

# **Processing and Interpretation of Momentum-Resolved Electron Energy-Loss Spectroscopy Data**

Jeroen Sangers

1st July 2021





# **Processing and Interpretation of Momentum-Resolved Electron Energy-Loss Spectroscopy data**

Jeroen Sangers  
4645197

Assessment committee :  
Dr. S. Conesa-Boj  
Dr. T. van der Sar  
Department of Quantum Nanoscience  
TU Delft, Faculty of Applied Sciences

Student:  
Sangers, Jeroen  
TU Delft, BSc program Applied Physics

## **Abstract**

Momentum resolved electron energy loss spectroscopy (MREELS) can provide a trove of valuable data about a specimen but often requires specialised and proprietary software to analyse. In this work Python methods that were designed for the analysis of MREELS data are explained and tested. This provided results consistent with earlier findings, namely the existence of the dispersionless 3.5 eV, 7.25 eV and 22 eV energy loss peaks in the low momentum transfer region of the q-EELS spectral data. The dispersion of the bulk plasmon was also tracked along different crystallographic directions and found to potentially be anisotropic in the high momentum transfer regime.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Table of contents</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Theory</b>	<b>2</b>
2.1 The Transmission electron microscope . . . . .	2
2.2 Crystal structure . . . . .	3
2.3 Electron scattering theory . . . . .	3
2.4 Momentum-resolved electron energy-loss spectroscopy . . . . .	4
2.4.1 Energy filtered transmission electron microscope . . . . .	4
2.5 Diffraction image . . . . .	5
2.6 q-EELS spectra and q-EELS map . . . . .	5
2.7 Physical relevance of (MR)EELS data . . . . .	6
<b>3 Experimental Method</b>	<b>6</b>
3.1 Data format . . . . .	6
3.2 Data correction techniques . . . . .	6
3.2.1 Removing or changing values . . . . .	6
3.2.2 Zero-loss peak and Batson correction . . . . .	7
3.3 Data processing techniques . . . . .	8
3.3.1 Integration techniques . . . . .	8
3.3.2 Slicing techniques . . . . .	8
3.4 Data extraction . . . . .	8
<b>4 Results</b>	<b>9</b>
4.1 Comparing data extraction techniques . . . . .	9
4.2 Reviewing Batson correction . . . . .	10
4.3 Interesting features from data . . . . .	11
<b>5 Conclusion</b>	<b>12</b>
<b>List of references</b>	<b>13</b>
<b>Appendix</b>	<b>15</b>

# 1 Introduction

Transmission electron microscopes are incredible machines capable of imaging structures at the atomic scale and can provide useful insight into the electronic properties of the sample being imaged.

The files produced by the transmission electron microscope are usually in a proprietary format and made to be analysed with specialised software that can often be expensive. Therefore this work will introduce some of the methods implemented in an open-source Python library, the methods to be discussed in this work are those that are used to analyse momentum resolved electron energy loss spectroscopy data sets. This work will use a data set procured by imaging a  $\gamma$ -indium selenide sample in an energy filtered transmission electron microscope set up for diffraction images.

The work starts by introducing all the necessary concepts in the theory section 2 such that his work can be understood without prior knowledge in the field of electron microscopy. In the experimental method section, chapter 3, a majority of the methods implemented will be presented and explained. The effect and usefulness of these methods are discussed in the results section, chapter 4. The work will be discussed, further intentions with respect to the software will be clarified and the final conclusion is presented in the conclusion chapter 5.

## 2 Theory

### 2.1 The Transmission electron microscope

The Transmission electron microscope (TEM) is a microscope that far exceeds the capabilities of a normal light microscope. Both types of microscope use a series of lenses to magnify the image of a specimen. A normal light microscope can amplify an image up to about  $1500\times$  and is limited by the diffraction limit of light. Assuming an average wavelength of  $550\text{nm}$  for green light, a high-end microscope is limited to resolving features  $100\text{nm}$  apart. This limit is insufficient for looking at atomic structures. [1]

An electron microscope circumvents this limit by using electrons, not light, to probe the specimen. Electrons when accelerated have a smaller wavelength than light thus allowing for images with resolved features as small as  $0.05\text{nm}$ . [2–4] The TEM works by releasing electrons from an electron source and accelerating them to an energy typically expressed in kilo-electronvolt. After being accelerated the electrons pass multiple electromagnetic lenses and a condenser aperture to shape the beam before it 'illuminates' the specimen as illustrated in figure 1. The beam incident on the sample is limited to a illumination semi-angle  $\alpha$  which is inversely proportional to the resolution, but limiting  $\alpha$  decreases the amount of electrons incident on the specimen and thus a frame needs more time for a exposure. After having interacted with the specimen the beam is again limited by an aperture, this aperture sets the collection semi-angle  $\beta$  which controls the limit of scattering angles allowed into the imaging lenses. After the beam is conditioned by the imaging lenses it passes through a electromagnetic prism which bends electrons with different energies in different directions. The energy filter is used for the energy filtered TEM images discussed in section 2.4.1.

Two types of images can be made with the TEM, a normal image which shows the magnified sample and a diffraction pattern image which can be made by placing the capture device in the focal point of the lens and filter system. A diffraction mode image shows the diffraction peaks that are characteristic of the sample and yields information on the reciprocal lattice of the sample. [5]

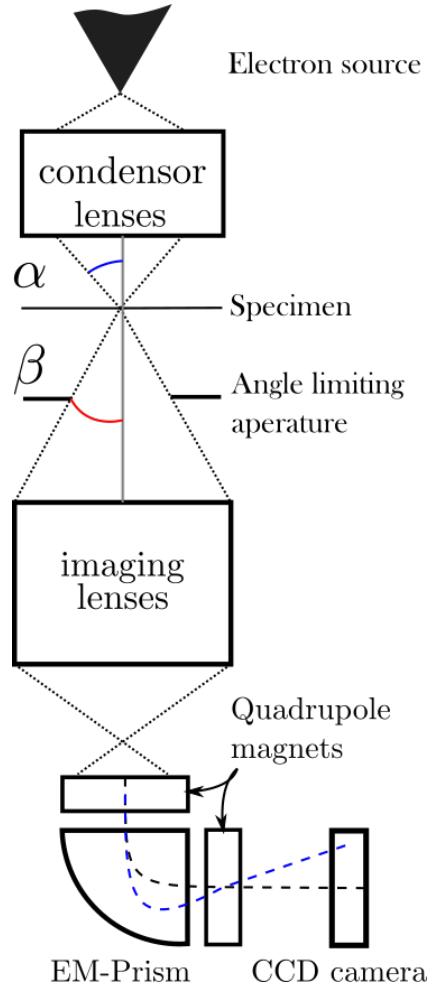


Figure 1: The optical column of the transmission electron microscope.

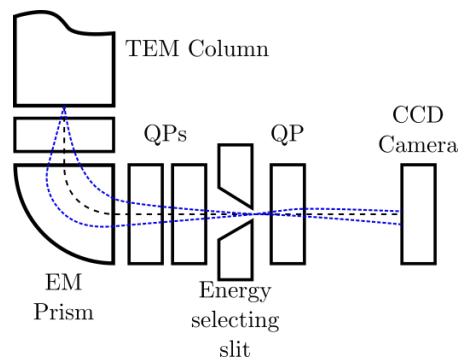


Figure 2: A post-column energy-filtering and imaging device in which electrons of a certain energy (blue dashed lines) are filtered by an electromagnetic prism and focused on the capture device by quadrupole magnets.

## 2.2 Crystal structure

A crystal is build up of unit cells in such a way that the whole crystal can be made by shifting and aligning unit cells. These unit cells are build up of atoms whose position in the unit cell can be fully expressed in terms of primitive basis vectors. If the crystal were to be cut in half by a plane this plane would be expressed by fractional coordinates, these fractional coordinates show where along a basis vector the plane and basis vector intersect. Such a plane is called a Miller plane and the corresponding coordinates are called the Miller indices. Miller planes that have the same orientation but are shifted to different unit cells share the same Miller indices and are called a family of planes. The crystal structure of  $\gamma$ -Indium Selenide with its basis vectors and unit cell is shown in figure 3. In reciprocal space the diffraction spots correspond to a certain family of planes, the position of diffraction spots can be expressed in reciprocal basis vectors. A reciprocal basis unit cell is shown in figure 4. In this figure the high-symmetry points are shown as well as the reciprocal basis vectors.

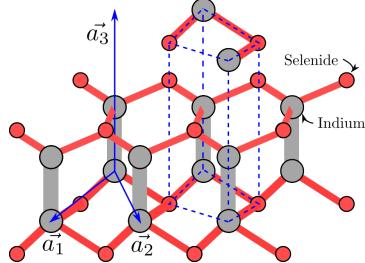


Figure 3: The crystal lattice of  $\gamma$ -InSe [7, 8]. The basis vectors are displayed as the blue arrows and a unit cell is outlined by the dashed blue box.

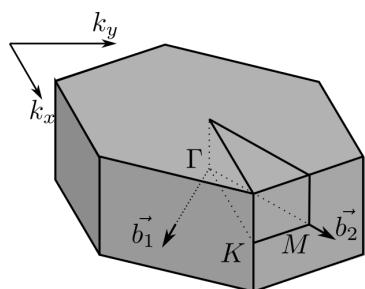


Figure 4: A reciprocal unit cell in momentum space containing the high-symmetry  $\Gamma$ -,  $M$ - and  $K$ -points. The reciprocal lattice vectors are denoted by the  $\vec{b}_1$  and  $\vec{b}_2$  vectors.

## 2.3 Electron scattering theory

In a TEM setup electrons are essentially shot through a sample in which the electrons can either simply pass through or scatter, in the latter scenario there are two possibilities, electrons can scatter elastically or inelastically. Scattering is a result of the interaction between the sampling electrons from the TEM source and the charged particles in the specimen. When scattering elastically the electrons interact with a nucleus of the specimen whose mass is many times greater than that of the sampling electron, resulting in a small and usually unmeasurable energy transfer. In a crystalline specimen electrons can only be scattered at certain angles due to the crystal structure creating a diffraction pattern of bright spots. In cases of large scattering angles the electron does transfer a significant amount of energy and can even reverse direction, this energy transfer can permanently displace atoms in the crystal structure causing a defect. [5, 9]

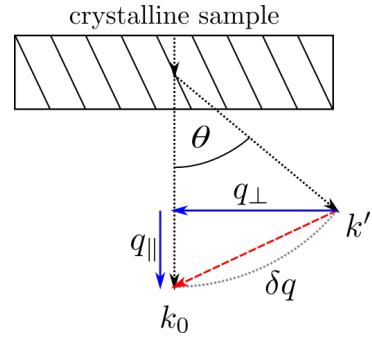


Figure 5: The elastic scattering  $k'$  of an electron over an angle  $\theta$  due to the interaction with a crystalline sample.

When the sampling electron interacts with an electron in the specimen's crystal lattice inelastic occurs due to the similarity in mass between the two electrons. The energy transfer of this interaction ranges from a few electronvolts up to multiple hundreds of electronvolts. Inelastic scattering not only results in an energy transfer but also in a momentum transfer as shown in figure 5, the  $k'$ -vector shows a scattered electron that deviates from the not scattered electron vector  $k_0$ . The total momentum transfer is the sum of the perpendicular momentum transfer  $q_{\perp}$  proportional to the scattering angle  $\theta$  and the momentum transfer parallel to the undisturbed path due to an energy transfer from the sampling electron to the sample [5, 9, 10]. This parallel momentum transfer is thus proportional to the energy-loss of the electron. Figure 6 shows the band structure of the crystalline sample of which the electrons scatter. In this figure two dispersion bands are shown, both bands can be occupied by electrons

of certain energies, to excite an electron from the blue band to the red band an electron needs either energy (path  $t_1$ ) or energy and momentum (path  $t_2$ ). The needed energy and momentum are transferred from an incident sampling electron in the inelastic interaction. By measuring the energy and momentum of a scattered electron it is possible to piece together all the combinations of energy and momenta transfer possible and thus find the band structure of the sample.

Another form of inelastic scattering is plasmon excitation. Since the outer-shell electrons of an atom are only weakly bound to the nucleus due to screening effects but are coupled together by electrostatic interaction. These delocalised electrons form an energy band similar to that shown in figure 6. When a fast-moving sampling electron is shot through the sample all nearby outer-shell electrons are displaced. If the sampling electron's velocity exceeds the fermi speed the displacement of outer-shell electrons creates an oscillating ripple creating waves of alternating positive and negative electric charge, this is known as a plasmon wake. [5]

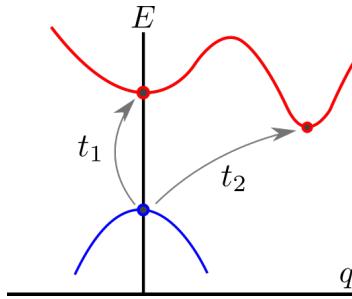


Figure 6: The band structure of the crystalline sample in fig. 5 showing both a direct bandgap  $t_1$  and an indirect bandgap  $t_2$ .

## 2.4 Momentum-resolved electron energy-loss spectroscopy

Momentum resolved electron energy-loss spectroscopy hereafter abbreviated as MREELS is a TEM imaging technique in which the imaging plane of the CCD camera is placed in the focal point of the imaging lenses and energy filter. This allows the camera to take a diffraction mode image in which the diffraction pattern of the scattering electrons is shown [10, 11]. This is illustrated in figure 7. In this figure the crystalline sample is shown as the box with the slanted lines that represent the Miller planes of the crystal structure. As shown in the figure all electrons that scatter off of the same family of Miller planes get focused on the same region on the imaging sensor, electrons that do not scatter also get focused in one spot at the centre of the diffraction pattern. A diffraction mode image does not im-

age normal space but instead shows reciprocal space which is also the reason this type of image is useful. In a normal image one would attribute lengths to the axes of an image but in a diffraction mode image the separation of features is given by a momentum difference. The separation of the light and dark grey arrows on the imaging plane in figure 7 is thus equal to the difference in perpendicular momentum transfer between scattering electrons (light grey) and electrons that do not scatter (dark grey), a 3D representation is presented in the Appendix. Since the momentum transfer for the electrons that do not scatter is zero the momentum transfer for scattering of a certain family of planes can be determined.

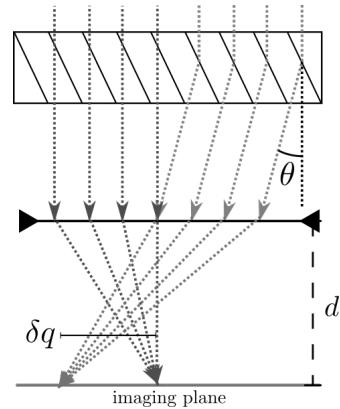


Figure 7: The focusing of scattered electrons on the imaging plane in discrete momenta transfers.

### 2.4.1 Energy filtered transmission electron microscope

An energy filtered transmission electron microscope is a microscope with an energy filter placed before the capture device. Energy filtering is accomplished by the use of an electromagnetic prism such as the one shown in figure 2. These prisms just like ordinary prism disperse the electrons with different wavelengths which are proportional to electron energy. By sliding a slit into the cone of dispersed electrons it is possible to choose a finite range of electron energies to image. After filtering by means of the prisms and the slit, the beam of electrons is conditioned by quadrupole electromagnets to shape the beam. [12] There also exist in-column energy filters such as an  $\Omega$ -filter [13] or a Wien-filter [14], in-column filters can be beneficial for limiting aberrations. [5] The EFTEM setup can be used in conjunction with the MREELS imaging technique to gather information on both the momentum transfer of the electron (via MREELS) and the energy-loss associated (via EFTEM) with that momentum transfer.

## 2.5 Diffraction image

In a diffraction mode image a set of bright spots for which scattering probability is high is present in momentum space. These bright spots correspond to a family of planes, to determine to which family of planes the bright spots correspond they first have to be indexed. Indexing is done by using the geometric properties of the image to narrow down how the image is oriented. By measuring the distance from the middle brightest spot to one of the closer outward ones, measuring the angle between lines connecting these spots and searching for these values in a database it is possible to determine to zone and orientation of the image<sup>1</sup>. Once the image is indexed the family of planes can be attributed to the bright diffraction spots as done in figure 8. The sample is oriented in such a way that the incident electron beam is parallel to the  $(\bar{1}11)$ -zone axis.

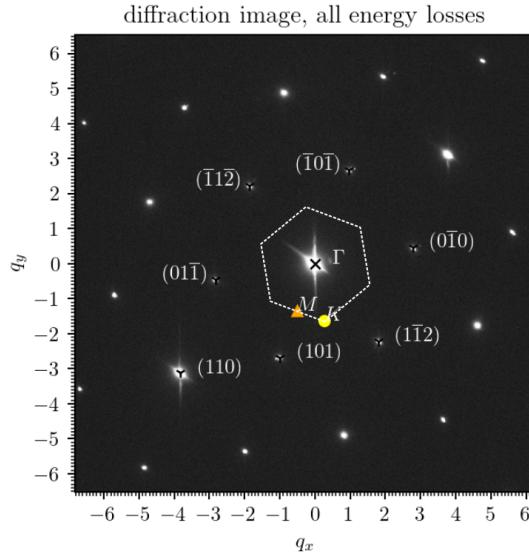


Figure 8: A diffraction mode image showing the total scattering intensity at momentum space positions. The diffraction spots are indexed with Miller indices. The  $M$ - and  $K$ -point and first Brillouin zone are also shown and are the same as in figure 4.

## 2.6 q-EELS spectra and q-EELS map

An electron energy-loss spectroscopy spectrum is a plot that show the scattering intensity at a certain energy-loss  $\Delta E$  of the electron, a q-EELS spectrum also has a momentum transfer value  $q_{\perp}$  that corresponds to the whole spectrum. A q-EELS spectrum is plotted in figure 9, this spectrum corresponds to a momentum transfer of zero and is thus in the middle of the centre diffraction spot. The spectrum starts with a high peak in the low-loss region, this peak is

called the zero-loss peak and shows the high intensity of electrons not losing energy and in this case momentum when passing through the sample. This peak is a problem since it masks the low-loss dispersion data partially. For high energy-loss resolution images it can be successfully removed using machine learning techniques. [15] In this work a simpler correction will be used since the energy-loss resolution is not high enough to benefit from this method.

The q-EELS map shows the same energy-loss spectra but for multiple momenta transfers at a time. For every combination of energy-loss and momentum transfer there is a single intensity value which is shown in colour, the brighter the pixel the more electrons scatter with that combination of energy-loss and momentum transfer. A q-EELS map is pictured in figure 10, the zero-loss peaks still show as the bright streak at 0 eV for all momenta transfers. This spectrum was created by walking the path from  $\Gamma$  to  $M$  and then to  $K$  in which the points are the same as in figure 8.

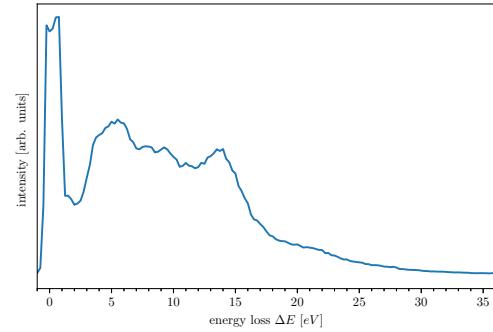


Figure 9: A q-EELS spectrum of  $\gamma$ -InSe for  $q_{\perp} = 0$

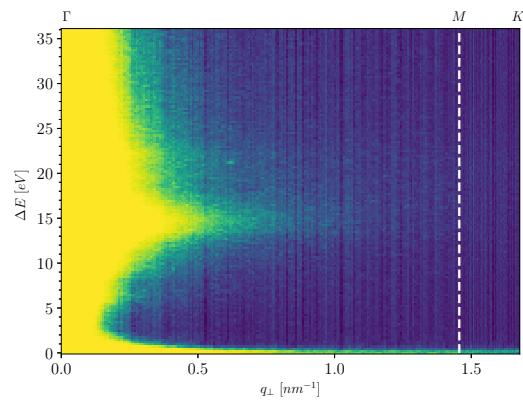


Figure 10: A q-EELS map of  $\gamma$ -InSe along  $\Gamma \rightarrow M \rightarrow K$

<sup>1</sup>A huge thank you to A. Brokkelkamp for showing how to index the diffraction image

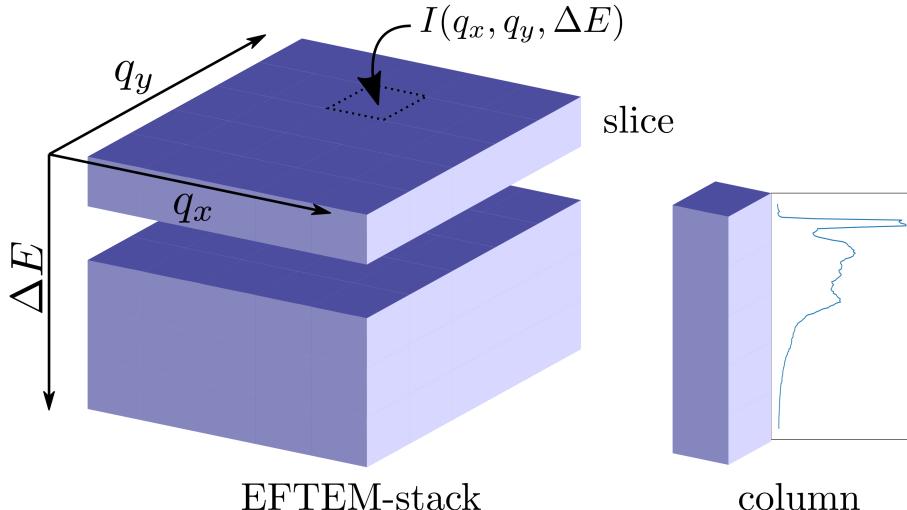


Figure 11: An illustration of an EFTEM-stack, slice, column and corresponding axes.

## 2.7 Physical relevance of (MR)EELS data

As hinted in section 2.4, the combination of both energy and momentum information allows for the reconstruction of the band structure of the specimen. This information can be used to determine the dens-

ity of states of the specimen [16] [5]. EELS spectra can also be used to determine the electronic properties of the specimen, the bandgap of a semiconductor and the dielectric function, as well as mechanical properties [5]. Different signature peaks in the EELS data can be used to determine the elemental makeup of the specimen [5].

## 3 Experimental Method

### 3.1 Data format

For this work a specimen of  $\gamma$ -InSe was imaged using the momentum resolved electron energy-loss technique in an energy filtered transmission electron microscope. The file was supplied with aligned zero-loss peaks in Gatan's .dm4 format which had to be converted to a python object for further data processing in python, for the conversion from Gatan's proprietary file to a python object the excellent ncempy package was used. [17]

The measurement data is gathered by taking energy filtered diffraction images at different energy-losses. All these images are stacked corresponding to their energy-loss value, resulting in a EFTEM image stack as illustrated in figure 11. In the illustrated cube the horizontal planes or slices are diffraction mode images associated with an energy-loss  $\Delta E$ , these energy slices have their own momentum axes that are not necessarily aligned with the whole stack. After alignment the individual pixels in the EFTEM stack can be fully expressed by four values, three coordinates:  $q_x$ ,  $q_y$ ,  $\Delta E$  and one scattering intensity  $I$ .

An example of such a spectrum is illustrated in fig-

ure 9. Another way is to extract multiple of these columns such that their total momentum increases, doing this yields a energy-momentum map as pictured in figure 10. Once the EFTEM stack is aligned it is possible to start extracting sets of values in meaningful ways. One way is to take a single column from the top of the stack to the bottom, doing this results in a 1D-array of values for a set position in momentum space and varying energy-loss, this is called an electron energy-loss spectrum.

### 3.2 Data correction techniques

#### 3.2.1 Removing or changing values

Before any meaningful results can be extracted from the EFTEM stack it is important to first correct the data. This process starts by removing all the values corresponding to negative energy-losses, these values are incorrectly indexed as a result of the microscope software aligning the slices such that the zero-loss peak is at an energy-loss of 0eV. It does this by shifting slices up or down to match them to their true energy-loss, slices corresponding to negative energy-losses are duplicates of slices at positive energy-losses.

After removing the negative energy-loss slices the

negative intensity values are raised by the minimum amount needed to make all values positive. This means that all values of the entire stack are raised by the absolute value of the most negative intensity. By making all intensities positive, the Batson correction and data processing techniques can better defined in code.

Since the acquisition time of the EFTEM stack is quite large and not all slices are taken at the same time it might happen that the sample moves due to disturbances. Correcting this is done by convoluting a slice with Sobel matrices to get an edge-detected image, these images are then aligned by means of Fourier phase correlation [18, 19].

### 3.2.2 Zero-loss peak and Batson correction

To try and remove the zero-loss peak of the q-EELS spectrum a Batson correction is performed. A Batson correction normally uses the sum of all EELS spectra in an EFTEM image of the sample to try and correct plural scattering and subtract the zero-loss peak [20]. Since such an image is not available the sum of all q-EELS spectra in the first Brillouin zone is used instead. This region was more likely to correspond to a true image of the sample, the sum of the q-EELS spectra in the first Brillouin zone is called the correction spectrum. The Batson correction is then performed by scaling the correction spectrum for each to be corrected q-EELS spectra in such a way that the integrated intensities of the zero-loss peaks match, then the "centre of intensity" of the zero-loss peaks is determined and the correction spectrum is subtracted from the to be corrected q-EELS spectra with the centres of intensity aligned. Looking at figure 12, in the top plot the Batson correction method will sum the intensities in between the vertical dashed lines, the window, and then scale the blue spectrum such that the sums of both spectra are equal in the window. The Batson method will then align the maxima, denoted by the dots, by shifting the spectra and then subtract the scaled blue spectrum from the original black spectrum. This would result in a corrected spectrum, shown in the bottom plot of figure 12. The Batson correction code was rewritten in python after being originally written in DigitalMicrographs scripting language for an earlier work. [21].

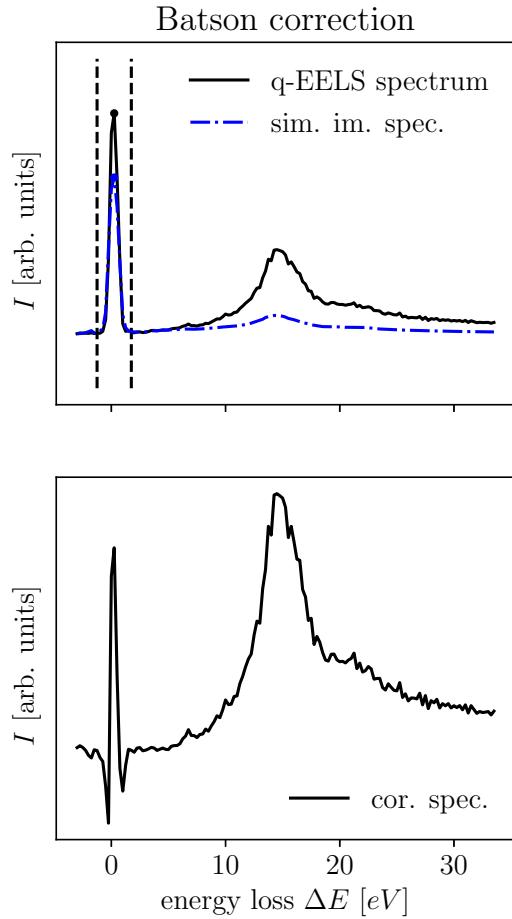


Figure 12: Top plot: Showing two q-EELS spectra. The black spectrum is the to be corrected spectrum and the blue spectrum is the simulated image spectrum. Bottom plot: Showing a Batson corrected q-EELS spectra.

### 3.3 Data processing techniques

Since the EFTEM data is essentially a 3D data cube of values it needs to be processed in such a way that the information it contains can be extracted and interpreted. This is something that can not be done with the cube itself since it is hard to represent 3D data in meaningful ways.

#### 3.3.1 Integration techniques

To achieve the goal of representing the data in a meaningful ways two integration techniques have been implemented in Python. A radial integration method that finds the centre of the image stack which is determined to be the brightest pixel of the unaltered EFTEM stack. Since this is most probably the zero-loss peak at the centre of the unscattered beam. From this centre outward the method sums all EELS spectra in circle from a certain radius to that radius plus a ringsize and attributes this value to the average value of  $q_{\perp}$  in the ring, as shown in figure 13. The starting and ending radius as well as the ringsize can be specified by the user. This method is the same as averaging over nearby momenta transfers. When this method is finished the result is an array of summed EELS spectra for a set of rings. This can be plotted as a momentum-energy EELS map as shown in figure 10. Instead of integration in circle segments over the entire stack the user might want to only do a line-like integration towards a single diffraction peak. If this is the case a similar method to the one described above is called except for the change that instead of circle segments it integrates over pie piece like segments, such that only the values in-between the green lines in figure 13 are used.

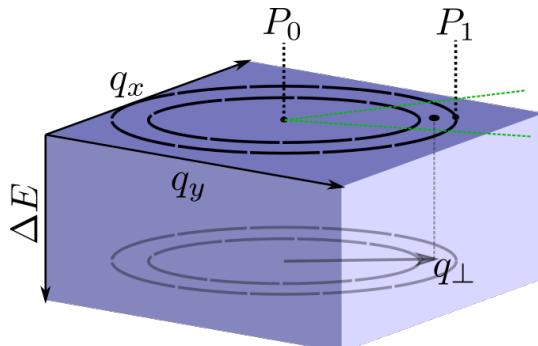


Figure 13: A 3D data cube or EFTEM stack with points  $P_0$  and  $P_1$  or the point from which and to which is being integrated by averaging q-EELS spectra.

#### 3.3.2 Slicing techniques

Slicing refers to the term of "array slicing" in Python/Numpy [22] which is the built-in way to select data from an array by specifying the positions of the data to select. [23–25] The slicing technique allows the user to specify one or two points and extracts all columns along that the line (arrow in figure 13) intersects between the two points. If one point is specified by the user the other point will be the centre of the EFTEM stack. This method returns an array of all the EELS spectra for the points along the line. This can again be plotted as a momentum-energy map.

### 3.4 Data extraction

Once the full 3D EFTEM stack is reduced to useful momentum-energy maps or  $q$ -EELS spectra it is possible to start identifying interesting features such as peaks in the  $q$ -EELS spectra at certain regions or bright spots in the momentum-energy maps. For instance, if long bright streaks can be observed in the momentum-energy map they might hint at a relation between energy and momentum of an often occurring reaction. One of such bright streaks that is present and will be tracked is the plasmon peak at roughly 14.5 eV energy-loss.

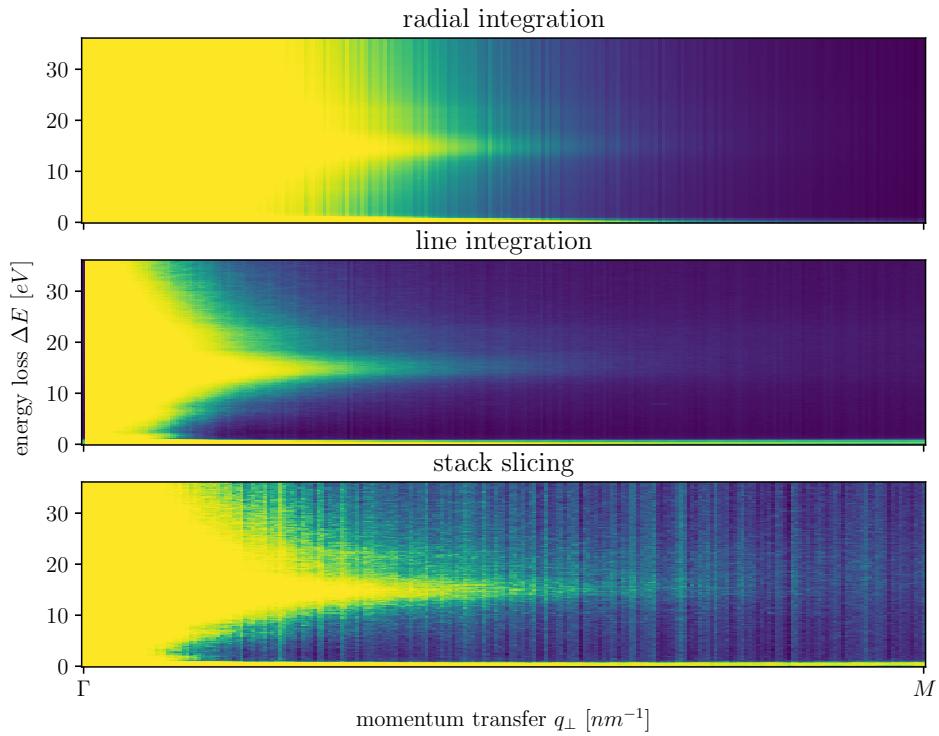


Figure 14: comparing various ways in which the q-EELS data can be extracted from the EFTEM stack

## 4 Results

### 4.1 Comparing data extraction techniques

In figure 14 three different q-EELS maps are plotted, the radial, linear and slicing technique from top to bottom respectively. It is immediately obvious that the radial and linear "integration" techniques have higher resolutions along the momentum transfer axis. This is due to the fact that slicing the EFTEM stack between two points yields a pixelated line, similarly to opening paint and drawing a line, whereas the integration techniques also extract information from pixels just besides the line between

two points resulting in smaller momentum transfer steps. One drawback to these techniques is that they also add intensity values from pixels not directly between to diffraction spots to the q-EELS map, this is especially so with the radial integration technique that uses all values on a circle outwards from the starting point. The integration techniques also yield a smoother gradient from low- $q_{\perp}$  to high- $q_{\perp}$  since it averages multiple q-EELS spectra in a ringsize, this again might introduce spectra not truly on the line in between two points but does average out any unwanted errors such as external radiation.

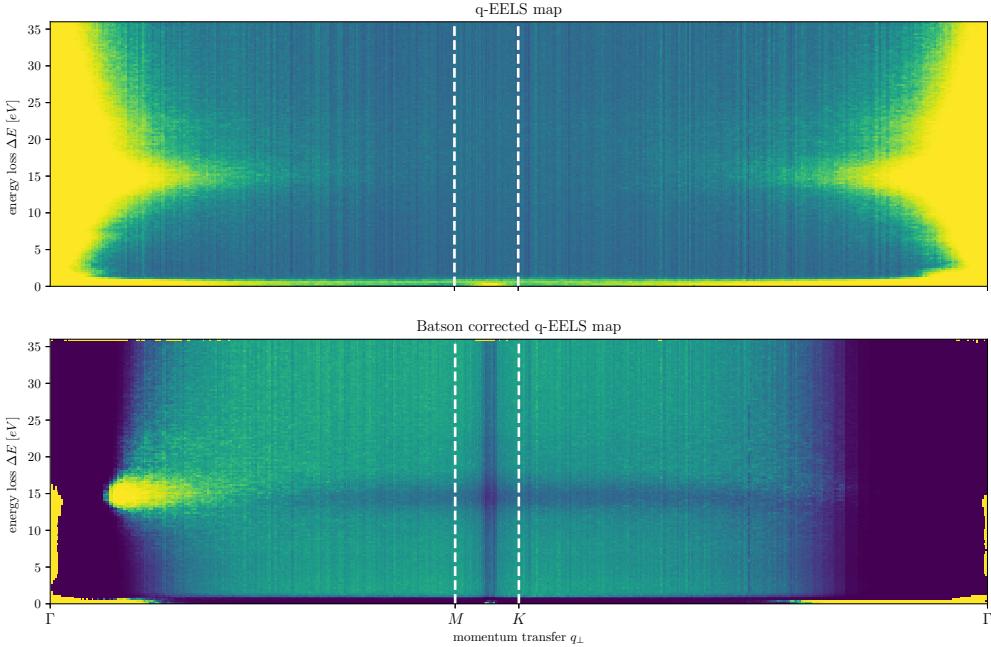


Figure 15: Top figure showing a q-EELS map extracted from the first Brillouin zone along the  $\Gamma, M, K$ -triangle edges by using the slicing technique. Bottom figure shows the Batson corrected q-EELS map.

## 4.2 Reviewing Batson correction

Carrying out the Batson correction on the sliced q-EELS map of the  $\gamma$ -InSe sample in the  $\Gamma \rightarrow M \rightarrow K$ -direction we can clearly see it has a big effect on the map. The q-EELS spectra near the low momentum transfer  $\Gamma$ -point are nearly fully erased. This is because of the high amount of elastically scattered electrons contributing to the zero-loss peak, creating a high zero-loss peak to low-loss region fraction. Since the amount of removed low-loss plural scattering is determined by this fraction which is too large here, the function overestimates the amount of plural scattering and thus subtracts too much from the q-EELS map. This was to be expected as it was also reported by earlier work which used this technique. [26] Amplifying this region was not the intention of applying the Batson correction since the original EFTEM stack's data was sufficient. The high momentum transfer region near the high symmetry  $M$ - and  $K$ -points has benefitted from the correction, the q-EELS spectra had their zero-loss peaks removed and contrast heightened making it easier to resolve features in these spectra.

An apparent but not unexplainable feature or artefact of the correction is the dark band in the middle between the  $M$ - and  $K$ -points. The original q-EELS map (top figure in 15) had a bright spot in the

middle of the zero-loss peak at the same position. The Batson correction uses an integral over the intensity of the zero-loss peak to determine how much plural scattering needs to be removed. Thus if there is an extra bright feature in the zero-loss peak the correction spectrum gets improperly scaled to be too large for what it needs to correct for and will upon subtraction remove too much intensity from the rest of the spectrum.

Another dark shadow can be seen spanning the whole momentum transfer range at about 14 eV energy-loss. This darker streak seems to curve upwards towards the middle and come back down, trailing nearly perfectly the bulk plasmon dispersion. This streak can not be explained by the subtraction since it curves upwards and the correction spectrum is subtracted centred around the zero-loss peak which curves downward towards negative energy-losses.

The zero-loss peak itself is removed nicely from the q-EELS map for a majority of the momentum transfers. The Batson correction seems to leave a bit of the zero-loss peaks for moderately low momentum transfers. This could be due to the fact that the Batson correction uses a simulated image spectrum for its correction which underrepresents the zero-loss peak. A simulated image spectrum is used for the reasons outlined in 3.2.2 and might not be accurate

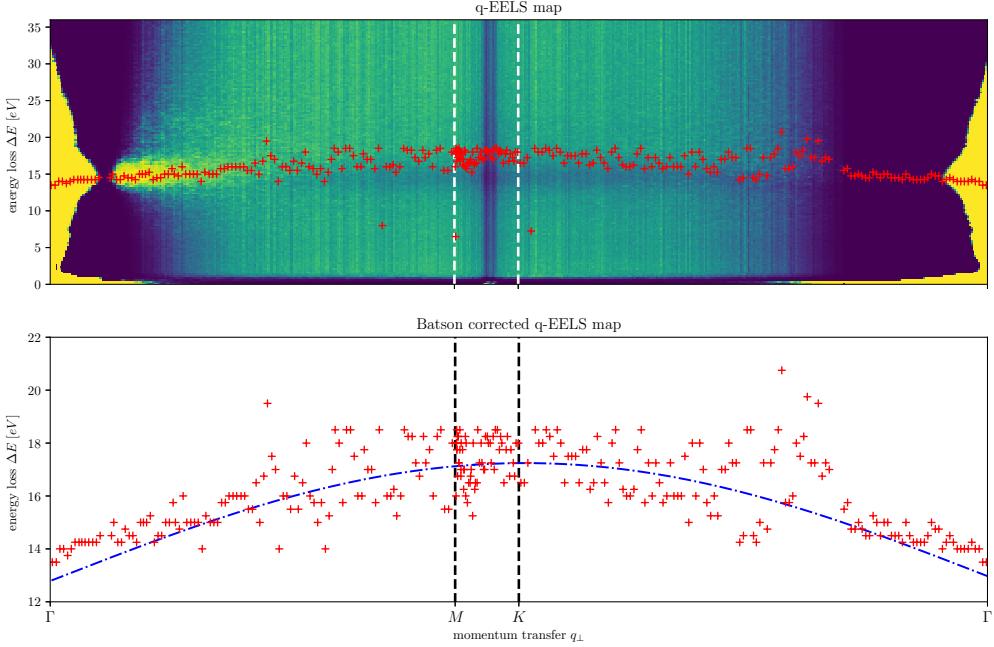


Figure 16: tracked peaks on stitched q-EELS map

enough as Batson himself reported in his work [20]. It could also be that the sum of smaller intense dispersions remains in the low-loss region. This would

be hard to tell without a higher energy resolution. Some more Batson corrected q-EELS maps have been included in the appendix 19.

### 4.3 Interesting features from data

For tracking the peak of the bulk plasmon a high enough separation between the peak and background spectrum was needed. The q-EELS map in figure 16 was made by stitching together the uncorrected q-EELS map in the low momentum transfer regime and the Batson corrected q-EELS map for high momenta transfers. The q-EELS data was gathered using the slicing method from the  $\Gamma$ -point to  $M$ , then  $K$ , and back to  $\Gamma$  with the definition of the points the same as earlier q-EELS maps and consistent with figure 8. The energy-loss value of the plasmon peak was found by searching for the highest intensity value in a window around a centre energy-loss value, this method does not always find the true peak as the outliers both far above and below the central point cloud show. The values for the peak of the plasmon were scrubbed from outliers and plotted in the bottom plot of figure 16, a sine centred around the point in the middle of the  $M$  and  $K$  lines was fitted and provides a decent result around the middle but deviates from the true function at the  $\Gamma$ -lines.

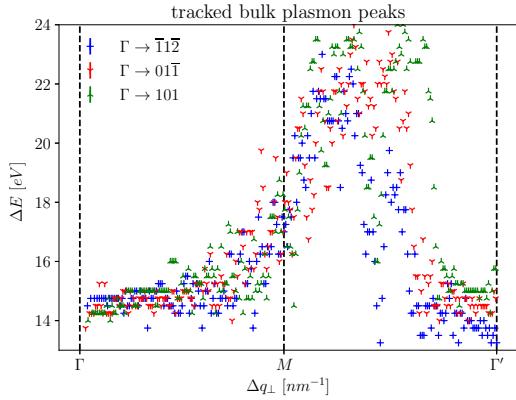


Figure 17: tracked peaks of the plasmon dispersion.

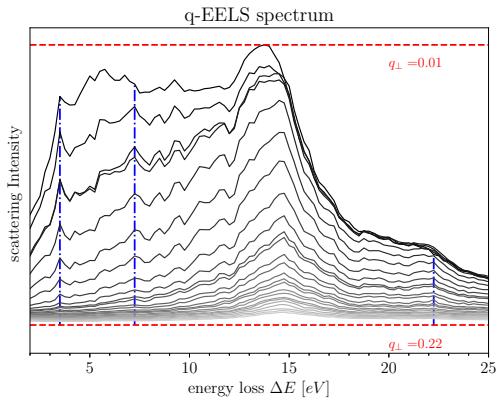


Figure 18: various tracked peaks not showing any dispersion.

To see whether the bulk plasmon dispersion is the same for multiple directions the line integration method was used to gather the q-EELS data. The path is taken from the centre diffraction spot  $\Gamma$  to the  $M$ -point and finally to  $\Gamma'$  which is the diffraction spot noted in the legend. The q-EELS data was not Batson corrected. As can be seen in figure 17 the dispersion of all the peaks seem to be somewhat similar at the start but diverges about halfway between the  $M$ - and  $\Gamma'$ -point, where the green and blue traces are highest and lowest respectively with red in between possibly hinting at the fact that the plasmon dispersion is anisotropic at high momenta transfers. Anisotropic electrical properties have been noted before in the thin-layer 2D material InSe. [27–30]

The bulk plasmon peak is not the only peak that can be tracked. However from all the peaks that were clear enough to track it was the only one that

showed any dispersion. As can be seen in figure 18 there are three peaks that can be seen in the low momentum transfer region. These peaks at 3.5 eV, 7.25 eV and 22 eV energy-loss can be seen at the same position across momentum transfers meaning that they do not disperse. Peaks at these energy-loss values have also been seen not dispersing in earlier literature [21] where indium selenide was studied using angle-resolved photoemission spectroscopy.

## 5 Conclusion

This work set out to introduce the reader to existing methods implemented in Python and validate the results with other earlier works.

From the data extraction techniques that were applied to the EFTEM stack the least useful one appeared to be the radial integration method since it was unable to isolate data in a single crystallographic direction. The other two extraction techniques are both useful for different purposes, the line integration technique was able to pool EFTEM columns of similar momentum transfer well and thus improved the momentum resolution of the q-EELS map whilst allowing the introduction of some data not truly in line with the crystallographic direction to be analysed. The stack slicing method delivered the ‘purest’ data but did so at the cost of momentum transfer resolution, the lack of pooling similar EELS spectra also reduced the signal with respect to the inherent background noise.

The zero-loss peak subtraction applied by the means of a Batson correction was useful for increasing the contrast of the data at the high momentum transfer region and was able to remove the zero-loss peak. Keeping in mind that the Batson correction needs an image spectrum to reach its desired accuracy, which was not available, the result were sufficient for tracking the relatively obscure plasmon peak in the high momentum transfer regime.

The analysed EFTEM stack showed similar features to those found in earlier works looking into the band structure of Indium Selenide, the low momentum transfer q-EELS spectra showed dispersionless peaks at energy-loss values similar to those found before. The bulk plasmon peak starting at roughly 14 eV energy loss was successfully tracked between diffraction spots and showed some anisotropic behaviours. Overall the results agree with earlier implementations of the methods applied, the results would benefit from a higher energy resolution to give the methods more information to work with. The Python methods developed for this work will continued to be worked on.

## References

- [1] E. G. van Putten, D. Akbulut, J. Bertolotti, W. L. Vos, A. Lagendijk, and A. P. Mosk, “Scattering lens resolves sub-100 nm structures with visible light,” *Phys. Rev. Lett.*, vol. 106, p. 193905, May 2011.
- [2] C. Kisielowski, B. Freitag, M. Bischoff, H. van Lin, S. Lazar, G. Knippels, P. Tiemeijer, M. van der Stam, S. von Harrach, M. Stekelenburg, and et al., “Detection of single atoms and buried defects in three dimensions by aberration-corrected electron microscope with 0.5-Å information limit,” *Microscopy and Microanalysis*, vol. 14, no. 5, p. 469477, 2008.
- [3] P. E. Batson, N. Dellby, and O. L. Krivanek, “Sub-ångstrom resolution using aberration corrected electron optics,” *Nature*, vol. 418, pp. 617–620, Aug. 2002.
- [4] O. Scherzer, “The theoretical resolution limit of the electron microscope,” *Journal of Applied Physics*, vol. 20, pp. 20–29, Jan. 1949.
- [5] R. F. Egerton, “Electron energy-loss spectroscopy in the TEM,” *Reports on Progress in Physics*, vol. 72, p. 016502, dec 2008.
- [6] S. H. Simon, *The Oxford solid state basics*. Oxford, UK: Oxford Univ. Press, 2013.
- [7] S. J. Magorrian, “Introduction,” in *Theory of Electronic and Optical Properties of Atomically Thin Films of Indium Selenide*, pp. 1–11, Springer International Publishing, 2019.
- [8] D. Ma, W. Ju, Y. Tang, and Y. Chen, “First-principles study of the small molecule adsorption on the inSe monolayer,” *Applied Surface Science*, vol. 426, pp. 244–252, 2017.
- [9] R. Egerton, *Electron Energy-Loss Spectroscopy in the Electron Microscope*. Springer US, 2011.
- [10] L. Bendersky and F. Gayle, “Electron diffraction using transmission electron microscopy,” *Journal of Research of the National Institute of Standards and Technology*, vol. 106, p. 997, Nov. 2001.
- [11] K. Kelton, “Quasicrystals,” in *Encyclopedia of Materials: Science and Technology* (K. J. Buschow, R. W. Cahn, M. C. Flemings, B. Ilschner, E. J. Kramer, S. Mahajan, and P. Veyssiére, eds.), pp. 7947–7952, Oxford: Elsevier, 2001.
- [12] F. Kahl, V. Gerheim, M. Linck, H. Müller, R. Schillinger, and S. Uhlemann, “Chapter three - test and characterization of a new post-column imaging energy filter,” in *Advances in Imaging and Electron Physics Including Proceedings CPO-10* (P. W. Hawkes and M. Häßch, eds.), vol. 212 of *Advances in Imaging and Electron Physics*, pp. 35–70, Elsevier, 2019.
- [13] W. Probst, G. Benner, J. Bühr, and E. Weimer, “An omega energy filtering tem principles and applications,” *Advanced Materials*, vol. 5, no. 4, pp. 297–300, 1993.
- [14] K. Tsuno, “Optical design of electron microscope lenses and energy filters,” *Microscopy*, vol. 48, no. 6, pp. 801–820, 1999.
- [15] L. I. Roest, S. E. van Heijst, L. Maduro, J. Rojo, and S. Conesa-Boj, “Charting the low-loss region in electron energy loss spectroscopy with machine learning,” *Ultramicroscopy*, vol. 222, p. 113202, 2021.
- [16] E. Gaufrès, F. Fossard, V. Gosselin, L. Sponza, F. Ducastelle, Z. Li, S. G. Louie, R. Martel, M. Côté, and A. Loiseau, “Momentum-resolved dielectric response of free-standing mono-, bi-, and trilayer black phosphorus,” *Nano Letters*, vol. 19, no. 11, pp. 8303–8310, 2019. PMID: 31603690.
- [17] P. Ercius, P. Gonzalo, T. Schoonjans, fniekiel, tcpekin, and koschie, “ncempy.” The last three authors are github usernames.
- [18] M. Sjödahl and L. R. Benckert, “Electronic speckle photography: analysis of an algorithm giving the displacement with subpixel accuracy,” *Appl. Opt.*, vol. 32, pp. 2278–2284, May 1993.
- [19] B. Schaffer, W. Grogger, and G. Kothleitner, “Automated spatial drift correction for efem image series,” *Ultramicroscopy*, vol. 102, no. 1, pp. 27–36, 2004.

- [20] P. E. Batson and J. Silcox, “Experimental energy-loss function,  $\text{Im}[-\frac{1}{\epsilon}(q, \omega)]$ , for aluminum,” *Phys. Rev. B*, vol. 27, pp. 5224–5239, May 1983.
- [21] A. Politano, D. Campi, M. Cattelan, I. B. Amara, S. Jaziri, A. Mazzotti, A. Barinov, B. Gürbulak, S. Duman, S. Agnoli, L. S. Caputi, G. Granozzi, and A. Cupolillo, “Indium selenide: an insight into electronic band structure and surface excitations,” *Scientific Reports*, vol. 7, June 2017.
- [22] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbas, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, Sept. 2020.
- [23] M. Lutz, *Learning Python*. Sebastopol, CA: O'Reilly, 2013.
- [24] W. McKinney, *Python for data analysis : data wrangling with pandas, NumPy, and IPython*. Sebastopol, CA: O'Reilly Media, Inc, 2018.
- [25] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
- [26] S. Schneider, “Angular resolved low loss eels for materials characterization,” 2013.
- [27] M. Brotons-Gisbert, R. Proux, R. Picard, D. Andres-Penares, A. Branny, A. Molina-Sánchez, J. F. Sánchez-Royo, and B. D. Gerardot, “Out-of-plane orientation of luminescent excitons in two-dimensional indium selenide,” *Nature Communications*, vol. 10, Sept. 2019.
- [28] G. W. Mudd, S. A. Svatek, T. Ren, A. Patanè, O. Makarovskiy, L. Eaves, P. H. Beton, Z. D. Kovalyuk, G. V. Lashkarev, Z. R. Kudrynskyi, and A. I. Dmitriev, “Tuning the bandgap of exfoliated InSe nanosheets by quantum confinement,” *Advanced Materials*, vol. 25, pp. 5714–5718, Aug. 2013.
- [29] A. I. Dmitriev, V. V. Vishnjak, G. V. Lashkarev, V. L. Karbovskyi, Z. D. Kovaljuk, and A. P. Bahtinov, “Investigation of the morphology of the van der waals surface of the InSe single crystal,” *Physics of the Solid State*, vol. 53, pp. 622–633, Mar. 2011.
- [30] J. Camassel, P. Merle, H. Mathieu, and A. Chevy, “Excitonic absorption edge of indium selenide,” *Phys. Rev. B*, vol. 17, pp. 4718–4725, Jun 1978.

## Appendix

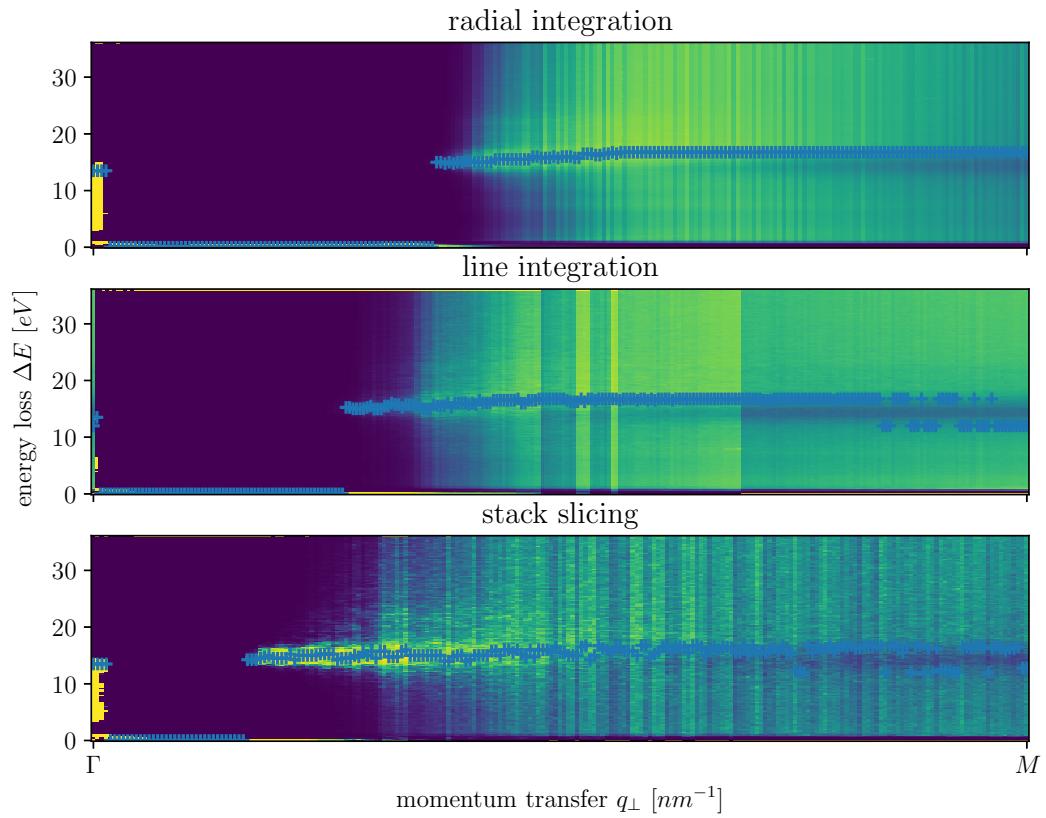


Figure 19: Batson correction applied to q-EELS maps constructed by different methods. The tracked plasmon peaks are denoted by the blue plusses.

# usage\_example

July 1, 2021

## 1 mreels.py usage and examples

```
[1]: #import the module the standard python way
import mreels

#importing other packages for examples
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
```

### 1.1 Dependencies

The package needs a few other python modules to be available in the current environment, namely:

- Numpy, for array calculations
- Scipy, for Fourier transforms and 2-dimensional convolution
- Matplotlib, for plotting
- Ncempy, for importing Gatan's proprietary .dm4 files
- Tqdm, for progressbars so you know that the code is just slow, not broken ;).

### 1.2 Starting out

#### 1.2.1 The data object

To start using the the mreels package you first need to create a data python object from the .dm4 file. The data object will be used by most functions since it holds all data and necessary metadata of the EFTEM stack.

To create a data object you need the .dm4 file in the same directory as your python file. Then you can call:

```
data = mreels.MomentumResolvedDataStack('string', preferred_frame)
```

with:  
- string : str , name of the dm4 file  
- preferred\_frame : int , index of the energy slice which will be used as reference point for other functions

preferred does not need to be set right away, but is used by build\_axes so setting pref\_frame to a energy slice with clear diffraction spots and bright centre gives better results.

to change to preferred frame you can call:

```
data.pref_frame = 25 #example
```

After the data-object is initialised a few functions can be called which change or append data to the data-object:

- `data.remove_neg_val()`, to raise all intensities by the minimum amount to make all intensities positive
- `data.rem_neg_el()`, removes the negative energy-loss slices from the EFTEM stack
- `data.build_axes()`, builds all the axis corresponding to the shape of `data.stack.shape`

```
[2]: data = mreels.MomentumResolvedDataStack('inse_example_stack.dmf4', pref_frame=10)

shape = data.stack.shape
print(shape)

data.rem_neg_el() #removes slices corresponding to negative energy-losses

shape = data.stack.shape
print(shape)

data.build_axes()
print("Energy losses are: \n", data.axis0)

(40, 2048, 2048)
(37, 2048, 2048)
Energy losses are:
[ 0.  1.  2.  3.  4.  5.  6.  7.  8.  9. 10. 11. 12. 13. 14. 15. 16. 17.
 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35.
 36.]
```

## 1.2.2 Plotting energy-slices

The EFTEM stack can be called from the data object by calling python `data.stack`, this returns a numpy.ndarray object with 3-axis. Axis-0 is the energy-loss and axis-1 and -2 are the first and second reciprocal space axis.

If the example above was executed you can see that there were 40 energy-loss slices and 2048 momentum coordinates for the first and second reciprocal space axis. The python `data.rem_neg_el()` removed 3 energy-slices that corresponded to negative energy losses.

Plotting can be done using the matplotlib plotting library.

```
[3]: ax1 = data.axis1
ax2 = data.axis2
stack = data.stack

fig, ax = plt.subplots(2, 2)

ax[0,0].imshow(stack[10,:,:].T, origin='lower')

ax[0,1].pcolorfast(ax1, ax2, stack[10,:,:].T)

sig = mreels.sigmoid
```

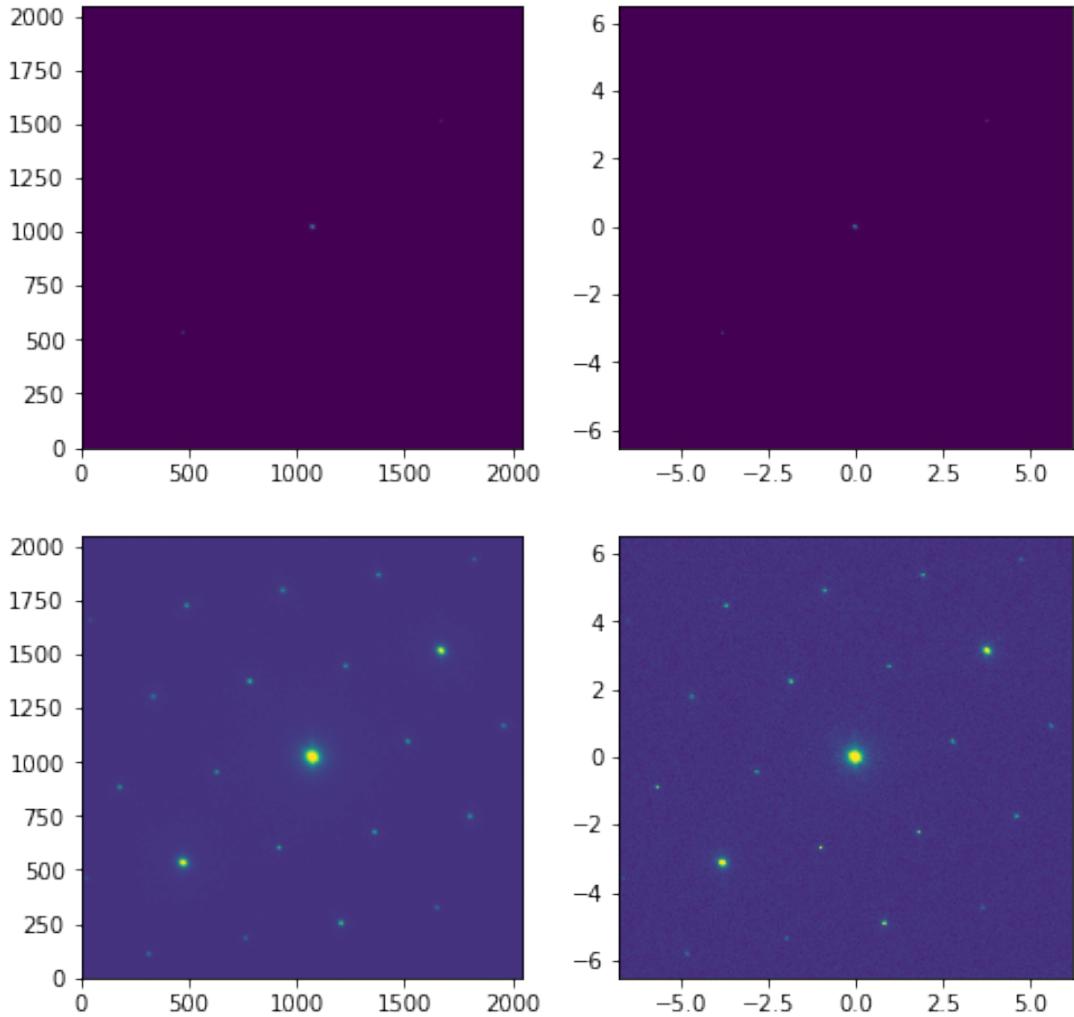
```

ax[1,0].imshow(sig(stack[10,:,:].T), origin='lower')

ax[1,1].pcolorfast(ax1, ax2, sig(stack[10,:,:].T))

plt.gcf().set_size_inches(8,8)
plt.show()

```



### 1.2.3 Finding features

After plotting a ‘normalised’ stack slice several diffraction peaks are visible. To find the centres of these diffraction peaks in their index coordinates a list with their false coordinates needs to be build by guessing the locations.

Coordinates are indexed as the stack was supplied, thus in (energy loss, ax1, ax2) format. For guessing peaks this means denoting the ‘x-axis’ value first.

A false centres list would be:

`[(500, 500), (900, 600)]` for the lower left bright peak and the dimmer peak to the right respectively.

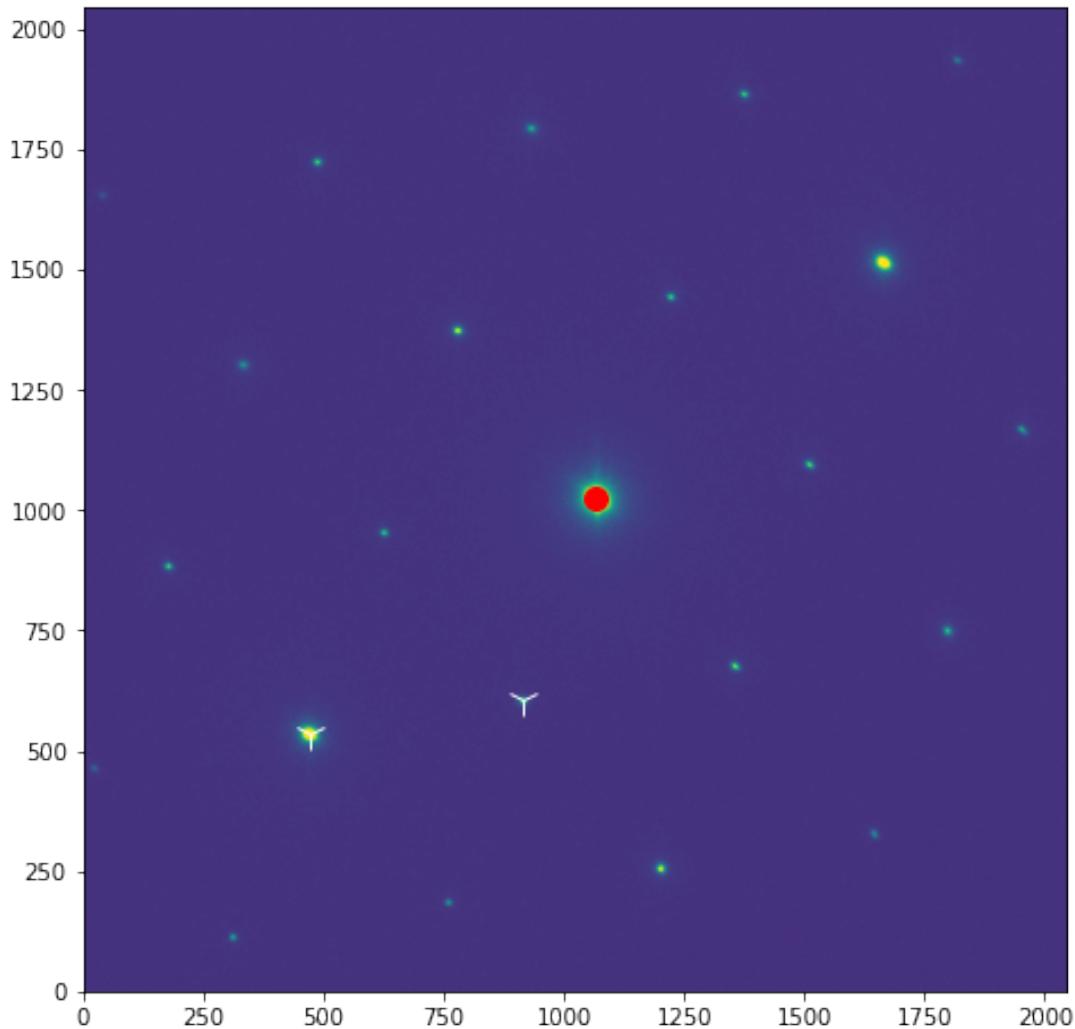
Finding the actual centres would then be as easy as specifying a energy-loss slice in which to find peaks, passing the list and specifying `leeway=` which is the search block size.

The centre of the diffraction image is assumed to be the brightest pixel in a slice, its coordinates can be requested by calling: `data.get_centre()`

```
[4]: fls_centres = [(500, 500), (900, 600)]  
  
tr_centres = mreels.get_true_centres(stack[10], fls_centres, leeway=100)  
  
print("True centres: \n", tr_centres)  
  
centre = data.get_centre()  
  
print("Stack centre: \n", centre)
```

True centres:  
[471 535 914 604]  
Stack centre:  
(1068, 1025)

```
[5]: fig, ax = plt.subplots(1,1)  
  
ax.imshow(sig(stack[10].T), origin='lower')  
ax.plot( 471, 535, marker='1', color='white', markersize=14)  
ax.plot( 914, 604, marker='1', color='white', markersize=14)  
ax.plot( 1068, 1025, marker='o', color='red', markersize=10)  
  
plt.gcf().set_size_inches(8,8)  
plt.show()
```



#### 1.2.4 Aligning the stack

The EFTEM stack in the data object can be aligned by calling `data.correct_drift()`. This function corrects for 2D-drift of the diffraction pattern, thus not for: warp, rotation, skewing etc.

It does this by means of Fourier phase correlation on one enhanced slice at a time, looping over the stack and using `data.pref_frame` as a reference slice.

```
[6]: data.correct_drift()
```

```
Aligning EFTEM stack: 100% | 37/37 [00:17<00:00, 2.11it/s]
```

#### 1.2.5 Extracting information

Information that can currently be extracted from the data object is limited to:

- q-EELS spectra, not automated but easily done by slicing.
- q-EELS maps using one of three functions.

- Radial averaging This method allows for the averaging of q-EELS spectra of a certain momentum transfer radially outwards from the centre.

Usage:

```
qmap, qaxis = mreels.get_qeels_data(data, r1, ringsize, pref_frame, forward_peak,
                                      method='radial', threads, starting_point, peak_width
                                      )
```

with:

- `data`, the data object
- `r1`, the staring radius in pixel lengths such that if the separation from the centre in x and y = 10 that  $r = \sqrt{10^2 + 10^2}$
- `pref_frame`, frame that will be used to find the peaks and centre
- `forward_peak`, not used if `method='radial'`
- `method='radial'`, to use radial averaging
- `threads`, number of cpu threads to use. function distributes energy-loss slices to threads so it should not be RAM limited.
- `starting_point`, will be used as centre. Can be left `None` to use stack centre
- `peak_width`, unused in `method='radial'`
- Line averaging This method allows for the averaging of q-EELS spectra of certain momentum transfer angle limited radially outwards from the centre in the direction of a specified point.

Usage:

```
qmap, qaxis = mreels.get_qeels_data(data, r1, ringsize, pref_frame, forward_peak,
                                      method='line', threads, starting_point, peak_width
                                      )
```

with:

- `data`, the data object
- `r1`, the staring radius in pixel lengths such that if the separation from the centre in x and y = 10 that  $r = \sqrt{10^2 + 10^2}$
- `pref_frame`, frame that will be used to find the peaks and centre
- `forward_peak`, peak to which will be integrated
- `method='line'`, to use line averaging
- `threads`, number of cpu threads to use. function distributes energy-loss slices to threads so it should not be RAM limited.
- `starting_point`, will be used as centre. Can be left `None` to use stack centre
- `peak_width`, width of the `forward_peak` in pixels. Needs to be set only if `forward_peak` is not a bright well-defined peak. Otherwise it will be determined as the amount of pixels from the `peak_centre` by which 1/3rd of the peak intensity is reached

- Line slicing This methods allows the user to extract all q-EELS spectra and corresponding momenta transfers in a line from point to point.

Usage:

```
qmap, qaxis = mreels.get_qeels_slice(data, point, use_k_axis, starting_point)
```

with:

- `data`, the data object
- `point`, the forward point to slice towards.
- `use_k_axis`, ignore this for now
- `starting_point`, to point from which to slice

### 1.2.6 Plotting a q-EELS map

A q-EELS map can be simply plotted using the function:

```
mreels.plot_qeels_data(data, qmap, qaxis, prefix, save)
```

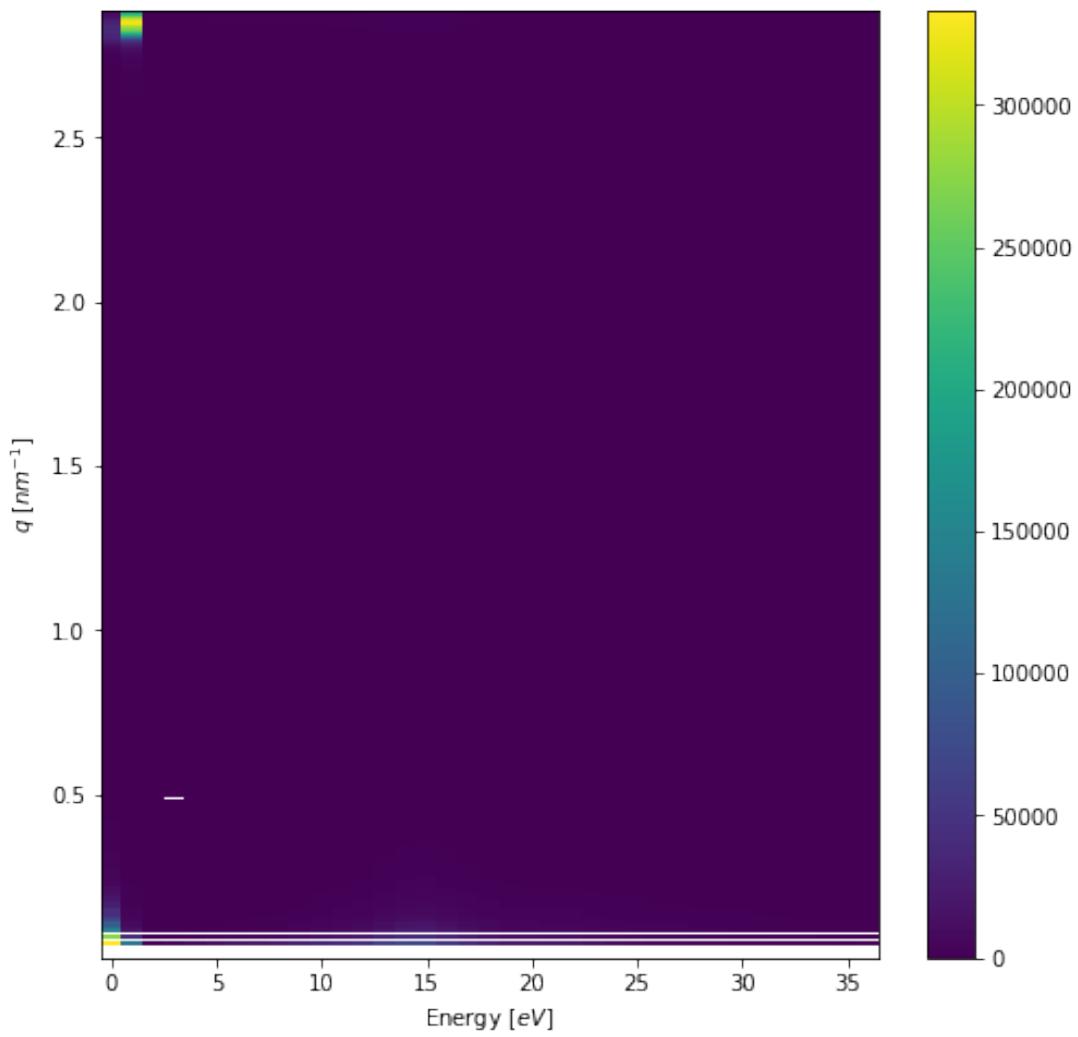
with: - `data`, data object - `qmap`, a qmap supplied by one of the previous functions - `qaxis`, qaxis supplied by previous functions corresponding to qmap - `prefix`, string to prefix to saved file if `save=True` also functions as plot title - `save`, boolean to toggle saving

```
[7]: fwd_peak0 = (914, 604)
fwd_peak1 = (471, 535)
```

```
qmapline, qaxisline = mreels.get_qeels_data(data, 100, 1, 10, fwd_peak0, method='line', threads=12)
```

```
Momentum axis: 100% | 454/454 [00:07<00:00, 64.73it/s]
 0% | 0/37 [00:00<?, ?it/s] c:\Users\Jeroe\Documents\Python\CBL-ML\MR-
EELS\mreels.py:327: RuntimeWarning: invalid value encountered in double_scalars
    integral = np.sum(integration_area)/np.sum(entries)
c:\Users\Jeroe\Documents\Python\CBL-ML\MR-EELS\mreels.py:327: RuntimeWarning:
divide by zero encountered in double_scalars
    integral = np.sum(integration_area)/np.sum(entries)
100% | 37/37 [00:56<00:00, 1.52s/it]
```

```
[8]: mreels.plot_qeels_data(data, qmapline, qaxisline, '')
```

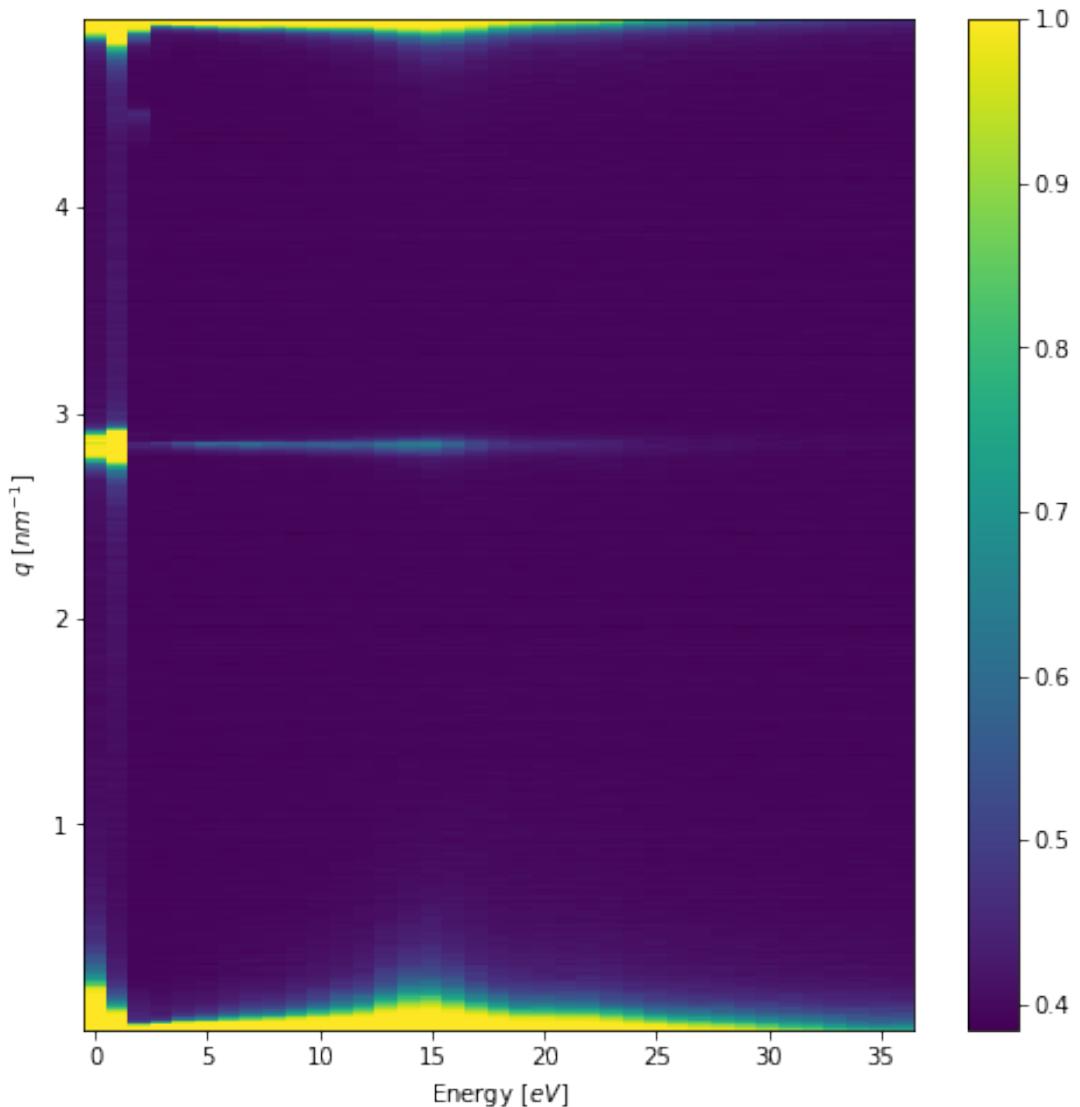


[9]: # *q-EELS slice maps can be easily chained*

```
qmapslice0, qaxisslice0 = mreels.get_qeels_slice(data, fwd_peak0)
qmapslice1, qaxisslice1 = mreels.get_qeels_slice(data, fwd_peak1, ↴
    starting_point=fwd_peak0)

qmapslice = np.append(qmapslice0, qmapslice1[1:], axis=0)
qaxisslice = np.append(qaxisslice0, qaxisslice1[1:])
```

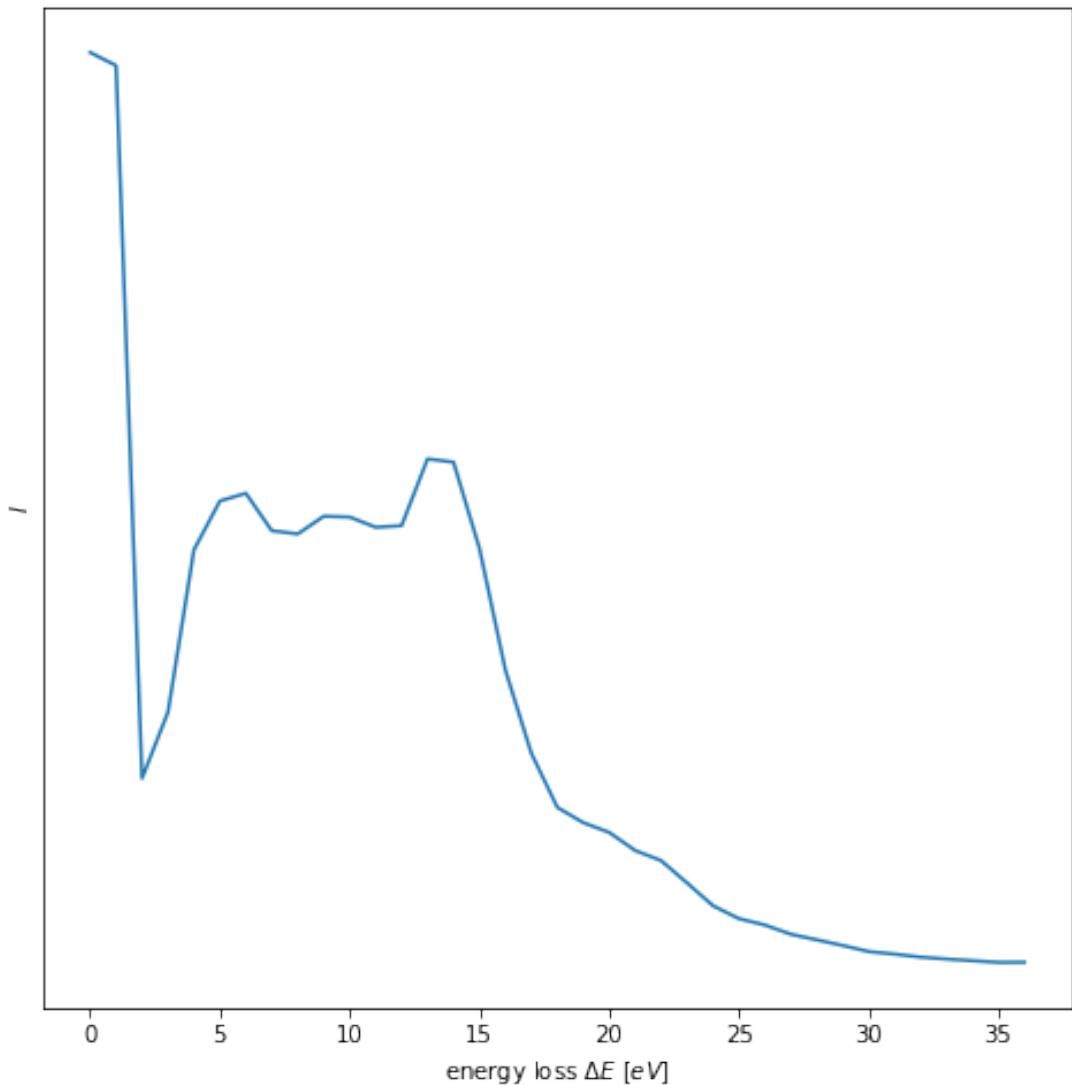
[10]: mreels.plot\_qeels\_data(data, sig(qmapslice, (20,50)), qaxisslice, '')



```
[11]: eax = data.axis0

fig, ax = plt.subplots(1,1)
ax.plot(eax, stack[:,centre[0], centre[1]])
ax.set_xlabel(r"energy loss $\Delta E$ [eV]")
ax.set_ylabel(r"$I$")
ax.set_yticks([])

plt.gcf().set_size_inches(8,8)
plt.show()
```



### 1.2.7 Batson correction

Batson correction uses an image spectrum of the same sample to remove the zero-loss peak and to try and remove plural scattering from the individual q-EELS spectra in the q-EELS map.

Usage:

```
batmap = mreels.batson_correct(data, energy_window, qmap, imspec)
```

with: - **data**, the data object - **energy\_window**, the size in *eV* of the zero-loss peak - **qmap**, the to be corrected q-EELS map - **imspec**, the image spectrum

[12] :

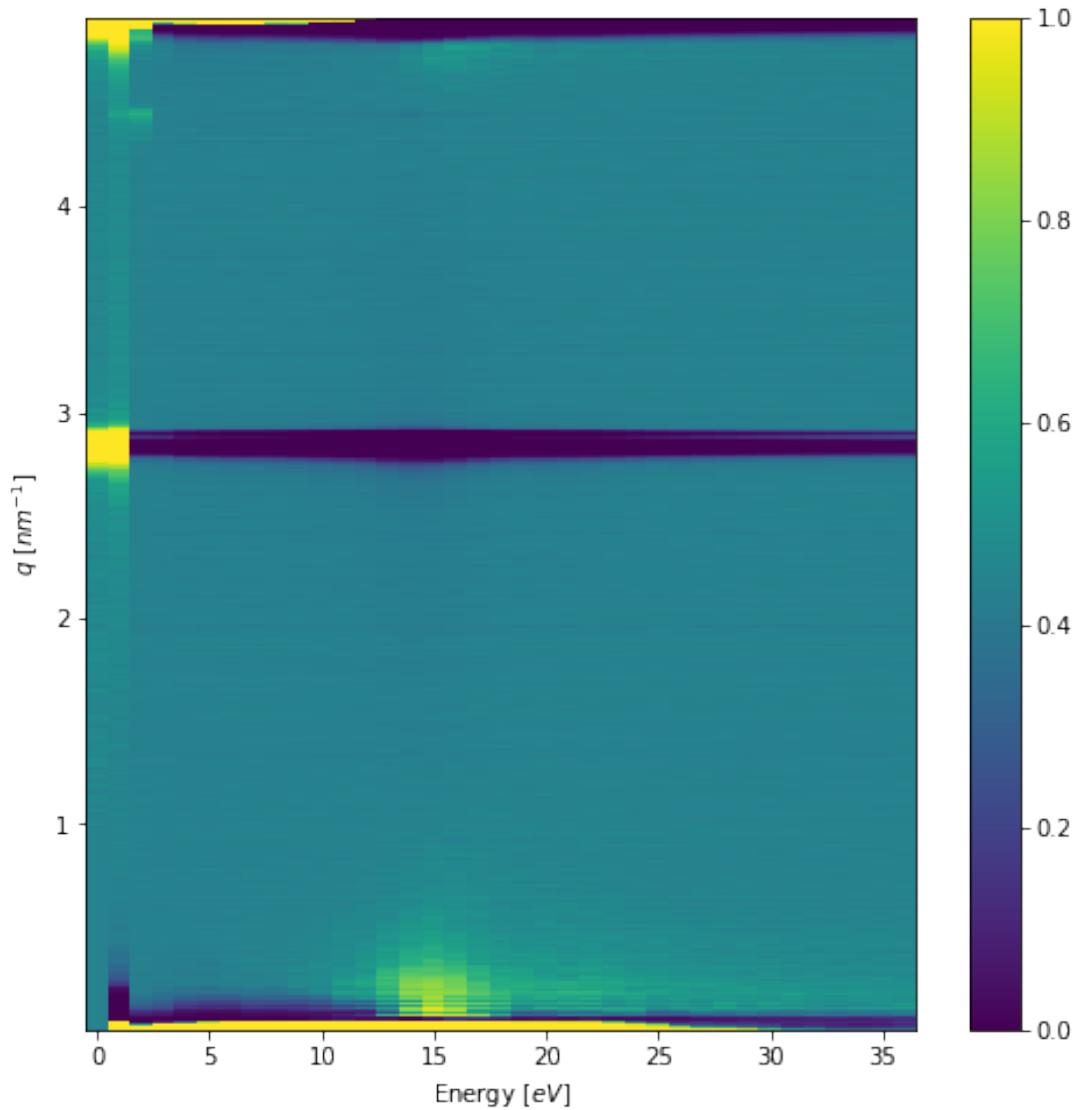
```

imspec = np.sum(stack[:,centre[0]-100:centre[0]+100,centre[1]-100:
                     ↵centre[1]+100], axis=(2,1)) #simulating an image spectrum, use an actual ↵
                     ↵image spectrum if you have one

batmap = mreels.batson_correct(data, 3, qmapslice, imspec)

mreels.plot_qeels_data(data, sig(batmap, (20,50)), qaxisslice, 'r')

```



### 1.2.8 Peak tracking

Peaks in the spectra can be tracked accross momentum space using the following functions:

```
    pqmap, pqaxis = mreels.pool_qmap( qmap, qaxis, poolsize)
```

with: - qmap, a q-EELS map - qaxis, a q-EELS momentum axis - poolsize, the size of pooling poolsize/2 spectra before and after are pooled together.

```
    ppos, perr = mreels.find_peak_in_range(qmap, centre, windows_size, adaptive_range)
```

with; - qmap, a (pooled) q-EELS map - centre, index around which to find the Peak - window\_size, size of window in which to search for peak centred around centre - adaptive\_range, boolean toggle for adaptive range. If True the centre for next search in a q-EELS spectra is the peak location in this spectra

Note: perr is currently poorly defined

```
[13]: pqmap, pqaxis = mreels.pool_qmap(qmapslice, qaxisslice, 4)
```

```
    ppos, perr = mreels.find_peak_in_range(pqmap, np.argwhere(eax==14)[0][0], 20)
    pposadapt, perradapt = mreels.find_peak_in_range(pqmap, np.
    ↪argwhere(eax==14)[0][0], 6, adaptive_range=True)
```

```
    fig, ax = plt.subplots(1,1)
    ax.plot(pqaxis, eax[ppos], label=r'pooled, adapt=False', linestyle='-.', ↪
    ↪color='red')
    ax.plot(pqaxis, eax[pposadapt], label=r'pooled, adapt=True', linestyle='--', ↪
    ↪color='blue')
    plt.legend()

    plt.ylim([12,28])

    plt.gcf().set_size_inches(8,8)
    plt.show()
```

```
c:\Users\Jeroe\Documents\Python\CBL-ML\MR-EELS\mreels.py:815: RuntimeWarning:
divide by zero encountered in double_scalars
    perr = np.append(perr, 1/(tmp[0][0]-np.mean(search_slice)))
```

