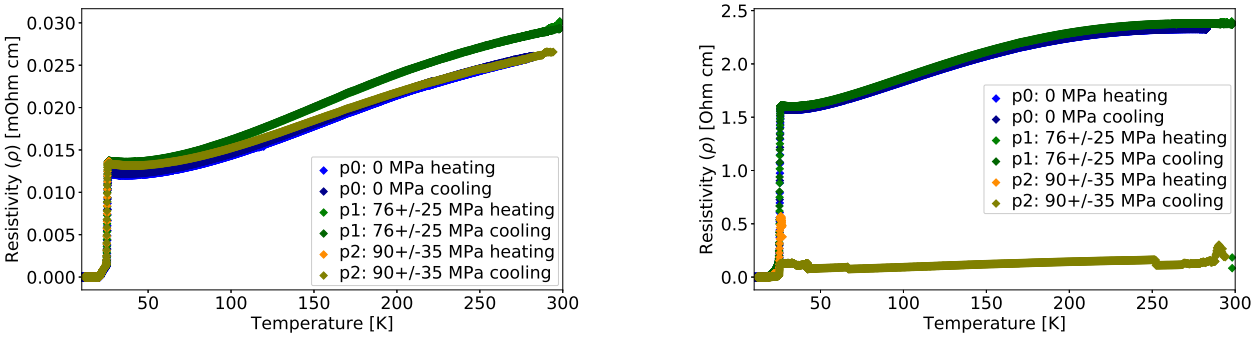
*155.* W. Tabis, *Manuscript in preparation.*

*156.* M. E. Barber, A. Steppke, A. P. Mackenzie, C. W. Hicks, *Review of Scientific Instruments* **90**, 023904, DOI 10.1063/1.5075485 (Feb. 2019).

# 6 Appendix

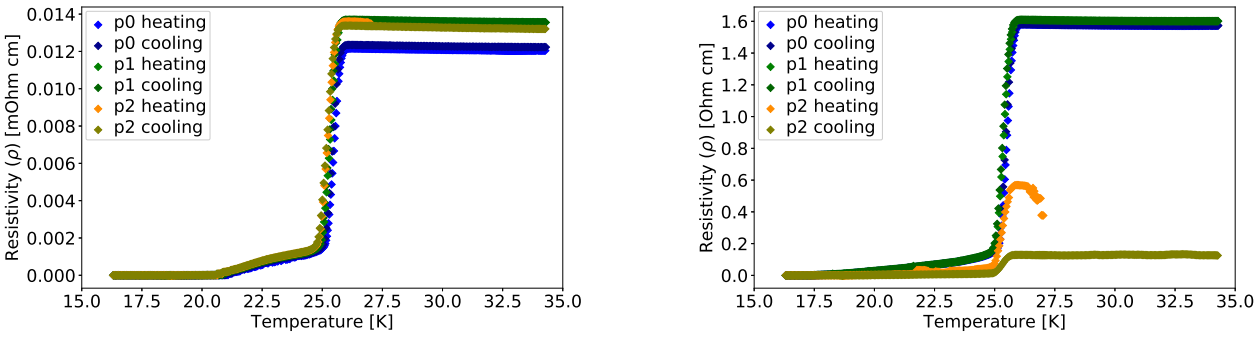
## A Data Analysis

In this section, the data analysis cycle as shown in section 4.1.1 (fig. 4.1) is demonstrated on the complete dataset.



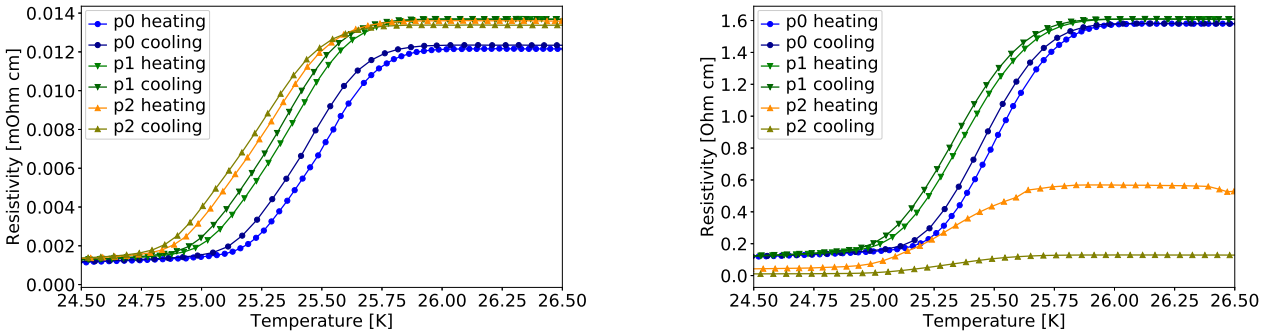

(a) *ab*-plane resistivity (b) *c*-axis resistivity

Figure A.1: Resistivity of the sample ICP13_s2 ($x = 0.13$), measured within the *ab*-plane (a) and along the *c*-axis (b) of the sample. p0 relates to ambient pressure, p1 and p2 are at $76 \pm 25$ MPa and $90 \pm 35$ MPa respectively. "heating" and "cooling" specify data which was measured during heating and cooling cycles, respectively. We did encounter difficulties with the measurement along the *c*-axis at p2, which strongly deviates from the other data. However, our main concern is the superconducting transition (which is not strongly affected), therefore we can safely neglect this issue.



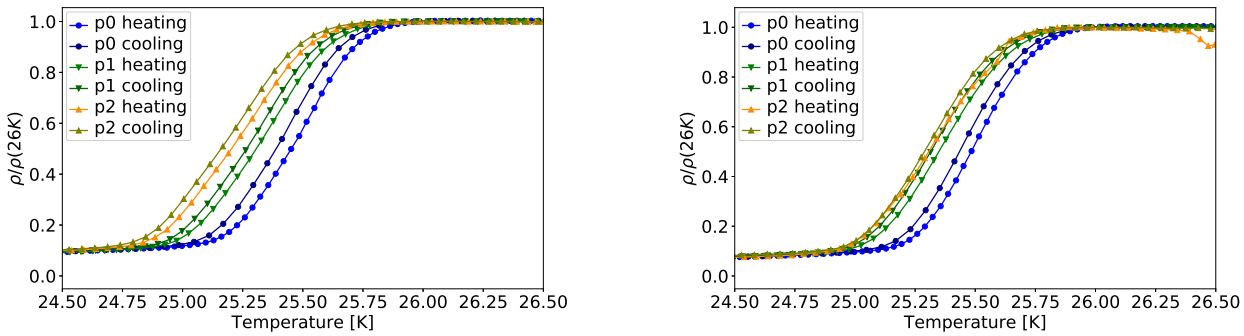

(a) *ab*-plane resistivity (b) *c*-axis resistivity

Figure A.2: The superconducting transition in greater detail. We estimate the transition midpoint around 25.3 K (see main text and subsequent figures). In the low-temperature side there is a tail which extends to about 21 K, which can be attributed to inhomogeneity in the doping of the sample. Lines are direct connections between measured points.



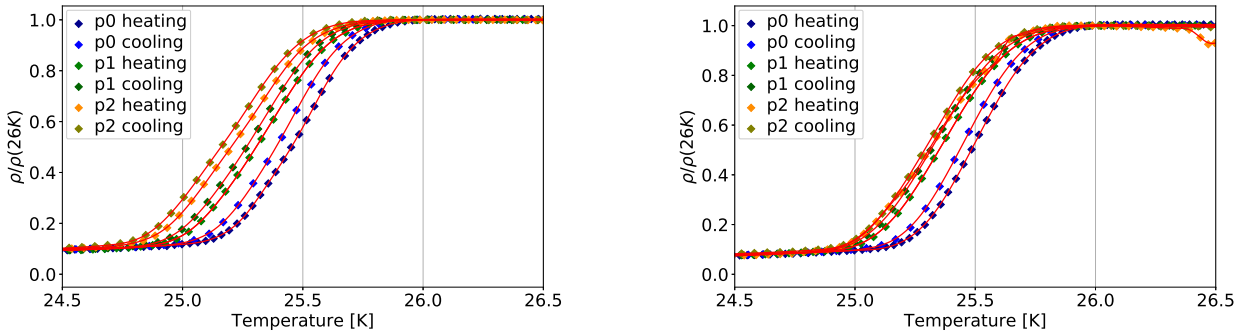

(a) *ab*-plane resistivity (b) *c*-axis resistivity

Figure A.3: The part of the raw data which is closely investigated, unchanged (see main text). The hysteresis due to small temperature gradients between sample and temperature sensor amounts to ∼0.06 K, ∼0.04 K and ∼0.038 K in p0, p1 and p2 respectively. Lines are direct connections between measured points.



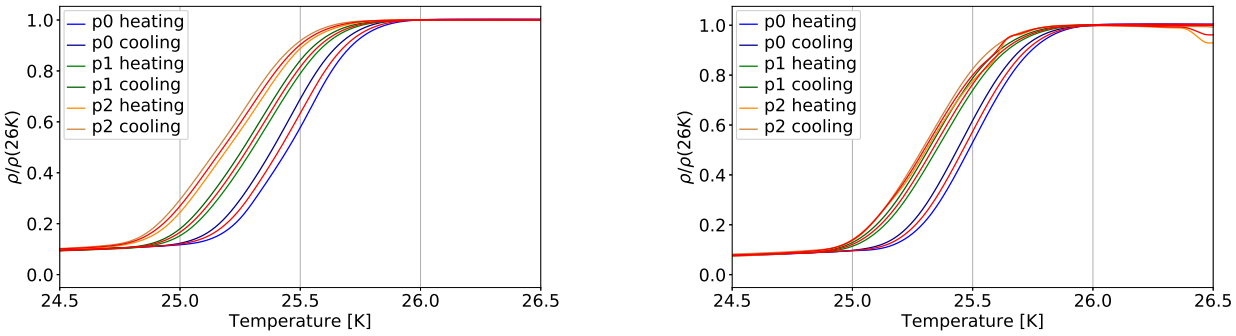

(a) *ab*-plane resistivity (b) *c*-axis resistivity

Figure A.4: Electrical resistivity around the transition temperature, divided by its value at 26 K. Lines are direct connections between measured points.




(a) *ab*-plane resistivity (b) *c*-axis resistivity

Figure A.5: Spline interpolation (red lines) of the electrical resistivity around the superconducting transition $T_c$. Data collected during heating and cooling is interpolated individually.




(a) *ab*-plane resistivity

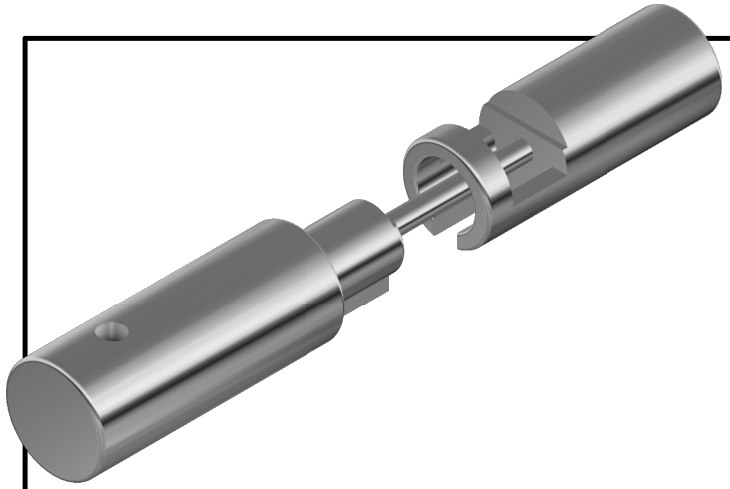
(b) *c*-axis resistivity

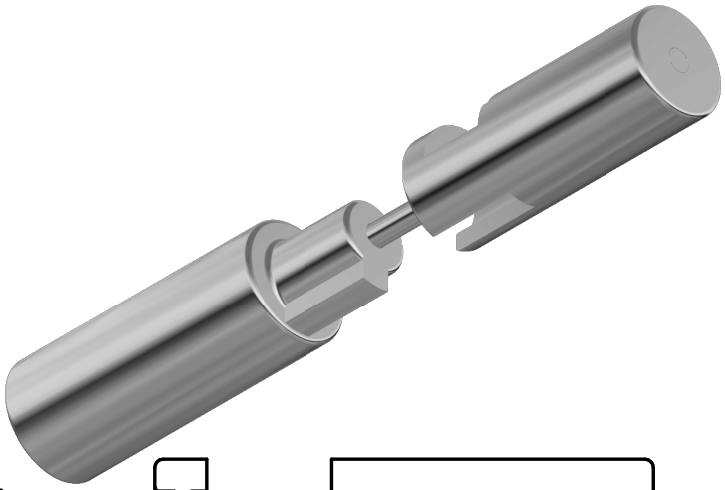
Figure A.6: Averaged values of the electrical resistivity from heating and cooling cycles (red line), together with the corresponding measured data.

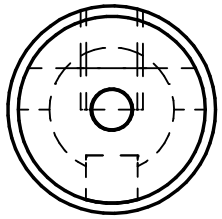
# B CAD drawings

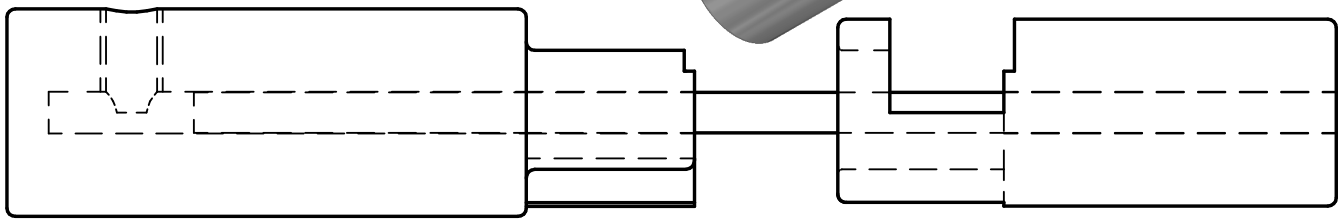
## B.1 New Sample Cartridge for the Gas Cell

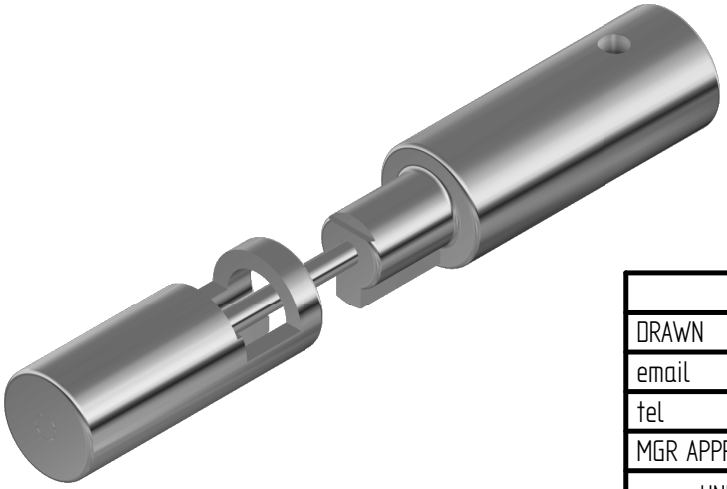
## B.2 Indium Press











| | NAME | DATE |
|---|---|---|
| DRAWN | Benjamin | 13/02/19 |
| email | | |
| tel | 13724 / 13157 | |
| MGR APPR | | |

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

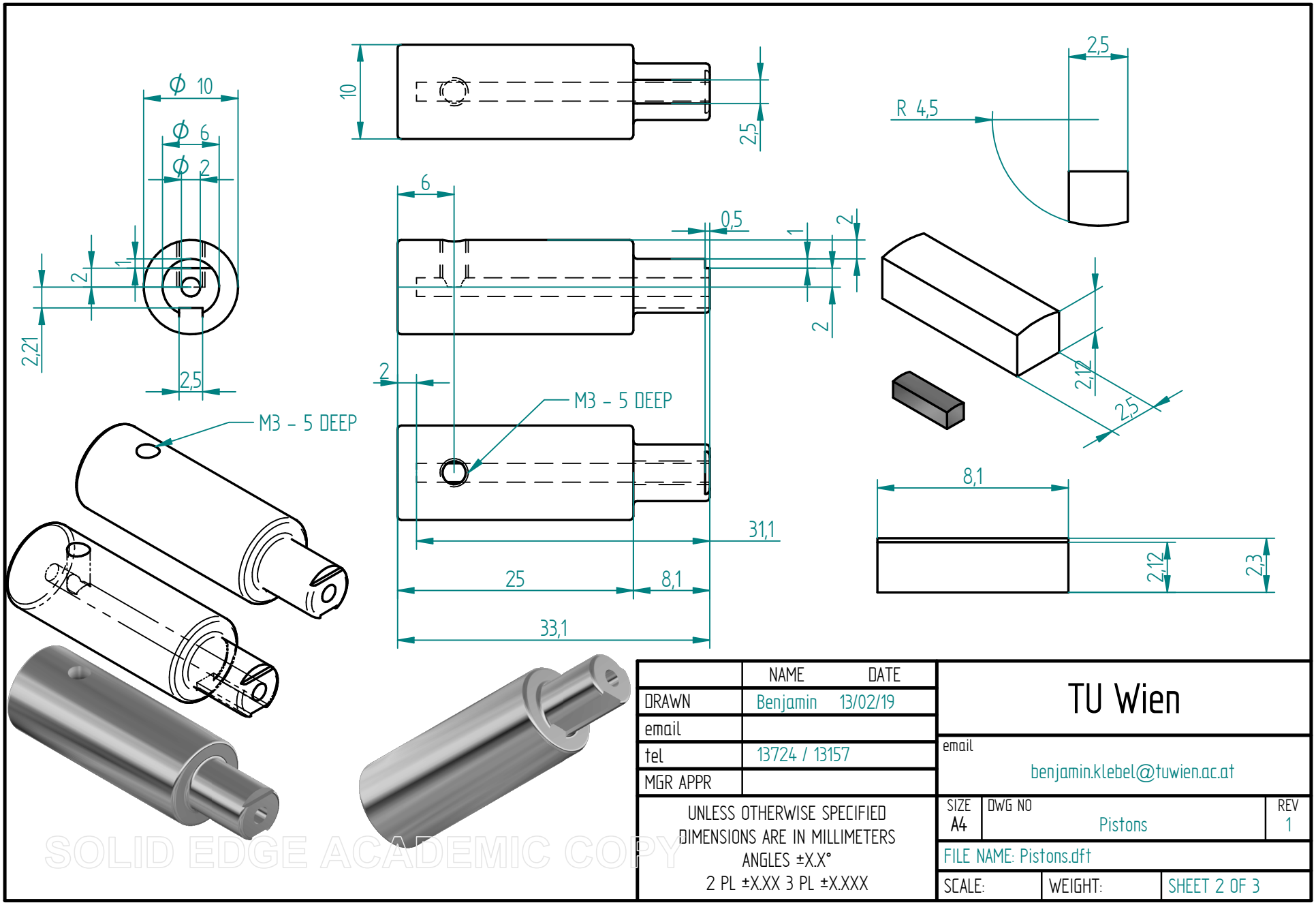
# TU Wien

email
benjamin.klebel@tuwien.ac.at

| SIZE | DWG NO | REV |
|---|---|---|
| A4 | Pistons | 1 |

FILE NAME: Pistons.dft

| SCALE: | WEIGHT: | SHEET 1 OF 3 |
|---|---|---|




| | NAME | DATE |
|---|---|---|
| DRAWN | Benjamin | 13/02/19 |
| email | | |
| tel | 13724 / 13157 | |
| MGR APPR | | |

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

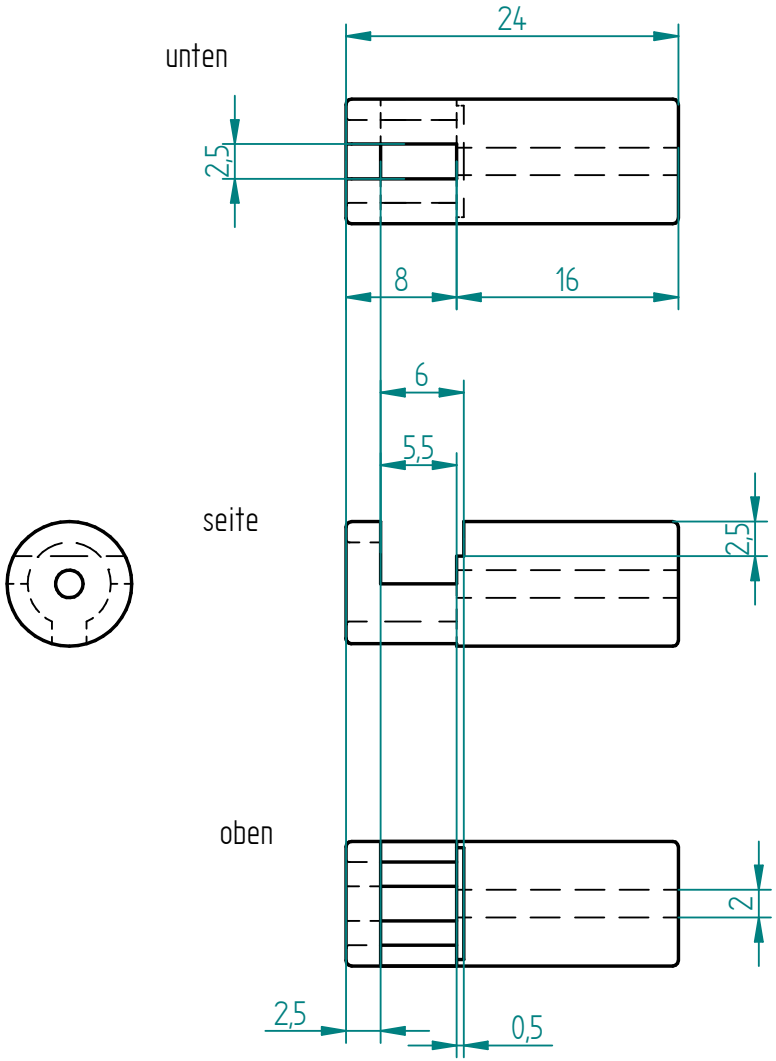
# TU Wien

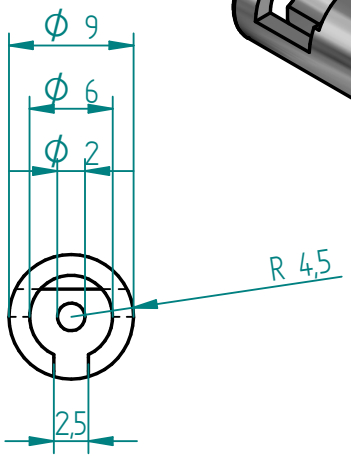
email
benjamin.klebel@tuwien.ac.at

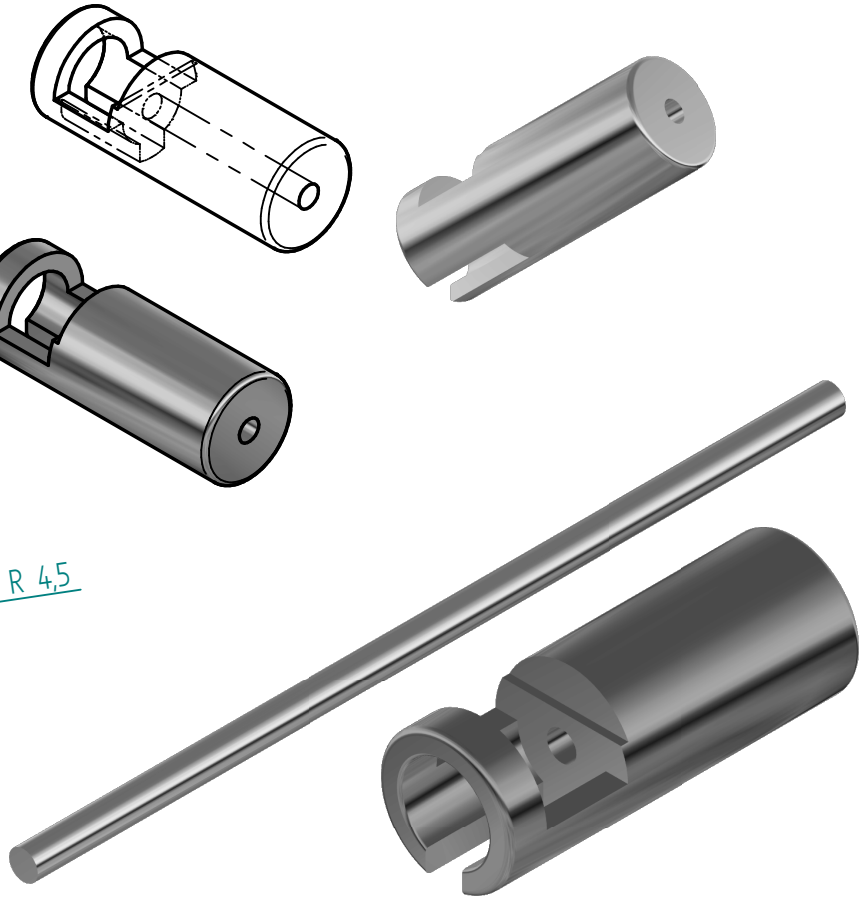
| SIZE | DWG NO | REV |
|---|---|---|
| A4 | Pistons | 1 |

FILE NAME: Pistons.dft

| SCALE: | WEIGHT: | SHEET 2 OF 3 |
|---|---|---|



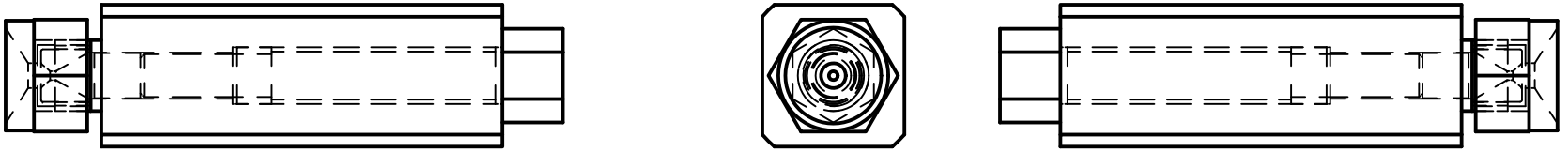






Der Stab sollte mit diesem Bauteil fix verbunden werden,
sodass das vorherige Bauteil daran geführt gleiten kann.

einfach etwas dünner als die 2mm Bohrung



| | NAME | DATE |
|---|---|---|
| DRAWN | Benjamin | 12/03/18 |
| email | | |
| tel | 13724 / 13157 | |
| MGR APPR | | |

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

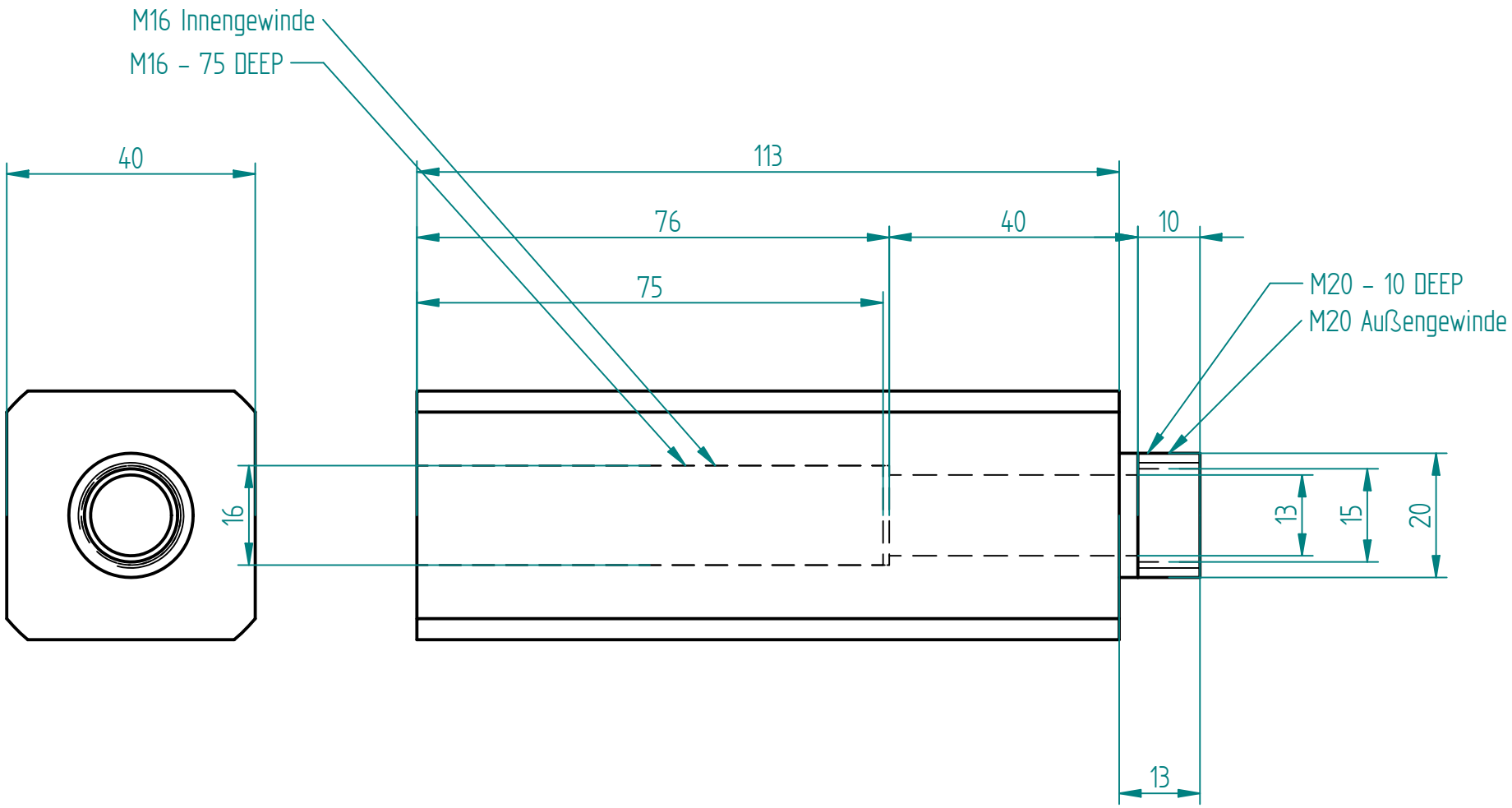
# TU Wien

email
benjamin.klebel@tuwien.ac.at

| SIZE | DWG NO | REV |
|---|---|---|
| A4 | Indium | Press |

FILE NAME: Indium-Press.dft

| SCALE: | WEIGHT: | SHEET 1 OF 7 |
|---|---|---|




| | NAME | DATE |
|---|---|---|
| DRAWN | Benjamin | 12/03/18 |
| email | | |
| tel | 13724 / 13157 | |
| MGR APPR | | |

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

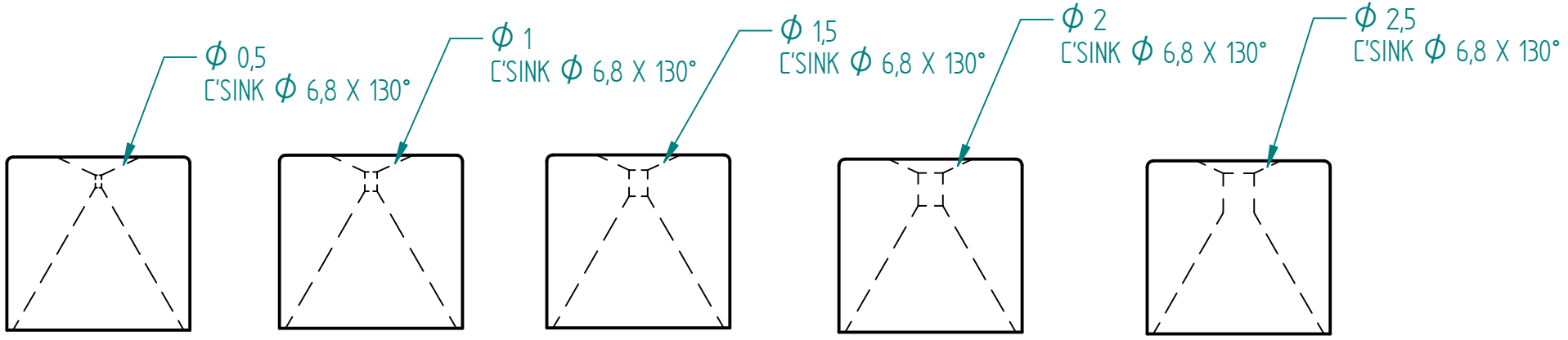
# TU Wien

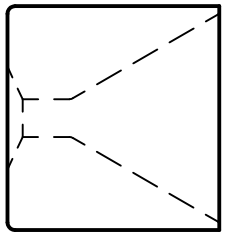
email
benjamin.klebel@tuwien.ac.at

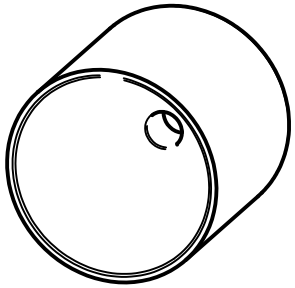
| SIZE | DWG NO | REV |
|---|---|---|
| A4 | Indium | Press |

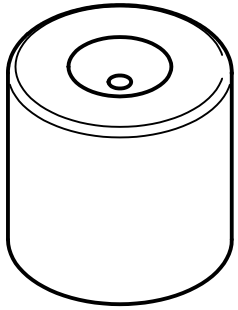
FILE NAME: Indium-Press.dft

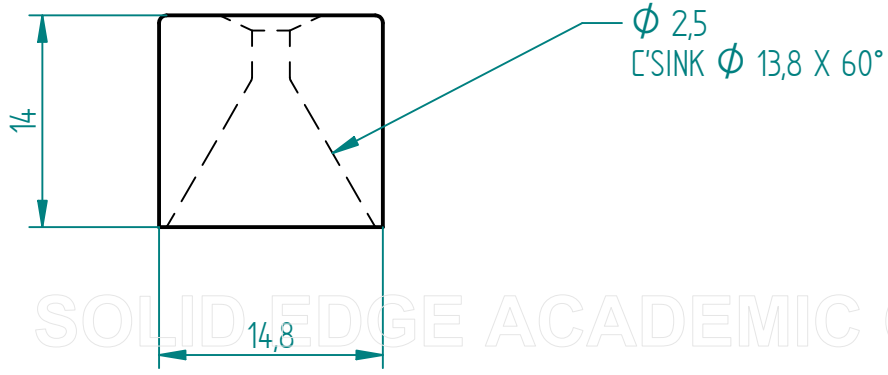
| SCALE: | WEIGHT: | SHEET 2 OF 7 |
|---|---|---|












| | NAME | DATE |
|---|---|---|
| DRAWN | Benjamin | 12/03/18 |
| email | | |
| tel | 13724 / 13157 | |
| MGR APPR | | |

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

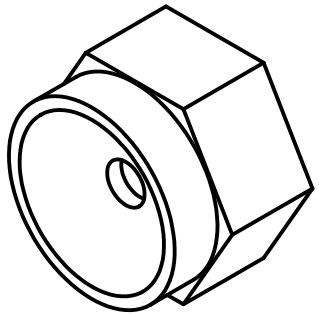
# TU Wien

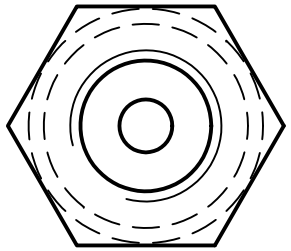
email
benjamin.klebel@tuwien.ac.at

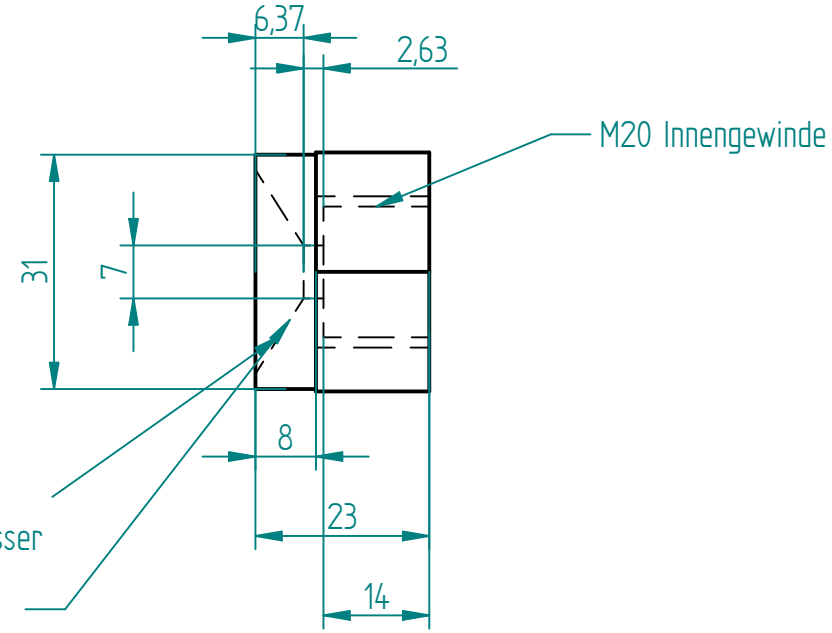
| SIZE | DWG NO | REV |
|---|---|---|
| A4 | Indium | Press |

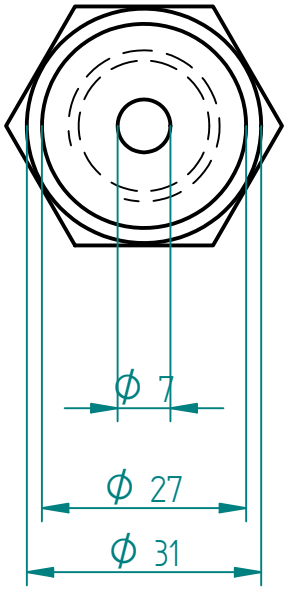
FILE NAME: Indium-Press.dft

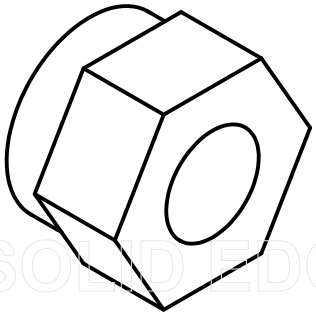
| SCALE: | WEIGHT: | SHEET 3 OF 7 |
|---|---|---|









| | NAME | DATE |
|---|---|---|
| DRAWN | Benjamin | 12/03/18 |
| email | | |
| tel | 13724 / 13157 | |
| MGR APPR | | |

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

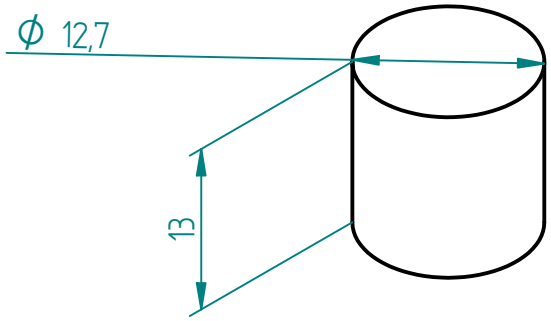
# TU Wien

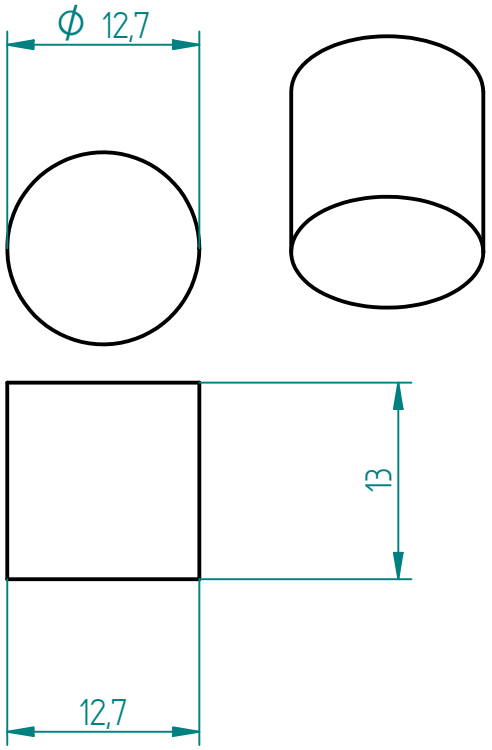
email
benjamin.klebel@tuwien.ac.at

| SIZE | DWG NO | REV |
|---|---|---|
| A4 | Indium | Press |

FILE NAME: Indium-Press.dft

| SCALE: | WEIGHT: | SHEET 4 OF 7 |
|---|---|---|






| | NAME | DATE |
|---|---|---|
| DRAWN | Benjamin | 12/03/18 |
| email | | |
| tel | 13724 / 13157 | |
| MGR APPR | | |

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

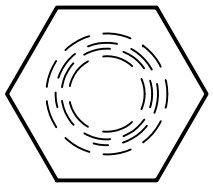
# TU Wien

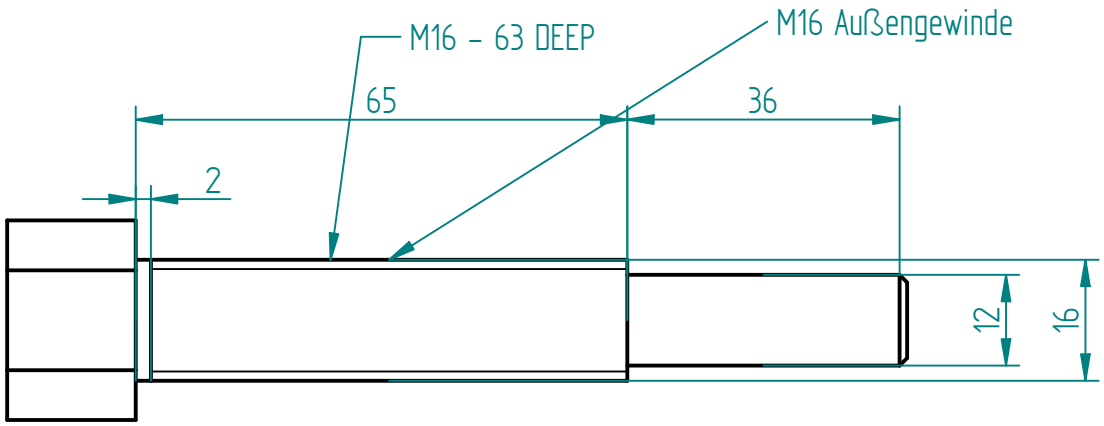
email
benjamin.klebel@tuwien.ac.at

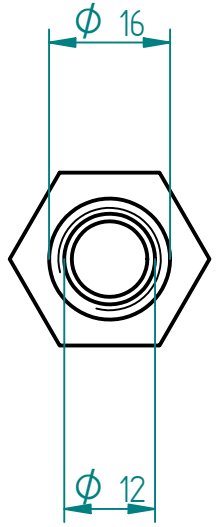
| SIZE | DWG NO | REV |
|---|---|---|
| A4 | Indium | Press |

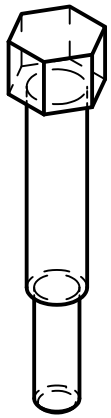
FILE NAME: Indium-Press.dft

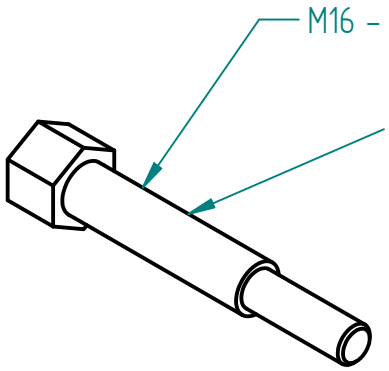
| SCALE: | WEIGHT: | SHEET 5 OF 7 |
|---|---|---|








| | NAME | DATE |
|---|---|---|
| DRAWN | Benjamin | 12/03/18 |
| email | | |
| tel | 13724 / 13157 | |
| MGR APPR | | |

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

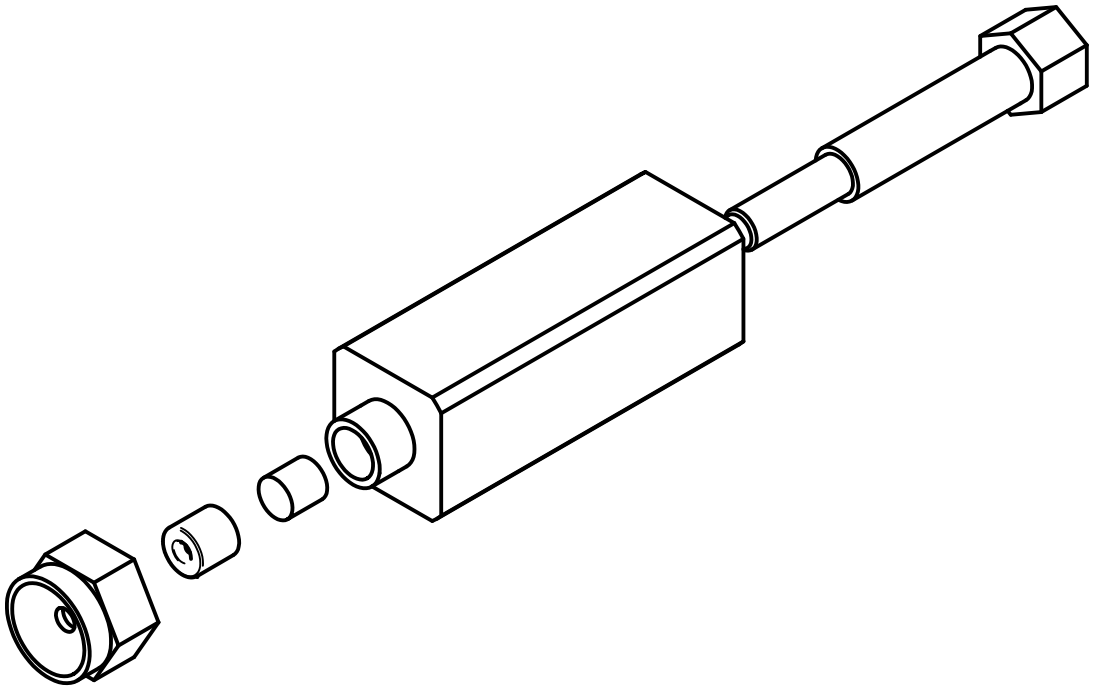
# TU Wien

email
benjamin.klebel@tuwien.ac.at

| SIZE | DWG NO | REV |
|---|---|---|
| A4 | Indium | Press |

FILE NAME: Indium-Press.dft

| SCALE: | WEIGHT: | SHEET 6 OF 7 |
|---|---|---|



| | NAME | DATE |
|---|---|---|
| DRAWN | Benjamin | 12/03/18 |
| email | | |
| tel | 13724 / 13157 | |
| MGR APPR | | |

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

# TU Wien

email
benjamin.klebel@tuwien.ac.at

| SIZE | DWG NO | REV |
|---|---|---|
| A4 | Indium | Press |

FILE NAME: Indium-Press.dft

| SCALE: | WEIGHT: | SHEET 7 OF 7 |
|---|---|---|

# C python and Graphical User Interface (GUI)

In the past, scientists have looked at instruments and noted values, staying in laboratories for extended amounts of time to log the progress of an experiment. Scientists still stay in laboratories for extended amounts of time, however this time can be spent more efficiently, as computers take the job of recording and controlling experiments. This frees up time for physicists to think about actual physics.

"A computer does it!" is now a bit simplistic, as "*a computer always does exactly what a human tells it to do – even if this human might not be aware of what they actually told the computer to do.*"[1] The important parts about experimental physics are still in the hands of physicists – decisions about what (and how) to measure in order to answer scientific questions. It is therefore very important to know how exactly to instruct computers for certain tasks.

## C.1 Demands of a Program and Programming Language

In the various laboratories around the world, scientists use a set of approaches to tell computers what to do. This is expressed by their choice of commercial programs, or, if no program is available for the specific task at hand, by their choice of programming language. Before the question is raised, which programming language should be used, the requirements for the program need to be declared. When deciding upon a programming language, in a laboratory where not only research but also education takes place, the interest of students to learn something useful for many purposes should be taken into account too. Thus, the principal requirements for the program and choice of programming language were decided upon as follows:

- The program should work seamlessly, and with a speed suited for controlling and measuring with the Oxford cryostat.
- It should be possibly to integrate all measuring instruments, old and new.
- The GUI should be as self-explanatory as possible, while it should behave as a normal user would expect it to, in 2019.
- The programming language should be versatile, enabling a programmer to adapt the program easily to most any use-case.
- Coding new parts of the program should not take too much time nor effort, to understand the existing code.
- It should be fairly easy to learn how to write code in the respective language, so students and scientists are not discouraged from doing so.
- Learning the programming language in use should benefit students once they leave the university.
- Data visualisation and some data analysis should be possible within the program.
- Additional instruments should be easy to integrate into the existing framework.

[1] A sentence most all students of programming courses will eventually hear.

There are several additional requirements to the capabilities of a programming language, specific to the experimental setup:

- Communication with devices should be possible over serial connections (RS-232), IEEE-488 connections (GPIB) and possibly Ethernet (Lan).
- Writing to different file formats for storing measurement data as well as logging general usage information – ideally with pre-defined program packages, without the need to access API's of file formats directly.
- The program should, in a final version, be capable of running complex measuring sequences independently. Thus, it must be possible to define a complex set of instructions, which can be read and executed in a most fail-safe manner.
- It would be best if this sequence of instructions could be platform independently shared and used – using an already existing file format which is used by the scientists from the laboratory itself would be the best option. This (not yet defined) file format needs to be read and written, in a similar manner as it is currently used.

It must be stressed that an object oriented programming language[2] seems to be most suitable, which should be ideally easy to learn, and heavily used both inside research and in the private sector. The programming language would ideally already have implementations of many of the required features easily available and ready to be used.

[2]Object oriented programming (in contrast to procedural programming) is a technique where abstract representations of arbitrary objects (abstract or with basis of objects in the world around us) can be built and manipulated. These are defined in "classes", which represent classes of objects. Objects may have attributes (a dog might have a colour and a weight for example), and methods (a dog may bark, from time to time).

## C.2 Comparison of different Programming Languages for Scientific Uses

There are a few programming languages to take into consideration (to name a few important ones):

- `python`
- LabView
- `C/C++`
- `Java`

**Java** is used very seldom in research applications, and in most cases because of its capability to produce "beautiful" user interfaces platform independently. It is an object oriented programming language, thus it has much of the versatility which is needed. However, `Java` is a rather "verbose" programming language, and not very friendly to new programmers, with a rather steep learning curve. All the more, `Java` is not known to many researchers. If a researcher is already proficient in `Java`, and it is unlikely that their code will be used by other researchers, it might be sensible to use `Java`. As this was not the case, `Java` was not considered fit to be used in this scope.

**C++** is usually used where speed is of utmost importance. Its advantage in speed mainly comes from its low-level approach: The code which is visible to a programmer is all there is. Nothing is running in the background by itself, everything that needs to happen must be programmed, including for example memory allocation for each individual variable. Thus, `C++` allows researchers to control measurements where control decisions need to happen at the nanosecond scale. However, this time-scale is of no interest regarding the cryostat system being built. Furthermore, all data visualisation, calculations in data analysis, communication with devices, writing to arbitrary file formats, would be required to be programmed. There are a number of pre-existing libraries for many of those tasks, however it is not apparently easy to integrate them in a platform independent manner. Additionally, only a small amount of code, which handles the relevant devices exists already. `C++` would be very educational for students, in regard to learning how to program, however it too has a steep learning curve, and researchers would not be encouraged to add to the program and maintain it – they would rather start from scratch with a different system. Thus, `C++` was not considered a good choice in this scope.

**LabView** is widely used in laboratories around the world. It is a "graphical programming language", as writing a program in LabView does not involve much code, but placing objects on a canvas, and connecting them with lines (abstract version of electrical circuit boards). There are many pre-defined GUI elements which come with some functionality already attached – those GUI elements can easily be placed and manipulated in a graphical editor. As LabView is provided by National Instruments, it comes with all necessary program libraries to control a big variety of instruments. Learning how to write a very simple program, which can perform simple tasks with instruments is quick, and any new researcher should not have problems following the logic of a LabView program. However, LabView is restricted to its designed elements, and by its very nature a procedural, and not object oriented language. The versatility of LabView is therefore very limited. For example, it is nigh impossible to program a sequence editor and sequence "runner", which allow for sequences of minor complexity, like nesting sweeps in temperature and field. Even more, what LabView gains in usability for small programs, it lacks once a much more complex issue has to be addressed in a complex way. To illustrate

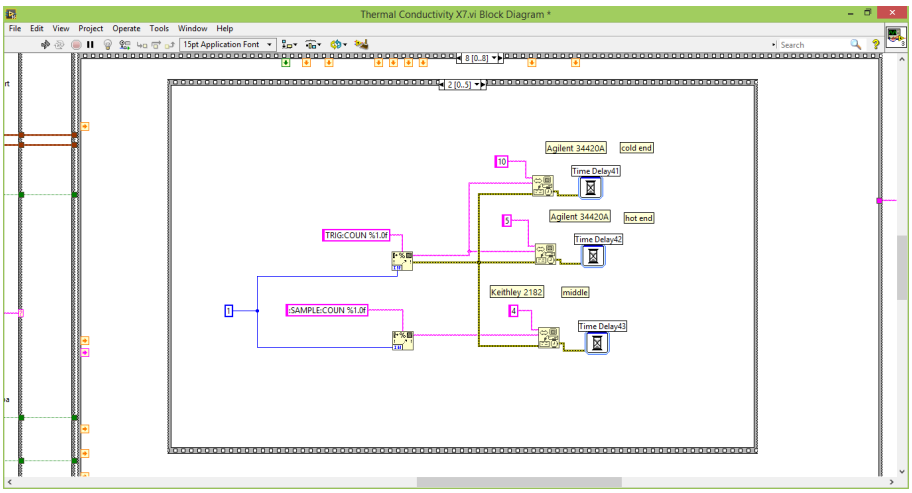



Figure C.7: Example of the blockdiagram (development environment) of part of a LabView VI. The small part in the middle of the screen shows the preparation of nanovoltmeters for heat conduction measurements. The blue line delivers the integer "1" to two control structures, which merge this integer into a string (pink boxes). From those control structures, the strings are handed to another control structure, which handles the device communication, and additionally requires the GPIB address (pink boxes with numbers). Borders of control structure boxes are visible on the sides – the corresponding control structures are too big to be displayed all at once. The complete program extends over much more screen-space than can be easily shown.

the usability problems with LabView, it is important to have an idea of how the programming environment is set up (fig. C.7):

In LabView, every piece of a program ("VI" - virtual instrument) has two layers: a front end, which a user will see, and a back end, where all front end actions are defined. The front end can be customised in a graphical editor, where GUI elements can be placed by drag&drop. The back end can only be visualised as an empty canvas, where objects can be placed. These objects may represent physical instruments, parts of GUI elements (which induce certain actions), functional parameters, or other VI's. The canvas may be dragged in any direction for almost any distance (if elements are placed there), however, no zoom option exists. Objects on the canvas need to be connected by lines – to be precise, every object may incorporate a high number of "connectors" where leads can be attached, to transport data to other objects. It is impossible to zoom to such an object, so that it can become hard to attach a line to the correct connector at the respective object.

Additionally, as LabView is a strictly procedural language, the GUI might not behave as expected by a "normal user" in 2019. Extra windows, which might be shown, may not be closed in a "normal fashion" with GUI elements provided by the operating system, as this might lead to freezing of the whole program (if it is not circumvented with effort). In the end, LabView might not aid students in their ongoing career at all (related to programming), as LabView is not used in most any field they are likely to enter. Any other "non-graphical", object oriented programming language might be much more beneficial for them.

For these reasons, LabView was disregarded.

**python** is a very widely used programming language, with an ever increasing user base in a variety of fields. As it can be used in a procedural and object oriented way – its syntax is (in nearly all ways) very similar to C/C++ (including very similar class structures). However, compared to C/C++ it is very easy to learn, the syntax does not contain many unintuitive complications which are necessary in C/C++. In general, programming with python is as intuitive as seems possible, for a code-based programming language (compared to graphical programming in LabView). python provides a very simple framework to import and use code, which was collected from a third party. There are a few package managers, where code solutions for a vast number of problems can be found. In the end, many of the high-performance packages are actually built on C++, so one can leverage the speed advantage, without entering details, for example how a cpu is designed. This rich environment of packages is a very strong argument for python, as there are already functions for many instruments which are to be controlled in the system at hand. Solutions for communication with instruments, accessing most any file format, visualisation of data, as well as performing various ways of analysing data in a very streamlined process exist. With PyQt 5 a wrapper for the Qt framework exists, which is the basis for many desktop applications. Qt is defined in C++ (it is fast), which would enable a student to quickly adapt to programming GUI applications in C++ in the future. Furthermore, there is a graphical editor, which can be used to place GUI elements by drag&drop in different windows, while predefining many values, which makes the production of the GUI itself as easy as in LabView. Incorporating the possibility for object oriented language, it is possible to define arbitrary objects, with individual attributes, which is very convenient if an abstract representation of physical instruments is required. Also, sequences for measurements can be defined in arbitrarily complex ways (regarding measurements), while the logic of traversing them can be programmed in a rather simple and concise way. It is possible to use pre-existing solutions for sequence editing and storing, as such sequences can be parsed without the need for a custom solution of writing sequences. The capabilities of such a sequence file format then only depend on a system already in use, or the imagination of the programmer. Additionally, python can leverage much of the capabilities of computer systems, including seamless multithreading and multiprocessing (also possible in Java/C++, not so easy in LabView).

There are some drawbacks to using python in a research lab: Though a great number of researchers are using python already, a large body of researchers who have never learned a code-based programming language, who are familiar with LabView. Any new programming language needs to be learned before it can be used, no matter its advantages over an older one. Apart from that, python is code, and anyone who wants to understand the logic behind it, needs to take a little bit of time to learn. LabView might be quicker in this regard (as the graphical visualisation of the programming logic seems to be grasped more easily), however the drawbacks of LabView play out in the long run.

Due to the above stated reasons, python was chosen as the programming language to develop the Cryostat-GUI control software.

## C.3 Cryostat-GUI

**GUI Programming: Threads** The GUI of a program should (in the best case) never become unresponsive. Thus, the part of the program which controls the GUI should always be disconnected from the rest of the program, logically as well as in its execution. If a program did everything on one core, with no concurrency at all, the GUI would become unresponsive, as soon as the program has to do anything. In order to avoid this, GUI and backend need to be separated, into at least two threads.

This principle was strictly followed: One thread controls the GUI, and all attached functions. This main GUI thread ("`mainthread`") then starts other threads if they are needed (for example for the communication with devices).

**`PyQt 5` – `Qt` Framework** `PyQt` is a `python` wrapper for the `Qt` framework. It allows for GUI windows in the typical style of many graphical computer programs. `Qt` is written in `C++`, and the wrapper enables a programmer to use its advantages directly in `python`. Thus, a very well-ordered, readable code in `python` can be seamlessly combined with the speed of `C++`.

`PyQt` provides a framework for threads, which is recommended to be used, instead of the "normal" `python` module `threading`. This framework implements signals, which can be broadcasted from any class, to which an arbitrary number of other classes can connect an arbitrary number of functions. These signals may contain and thus transmit data, however they do not need to. In the `threading` module, a thread can be created with a certain target function, which is called once the thread is started. This is not possible in the `PyQt` framework: Here, a thread must contain a class. This class can then emit and connect to signals. Thus, data and timings can be exchanged between threads in a fast and thread-safe way.

The whole GUI works through these signals: Pressing a button lets the corresponding code-object emit a signal, to which functions can be connected, so they are invoked on pressing that button.

**Inheritance in Object Oriented Programming** Object oriented programming has a few strengths. Inheritance is a very important one:

When defining various classes, a programmer might realise that they have many attributes and methods in common. These common characteristics should then be transferred to a more abstract class, which holds the information which would otherwise be defined at least two times. Both the original classes would inherit from this new "parent" class, with which all its attributes and methods are incorporated into the classes in use.

In the Cryostat-GUI for example, classes which represent measuring devices, usually inherit from classes which incorporate the definitions for communications with the corresponding devices.

**Distributed Architecture** The object oriented approach has a serious impact on the implementation: Every measuring device is represented by a class. These classes define how the program can communicate with the devices, supplying the corresponding functions. In all cases, the classes for individual instruments inherit generic functions of "how to communicate with a certain class of instrument" (via GPIB, serial or Ethernet connection), from the corresponding communication specific class. Those functions (for example `query()`) are then applied in defining in the device class, what exactly will be sent to a specific device, over the arbitrary communication channel, in order to evoke the expected response.

In these two levels of one parent and one child class, the communication with devices is exposed. Those classes can then be used in an arbitrary context to communicate with a device,

which can be in the case of the Cryostat-GUI, but also with any other small script, in any other laboratory.

In the Cryostat-GUI, one more level of abstraction in the device communication is present: In order to continuously supply the GUI with measurement data (for example temperature readings), some acting element of the program ("Updater" classes) needs to query data from the instruments, and send it to the `mainthread`, which can integrate it into the GUI. Any command (like changing the set temperature) which the `mainthread` receives is sent to this "Updater" class, which can fit it between queries. The "Updater" class lives in a thread which is separate from the `mainthread`, and inherits from classes like the `AbstractLoopThread`, which implement behaviour of running one function endlessly repeated. This `AbstractLoopThread` class allows for continuous queries to the instruments, while also allowing to pause the querying, for example when a time-critical measurement should not be interrupted by a regular query.

**Software Class Threads** Additionally, there are several classes, which fulfil various different tasks:

- Logging all current data to a database. (`main_Logger` in `logger.py`)
- Logging current data to a list which can be plotted for a live feed of measured data. (`live_Logger` in `logger.py`)
  - Here also calculations like the mean or the slope of the data in the list are performed.
- Logging measurement data in a specific file format, which is sent from a measuring class. (`measurement_Logger` in `logger.py`)
- Performing specific measurements, defined in other functions, sending the individual data to the measurement logger. (`Oneshot_Thread_multichannel` in `Sequence.py`)
- A plotting window, where the whole `matplotlib` functionality is implemented, and only the data, and labels need to be supplied. (`Window_plotting` in `util.py`)

### Room for Improvements

A program which is developed concurrently with actual usage will almost never be perfect. There is always room for improvement, especially related to intuitiveness of the user interface. Currently pending issues can be found at the GitHub repository – since a laboratory is a fast moving environment, not all issues might be up to date however.

Improvements planned for the future include:

- A sequence running scheme, where a sequence can be defined in advanced, and the device applies all instructions.
- A general "preferences" window, which collects all preferences which are not being changed in everyday circumstances.
- Export of measurement (and logging) data in different file formats.
- A more versatile plotting capability for the universal logging feature.
- Advanced automation procedures for older control devices (such as the Oxford Instruments) (e.g. automatic selection of PID values for the Oxford temperature controller).

Apart from openly visible changes and additions to the GUI, there are several issued concerned with the program's backend:

- A possibility of intra-application SQL-injections.
- Date and time formats should be updated and ported to a concise framework (`python datetime`)
- The calculations corresponding to the live-plotting feature should be outsourced to different cpu cores with `multiprocessing`

Additionally, the quick pace with which this laboratory develops entails the need to adapt the program to new devices, and measurement procedures. More adjustments which currently can not be foreseen are therefore only to be expected.

## C.4 Cryostat-GUI: Selected Windows

In the following section, selected windows of the Cryostat-GUI program are shown. These include:

- the main window, closing of which turns off the application – fig. C.8
- temperature control system – fig. C.9
- Keithley current sources and nanovoltmeters – fig. C.10
- measurement control system – fig. C.11
- limited preferences window for control over software systems – fig. C.12
- Cryogenic liquid control system – fig. C.13
- the plotting feature – fig. C.14

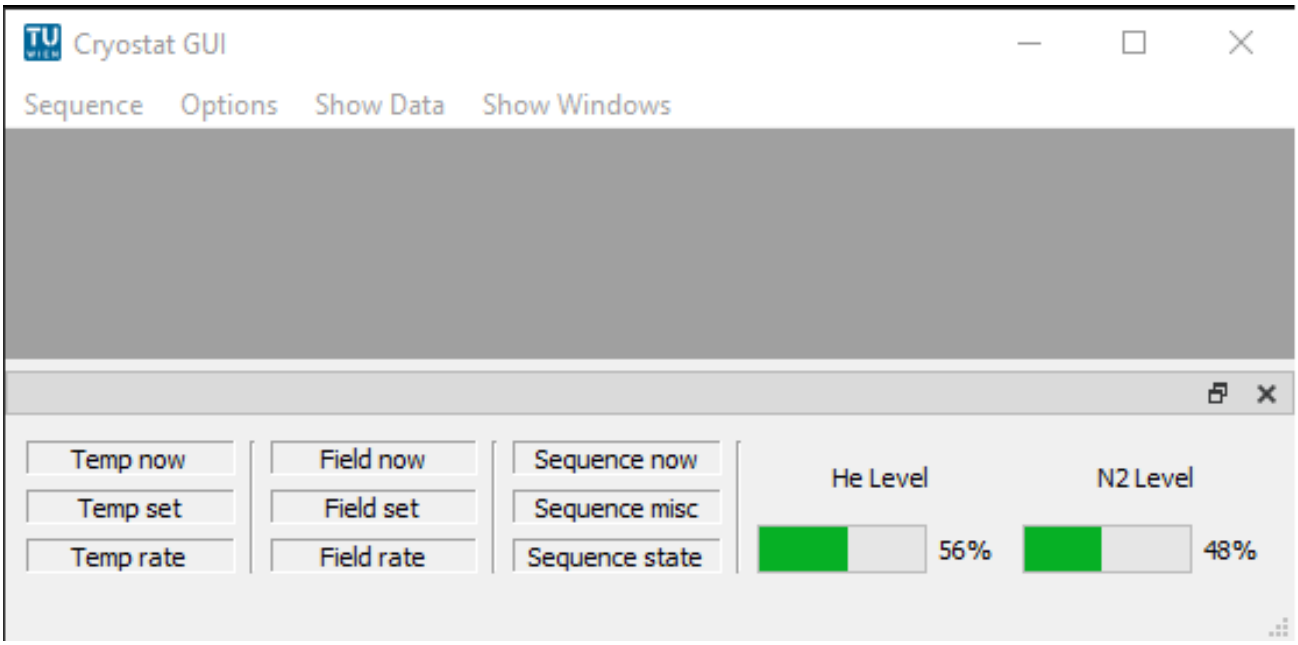



Figure C.8: The main Cryostat-GUI window. The different additional control windows can be opened from the menu bar at the top. At the bottom is an information dock, which currently is capable of showing the cryogenic liquid status. There are place holders for more information to be seen on the first glance. All additionally opened windows are disconnected from this main window, and act independently – as long as the shown main window is not closed, which results in turning off the application completely. The grey area in the middle is planned to serve as canvas for future subwindows: Sequence editing/running and display of measurement data.

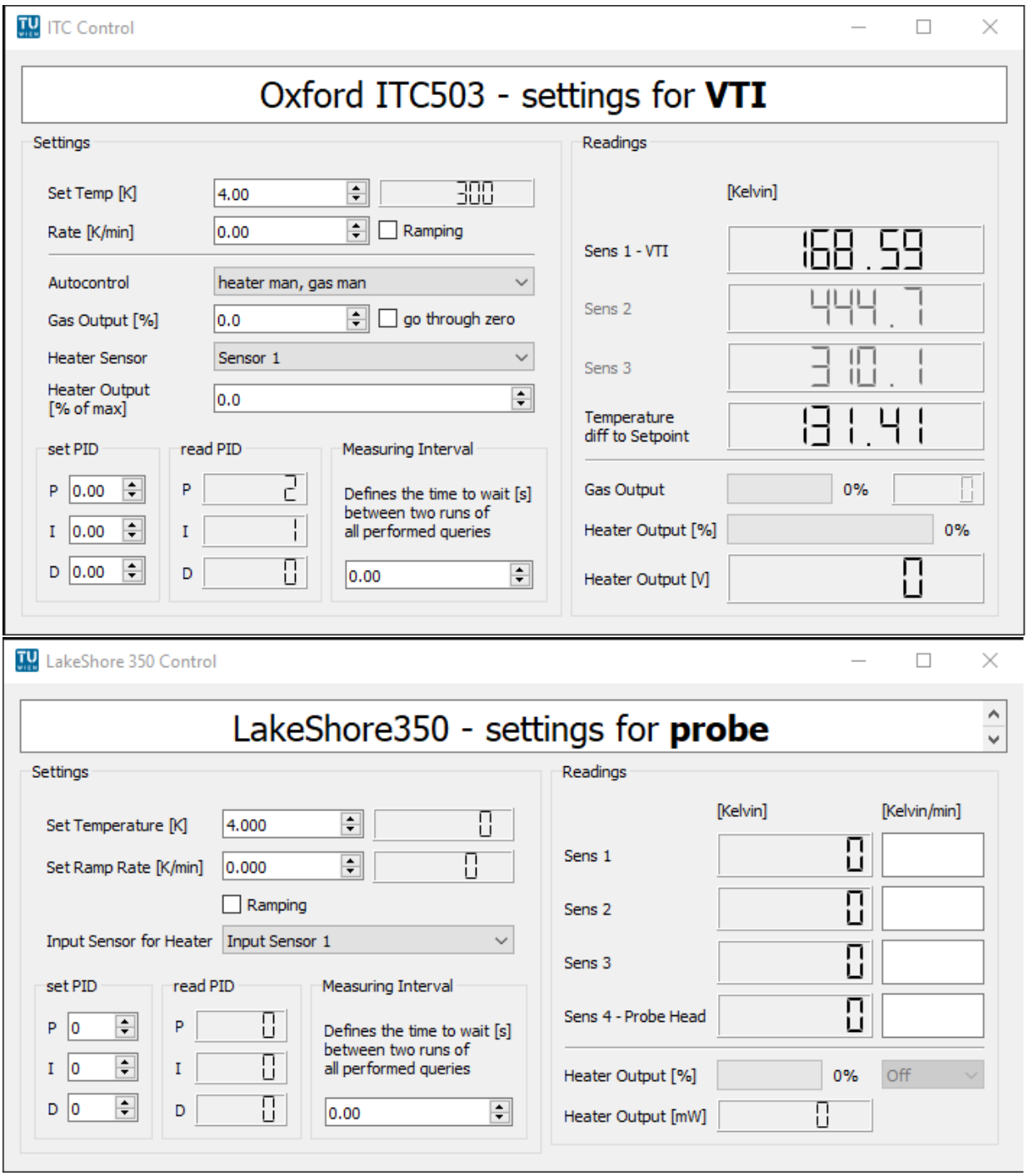



Figure C.9: The temperature control and observe windows, used to address the Oxford ITC 503 (top) and the LakeShore 350 (bottom). In the case of the Oxford device, it is feasible to directly control the heater and needle valve, thus the corresponding control elements were introduced. Sensor 2 and 3 of the Oxford device do not have physical representations in the cryostat in use, nevertheless we left the corresponding GUI elements in place for the sake of generality. It is planned to move the PID values (both reading and setting) to a general preferences window, together with the data query interval. As stepper motors are not necessarily always entirely exact in their steps, over multiple changes of the needle valve opening percentage (which is controlled via a stepper motor), arbitrary values might be reached. Therefore, an option was introduced (go through zero), which changes the device behaviour to close the needle valve every time the opening percentage is changed, to only afterwards proceed to the respective new value. Like this, total reproducibility of opening percentages was ensured, in case this was needed.

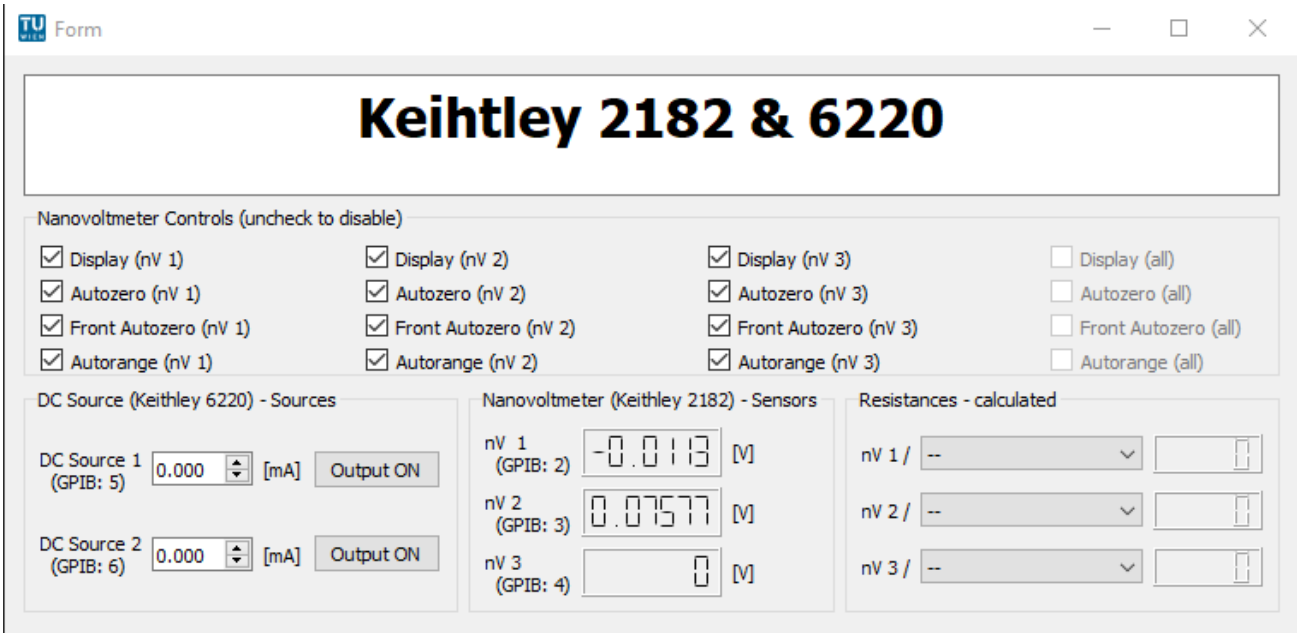



Figure C.10: Control window of both current sources and nanovoltmeters from Keithley. This window is meant for testing, and a possible speed increase of measurements: The tick options of the nanovoltmeters all have an impact on the measuring rate – in the current measuring setup (and the IV-linear technique in use) the "Display" and "Front Autozero" can be safely disabled. These options are the only ones which might influence the measuring behaviour of the system – all other parameters (what current, whether a current source is on or off, which current source is paired with which voltmeter to measure resistances) are defined in the measuring procedure.

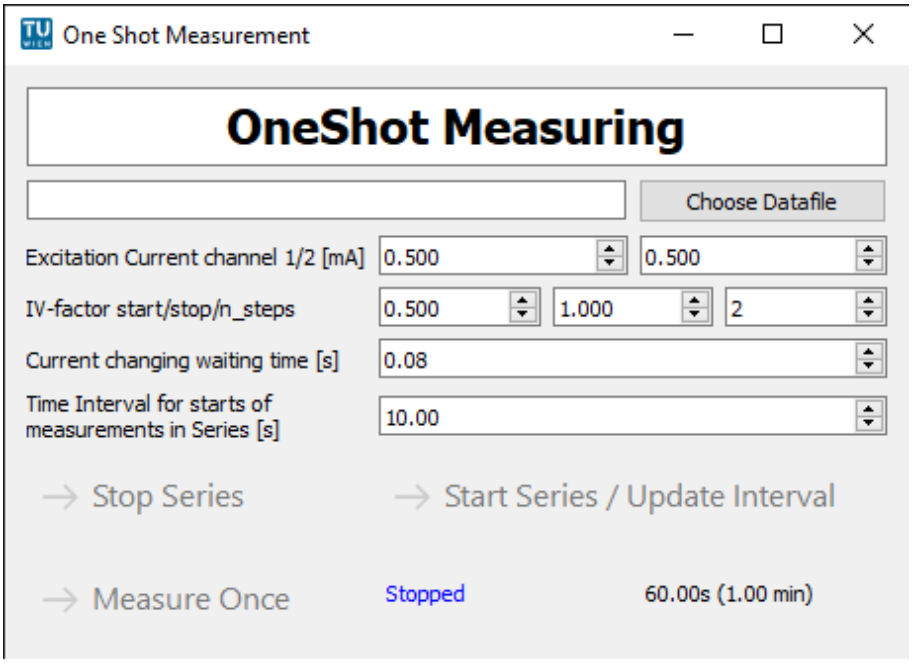



Figure C.11: The control window for measurements. It was called "Oneshot", in order to separate it from future "sequence"-like measurements, and because initially only allowed individual measurements were possible. The maximum excitation current can be specified in milli Amperes for two separate measuring channels individually. In the third line of inputs, the IV-curve can be set up. A sequence of factors can be created, with which the maximum excitation current is multiplied to acquire a sequence of excitation currents for an IV-curve. For any such sequence, both polarities are measured for resistivity measurements. The current changing waiting time specifies how long the program waits before a voltage measurement is taken after the current was changed (so the current is guaranteed to be stable). The last control line introduces the time interval between commands of the program to start a single measurement of resistivity. In case the specified time is shorter than a single measurement lasts, the corresponding queries are queued, and the program will continue to measure after the measuring series was stopped.

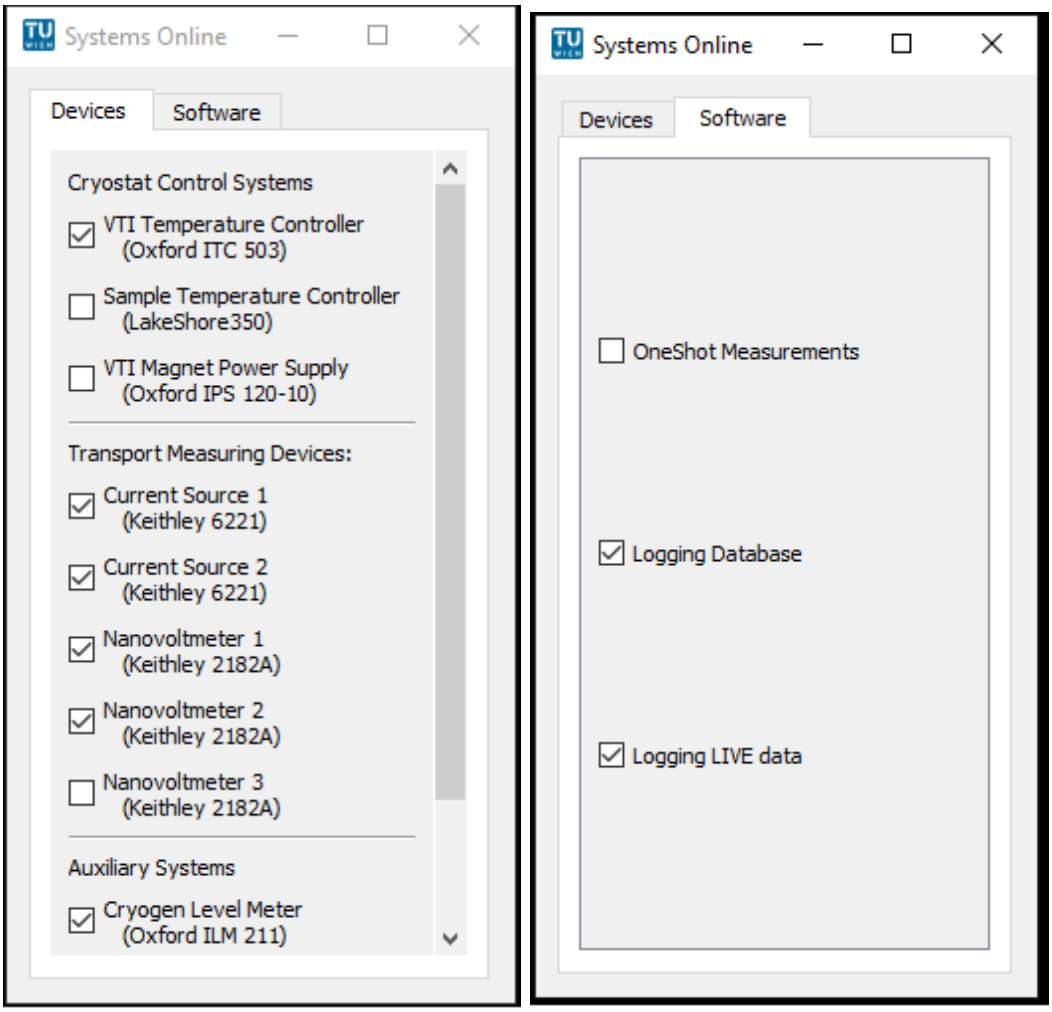



Figure C.12: Control window for turning on/off different systems of the application – this includes communication with certain devices, as well as logging and measurement systems.

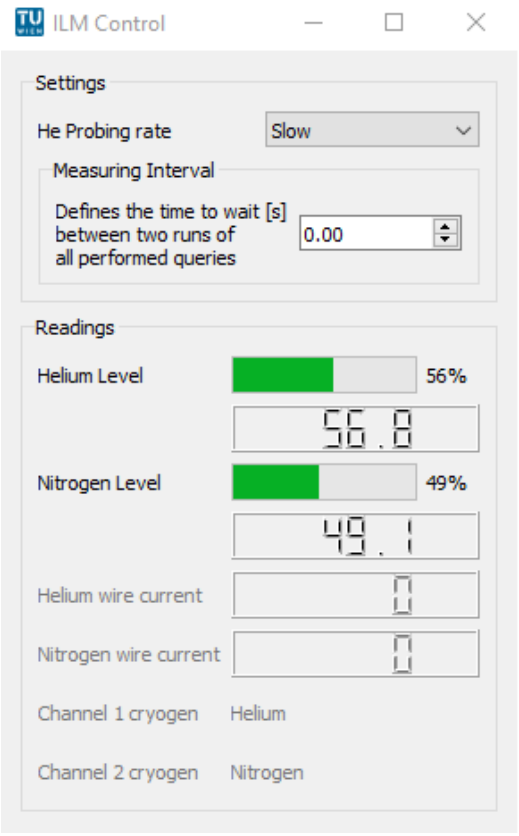



Figure C.13: Window to monitor the cyrogenic liquids of the cryostat, including an option to choose between slow and fast modes of the helium level probe.

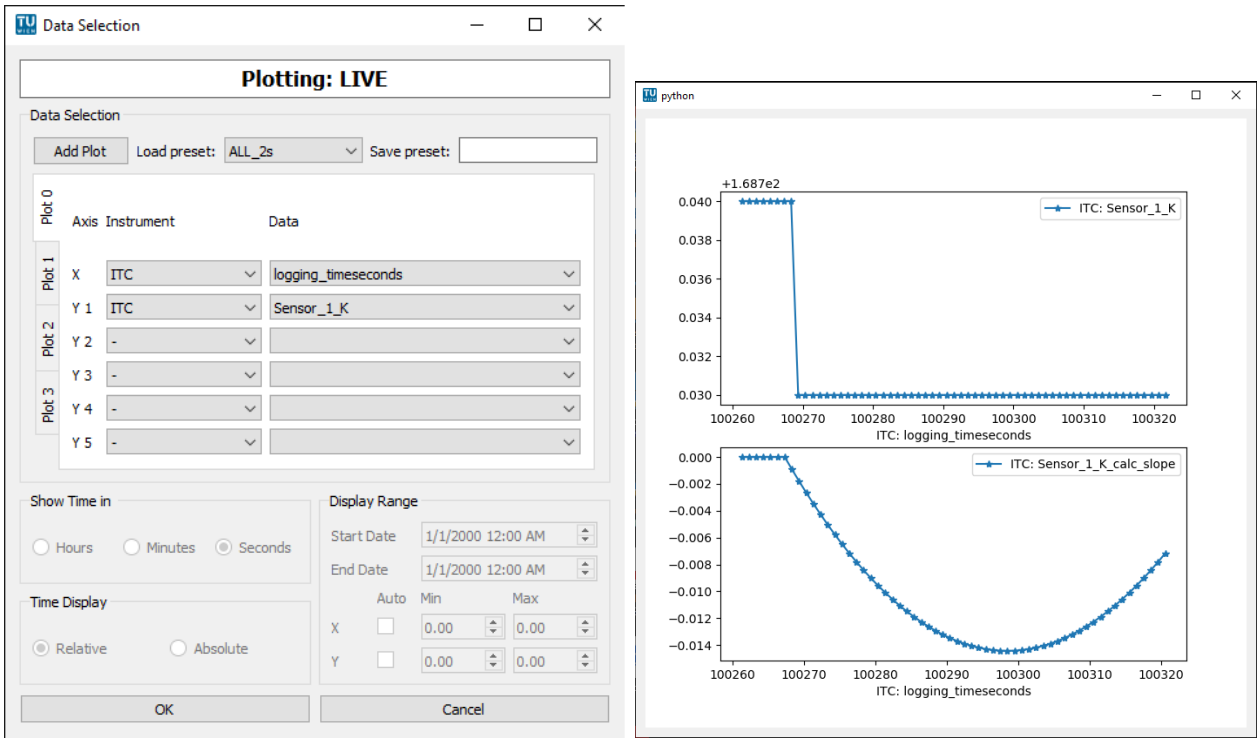



Figure C.14: Plotting feature of the Cryostat-GUI. Presets of data display configurations can be saved and loaded, with up to five datasets in one single figure. It is possible to show as many figures at a time as wanted, however it is not advisable to use more than 4 at a time, because of readability issues arising from a limited screen resolution and size.

## C.5 Code

*In this section of the appendix, selected portions of the* python *code which can be found on* GitHub *are shown. In order to facilitate readability, the indentation spacing was set to 2 spaces, and several lines which were otherwise slightly above 80 characters long were rearranged, upholding correct syntax. As can be seen on GitHub in closer detail, the syntax format guidelines PEP8 were generally followed.*

### Resistance Measuring

The following code segment is an exact representation of the code used for the resistivity measurements. Indeed, the resistance is being measured here, the geometrical factor of the sample was introduced later in the data analysis.

```
import time
from copy import deepcopy
import numpy as np
from numpy.polynomial.polynomial import polyfit
from itertools import combinations_with_replacement as comb


from util import loops_off

def measure_resistance_multichannel(threads,
          excitation_currents_A,
          threadnames_RES,
          threadnames_CURR,
          iv_characteristic,
          threadname_Temp='control_LakeShore350',
          current_reversal_time=0.08,
          **kwargs):
  """conduct one 'full' measurement of resistance:
    arguments: dict conf
      threads = dict of threads running of the mainWindow class
      threadname_Temp  = name of the (LakeShore) Temperature thread
      threadnames_RES  = list of names
                of the (Keithley) Voltage measure threads
      threadnames_CURR  = list of names
                of the (Keithley) Current set threads
      n_measurements  = number of measurements (dual polarity)
              to be averaged over
              default = 1 (no reason to do much more)
      excitation_currents_A = list of excitations currents
              for the measurement
    returns: dict data
      T_mean_K : dict of means of temperature readings
          before and after measurement [K]
      T_std_K : dict of stds of temperature readings
          before and after measurement [K]
      resistances, voltages, currents:
          dicts with corresponding values for all measurement channels
      timeseconds: pythons time.time()
      ReadableTime: Time in %Y-%m-%d %H:%M:%S
      SearchableTime: Time in %Y%m%d%H%M%S
  """
  # measured current reversal = 40ms.
  # reversal measured with a DMM 7510 of a 6221 Source (both Keithley)
  lengths = [len(threadnames_CURR),
         len(threadnames_RES),
```

```
        len(excitation_currents_A)]
for c in comb(lengths, 2):
  if c[0] != c[1]:
    raise AssertionError(
      'number of excitation currents, current sources'
      ' and voltmeters does not coincide!')

data = dict()
resistances = {key: dict(coeff=0, residuals=0, nonohmic=0)
         for key in threadnames_RES}
voltages = {key: [] for key in threadnames_RES}
currents = {key: [] for key in threadnames_CURR}

with loops_off(threads):

  temp1 = threads[threadname_Temp][0].read_Temperatures()
  temps = {key: [val] for key, val in zip(temp1.keys(),
                                          temp1.values())}

  for ct, (name_curr,
           exc_curr,
           name_volt) in enumerate(zip(threadnames_CURR,
                                       excitation_currents_A,
                                       threadnames_RES)):
    threshold_residuals = 1e4
    # threshold_coefficients = 1e4

    threads[name_curr][0].enable()
    # for idx in range(n_measurements):
    # as first time, apply positive current --> pos voltage
    for current_base in iv_characteristic:
      for currentfactor in [-1,  1]:
        current = exc_curr * currentfactor * current_base
        # print(current)
        currents[name_curr].append(current)
        threads[name_curr][0].gettoset_Current_A(current)
        threads[name_curr][0].setCurrent_A()
        # wait for the current to be changed:
        time.sleep(current_reversal_time)
        voltage = threads[name_volt][0].read_Voltage()
        voltages[name_volt].append(voltage)
    c, stats = polyfit(currents[name_curr], voltages[
               name_volt], deg=1, full=True)
    resistances[name_volt]['coeff'] = c[1]
    resistances[name_volt]['residuals'] = stats[0][0]
    c_wrong = polyfit(currents[name_curr], voltages[
              name_volt], deg=4)
    # print(stats[0], c_wrong)

    if stats[0] > threshold_residuals:
      resistances[name_volt]['nonohmic'] = 1

    threads[name_curr][0].disable()

  temp2 = threads[threadname_Temp][0].read_Temperatures()
  for key in temps:
    temps[key].append(temp2[key])

data['T_mean_K'] = {key + '_mean': np.mean(temps[key]) for key in temps}
data['T_std_K'] = {
```

```
    key + '_std': np.std(temps[key], ddof=1) for key in temps}

data['resistances'] = {key.strip('control_'): value
                       for key, value in zip(resistances.keys(),
                                             resistances.values())}
data['voltages'] = {key.strip('control_'): value
                    for key, value in zip(voltages.keys(),
                                          voltages.values())}
data['currents'] = {key.strip('control_'): value
                    for key, value in zip(currents.keys(),
                                          currents.values())}

data['datafile'] = kwargs['datafile']
timedict = {'timeseconds': time.time(),
      'ReadableTime': convert_time(time.time()),
      'SearchableTime': convert_time_searchable(time.time())}
data.update(timedict)
return data
```

### Live Logging and Calculation Class

Here, the portion of the code which is responsible for the calculation of live data analysis is shown. This feature of the program is intended to give the user a fast means of checking the behaviour of the monitored system.

```
from util import AbstractLoopThread
import numpy as np
from numpy.polynomial.polynomial import polyfit as nppolyfit
from copy import deepcopy
import time

class live_Logger(AbstractLoopThread):
  """docstring for live_Logger"""

  def __init__(self, mainthread, **kwargs):
    super().__init__()
    self.mainthread = mainthread
    self.interval = 1
    self.length_list = 60
    self.data = mainthread.data
    self.dataLock = mainthread.dataLock
    self.dataLock_live = mainthread.dataLock_live

    self.calculations = {
      'ar_mean': lambda time, value:
                  np.nanmean(value),
      'stddev': lambda time, value:
                  np.nanstd(value),
      'stderr': lambda time, value:
                  np.nanstd(value) / np.sqrt(len(value)),
      'stddev_rel': lambda time, value:
                  np.nanstd(value) / np.nanmean(value),
      'stderr_rel': lambda time, value:
                  np.nanstd(value) / (np.nanmean(value) *
                                      np.sqrt(len(value))),
      # 'test': lambda time, value: print(time),
      'slope': lambda time, value:
                  nppolyfit(time, value, deg=1, full=True),
      'slope_of_mean': lambda time, value:
                  nppolyfit(time, value, deg=1)[1] * 60
    }
    self.slopes = {
      'slope': lambda value, mean:
                  value[0][1] * 60,  # minutes,
      'slope_rel': lambda value, mean:
                  value[0][1] / mean * 60,
      'slope_residuals': lambda value, mean:
                  value[1][0][0] * 60 if \
                      len(value[1][0]) > 0 else np.nan}
    self.noCalc = ['time', 'Time', 'logging',
                   'band', 'Loop', 'Range', 'Setup', 'calc']
    self.pre_init()
    self.initialisation()
    mainthread.sig_running_new_thread.connect(self.pre_init)
    mainthread.sig_running_new_thread.connect(self.initialisation)
    mainthread.sig_logging_newconf.connect(self.update_conf)

  @pyqtSlot()  # int
  def work(self):
```

```
    """
      class method which (here) starts the run,
      as soon as the initialisation was done.
    """
    try:
      while not self.initialised:
        time.sleep(0.02)
      self.running()
    except AssertionError as assertion:
      self.sig_assertion.emit(assertion.args[0])
    finally:
      QTimer.singleShot(self.interval * 1e3, self.worker)

  @pyqtSlot()  # int
  def worker(self):
    """
      class method which is working all the time,
      while the thread is running, keeping the event loop busy
    """
    try:
      while not self.initialised:
        time.sleep(0.02)
      self.running()
    except AssertionError as assertion:
      self.sig_assertion.emit(assertion.args[0])
    finally:
      QTimer.singleShot(self.interval * 1e3, self.worker)

  def running(self):
    """
      go through all stored values for every instrument,
      and append them to the list which will be plotted
    """
    try:
      # print("live logger trying to log")
      with self.dataLock_live:
        with self.dataLock:
          # print(self.data_live)
          for instr in self.data:
            timedict = dict(
              logging_timeseconds=time.time() - self.startingtime,)
            dic = deepcopy(self.data[instr])
            dic.update(timedict)

            # print(times[0])
            for varkey in dic:
              # print(instr, varkey)
              self.data_live[instr][
                varkey].append(dic[varkey])
            if self.time_init:
              times = [float(x) for x in self.data_live[
                instr]['logging_timeseconds']]
            else:
              times = [0]
        for instr in self.data_live:
          for varkey in self.data_live[instr]:
            for calc in self.calculations:
              if all([x not in varkey for x in self.noCalc]):
                if self.time_init:
                  self.calculations_perform(
```

```
                    instr, varkey, calc, times)
            if self.count > self.length_list:
              self.counting = False
              self.data_live[instr][varkey].pop(0)

    except AssertionError as assertion:
      self.sig_assertion.emit(assertion.args[0])
    except KeyError as key:
      self.sig_assertion.emit("live logger" + key.args[0])
    self.time_init = True
    if self.counting:
      self.count += 1

  def calculations_perform(self, instr, varkey, calc, times):
    """
      perform one specified calculation on all corresponding datasets

      return: None
    """
    if calc == 'slope':
      fit = self.calculations[calc](times, self.data_live[instr][varkey])
      for name, calc_slope in zip(self.slopes.keys(),
                                  self.slopes.values()):
        self.data_live[instr][
          '{key}_calc_{c}'.format(key=varkey, c=name)].append(calc_slope(
            fit, self.data_live[instr][
              '{key}_calc_{c}'.format(key=varkey, c='ar_mean')][-1]))
    elif calc == 'slope_of_mean':
      times_spec = deepcopy(times)
      while len(times_spec) > len(self.data_live[instr][
        '{key}_calc_{c}'.format(key=varkey, c='ar_mean')]):
        times_spec.pop(0)
      fit = self.data_live[instr][
        '{key}_calc_{c}'.format(key=varkey, c=calc)].append(
          self.calculations[calc](
            times_spec, self.data_live[instr][
              '{key}_calc_{c}'.format(key=varkey, c='ar_mean')]))
    else:
      try:
        self.data_live[instr][
            '{key}_calc_{c}'.format(key=varkey, c=calc)].append(
                 self.calculations[calc](
                     times, self.data_live[instr][varkey]))

      except TypeError:
        # raise AssertionError(e_type.args[0])
        # print('TYPE CALC')
        pass
      except ValueError as e_val:
        raise AssertionError(e_val.args[0])

  def pre_init(self):
    self.initialised = False

  def initialisation(self):
    """
       copy the current data-dict,
       update for logging times,
       insert empty lists in all values
    """
```

```
    self.startingtime = time.time()
    timedict = dict(logging_timeseconds=0,)
    self.time_init = False
    self.count = 0
    self.counting = True
    with self.dataLock:
      with self.dataLock_live:
        self.mainthread.data_live = deepcopy(self.data)
        self.data_live = self.mainthread.data_live
        for instrument in self.data:
          dic = self.data[instrument]
          dic.update(timedict)
          self.data_live[instrument].update(timedict)
          for variablekey in dic:
            self.data_live[instrument][variablekey] = []
            if all([x not in variablekey for x in self.noCalc]):
              for calc in self.calculations:
                self.data_live[instrument][
                  '{key}_calc_{c}'.format(key=variablekey, c=calc)] = []
              for calc in self.slopes:
                self.data_live[instrument][
                  '{key}_calc_{c}'.format(key=variablekey, c=calc)] = []
    self.initialised = True

  def setLength(self, length):
    """set the number of measurements the calculation should be conducted
                                          over"""

    if self.length_list > length:
      with self.dataLock_live:
        for instr in self.data_live:
          for varkey in self.data_live[instr]:
            self.data_live[instr][varkey] = self.data_live[
              instr][varkey][(self.length_list - length):]
    elif self.length_list < length:
      with self.dataLock_live:
        for instr in self.data_live:
          for varkey in self.data_live[instr]:
            self.data_live[instr][varkey] = [
              np.nan] * (length - self.length_list) + \
                  self.data_live[instr][varkey]
    self.length_list = length

  def update_conf(self, conf):
    """
      - update the configuration with one being sent.
    """
    self.interval = conf['general']['interval_live']
```

### Exception handling

In this code snippet, a custom decorator is shown. This decorator is used for exception handling purposes: Any method of a QThread object in which such an exception might occur may be decorated with it, ensuring an undisturbed program execution in case of failure, with the possibility to display occurring errors to the user.

```
import functools
import inspect
from visa import VisaIOError


def ExceptionSignal(thread, func, e_type, err):
  thread.sig_assertion.emit('{}: {}: {}: {}'.format(
    thread.__name__,
    func.__name__,
    e_type,
    err.args[0]))


def ExceptionHandling(func):
  @functools.wraps(func)
  def wrapper_ExceptionHandling(*args, **kwargs):
    if inspect.isclass(type(args[0])):
      try:
        return func(*args, **kwargs)
      except AssertionError as e_ass:
        ExceptionSignal(args[0], func, 'Assertion', e_ass)

      except TypeError as e_type:
        ExceptionSignal(args[0], func, 'Type', e_type)

      except KeyError as e_key:
        ExceptionSignal(args[0], func, 'Key', e_key)

      except ValueError as e_val:
        ExceptionSignal(args[0], func, 'Value', e_val)

      except AttributeError as e_attr:
        ExceptionSignal(args[0], func, 'Attribute', e_attr)

      except NotImplementedError as e_implement:
        ExceptionSignal(args[0], func, 'NotImplemented', e_implement)

      except VisaIOError as e_visa:
        if isinstance(e_visa, type(args[0].timeouterror)) and \
            e_visa.args == args[0].timeouterror.args:
          args[0].sig_visatimeout.emit()
        else:
          ExceptionSignal(args[0], func, 'VisaIO', e_visa)
    else:
      print('There is a bug!! ' + func.__name__)
  return wrapper_ExceptionHandling
```

### Template Control Thread

This code segment may serve as a template for control threads of instruments which might be introduced to the system in the future.

```
from PyQt5.QtCore import pyqtSlot

from pyvisa.errors import VisaIOError

from copy import deepcopy
from importlib import reload

from util import AbstractLoopThread
from util import ExceptionHandling

import Dummyfolder


class Template_Updater(AbstractLoopThread):
  """Updater class to update all instrument data of a Dummy instrument.

  For each function (except collecting data), there is a wrapping method,
  which we can call by a signal, from the main thread. This wrapper sends
  the corresponding value to the device.

  There is a second method for all wrappers, which accepts
  the corresponding value, and stores it,
  so it can be sent upon acknowledgement

  The information from the device is collected in regular intervals
  (method "running"),
  and subsequently sent to the main thread. It is packed in a dict,
  the keys of which are displayed in the "sensors" dict in this class.
  """

  sensors = dict(
    channel_1=1,
    channel_2=2)

  def __init__(self, InstrumentAddress='', **kwargs):
    super().__init__(**kwargs)
    global Dummyfolder
    dummy = reload(Dummyfolder.dummy).dummy
    self.Dummy = dummy(InstrumentAddress=InstrumentAddress)
    self.__name__ = 'Template_Updater ' + InstrumentAddress
    self.interval = 3
    self.control_state = True
    self.setControl()

  @ExceptionHandling
  def running(self):
    """extract all data from the Dummy, emit signal, sending the data"""
    data = dict()

    for key in self.sensors:
      try:
        data[key] = self.Dummy.getValue(self.sensors[key]) * 0.1
      except AssertionError as e_ass:
        self.sig_assertion.emit(e_ass.args[0])
      except VisaIOError as e_visa:
```

```
        if (isinstance(e_visa, type(self.timeouterror)) and
            e_visa.args == self.timeouterror.args):
          self.sig_visatimeout.emit()
          self.Dummy.clear_buffers()
        else:
          self.sig_visaerror.emit(e_visa.args[0])
    self.sig_Infodata.emit(deepcopy(data))

  @pyqtSlot()
  @ExceptionHandling
  def setControl(self):
    """set Control status of the instrument"""
    self.Dummy.setControl(self.control_state)

  @pyqtSlot(int)
  @ExceptionHandling
  def setProbingSpeed(self, speed, channel=1):
    """set probing speed for a specific channel

      for fast probing, speed = 1
      for slow probing, speed = 0
      this comes from the order in the comboBox in the GUI
    """
    if speed == 1:
      self.Dummy.setFast(channel)
    elif speed == 0:
      self.Dummy.setSlow(channel)

  @pyqtSlot(int)
  def gettoset_Control(self, value):
    """receive and store the value to set the Control status"""
    self.control_state = value
```

### Sequence Parser

A sequence parser for measuring sequences of the Quantum Design PPMS is presented here. All generic PPMS commands are covered, as well as the commands for the resistivity option. The sequence is converted into a `json` format, and includes the complete logic (including nested scans) of the original sequence.

```
import pickle
import os
import re
import json


dropstring = re.compile(r'([a-zA-Z0-9])')
searchf_number = re.compile(r'([0-9]+[.]*[0-9]*)')
searchf_string = re.compile(r'''["']{2}(.*?)["']{2}''')


class EOSException(Exception):
  """Exception to raise if an EOS was encountered"""
  pass


def parse_binary(number):
  """parse an integer number which represents a sum of bits
    returns a list with True and False, from back to front
  """
  # print(number)
  number = int(number)
  nums = list(reversed('{:b}'.format(number)))
  # print(nums)
  for ct, num in enumerate(nums):
    nums[ct] = True if int(num) else False
  return nums


class Sequence_parser(object):
  """Abstract Sequence parser, without GUI"""

  def __init__(self, sequence_file=None, textnesting='    ', **kwargs):
    """initialise important attributes"""
    super(Sequence_parser, self).__init__(**kwargs)

    self.sequence_file = sequence_file
    self.textnesting = textnesting
    self.initialize_sequence(self.sequence_file)

  def saving(self):
    """save serialised versions of a sequence"""
    with open(self.sequence_file_p, 'wb') as output:
      pickle.dump(self.data, output, pickle.HIGHEST_PROTOCOL)
    with open(self.sequence_file_json, 'w') as output:
      output.write(json.dumps(self.data))

  def change_file_location(self, fname):
    self.sequence_file = os.path.splitext(
                  fname)[0] + '.seq'
    self.sequence_file_p = os.path.splitext(
                  self.sequence_file)[0] + '.pkl'
    self.sequence_file_json = os.path.splitext(
```

```
      self.sequence_file)[0] + '.json'

  @staticmethod
  def construct_pattern(expressions):
    pat = ''
    for e in expressions:
      pat = pat + r'|' + e
    return pat[1:]

  def initialize_sequence(self, sequence_file):
    """parse a complete file of instructions"""
    if sequence_file:
      self.change_file_location(sequence_file)

      exp = [r'TMP TEMP(.*?)$', r'FLD FIELD(.*?)$', r'SCAN(.*?)$',
           r'WAITFOR(.*?)$', r'CHN(.*?)$', r'CDF(.*?)$', r'DFC(.*?)$',
           r'LPI(.*?)$', r'SHT(.*?)DOWN', r'EN(.*?)EOS$', r'RES(.*?)$',
           r'BEP BEEP(.*?)$', r'CMB CHAMBER(.*?)$', r'REM(.*?)$']
      self.p = re.compile(self.construct_pattern(
        exp), re.DOTALL | re.M)  # '(.*?)[^\S]* EOS'

      self.data, self.textsequence = self.read_sequence(sequence_file)
    else:
      self.textsequence = []
      self.data = []
      self.sequence_file = ''

  def read_sequence(self, file):
    """read the whole sequence from a file"""
    with open(file, 'r') as f:
      data = f.readlines()  # .replace('\n', '')

    # preparing variables
    self.jumping_count = [0, 0]
    self.nesting_level = 0
    # parse sequence
    commands, textsequence = self.parse_nesting(data, -1)
    return commands, textsequence

  def parse_nesting(self, lines_file, lines_index):
    """parse a nested command structure"""
    commands = []
    if lines_index == -1:
      textsequence = []
    else:
      textsequence = None
    for ct, line_further in enumerate(lines_file[lines_index + 1:]):
      if self.jumping_count[self.nesting_level + 1] > 0:
        self.jumping_count[self.nesting_level + 1] -= 1
        continue
      for count, jump in enumerate(self.jumping_count[:-1]):
        self.jumping_count[count] += 1
      try:

        dic_loop = self.parse_line(
          lines_file, line_further, lines_index + 1 + ct)
      except EOSException:
        self.nesting_level -= 1
        dic_loop = dict(
          typ="EOS",
```

```
          DisplayText=self.textnesting * (self.nesting_level) + 'EOS')
        commands.append(dic_loop)
        break
      if dic_loop is not None:
        commands.append(dic_loop)
        if lines_index == -1:
          textsequence.append(dic_loop)
          self.add_text(textsequence, dic_loop)
    del self.jumping_count[-1]
    return commands, textsequence

  def add_text(self, text_list, dic):
    """build the un-nested list of displayed commands"""
    if 'commands' in dic:
      for c in dic['commands']:

        try:
          text_list.append(dict(DisplayText=c['DisplayText']))
        except KeyError:
          print(c)
        self.add_text(text_list, c)

  def parse_line(self, lines_file, line, line_index):
    """parse one line of a sequence file, more if it is a scan"""
    line_found = self.p.findall(line)

    try:
      line_found = line_found[0]
    except IndexError:
      return None

    # print(line_found)
    if line_found[0]:
      # set temperature
      dic = self.parse_set_temp(line)
    elif line_found[1]:
      # set field
      dic = self.parse_set_field(line)
    elif line_found[2]:
      # scan something
      self.jumping_count.append(0)
      dic = self.parse_scan_arb(lines_file, line, line_index)
    elif line_found[3]:
      # waitfor
      dic = self.parse_waiting(line)
    elif line_found[4]:
      # chain sequence
      dic = self.parse_chain_sequence(line)
    elif line_found[5]:
      # resistivity change datafile
      dic = self.parse_res_change_datafile(line)
    elif line_found[6]:
      # resistivity datafile comment
      dic = self.parse_res_datafilecomment(line)
    elif line_found[7]:
      # resistivity scan excitation
      dic = self.parse_res_scan_excitation(line)
    elif line_found[8]:
      # Shutdown to a standby configuration
      dic = dict(typ='Shutdown')
```

```
    elif line_found[9]:
      # end of a scan
      raise EOSException()
    elif line_found[10]:
      # resistivity - measure
      dic = self.parse_res(line)
    elif line_found[11]:
      # beep of certain length and frequency
      dic = self.parse_beep(line)
    elif line_found[12]:
      # chamber operations
      dic = self.parse_chamber(line)

    elif line_found[13]:
      # remark
      dic = dict(typ='remark', DisplayText=line_found[13])

    # try:
    #     print(dic)
    # except NameError:
    #     print(line_found)
    # if dic['typ'] is None:
    #     print(line_found)

    return dic

  def parse_scan_arb(self, lines_file, line, lines_index):
    """parse a line in which a scan was defined"""
    # parse this scan instructions
    line_found = self.p.findall(line)[0]

    dic = dict(typ=None)
    if line_found[2][0] == 'H':
      # Field
      dic = self.parse_scan_H(line)

    if line_found[2][0] == 'T':
      # temperature
      dic = self.parse_scan_T(line)

    if line_found[2][0] == 'P':
      # position
      dic = self.parse_scan_P(line)

    if line_found[2][0] == 'C':
      # time
      dic = self.parse_scan_time(line)

    self.nesting_level += 1

    commands, _ = self.parse_nesting(lines_file, lines_index)

    dic.update(dict(commands=commands))
    return dic

  @staticmethod
  def read_nums(comm):
    """convert a string of numbers into a list of floats"""
    return [float(x) for x in searchf_number.findall(comm)]
```

```
@staticmethod
def parse_binary_dataflags(number):
  """parse flags what to store"""
  nums = parse_binary(number)
  names = ['General Status', 'Temperature',
       'Magnetic Field', 'Sample Position',
       'Chan 1 Resistivity', 'Chan 1 Excitation',
       'Chan 2 Resistivity', 'Chan 2 Excitation',
       'Chan 3 Resistivity', 'Chan 3 Excitation',
       'Chan 4 Resistivity', 'Chan 4 Excitation']
  empty = [False for x in names]
  bare = dict(zip(names, empty))
  bare.update(dict(zip(names, nums)))
  return bare

@staticmethod
def displaytext_waiting(data):
  """generate the displaytext for the wait function"""
  string = 'Wait for '
  separator = ', ' if data['Temp'] and data['Field'] else ''
  sep_taken = False

  if data['Temp']:
    string += 'Temperature' + separator
    sep_taken = True
  if data['Field']:
    string = string + 'Field' if sep_taken else \
             string + 'Field' + separator
    sep_taken = True
  string += ' & {} seconds more'.format(data['Delay'])
  return string

@staticmethod
def displaytext_scan_T(data):
  """generate the displaytext for the temperature scan"""
  return 'Scan Temperature from {start} to {end} in'.format(**data) + \
         ' {Nsteps} steps, {SweepRate}K/min, '.format(**data) + \
         '{ApproachMode}, {SpacingCode}'.format(**data)

@staticmethod
def displaytext_scan_H(data):
  """generate the displaytext for the field scan"""
  return 'Scan Field from {start} to {end} in'.format(**data) + \
         ' {Nsteps} steps, {SweepRate}K/min, '.format(**data) + \
         '{ApproachMode}, {SpacingCode}, {EndMode}'.format(**data)

@staticmethod
def displaytext_set_temp(data):
  """generate the displaytext for a set temperature"""
  return 'Set Temperature to {Temp} at'.format(**data) + \
         ' {SweepRate}K/min'.format(**data) + \
         ' (rate is only a wish...)'

@staticmethod
def displatext_res_scan_exc(data):
  """generate the displaytext for an excitation scan"""
  # TODO - finish this up
  return 'Scanning RES Excitation'

@staticmethod
```

```
  def displaytext_res(data):
    """generate the displaytext for the resistivity measurement"""
    # TODO - finish this up
    text = 'Resistivity '
    chans = []
    chans.append('Ch1 ')
    chans.append('Ch2 ')
    chans.append('Ch3 ')
    chans.append('Ch4 ')

    for ct, chan_conf in enumerate(data['bridge_conf']):
      if chan_conf['on_off'] is False:
        chans[ct] += 'Off, '
        continue
      chans[ct] += '{limit_current_uA}uA, '.format(**chan_conf)
    chans[-1].strip(',')
    for c in chans:
      text += c
    return text

  @staticmethod
  def displaytext_set_field(data):
    """generate the displaytext for a set field"""
    return 'Set Field to {Field} at {SweepRate}T/min '.format(**data)

  def parse_chamber(self, comm):
    '''parse a command for a chamber operation'''
    nums = self.read_nums(comm)
    dic = dict(typ='chamber_operation')
    if nums[0] == 0:
      dic['operation'] = 'seal immediate'
    if nums[0] == 1:
      dic['operation'] = 'purge then seal'
    if nums[0] == 2:
      dic['operation'] = 'vent then seal'
    if nums[0] == 3:
      dic['operation'] = 'pump continuous'
    if nums[0] == 4:
      dic['operation'] = 'vent continuous'
    if nums[0] == 5:
      dic['operation'] = 'high vacuum'

    dic['DisplayText'] = self.textnesting * self.nesting_level + \
      'Chamber Op: {operation}'.format(**dic)
    return dic

  def parse_set_temp(self, comm):
    """parse a command to set a single temperature"""
    # TODO: Fast settle
    nums = self.read_nums(comm)
    dic = dict(typ='set_T', Temp=nums[0], SweepRate=nums[1])
    dic['DisplayText'] = self.textnesting * \
      self.nesting_level + self.displaytext_set_temp(dic)
    return dic

  def parse_set_field(self, comm):
    """parse a command to set a single field"""
    nums = self.read_nums(comm)
    dic = dict(typ='set_Field', Field=nums[0], SweepRate=nums[1])
    dic['DisplayText'] = self.textnesting * \
```

```
      self.nesting_level + self.displaytext_set_field(dic)
    return dic

  def parse_waiting(self, comm):
    """parse a command to wait for certain values"""
    nums = self.read_nums(comm)
    dic = dict(typ='Wait',
           Temp=bool(int(nums[1])),
           Field=bool(int(nums[2])),
           Position=bool(int(nums[3])),
           Chamber=bool(int(nums[4])),
           Delay=nums[0])
    dic['DisplayText'] = self.textnesting * \
      self.nesting_level + self.displaytext_waiting(dic)
    return dic

  def parse_chain_sequence(self, comm):
    """parse a command to chain a sequence file"""
    file = comm[4:]
    return dict(
          typ='chain sequence',
          new_file_seq=file,
          DisplayText=self.textnesting * \
              self.nesting_level + 'Chain sequence: {}'.format(comm))
    # print('CHN', comm, dic)
    # return dic

  def parse_scan_T(self, comm):
    """parse a command to do a temperature scan"""
    temps = self.read_nums(comm)
    # temps are floats!
    if len(temps) < 6:
      raise AssertionError(
        'not enough specifying numbers for T-scan!')

    dic = dict(typ='scan_T', start=temps[0],
           end=temps[1],
           SweepRate=temps[2],
           Nsteps=temps[3],
           SpacingCode=temps[4],
           ApproachMode=temps[5])
    if int(temps[4]) == 0:
      dic['SpacingCode'] = 'uniform'
    elif int(temps[4]) == 1:
      dic['SpacingCode'] = '1/T'
    elif int(temps[4]) == 2:
      dic['SpacingCode'] = 'logT'

    if int(temps[5]) == 0:
      dic['ApproachMode'] = 'Fast'
    elif int(temps[5]) == 1:
      dic['ApproachMode'] = 'No O\'Shoot'
    elif int(temps[5]) == 2:
      dic['ApproachMode'] = 'Sweep'
    dic['DisplayText'] = self.textnesting * \
      self.nesting_level + self.displaytext_scan_T(dic)
    return dic

  def parse_scan_H(self, comm):
    '''parse a command to do a field scan'''
```

```
    numbers = self.read_nums(comm)
    if len(numbers) < 7:
      raise AssertionError('not enough specifying numbers for H-scan!')

    dic = dict(typ='scan_H',
           start=numbers[0],
           end=numbers[1],
           SweepRate=numbers[2],
           Nsteps=numbers[3])
    if int(numbers[4]) == 0:
      dic['SpacingCode'] = 'uniform'
    elif int(numbers[4]) == 1:
      dic['SpacingCode'] = 'H*H'
    elif int(numbers[4]) == 2:
      dic['SpacingCode'] = 'H^1/2'
    elif int(numbers[4]) == 3:
      dic['SpacingCode'] = '1/H'
    elif int(numbers[4]) == 4:
      dic['SpacingCode'] = 'logH'

    if int(numbers[5]) == 0:
      dic['ApproachMode'] = 'Linear'
    if int(numbers[5]) == 1:
      dic['ApproachMode'] = 'No O\'Shoot'
    if int(numbers[5]) == 2:
      dic['ApproachMode'] = 'Oscillate'
    if int(numbers[5]) == 3:
      dic['ApproachMode'] = 'Sweep'

    if int(numbers[6]) == 0:
      dic['EndMode'] = 'persistent'
    if int(numbers[6]) == 1:
      dic['EndMode'] = 'driven'
    dic['DisplayText'] = self.textnesting * \
      self.nesting_level + self.displaytext_scan_H(dic)
    return dic

  def parse_scan_time(self, comm):
    nums = self.read_nums(comm)
    if len(nums) < 3:
      raise AssertionError(
        'not enough specifying numbers for time-scan!')

    dic = dict(typ='scan_time', time=nums[0], Nsteps=nums[1])

    if int(nums[2]) == 0:
      dic['SpacingCode'] = 'uniform'
    if int(nums[2]) == 1:
      dic['SpacingCode'] = 'ln(t)'

    dic['DisplayText'] = self.textnesting * self.nesting_level + \
      'Scan Time {time}secs in {Nsteps} steps, {SpacingCode}'
    return dic

  def parse_scan_P(self, comm):
    nums = self.read_nums(comm)
    if len(nums) < 3:
      raise AssertionError(
        'not enough specifying numbers for position-scan!')
```

```
    dic = dict(typ='scan_position',
               start=nums[0],
               end=nums[1],
               speedindex=nums[2],
               Nsteps=nums[3])

    dic['ApproachMode'] = 'Sweep' if len(nums) > 3 else 'Pause'
    dic['DisplayText'] = self.textnesting * self.nesting_level + \
      'Scan Position from {start} to {end} in '.format(**dic) + \
      '{Nsteps} steps, {speedindex}, {ApproachMode} '.format(**dic)
    return dic

  def parse_beep(self, comm):
    '''parse a command to beep for a certain time at a certain frequency'''
    nums = self.read_nums(comm)
    if len(nums) < 2:
      raise AssertionError('not enough specifying numbers for beep!')

    dic = dict(typ='beep', length=nums[0], frequency=nums[1])
    dic['DisplayText'] = self.textnesting * self.nesting_level + \
      'Beep for {length}secs at {frequency}Hz'.format(**dic)
    return dic

  def parse_res_change_datafile(self, comm):
    """parse a command to change the datafile"""
    file = searchf_string.findall(comm)
    return dict(typ='res_change_datafile', new_file_data=file,
          mode='a' if comm[-1] == '1' else 'w',
          # a - appending, w - writing, can be inserted
          # directly into opening statement
          DisplayText=self.textnesting * self.nesting_level + \
                'Change data file: {}'.format(file))
    # print('CDF', comm, dic)
    # return dic

  def parse_res_datafilecomment(self, comm):
    """parse a command to write a comment to the datafile"""
    comment = searchf_string.findall(comm)[0]
    dic = dict(typ='res_datafilecomment',
           comment=comment,
           DisplayText=self.textnesting * self.nesting_level + \
                 'Datafile Comment: {}'.format(comment))
    return dic

  @staticmethod
  def parse_res_bridge_setup(nums):
    """parse the res bridge setup for an excitation scan"""
    bridge_setup = []
    bridge_setup.append(nums[:5])
    bridge_setup.append(nums[5:10])
    bridge_setup.append(nums[10:15])
    bridge_setup.append(nums[15:20])
    for ct, channel in enumerate(bridge_setup):
      bridge_setup[ct] = dict(limit_power_uW=channel[1],
                  limit_voltage_mV=channel[4])
      bridge_setup[ct]['ac_dc'] = 'AC' if channel[2] == 0 else 'DC'
      bridge_setup[ct]['on_off'] = True if channel[0] == 2 else False
      bridge_setup[ct]['calibration_mode'] = 'Standard' if channel[
        3] == 0 else 'Fast'
    return bridge_setup
```

```
def parse_res(self, comm):
  """parse a command to measure resistivity"""
  nums = self.read_nums(comm)
  dataflags = self.parse_binary_dataflags(int(nums[0]))
  reading_count = nums[1]
  nums = nums[2:]
  bridge_conf = []
  bridge_conf.append(nums[:6])
  bridge_conf.append(nums[6:12])
  bridge_conf.append(nums[12:18])
  bridge_conf.append(nums[18:24])
  for ct, channel in enumerate(bridge_conf):
    bridge_conf[ct] = dict(limit_power_uW=channel[2],
                limit_current_uA=channel[1],
                limit_voltage_mV=channel[5])
    bridge_conf[ct]['on_off'] = True if channel[0] == 2 else False
    bridge_conf[ct]['ac_dc'] = 'AC' if channel[3] == 0 else 'DC'
    bridge_conf[ct]['calibration_mode'] = 'Standard' if channel[
      4] == 0 else 'Fast'
  data = dict(typ='res_measure',
        dataflags=dataflags,
        reading_count=reading_count,
        bridge_conf=bridge_conf)
  data['DisplayText'] = self.textnesting * \
    self.nesting_level + self.displaytext_res(data)
  return data

def parse_res_scan_excitation(self, comm):
  """parse a command to do an excitation scan"""
  nums = self.read_nums(comm)
  scan_setup = []
  scan_setup.append(nums[:3])  # 1
  scan_setup.append(nums[3:6])  # 2
  scan_setup.append(nums[6:9])  # 3
  scan_setup.append(nums[9:12])  # 4
  for ct, channel in enumerate(scan_setup):
    scan_setup[ct] = dict(start=channel[0], end=[channel[1]])
    if channel[-1] == 0:
      scan_setup[ct]['Spacing'] = 'linear'
    if channel[-1] == 1:
      scan_setup[ct]['Spacing'] = 'log'
    if channel[-1] == 2:
      scan_setup[ct]['Spacing'] = 'power'

  dataflags = self.parse_binary_dataflags(nums[14])
  n_steps = nums[12]
  reading_count = nums[13]
  bridge_setup = self.parse_res_bridge_setup(nums[15:35])
  data = dict(typ='res_scan_excitation',
        scan_setup=scan_setup,
        bridge_setup=bridge_setup,
        dataflags=dataflags,
        n_steps=n_steps,
        reading_count=reading_count)
  data['DisplayText'] = self.textnesting * \
    self.nesting_level + self.displatext_res_scan_exc(data)
  return data
```