universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Real-time drift compensation for focused electron irradiation

verfasst von | submitted by

Eugen Kozyrau BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 876
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Physics
Betreut von | Supervisor:	Assoz. Prof. Dr. Toma Susi

Abstract

Manipulation of single atoms in a scanning transmission electron microscope requires a stable stage to accurately target irradiation at a chosen atom. This is done by parking the electron beam at a desired position.

However, despite the industry-leading stability of the Nion microscope, drift of the stage is difficult to eliminate completely. With ideal values of 0.5 Å per minute, the sample constantly moves relative to the electron beam. Furthermore, while the atom is irradiated, we currently cannot measure the drift. To be better able to automate STEM manipulation of impurity atoms in 2D materials, this drift must be monitored and corrected in real time.

One possible way of achieving this is by monitoring the convergent beam electron diffraction pattern of the stationary beam during manipulation. In a crystal, the diffraction pattern is expected to be symmetrical if the beam is centered on a lattice atom. If the stage holding the sample moves relative to the beam, the diffraction pattern will change accordingly. Detecting this change may allow for a viable way of drift monitoring, even when the probe is stationary. In practice, this may require the use of a fast direct-electron detector, which are increasingly available for use in so-called 4D-STEM.

The aim of this thesis is to analyze the effects of stage drift on the convergent beam electron diffraction pattern, and to find ways to incorporate this information into software used for the automation of atom manipulation. We perform experiments using an sCCD camera and the newly installed state-of-the-art Dectris ARINA direct-electron detector in the Vienna Nion UltraSTEM100. The experimental work is supported by quantitative simulations using the in-house *ab*TEM code.

Zusammenfassung

Manipulation einzelner Atome in einem Rastertransmissionselektronenmikroskop erfordert einen stabilen Objekthalter, um ein ausgewähltes Atom präzise bestrahlen zu können. Dies erfolgt durch "Parken" des Elektronenstrahls an der gewünschten Position.

Doch trotz der branchenführenden Stabilität des Nion Mikroskops, ist es schwierig den Drift des Mikroskoptisches komplett zu eliminieren. Mit idealen Werten von 0.5 Å pro Minute bewegt sich die Probe ständig relativ zum Elektronenstrahl. Noch dazu kann der Drift zurzeit nicht während der Bestrahlung gemessen werden. Um die Manipulation von Fremdatomen in 2D-Materialien besser automatisieren zu können, muss dieser Drift in Echtzeit überwacht und korrigiert werden.

Eine Möglichkeit dies zu bewerkstelligen, besteht darin das konvergente Elektronenbeugungsbild (CBED) des stationären Strahls während der Manipulation zu beobachten. Es wird damit gerechnet, dass das Beugungsbild eines Kristalls symmetrisch ist, wenn der Strahl über einem Atom des Gitters positioniert ist. Falls sich der Objekthalter mit der Probe relativ zum Strahl bewegt, wird sich das Beugungsbild entsprechend ändern. Das Erkennen dieser Änderung könnte eine praktikable Möglichkeit zur Driftüberwachung bieten, sogar mit stationärem Strahl. In der Praxis könnte dies einen schnellen, direkten Elektronendetektor erfordern. Diese sind zunehmend zur Verwendung in der sogenannten 4D-STEM verfügbar.

Das Ziel dieser Diplomarbeit ist es, die Auswirkungen des Objekthalter-Drifts auf das konvergente Beugungsbild zu analysieren und Wege zu finden, diese Information in Atommanipulationssoftware einzubauen. Wir führen Experimente mit einer sCCD Kamera und dem neulich installierten hochmodernen Dectris ARINA Direct-Electron Detektor am Wiener Nion UltraSTEM 100 durch. Die Versuche werden durch quantitative Simulationen mittels hauseigenem *ab*TEM code unterstützt.

Acknowledgements

Working on this thesis was the single greatest challenge, and simultaneously, the single greatest opportunity for growth, academic and personal, in my life so far. I met many new people who taught me, guided me and accompanied me during this time.

I would first like to thank my supervisor, Prof. Dr. Toma Susi, for this opportunity, for conducting the experimental work together with me, and for his guidance throughout all stages of this project. Next, I would like to thank Manpreet Kaur for providing the core idea for this thesis. I am also grateful to Dr. Manuel Längle and Prof. Dr. Jani Kotakoski for their advice on organising and writing this thesis. I would especially like to thank the colleagues with whom I shared many conversations and an office space: Nandhini Ravindran, Diego Bautista, Florian Wolf, Erik Giménez Andreu, Fabian Kraft and Alissa Freilinger, as well as all the other members of the PNM research group that I had the pleasure of meeting and talking to, such as Dr. Clemens Mangler, Dr. Jana Dzíbelová, Dr. Jacob Madsen, Dr. Vladimír Zobač, Shrirang Chokappa, Somar Dibeh, Philipp Irschik, Umair Javed, Wael Joudi, Clara Kofler, David Lamprecht, Diana Propst, Thuy An Tripathi-Bui, Barbara Maria Mayer, Morris Weimerskirch, Helena Kempf and Kathrin Linke.

Contents

List of Acronyms

2D two-dimensional

3D three-dimensional

4D four-dimensional

ADF annular dark field

BF bright-field

CBED convergent beam electron diffraction

CCD charge coupled device

CFEG cold field electron gun

CMOS complementary metal–oxide–semiconductor

CoM center of mass

CVD chemical vapor deposition

DQE detective quantum efficiency

EELS electron energy loss spectroscopy

FWHM full width at half maximum

GUI graphical user interface

HAADF high angle annular dark field

IAM independent atom model

iCoM integrated center of mass

MAADF medium angle annular dark field

PAD pixel array detector

STEM scanning transmission electron microscope/microscopy

STM scanning tunneling microscope/microscopy

TEM transmission electron microscope/microscopy

UHV ultra-high vacuum

UI user interface

VOA virtual objective aperture

Chapter 1

Introduction

In the late 1980s, researchers first managed to move individual atoms using scanning tunneling microscopes (STM) [1]. These were the first successful attempts at controlled atom manipulation. The atomically sharp tip of an STM can be used to pick up an atom from the surface of a substrate, carry it over a set distance and place it at a new position. This feat often required the use of cryogenic temperatures and is only possible for weakly bound atoms.

Thanks to recent improvements to resolution in a completely different microscopy technique called scanning transmission electron microscopy (STEM), such as the introduction of aberration-correctors [2], it is now possible to target individual atoms with a beam of high-energy electrons. Their kinetic energy of tens, sometimes hundreds of keV is capable of breaking covalent bonds.

In 2014, during atomic-resolution STEM experiments, graphene dopants, specifically Si, were found to occasionally jump from lattice site to lattice site under 60 keV electron irradiation [3]. Subsequent papers further explored the ability of controlling where these jumps or bond reversals occurred, by irradiating specific atoms using the microscope's beam, with promising results. Research on this phenomenon has also extended to other dopants [4, 5], with silicon remaining the most resilient to damage.

Due to the stochastic nature of electron scattering, activation of this process requires a significant amount of time spent irradiating an atom. More so for lower electron energies, where the damage to the material is reduced.

A well-known problem in STEM is the presence of stage drift and drift in general. In the context of atom manipulation, drift may hinder efficient irradiation of a targeted atom, making both manual and automated experiments difficult. Currently, drift correction methods entail analyzing and aligning STEM images, which is not a real-time method that can be used during in-situ experiments with a stationary beam.

This work aims to implement real-time drift compensation using an approach similar to gradient descent. Instead of derived gradients, however, normalised center of mass (CoM) vectors calculated from convergent beam electron diffraction (CBED) patterns are used to determine the step size and direction of the descent. The minima targeted by this method represent minima of the projected electric field inside a sample, specifically minima at the positions of atoms as opposed to vacancies. By tracking such atomic sites with the microscope's probe, drift can be compensated and efficiency of atom manipulation increased.

1.1 Electron scattering

The scattering of electrons by atoms is the core concept in electron microscopy. Scattering events can be categorized into elastic and inelastic. Electrons that lost no (or almost no) energy during interaction with an atom are considered elastically scattered, whereas electrons that lost kinetic energy count as inelastically scattered. While the latter can be used for spectroscopy methods such as electron energy loss spectroscopy (EELS), the former is commonly used for imaging.

Let us now consider an electron passing by a single, unbound atom. Upon entering the electron cloud of the atom, it will experience a Coulomb attraction towards the core. As illustrated by Williams and Carter in Figure (1.1), this attraction is greater the closer the impinging electron passes by the core.

Cross section

There is however no guarantee that an electron will be scattered at all. The probability of an electron interacting with an atom's nucleus or bound electrons elastically or inelastically is described using an interaction cross section $\sigma(\theta)$. This probability depends on the charge distribution of the scattering atoms as well as the electron's kinetic energy. At high energies

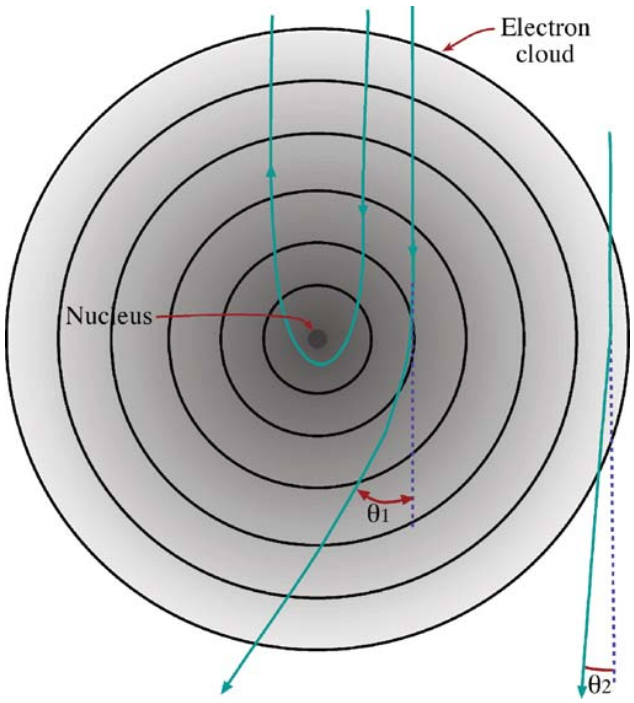

Figure 1.1: Simplified illustration of electrons elastically scattered by a single, isolated atom due to Coulombic attraction. From Williams & Carter [6], chapter 3.2.

in the range of keV, most electrons will only be scattered weakly or not scattered at all. The cross section of inelastically scattered electrons is even smaller than its elastic counterpart, at times by orders of magnitude.

To determine the cross section of an electron scattering to a specific range of angles $[\theta_1, \theta_2]$ in all directions φ, one must know the differential cross section:

$$\frac{d\sigma(\theta)}{d\Omega} = \frac{1}{2\pi \sin\theta} \frac{d\sigma(\theta)}{d\theta} = \frac{1}{\sin\theta} \frac{d\sigma(\theta)}{d\theta\, d\varphi}\,, \tag{1.1}$$

where $\frac{d\sigma}{d\Omega}$ describes the angular distribution of scattering from a single spherically symmetric atom, Ω a solid angle and θ the scattering semiangle. The solid angle Ω can represent the extent of a flat aperture or detector, orthogonal to an electron beam. The conversion term between the differential fractions stems from the geometrical relationship between Ω and θ, as well as an azimuthal symmetry in φ. Integrating over a detector's solid angle yields the following form of the scattering cross section for a detector:

$$\sigma_{\text{detector}} = \oint_{\text{detector}} \frac{d\sigma(\theta)}{d\Omega} d\Omega = 2\pi \int_{\theta_1}^{\theta_2} \frac{d\sigma(\theta)}{d\Omega} \sin\theta\, d\theta\,. \tag{1.2}$$

The differential cross section $\frac{d\sigma}{d\Omega}$ defines the atomic scattering factor $f(\theta)$ via the following relation:

$$|f(\theta)|^2 = \frac{d\sigma(\theta)}{d\Omega}, \tag{1.3}$$

where $f(\theta)$ is a measure of the amplitude of a scattered electron wave and $|f(\theta)|^2$ is proportional to the scattered intensity.

Atomic scattering factor

The atomic scattering factor or atomic form factor $f(\theta)$ is preferred over the more commonly known Rutherford cross section for low-angle scattering at $\theta \lesssim 3°$, as the latter neglects the wave nature of an electron beam. For context, common scanning transmission electron microscopy (STEM) setups use high angle annular dark field (HAADF) detectors to detect Rutherford scattered electrons, with solid angles between $\theta_1 \approx 90$ mrad, $\theta_2 \approx 300$ mrad, equating to scattering semiangles of roughly $\theta \approx 5$ to $17°$. Smaller angles are covered by bright-field (BF) and intermediate annular dark field (ADF) detectors.

A useful approximation for the atomic scattering factor $f(\theta)$ is the first Born approximation (1st BA), which assumes that for a scattering event, the incoming electron wave is a plane wave that is only scattered once and does not diminish in amplitude after scattering. It assumes furthermore that a detector would be far away from the scattering origin, allowing the usually spherical exit waves to also be treated as plane waves [7]. A scattered wave in the 1st BA thus looks as follows:

$$\begin{aligned} \psi(\vec{x}) &= e^{2\pi i k_z z} \quad \text{incident} \\ &\approx e^{2\pi i k_z z} + f_e(\Delta\vec{k})\frac{e^{2\pi i \Delta\vec{k}\cdot\vec{r}}}{|\vec{r}|} \quad \text{scattered}\,, \end{aligned} \tag{1.4}$$

where $f_e(\Delta\vec{k})$ is the atomic scattering factor for electron scattering as a function of the difference $\Delta\vec{k} = \frac{2\sin\theta}{\lambda}$ between incident and scattered wave vectors:

$$f_e(\Delta\vec{k}) = \frac{1}{2\pi e a_0}\int V_a(\vec{r}) e^{2\pi i \Delta\vec{k}\cdot\vec{r}} d^3r, \tag{1.5}$$

where e is the magnitude of the electron charge, a_0 is the Bohr radius and $V_a(\vec{r})$ is the total electrostatic potential. $f_e(\Delta\vec{k})$ is in units of Å and must be multiplied by the relativistic mass ratio m/m_0 for different incident electron energies. Equation 1.5 is in essence a Fourier transform of the electrostatic potential, which makes it convenient for computational methods. It is however not suited to calculate scattering in a microscope image directly [8–11].

$$\begin{array}{ccc} V_a(\vec{r}) & \xleftrightarrow{\text{Poisson}} & \rho(\vec{r}) \\ \text{1BA } \mathcal{F}^{-1} \uparrow & & \downarrow \mathcal{F} \\ f_e(\Delta\vec{k}) & \xleftarrow[\text{Mott-Bethe}]{} & f_x(\Delta\vec{k}) \end{array} \tag{1.6}$$

Because the total electrostatic potential is only analytically solvable for the hydrogen atom and otherwise time-consuming to calculate numerically, eq. 1.5 is commonly used to approximate the potential $V_a(\vec{r})$ or projected potential $V_z(\vec{r})$ from the scattering factor $f_e(\Delta\vec{k})$ in its inverse form. In turn, the Mott-Bethe formula calculates the 1st BA factor $f_e(\Delta\vec{k})$ by applying the Poisson equation to the x-ray scattering factor $f_x(\Delta\vec{k})$ in Fourier space. The latter is a straightforward Fourier transform of the electron charge distribution $\rho(\vec{r})$.

Plots of the atomic scattering factor generally show a maximum scattering magnitude at an angle of 0, decreasing smoothly and rapidly with increasing scattering angles.

The concept of an atomic scattering factor can be further expanded to a crystal lattice by introducing the structure factor $F(\theta)$. This new amplitude can be defined as the sum of all n atoms of a unit cell:

$$F(\theta) = \sum_i^n f_i(\theta)\ e^{2\pi i(hx_i+ky_i+lz_i)}\,. \tag{1.7}$$

Here i is the index of the i'th atom in a given unit cell, θ the scattering angle, x_i, y_i, z_i the atom's coordinates and h, k, l the Miller indices of a given crystal lattice. The exponential term is a phase factor, accounting for constructive and destructive interference.

$F(\theta)$ is a linear superposition of the atomic scattering factors $f_i(\theta)$ that assumes large distances between atoms compared to their size. This is of course inaccurate, and valence electrons will re-arrange upon forming bonds, so the atomic potentials will change as well. Investigations by McWeeny [12], Freeman [13] and Meyer et al. [14], Susi et al. [15] have shown an error of 5-10% in x-ray and electron scattering factor for low-Z atoms at small scattering angles. Nevertheless, this concept is well established in microscopy simulations as the independent atom model (IAM).

1.1.1 Deflections due to electric & magnetic fields

While typical ray diagrams of electron optics draw electron paths as straight lines diverted by lenses, in reality, magnetic fields also induce a rotation according to the Lorentz force $\vec{F}$:

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B}) \,, \tag{1.8}$$

with q being an electric charge, $\vec{E}$ and $\vec{B}$ electric and magnetic fields respectively and $\vec{v}$ the velocity of the charge. For an electron with the charge $q = -e$ and upon entering a uniform magnetic field at an angle θ close to 90° with respect to $\vec{B}$, the centripetal force is:

$$F_\perp = eB \cdot v \sin\theta = evB = \frac{m_e v^2}{r} \,, \tag{1.9}$$

where m_e is the electron mass and r the distance from the optical axis. Rewriting equation [1.9] gives us an expression for r:

$$r = \frac{m_e v}{eB} \,. \tag{1.10}$$

Rearranging it again yields the angular velocity ω_c , which is related to the cyclotron frequency f_c by a factor of 2π:

$$\omega_c = \frac{v}{r} = \frac{eB}{m_e} \,. \tag{1.11}$$

In short, the electrons follow a helical path along the lens axis, as shown in Figure (1.2). As a result, an image in electron microscopy may appear rotated with respect to the physical orientation of a sample, depending on the strength of the magnetic field B and the traveled distance z. Keep in mind that the above equations do not consider relativistic speeds.

A derivation by P.W. Hawkes [16] from Williams & Carter's book on transmission electron microscopy [6], chapter 6.3C,D, shows the change in rotation angle for a rotationally symmetrical field $\vec{B}$ near the optical axis:

$$\frac{d\theta}{dz} = \frac{\eta B}{2\sqrt{V}}, \quad \eta = \sqrt{\frac{e}{2m_0 c^2}} \,. \tag{1.12}$$

Integrating along the extent of the magnetic field yields an angle of rotation θ that is proportional the strength of the magnetic field B, the distance z traveled through the field and inversely proportional to the root of the acceleration voltage V.

A change of this rotation angle θ can easily be encountered when changing the defocus or camera length in a transmission electron microscope, as the image appears to rotate due to changes in B or z.

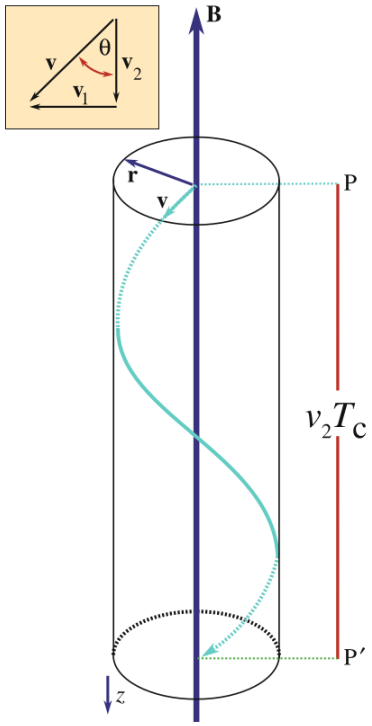

Figure 1.2: Electron trajectories in a homogeneous magnetic field, spiraling with a period of rotation T_c from the on-axis point **P** and intersecting the axis again at point **P'**. From Williams & Carter [6] chapter 6.3C,D.

1.2 Scanning transmission electron microscopy

In the first half of the 20th century, scientists realised the potential of using electrons for microscopy. Abbe's diffraction limit postulates that the smallest resolvable distance in an optical microscope is proportional to the wavelength of light. For visible light, that limit is on the order of 200 nm, far too large to image molecules, let alone atoms. Electrons, while being fermionic particles with mass, have also been shown to exhibit wave properties. When accelerated to a kinetic energy of 100 keV, their de Broglie wavelength would equate to just a few picometers. Thus, according to the Abbe limit, an electron microscope should be able to reach far higher resolutions than possible with light.

One of the first electron microscopes was built in 1931 by Knoll and Ruska [17]. It accelerated electrons through an anode aperture and focused them with electromagnetic lenses into a parallel beam onto a sample. The electrons transmitted through the sample were then imaged on a fluorescent screen.

Not long after this first wave of transmission electron microscopes, in 1938 the scanning transmission electron microscope (STEM) was invented by Von Ardenne [18]. It uses a con-

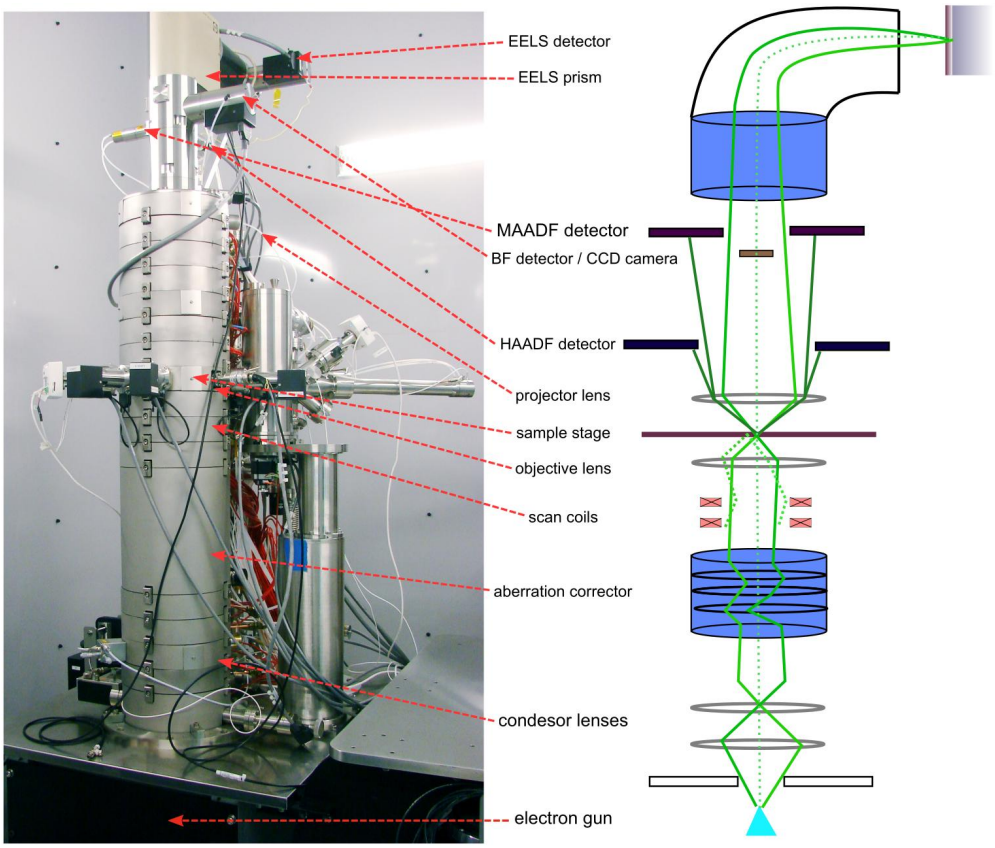

Figure 1.3: An early Vienna Nion UltraSTEM 100 column setup and a simplified schematic of its key components, along with the paths of electron rays through the instrument, by Tripathi [19].

vergent rather than a parallel electron beam, which illuminates a single point on the sample. This beam is then scanned across an area of the sample and a scattering intensity is recorded for each scan position for a scattering solid angle Ω depending on the detector. The end result forms a 2D image. Von Ardenne's prototype microscope underperformed compared to TEMs at the time and was unfortunately destroyed during the war. The STEM saw few further developments until the introduction of the field emission gun (FEG) by Albert Crewe in 1969, a more performant, coherent and durable electron source than thermionic filaments made of tungsten and lanthanium hexaboride (LaB_6). He then demonstrated imaging of individual atoms using a modern STEM design with an annular dark field (ADF) detector [20] and proceeded working on the improved cold field electron gun (CFEG) with his colleagues at the University of Chicago. Twenty years later, resolutions achievable with scanning transmission electron microscopes reached below 2 Å [21]. Then, about ten years later, microscopes with aberration correctors began inching closer to 1 Å resolutions [2] by correcting primarily for spherical aberrations that have been the limiting factors of STEM resolution. Today's microscopes are pushing past 0.5 Å resolution using increasingly com-

plex and expensive aberration correctors. At the same time, thanks to the introduction of segmented ADF detectors and pixelated bright-field (BF) cameras, fidelity and resolution can be further improved using computational techniques such as differential phase contrast (DPC) and ptychography [22].

Figure (1.3) depicts a Nion UltraSTEM 100 scanning transmission electron microscope, with a simplified schematic by Tripathi [19] next to it. Electrons are ejected at energies of tens to hundreds of keV by the CFEG at the bottom and pass an aperture that limits their angular spread. The spherical aberrations of the electron beam are then reduced by the condenser lenses and the aberration corrector, before it is deflected by the scan coils and focused onto the sample inside the stage by the strong objective lens. The transmitted electron beam then passes through the projector lens, which sets the camera length, before the electrons are collected by the detectors for imaging.

While most scanning transmission electron microscopes only detect transmitted electrons, multiple detectors can be used at the same time for different scattering solid angles. A popular combination is a high angle annular dark field (HAADF) detector working in tandem with an electron energy loss spectroscopy (EELS) camera as a BF detector at low scattering angles. This way the chemical composition of tiny areas or even atoms can be identified more accurately than just using Z-contrast on ADF images.

1.2.1 Convergent beam electron diffraction pattern

Alternatively to an EELS camera or a simple BF detector plate, weakly scattered electrons can be detected using a pixelated detector, such as a charge coupled device (CCD) camera or a pixel array detector (PAD). Doing so in a scanning transmission electron microscope enables recordings of convergent beam electron diffraction (CBED) patterns. The microscope beam's convergence, measured by the incident beam semiangle α, decreases the diameter of the beam spot and increases the diffraction spots on the detector plane behind the sample. What are typically just dots on a TEM's diffraction pattern appear as disks in STEM.

Diffracted disks may be faint or overlap with the bright-field disk. Such overlapping regions exhibit interference fringes, which can be used in more advanced, computationally expensive phase reconstruction methods in ptychography [23] to achieve super-resolution images.

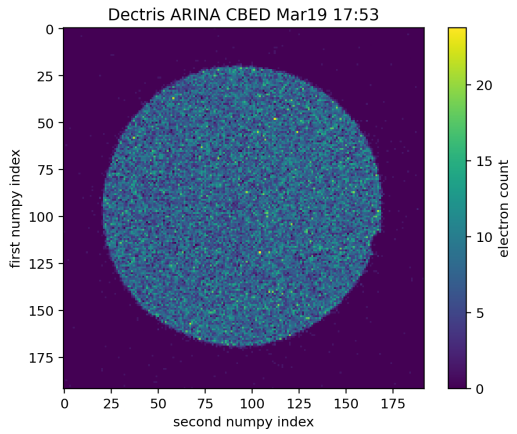

Figure 1.4: A CBED pattern captured using a Dectris ARINA hybrid pixel detector at an integration time of 0.1 ms.

Another way to extract information about a sample from a CBED is by looking at the effects of electric fields on a BF disk [24]. Electric fields much larger in extent than the electron probe's size shift the BF disk uniformly in reciprocal space. This can be modeled using a strong phase approximation. Fields much smaller than the probe, on the other hand, such as atomic fields, redistribute the intensity inside the disk. In that case, the weak phase approximation (WPA) is appropriate. This redistribution contains information about the position of the electron beam relative to the lattice and it is this information that this work aims to utilize to perform real-time drift correction.

Other than information about the sample, diffraction patterns exhibit aberrations and other imperfections pertaining to the microscope optics. A CBED pattern such as the one in Figure 1.4 has two reciprocal space axes and an intensity value for each scattering vector. In this case the axes are simply the indices of the numpy data object in a pixel coordinate system. Applying a conventional calibration to the plot axes positions the origin at the center of the image and introduces units of mrad for scattering angles or inverse Å for scattering vectors. The intensity can sometimes be directly expressed in the number of electrons detected, depending on the type of detector used and its detective quantum efficiency (DQE).

Acquiring CBED patterns during a STEM scan yields a 4D dataset consisting of intensity data in two real-space dimensions and two reciprocal-space dimensions, giving this method the name 4D-STEM.

During this thesis, most of the experimental data recorded consists of CBED patterns and

4D datasets. The goal of real-time drift compensation, however, is to extract information from CBED patterns recorded during a parked electron probe, that is, without the need for scanning.

1.2.2 Center of mass

Definition

In classical mechanics, the center of mass, or barycenter, is a unique point of a mass distribution in real space. It is defined as the point where all relative position vectors weighted by the corresponding mass sum to zero:

$$\sum_{i=1}^{n} m_i(\vec{r_i} - \vec{R}) = 0\,. \tag{1.13}$$

Here $\vec{r_i}$ is the relative position vector of a mass m_i and $\vec{R}$ the position of the center of mass.

For a continuous solid Q, with density $\rho(\vec{r})$, this condition is written as follows:

$$\int_Q \rho(\vec{r}) \left(\vec{r} - \vec{R}\right) dV = 0\,. \tag{1.14}$$

Solving equation (1.13) for $\vec{R}$ yields:

$$\vec{R} = \frac{\sum_{i=1}^{n} m_i \vec{r_i}}{\sum_{i=1}^{n} m_i} = \frac{\sum_{i=1}^{n} m_i \vec{r_i}}{M}\,, \tag{1.15}$$

where M is the total mass.

For the continuous case, M is the integral of the density $\rho(\vec{r})$ over the whole solid. Solving (1.14) thus results in:

$$\vec{R} = \frac{1}{M} \iiint_Q \rho(\vec{r})\vec{r}\ \, dV\,. \tag{1.16}$$

This is the expression for the center of mass in three-dimensional real space. Depending on the coordinate system, this point in space can be written in units of length.

Application in TEM

In the context of TEM microscopy and 4D-STEM, center of mass (CoM) is often used to find the center of diffraction disks or the bright-field disk. It can also be used to acquire

information about the structure of a sample by measuring the mean re-distribution of intensity inside of a bright-field disk. Contrast of weak higher-angle scattering in dark-field areas has little effect on the center of mass, however.

In this context, center of mass is applied to two-dimensional images in reciprocal space. In the case of a 4D-STEM dataset, CoM is calculated for diffraction patterns at every probe position, connecting the extracted information to a real-space structure. Let us rename the terms in the previous expression for the center of mass (1.16) to apply them to reciprocal space.

The density $\rho(\vec{r})$ becomes the intensity $I(\vec{k})$ at the reciprocal space position $\vec{k}$, the 3D integral over a continuous mass Q becomes a 2D integral over a diffraction pattern P and the total mass M becomes the total intensity I_{tot}:

$$\overrightarrow{\mathrm{CoM}} = \frac{1}{I_{tot}} \iint_P I(\vec{k})\vec{k} \; d^2\vec{k} \tag{1.17}$$

This form of the center of mass is often called $\overrightarrow{I^{\mathrm{CoM}}}$ [25] or $\overrightarrow{\mathrm{CoM}}$ [24] in literature, and will henceforth be referred to as such. Analogous to its classical counter part, $\overrightarrow{\mathrm{CoM}}$ is expressed in the same units as $\vec{k}$. An application of this equation is found in the py4DSTEM code, where it is used to find the approximate center of a bright-field disk or probe.

For a 4D-STEM dataset, the center of mass is a 2D vector field as a function of the probe position:

$$\overrightarrow{\mathrm{CoM}}(\vec{r_p}) = \frac{1}{I_{tot}} \iint_P I(\vec{k}, \vec{r_p})\vec{k} \; d^2\vec{k}\,, \tag{1.18}$$

with $\vec{r_p}$ being the probe position relative to the sample. The resulting vector field has two scalar components, commonly referred to as CoM_x and CoM_y.

Integrated center of mass

The center of mass can also be used to generate a real-space image of a STEM sample. Performing a 2D integration on $\overrightarrow{\mathrm{CoM}}(\vec{r_p})$ results in an image $\mathrm{iCoM}(\vec{r_p})$ that looks similar (but not equal) to that of an annular dark field detector. This image reconstruction method is called center of mass imaging or integrated center of mass (iCoM). While iCoM images are slower to record and more computationally expensive to form, they exhibit advantages over ADF such as increased signal to noise ratio and more information contained about the sample.

Center of mass imaging can also be approximated by Differential Phase Contrast imaging (iDPC) [25]. With an increasing amount detector segments, a DPC signal will approach that of CoM. A key advantage of DPC using segmented detectors remains the considerably faster acquisition time.

Physical interpretation

Lazić et al. [25] derive the center of mass from an image-formation point of view, without using 1st Born and weak phase object approximations, based on work by Waddell and Chapman [26, 27]. The only assumptions made are that a sample is thin and non-magnetic.

Starting from eq. (1.18), they come to an expression for the center of mass that expresses it as a function of an incoming electron wave and the gradient of the phase shift imprinted on the wave as it passes through the sample:

$$\overrightarrow{\mathrm{CoM}}(\vec{r_p}) = \vec{j_0} + \frac{1}{2\pi}\left(|\psi_{in}(\vec{r})|^2 \star \vec{\nabla}\varphi(\vec{r})\right)(\vec{r_p}). \tag{1.19}$$

Here $\psi_{in}(\vec{r})$ is the input wave function, with $\vec{r}$ being its real-space position on the plane of the sample, $\vec{r_p}$ is the probe position and $\vec{j_0}$ an offset vector that does not contain any sample contrast. The operator $\star$ represents a cross-correlation operation.

Cao et al. [24] attribute $\vec{j_0}$ to a non-symmetrical probe shape, caused by the presence of aberrations. In practice, there may be a multitude of factors that can exert an offset to the center of mass. Most notably, a positionally offset BF disk can introduce a CoM-offset $\vec{j_0}$ that is larger than the sample's CoM contrast magnitude. This type of offset is introduced by large, uniform fields.

Lazić et al. [25] further integrated eq. (1.19) without the offset term $\vec{j_0}$ in the Fourier domain to obtain the following expression for the integrated center of mass iCoM($\vec{r_p}$):

$$\mathrm{iCoM}(\vec{r_p}) = \frac{1}{2\pi}\left(|\psi_{in}(\vec{r})|^2 \star \varphi(\vec{r})\right)(\vec{r_p}), \tag{1.20}$$

proving that an iCoM image is proportional to the phase shift $\varphi(\vec{r})$. This type of 2D integration is a way of performing a phase reconstruction.

Next, a non-magnetic sample only exhibits an electrostatic potential field $\phi(\vec{r})$, with $\vec{r}$ being a 3D-vector in real space. It is related to the sample's electric field $\vec{E}(\vec{r})$, and the charge

distribution $\rho(\vec{r})$ within the sample, though the Poisson equation and Gauss' law:

$$\nabla^2\phi(\vec{r}) = -\vec{\nabla}\vec{E}(\vec{r}) = -\frac{1}{\varepsilon_0}\rho(\vec{r}). \tag{1.21}$$

For thin samples, a phase shift $\varphi(\vec{r})$ of the electron wave is proportional to the projected potential $\phi_z(\vec{r})$. Here $\vec{r}$ is again a 2D vector , as it was in equation (1.19). The corresponding proportionality constant is defined as $\sigma = 2\pi me\lambda/h^2$, with the electron's de Broglie wavelength λ and Planck's constant h. Inserting this into eq. (1.21) lets us rewrite it in the form:

$$\nabla^2\varphi(\vec{r}) = \sigma\nabla^2\phi_z(\vec{r}) = -\sigma\vec{\nabla}\vec{E_z}(\vec{r}) = -\frac{\sigma}{\varepsilon_0}\rho_z(\vec{r}). \tag{1.22}$$

From this relation and the fact that an iCoM($\vec{r_p}$) image is directly proportional to the phase shift $\varphi(\vec{r})$, we can conclude that a recorded vector image $\overrightarrow{\mathrm{CoM}}(\vec{r_p})$ is directly proportional to the projected electric field $\vec{E_z}(\vec{r})$ of a thin sample, save for an offset $\vec{j_0}$. Furthermore, calculating a divergence of $\overrightarrow{\mathrm{CoM}}(\vec{r_p})$ will result in a differentiated CoM image dCoM($\vec{r_p}$) that is directly proportional to the projected charge density $\rho_z(\vec{r})$.

Cao et al. [24] expanded on the quantum-mechanical wave nature of the center of mass being a function of the outgoing wave $\Psi(\vec{k}, \vec{r_p})$, and show using commutation:

$$\begin{aligned}\overrightarrow{\mathrm{CoM}}(\vec{r_p}) &= \int \vec{k}\,|\Psi(\vec{k},\vec{r_p})|^2\,d\vec{k} \\ &= \int \Psi^*(\vec{k},\vec{r_p})\,\vec{k}\,\Psi(\vec{k},\vec{r_p})\,d\vec{k} \\ &= \frac{\langle\,\hat{p}\,\rangle}{\hbar},\end{aligned} \tag{1.23}$$

with Ψ being the electron wavefunction at the detector and $\hat{p}$ the momentum operator. The result is a direct relation of the center of mass to the expectation value of the electronic momentum. This gives an intuitive explanation of CoM depending on the re-distribution of electrons within the bright-field disk.

1.3 Graphene

2D materials such as graphene can be studied at atomic scale using STEM. With properties facilitating controlled atomic manipulation, graphene in particular provides a use case for real-time drift correction.

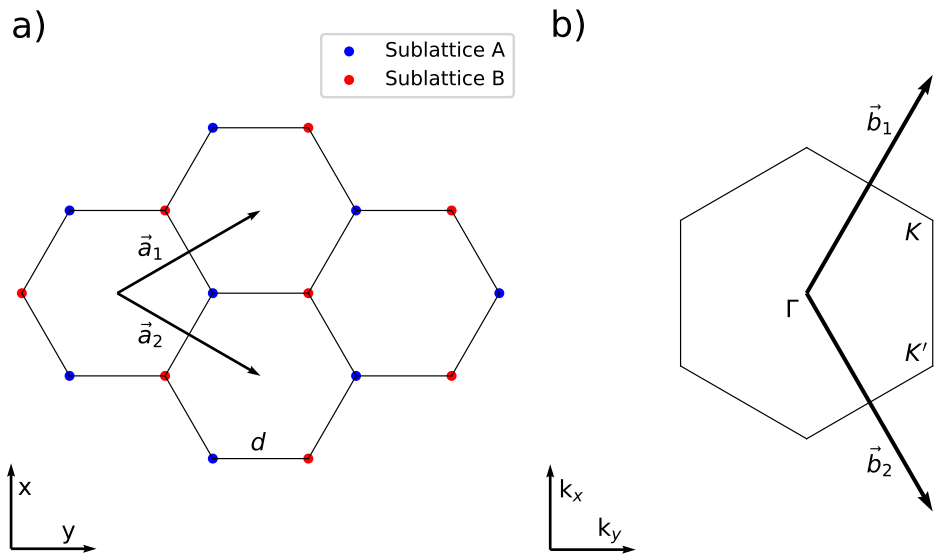

Figure 1.5: a) Real-space lattice of graphene with lattice vectors and sublattices in different colors. b) Reciprocal lattice vectors of graphene with the 1st Brillouin zone.

Carbon, a light nonmetallic element with atomic number $Z = 6$, can exist in a multitude of allotropes, such as diamond or graphite. The difference lies in the crystal structure with consequently different chemical, electronic and mechanical properties. Both diamond and graphite are 3D allotropes. Inspecting graphite, however, reveals a stack of atomically thin 2D sheets. A 2D material would exhibit a different electronic structure from its 3D counterparts, significantly changing its properties. For a long time, 2D materials were presumed to neither exist in a free state nor be thermodynamically stable due to thermal fluctuations [28, 29]. Although the possibility of a one-atom-thick carbon crystal has been studied as early as the 1930s [28], it was only in 2004 that Novoselov et al. [30] confirmed the existence of this material called graphene by pulling apart graphite layers using adhesive tape. Other than using this mechanical method called exfoliation, graphene can also be grown using chemical vapor deposition (CVD).

The carbon atoms in graphene are arranged in a honeycomb lattice. Graphene's lattice can be described using two real-space unit vectors:

$$\vec{a_1} = \frac{d}{2}(3, \sqrt{3}), \quad \vec{a_2} = \frac{d}{2}(3, -\sqrt{3}), \tag{1.24}$$

where $d = 1.42$ Å is the carbon-carbon distance. The corresponding reciprocal lattice vectors are rotated by 30° with respect to lattice orientations as shown in Figure 1.5 and can be written as:

$$\vec{b_1} = \frac{2\pi}{3d}(1, \sqrt{3}), \quad \vec{b_2} = \frac{2\pi}{3d}(1, -\sqrt{3}). \tag{1.25}$$

Graphene's orbitals are sp^2 hybridized, resulting in three hybrid orbitals aligned on a single plane and oriented at 120° angles. These three 1.42 Å long covalent σ bonds are responsible for graphene's high Young's modulus of 1 TPa and achievable tensile strengths upwards of 50 GPa [31].

The remaining non-hybridized out-of-plane p_z orbital forms a weaker covalent π bond to neighbouring atoms, hybridizing into to the π and π^* valence and conduction bands and giving rise to most of graphene's extraordinary electronic properties [32].

Pristine graphene is a zero-gap semiconductor. Its band structure does not have a gap, instead the valence and conduction bands cross at the so-called Dirac points (K, K'). The positions of the Dirac points in momentum space are

$$\vec{K} = \frac{2\pi}{3d}\left(1, \frac{1}{\sqrt{3}}\right), \quad \vec{K'} = \frac{2\pi}{3d}\left(1, -\frac{1}{\sqrt{3}}\right). \tag{1.26}$$

The unusual linear dispersion at these points allow charge carriers to behave like relativistic particles with very large Fermi velocity and to exhibit various fascinating phenomena, such as the integer quantum hall effect (IQHE) [33].

1.3.1 Impurity and dopant atoms

In practice, graphene may exhibit defects and impurities stemming from thermodynamics and the production process. Defects are irregularities in the structure of a crystal that break the regular symmetry. In atomically thin 2D materials such as graphene, defects are limited to holes, line and point defects [34].

Impurities and their intentionally introduced counterpart called dopants, on the other hand, consist of atoms that are foreign to the structure of a pristine crystal. They can range from single atoms to large clusters of impurity atoms. Substitutional impurities, in particular, replace regular carbon atoms at their respective sites.

Introducing vacancies and impurities into graphene affects the electronic structure of the material, for instance by opening a band gap in the case of boron [35] and nitrogen [36]. This makes doping a potential tool for a variety of electronic and chemical applications. Other impurities such as silicon can enhance plasmon resonances and act as atomic antennae [37].

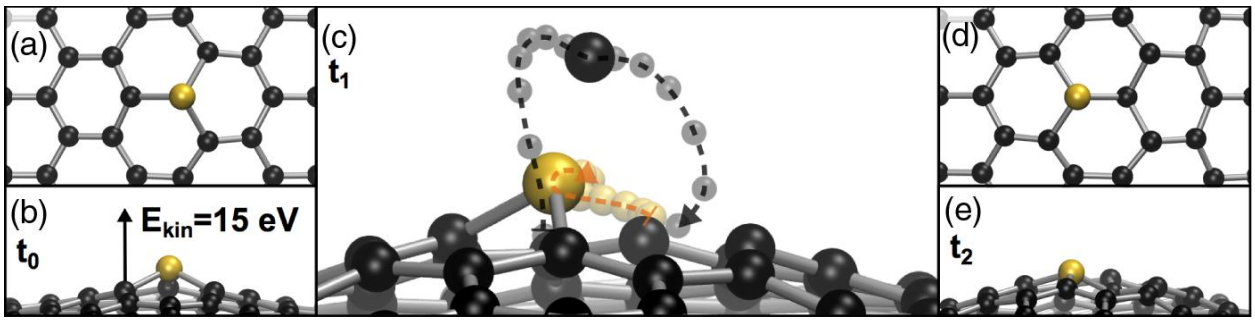

Figure 1.6: Molecular dynamics simulation of the direct exchange proccess by Susi et al. [3]. (a) Top-down view of the Si-C_3 starting configuration. (b) Side view of an electron impact transferring 15 eV of kinetic energy to a C neighbor of the impurity. (c) Trajectories of the ejected C atom and Si impurity overlaid on a snapshot at $t_1 \sim 700$ time steps into the simulation. (d) Top-down view at $t_2 \sim 1400$ time steps and (e) side view at t_2.

1.3.2 Single-atom manipulation

A peculiar phenomenon of graphene impurities, is the ability to swap lattice sites with a neighboring carbon atom. Discovered in 2014 by Susi et al. [3], electron irradiation of the area around a substitutional silicon impurity caused it to move to a different position of the lattice. This particular three-fold configuration (Si-C_3) is non-planar due to silicon's larger covalent radius. Subsequent studies and attempts at making controlled movements of a silicon atom achieved a statistically significant success rate in the manipulation of the impurity, particularly by moving it around a graphene hexagon several times [38]. Experiments using other impurity atoms such as aluminium [4] and phosphor [5] have also been conducted, but resulted in far more cases of beam damage to the lattice and loss of the impurities respectively.

The underlying direct exchange process is too fast to be observed, but attempts to simulate it using ab-initio molecular dynamics (MD) depicted in Figure (1.6) show the carbon neighbor of a silicon atom jumping out of the graphene lattice plane and the impurity relaxing into the created vacancy. The launched carbon atom is then pulled back into the lattice due to their interaction, effectively swapping their positions. More recent DFT/MD simulations, including phonon dispersion, by Tripathi et al. [38] indicated a transferred kinetic energy threshold $T = 14.8$ eV for incoming electrons to trigger a direct exchange and a knock-on threshold $T_d = 16.9$ eV. Least-squares fitting of an event cross section σ_d to experimental data, however, yielded lower values of $T = 13.04$ and $T_d = 14.44$. This is believed to stem from inelastic scattering effects weakening Si-C bonds [39].

Successful activation of direct exchange requires irradiating the desired carbon neighbor for hundreds of seconds with a 55 keV beam or tens of seconds at 60 keV, but with a higher chance of the silicon atom being locked into a stable Si-C_4 configuration by the ejection of the irradiated carbon atom.

On one hand, this beam needs to be sufficiently small on the sample plane to efficiently target the correct carbon neighbor and deposit most of its dose at that site. On the other hand, the probe needs to stay on said C atom for an extended period of time without drifting away. Drift commonly means mechanical drift of the stage, caused by factors such as mechanical lash, oscillations, hysteresis of piezo-electric motors, thermal and gravitation effects or the stiffness and consistency of o-rings [40]. There are also other types, such as lens voltage drift, leading to parasitic aberrations or probe drift.

Established methods of drift correction work on the basis of analyzing STEM images. Atom manipulation on the other hand requires a way of compensating drift in real time and with a stationary "parked" probe, forming the motivation of this thesis.

Direct exchange as a method of controlled single-atom manipulation opens up the potential for a multitude of new applications and technological advances. Atomic memory arrays [41] with stronger covalent bonds for temperature stability, exploration of quantum many-body effects [42, 43] and development of atomic antennae [37].

Chapter 2

Materials and Methods

2.1 STEM simulations

Planning, prototyping and analyzing experimental data was aided by *ab*TEM simulations. *ab*TEM is a Python-based image simulation package [44] aimed at providing an easy way to simulate (S)TEM images. It also incorporates the option of using ab-initio potentials via the DFT code GPAW. Simulations can be performed using the multislice or more recent PRISM algorithm for larger systems.

While the more advanced features for *ab*TEM are impressive, this work will utilize the simpler independent atom model (IAM) to approximate the potential of carbon and graphene and the multislice algorithm to simulate images of pristine graphene and its center of mass.

The workflow using *ab*TEM consists of setting up a lattice model with the ASE package [45], using it to generate a potential, preparing a wave function representing the probe, performing the multislice algorithm, and applying a detector. Simulating a STEM scan is also an option, in which case a multislice simulation is performed for each probe position or compounded using PRISM. An official guide to *ab*TEM can be found online [46] .

2.1.1 Independent atom model

The electron beam of a microscope interacts with the sample primarily by interacting with the Coulomb potential of its electrons and nuclei. Thus, simulations require the total electrostatic potential of the specimen. A commonly used approximation is the independent atom

model (IAM), which calculates the potential as a superposition of single atom potentials arranged in a crystal. While this method is fast, it neglects valence bonding effects that some materials exhibit.

The full specimen potential $V(\mathbf{r})$ is:

$$V(\mathbf{r}) = \sum_i V_i(\mathbf{r} - \mathbf{r}_i) \quad , \tag{2.1}$$

with the atomic potential $V_i(\mathbf{r})$ of the i'th atom at position $\mathbf{r}_i$.

The potentials of individual atoms $V_i(\mathbf{r})$ can be obtained by applying Poisson's equation to their electron charge distribution, which in turn can be approximated using computational methods such as Hartree-Fock or density functional theory (DFT). However, performing such calculations repeatedly can be computationally expensive and time-consuming. Thus, similar to most multislice simulation codes, *ab*TEM includes a parametrization of the atomic potentials.

Parametrizations entail fitting a set of simpler functions such as Gaussians, Lorentzians or the analytically solvable hydrogen form factor [47] to an atomic form factor $f_e(\theta)$ (see chapter 1.1). The corresponding fitting parameters can then be conveniently used to reconstruct the atomic potential. By default, *ab*TEM uses the parametrization by Lobato and Van Dyck [47], with parametrizations by Kirkland [8] and Peng [48] also implemented and selectable.

2.1.2 Multislice simulations

To simulate how the electron beam of a scanning transmission electron microscope interacts with a sample, the multislice algorithm, first described by Cowley and Moodie [49], propagates a wave function representing the beam through slices of the sample's electrostatic potential.

Two assumptions are made in this method. The first is that the wavefunction varies slowly along the z-axis, meaning the term proportional to $\partial^2/\partial z^2$ of the Laplacian in the Schrödinger equation can be neglected. The second is the high-energy approximation, in which the electron's kinetic energy and wave vector are much larger than any transverse variations, including from relatively weak potentials involved in scattering. Additionally, assumptions made for the atomic form factor (1.1) such as the first Born approximation apply here as well.

A great recent guide on this topic was made by Ophus et al. [7] with in-depth derivations available in the book by Kirkland [8].

Potential discretisation

Slicing the potential of a material is done by discretising it into finite slices along the z-axis. The slice thickness Δz presents a trade-off between computational cost and accuracy, with reasonable values generally between 0.5 Å and 2 Å. *ab*TEM uses a numerical integral using the Gaussian quadrature rule by default. Additionally, because simulated potentials use a cut-off radius, *ab*TEM implements a tapering cut-off starting at 85% of the full cut-off to prevent discontinuities. The potential slices are calculated using integrals of the form:

$$V_n(x,y) = \int_{z_n}^{z_n+\Delta z} V(x,y,z)dz\,, \tag{2.2}$$

where V_n is the projected potential at slice n, z_n the z-position at the entrance of the n'th slice and Δz the slice thickness.

A computationally faster, but potentially less accurate alternative are infinite projection integrals. The finite integrals are replaced with infinite ones that stay well-behaved due to only Gaussians and Lorenzians being used in the potential parametrization:

$$\int_{-\infty}^{\infty} V(x,y,z)dz \approx \int_{z_i}^{z_i+\Delta z} V(x,y,z)dz\,. \tag{2.3}$$

The integrals are calculated using an approach by Van Den Broek et al. [50], which is a hybrid method combining real-space and Fourier space calculation.

Wave propagation

After establishing a set of sufficiently thin slices and choosing a wave function representative of the desired electron beam, scattering interactions are calculated at the entrance to each real-space slice using the respective potential slice projection. The resulting output wave is then fed into the next slice as input.

The scattering is represented by a transmission operator

$$t_n(\vec{r}) = \exp\left[i\sigma V_n(\vec{r})\right], \tag{2.4}$$

that applies a phase shift, proportional to the potential slice $V_n(\vec{r})$, with the interaction constant σ, as well as a propagation operator

$$p(\vec{r}) = \frac{1}{i\lambda\Delta z} \exp\left[\frac{i\pi}{\lambda\Delta z} r^2\right], \tag{2.5}$$

which is the Fresnel free-space operator for propagation by a distance Δz, to form an expression for the slice's exit wave function

$$\psi_{n+1}(\vec{r}) = p(\vec{r}) * \left[t_n(\vec{r})\psi_n(\vec{r})\right] , \tag{2.6}$$

with $*$ being the convolution operator. Using the Fourier convolution theorem on eq. (2.6) converts this equation into a simple multiplication in reciprocal space, which can be applied sequentially and calculated efficiently.

The final exit wave function can then be modulated by a transfer function (MTF) to reflect microscope optics effects, and collapsed into a real-valued intensity using a detector function.

2.1.3 Center of mass

*ab*TEM's implementation of the center of mass takes a set of diffraction patterns or a 4D dataset as input and returns a complex image. The real and imaginary parts of the image then correspond to the center of mass component in x- and y-direction respectively.

This discrete form of equation 1.18 is implemented as the following Python function:

```
def _com(array, x, y):
    com_x = (array * x[:, None]).sum(axis=(-2, -1))
    com_y = (array * y[None]).sum(axis=(-2, -1))
    com = com_x + 1.0j * com_y
    return com
```

with `array` being a numpy array containing diffraction pattern intensities and `x`, `y` vectors containing the corresponding coordinates in reciprocal space.

For a single diffraction pattern, the above function can be written as follows:

$$\mathrm{CoM}_x = \sum_{i,j=0}^{n-1,m-1} I_{i,j}\, k_{x,i} \tag{2.7}$$

$$\mathrm{CoM}_y = \sum_{i,j=0}^{n-1,m-1} I_{i,j}\, k_{y,j}\,, \tag{2.8}$$

where $I_{i,j}$ are the CBED pattern intensities at indices i, j with dimensions n, m, and k_x, k_y are the reciprocal-space axes vectors, corresponding to diffraction pattern calibration.

When using a 4D dataset for the `array` parameter, the calculation is performed per diffraction pattern, because `.sum(axis=-2, -1)` only sums over reciprocal space dimensions, represented by the last two array indices by convention.

Due to the fact that *ab*TEM's simulated diffraction patterns are generated in a normalized state, the total intensity I_{tot} from equation 1.18 is set to 1. Additionally, keep in mind that *ab*TEM's center of mass function expects array indices i, j to map to x, y, as opposed to the y, x convention often used for digital imaging.

2.1.4 Diffraction-space rotation correction

Due to phenomena discussed in section 1.1.1, the rotation offset between the real space image and the diffraction-space image (henceforth referred to as "QR-rotation") needs to be accounted for. Otherwise, a connection of the center of mass of CBED patterns recorded in reciprocal space to real-space displacement cannot be made.

For this purpose, code from a previous version of the py4DSTEM dpc.py [51] module was used. More specifically, to approximate the QR-rotation angle θ_{min}, two different methods were implemented. The first method finds the angle θ_{min} using the smallest curl of the CoM images via gradient descent. The second method chooses θ_{min} based on the phase reconstruction image with the highest contrast.

An QR-rotation is performed by applying a rotation matrix that is a function of θ_{min} to all CoM vectors $\vec{I}_{\mathrm{CoM}}(\vec{r_p})$ of a dataset:

$$\vec{I}_{\mathrm{CoM,rot}}(\vec{r_p}) = \begin{pmatrix} \cos(\theta_{\mathrm{min}}) & -\sin(\theta_{\mathrm{min}}) \\ \sin(\theta_{\mathrm{min}}) & \cos(\theta_{\mathrm{min}}) \end{pmatrix} \cdot \vec{I}_{\mathrm{CoM}}(\vec{r_p})\,. \tag{2.9}$$

Here $\vec{I}_{\mathrm{CoM}}$ is a 2D vector field with the probe position $\vec{r_p}$ as parameter. This is also applicable to single CoM vectors of single diffraction patterns, as is later used in tracking scenarios.

If one were to plot $\vec{I}_{\mathrm{CoM,rot}}$ as a CoMx and a CoMy image, then an arbitrary QR-rotation would result in the images looking like superpositions of each other.

Additionally, a flip may be required if the real and diffraction space axes are flipped with respect to each other:

$$\vec{I}_{\mathrm{CoM,flipped}}(\vec{r_p}) = \begin{pmatrix} 0 \\ (-1)^{flip} \end{pmatrix} \cdot \vec{I}_{\mathrm{CoM}}(\vec{r_p})\,. \tag{2.10}$$

Here $flip$ is a boolean variable.

Both methods of QR-rotation correction work under the assumption that the center of mass is proportional to the gradient of the scalar phase-shift that a sample imposes on an electron beam. This further assumes that the specimen is thin and can be treated as a pure phase object.

Minimizing curl

The goal of the curl method is to find the rotational offset θ_{min} and the $flip$ by checking if the center of mass is a conservative vector field – one that can be a gradient to a scalar phase. In other words, line integrals along the field must be path-independent. This condition cannot be met if θ_{min} is non-zero, meaning diffraction space is rotated by an angle QR-rotation with respect to real-space coordinates.

py4DSTEM's algorithm first performs a gradient descent of cost functions representing all 4-pixel closed-loop line integrals, as a function of a QR-rotation angle θ, starting at $\theta = 0$. This is done once for a non-flipped case and once more for a flipped case. At the end, the square of sums of all cost functions is compared between the flipped and un-flipped case and the smaller of the two is chosen.

Maximizing contrast

The maximum contrast method of finding the QR-rotation is based on the premise that the phase reconstruction image of the center of mass exhibits the highest contrast when θ_{min} and $flip$ are compensated.

The corresponding py4DSTEM algorithm first performs a phase reconstruction for each given

combination of QR-rotation angles θ and $flip$ values. The number of angles to be tested amounts to 100 by default and is evenly distributed from 0 to 2π. The resulting phase images are then used to calculate the respective standard deviations as a measure of contrast. θ_{min} and $flip$ are then chosen based on the combination with the highest standard deviation value.

The phase reconstruction function py4DSTEM uses here is a Fourier-space integration of the center of mass, similar to that of *ab*TEM. It features an optional iterative field of view optimization, which is not used here. That particular version of py4DSTEM was also missing a factor of 2π in phase magnitude compared to *ab*TEM.

Note that the use of a predetermined interval $[0, 2\pi]$ for θ results in two degenerate solutions, 180° apart. This corresponds to a complete phase-contrast reversal and can be seen as such in the iCoM image. The correct solution has to be picked manually, by inverting the final center of mass $\vec{I}_{\mathrm{CoM,rot}}$, if the phase or iCoM image does not exhibit bright atomic sites and dark open spaces. Physically, the second solution would imply an opposite, positive charge to the electron, which is not applicable to an electron microscope.

These QR-rotation correction methods are computationally expensive for large datasets, especially the more sophisticated contrast method.

2.1.5 Poisson noise

A more realistic simulated TEM image can be achieved by applying Poisson noise. Poisson noise or shot noise in a microscope is introduced by the electron gun. Electrons ejected by the gun follow a Poisson distribution, which leads to a noisy image, especially at lower currents. *ab*TEM provides a function to apply Poisson noise to diffraction patterns or images. The function accepts a total electron dose or a dose per area and an optional seed for the random number generator as parameters. In this work, this function is used to approximate noisy diffraction patterns and their effect on the drift-tracking algorithm.

Real detectors generally introduce their own Gaussian noise introduced through their electronic components, though this analysis has been left out for brevity. Additionally, the detector used in the latter half of experimentation (see chapter 2.2.2) uses a counting methodology to eliminate most Gaussian noise.

2.2 Nion UltraSTEM 100

The UltraSTEM 100 is an aberration-corrected scanning transmission electron microscope developed and built by Nion [52]. At the Vienna Sternwarte location, the microscope is installed inside a shielded room on a dedicated vibration-free foundation with the electron gun located at the bottom of the column, oriented upwards.

Additionally, the microscope stage is connected to a custom UHV sample transfer system consisting of flexible bellows, gate valves and branching pipes. This "CANVAS" system [53] is further connected to multiple other devices for sample preparation and alteration, as well as sample storage. This ensures minimal contamination of low-dimensional materials during sample transfer. The transfer itself is done by loading sample pucks that are compatible with the microscope stage, onto small cars, up to three pucks at a time, and manually moving the cars along the bellows using magnetic coupling. Exchanging sample pucks between the car and the microscope stage is also done manually using a retractable gripper arm.

The microscope's electron source is a cold field electron gun (CFEG) rated for 100 kV acceleration voltage with an energy spread of roughly 300 meV. In constant-current mode, it can be operated between 40 kV and 100 kV .

Inside the microscope, the gun is kept at an UHV in the low 10^{-11} Torr range. It is separated from the sample chamber by two intermediate pumping volumes, enabling experimentation at pressures six magnitudes higher than the gun's operating pressure.

After leaving the gun, electrons first pass through a set of three condenser lenses and a virtual objective aperture (VOA), allowing for a wide range of illumination semi-angles from < 1 mrad to ~ 35 mrad.

Next is the C_3/C_5 aberration corrector. It consists of three quadrupole-octupole lenses handling third-order (C_3) aberrations and twelve rotatable quadrupoles correcting up to fifth-order (C_5) aberrations. Tuning is done using software due to the sheer complexity of the lens system.

From there, electrons pass the scan coils and enter the strong objective lens, which focuses the beam onto the sample. The custom-made sample chamber [54] is meant for controlled internal pressures between 10^{-10} and 10^{-6} mbar, reaching ultra-high vacuum (UHV). Standing at 120 mm, the chamber is twice as high as the standard version and its condenser-objective

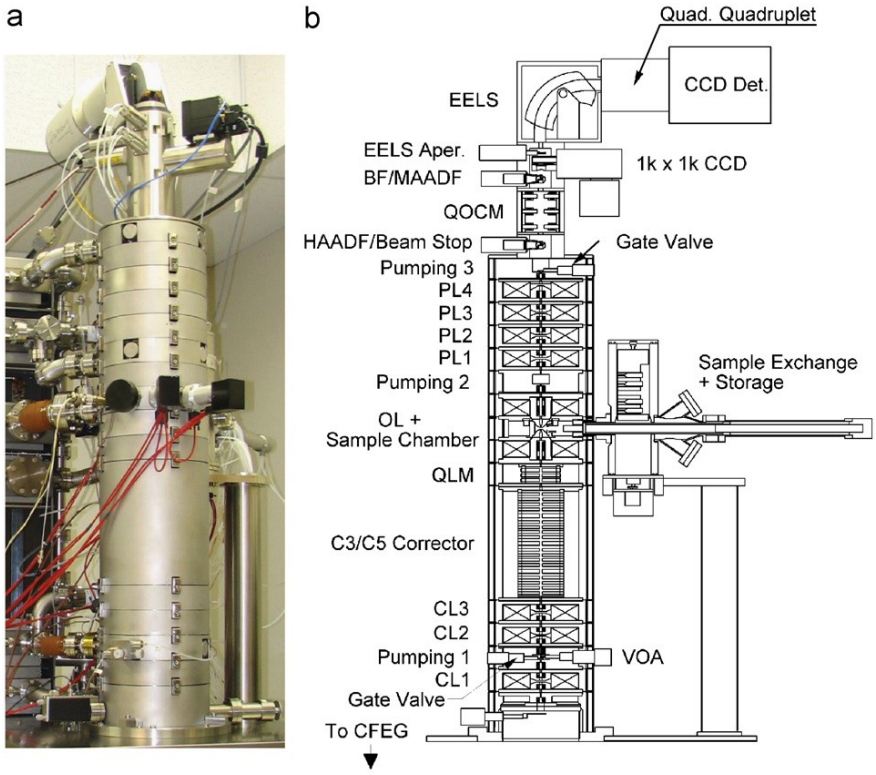

Figure 2.1: a) UltraSTEM 100 default column and b) schematic crossection of the column.

lense (OL) has been optimized for chromatic aberrations as low as $C_c = 1.0$ mm. The whole setup is designed to be modular and customizable, as well as completely computer-controlled, allowing for remote access.

The transmitted signal then enters a set of four projector lenses that offer a range of virtual camera lengths from less than 10 mm to over 1 m.

Finally, at the top of the column is the detector section. The first detector has three positions: a HAADF scintillator, a position to act as a beam-stop and the retracted position. Located between the first and second detector is a quadrupole-octupole coupling module (QOCM). Its purpose is to adjust second and third order aberrations for use with electron energy loss spectroscopy (EELS). Additionally, it provides a lens for modifying the camera length relative to the HAADF detector. This way, diffraction patterns can be expanded to cover most of a subsequent pixelated detector's area. The second detector section also has three positions: a medium angle annular dark field (MAADF) scintillator, a bright-field (BF) scintillator and the retracted position. Mounted on top of that are modules containing the CCD camera and other newer pixelated detectors, also retractable. The column is then topped off by an EELS aperture followed by the electron energy-loss spectrometer.

Figure 2.2: Front view of the Hamamatsu ORCA-Flash4.0 V3 CMOS camera without a scintillator. Courtesy of the manufacturer.

2.2.1 Hamamatsu ORCA sCCD camera

Hamamatsu's ORCA line of scientific CCD (sCCD) cameras consists of CMOS-type indirect detectors. As such, they require a scintillator layer in order to detect electrons. Scintillators such as the P32 powder scintillator convert electrons to photons, which are then delivered onto a semi conductor camera sensor via fiber-optic coupling.

The particular model in use is a Hamamatsu ORCA-Flash4.0 [55], with a pixel array size of 2048 by 2048 pixels and a maximum output refresh rate of 100 frames per second. The pixel size is 6.5 by 6.5 µm. The sensor's rms readout noise is rated at roughly 1.5 electrons. It can be connected to a PC using either a USB 3.0 cable or a Camera Link interface board.

The sensor's detective quantum efficiency (DQE) is strongly dependent on the wavelength of captured photons, with a maximum of 82% at 560 nm for the latest generation.

2.2.2 Dectris ARINA direct electron detector

The Dectris ARINA is an electron-counting hybrid-pixel or pixel array detector (PAD) [56]. This type of detector does not utilize a scintillator, instead it detects electrons directly using a dedicated semi conductor chip, bonded to a KITE electron counting application-specific integrated circuit (ASIC) [57]. Instead of digitizing an analog intensity signal, using the electron counting method intensity peaks are registered as electron impact events. This approach has the advantages of nearly eliminating the inherent electronic Gaussian noise of a detector, as well as any temporal smearing effects [58].

To allow for direct absorption of high-energy electrons and to enable a high dynamic range,

Figure 2.3: Front view of the Dectris ARINA direct electron detector.
Courtesy of the manufacturer.

the physical pixel size of these detectors is much larger than that of indirect detectors. In case of the Dectris ARINA, the pixel size is 100 by 100 µm. Due to the much larger area required by individual pixels, the size of the pixel array that can reasonably fit on a single chip is greatly reduced compared to CCD cameras. The array size of the ARINA is thus 192 by 192 pixels.

Zambon et al. [57] assessed the DQE of the detector to be 0.80 for a 60 keV beam, when aligned perpendicular to it. The Dectris ARINA can achieve a frame rate of 20 kHz or up to 120 kHz with in-pixel compression and 2 by 2 hardware-binning enabled. It is connected to a dedicated detector control unit of standard 19 inch rack size at 2U height via optical fibre, which in turn can be reached by the User PC through a network connection at up to 100 gigabits per second.

2.3 Nion Swift

Nion Swift is an open-source image-processing and electron microscope control software released by Nion for use with its electron microscopes. It is written in Python and uses Qt as backend for its Nion UI. To communicate with and control the microscope, Nion Swift connects to Nion's proprietary hardware control software AS2. In addition to a user-friendly interface, it provides access to data and control elements through its integrated Python console. This feature was essential in creating a plugin meant to interface with instruments specific to the UltraSTEM 100 at Vienna's Sternwarte, such as the newly installed Dectris ARINA detector discussed in the previous chapter.

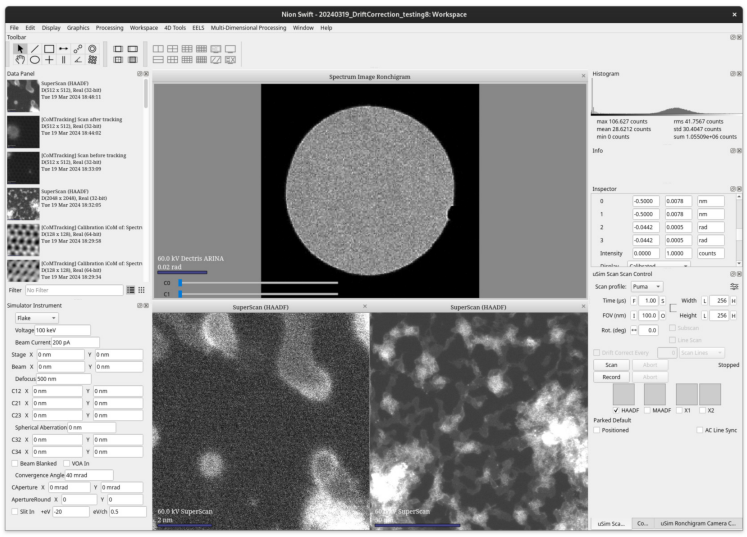

Figure 2.4: Screenshot of the nionswift UI, featuring the usim microscope simulator.

Nionswift API

The nionswift API object is the main way to interface with nionswift data elements and instruments inside its Python console. It is also used by plugins to access them, create new ones and display their own graphical user interface.

Nionswift plugins are placed inside of the `nionswift_plugin` directory of the (virtual) Python environment containing nionswift. They require their own folder, containing an `__init__.py` file, that imports or contains an '-Extension' class and a '-Delegate' class. The former grabs the API object, creates the GUI panel object containing the '-Delegate' class with the API as argument, and closes it once the plugin is unloaded, whereas the latter defines the GUI elements to be populated on the panel. A plugin template can be found on nionswift's GitHub webpage [59].

usim

The Nion Swift STEM Microscope Simulator usim is a library for nionswift that provides a few virtual specimens as well as simulated microscope and detector control elements. It was used to prototype the CoM-tracking plugin's access to the beam shifter controls and detector images. It was also useful in simulating the beam movements of a tracking loop, before CoM-tracking had its own tracking simulation routine inside nionswift.

Chapter 3

Results and Discussion

3.1 Simulations

For this thesis, most of the simulation procedure adheres to the walkthrough section of the *ab*TEM documentation [46].

Simulating graphene starts with creating the graphene structure itself. The `ase` module was used to create an object using the `ase.build.graphene(vacuum=2)` method, representing a pre-defined graphene unit cell. This object contains the positions of the two atoms. The parameter `vacuum=2` sets the extent of the unit cell to 4 Ångströms in z-direction and positions the structure at the z-coordinate 2 Å.

Next, because the multislice algorithm used by *ab*TEM requires an orthogonal and periodic structure as input, the hexagonal graphene unitcell must be transformed. This was done

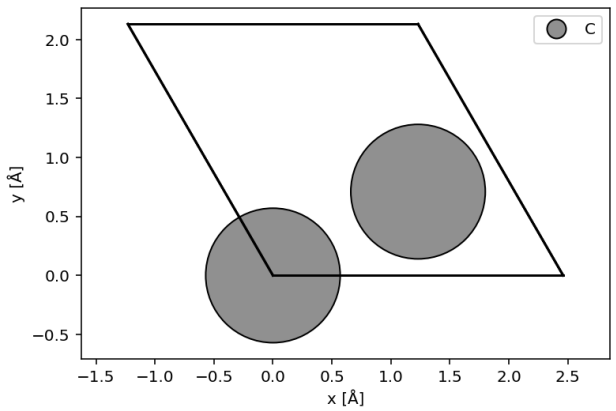

Figure 3.1: The unit cell of graphene according to the ASE module.

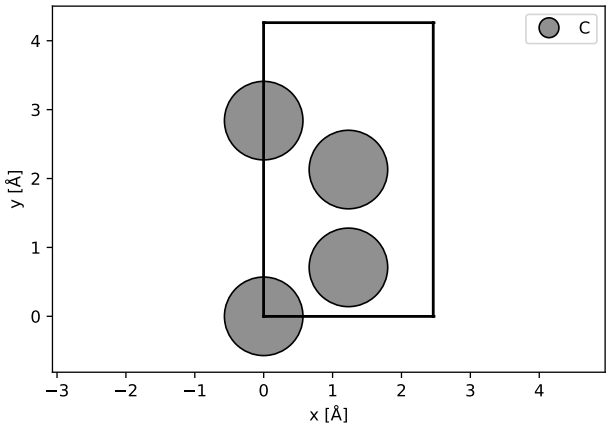

Figure 3.2: An orthogonalised unit cell of graphene.

using *ab*TEM's `orthogonalize_cell` function. By doubling the cell size, hexagonal unit cells can be made orthogonal without introducing strain [60]. The result can be seen in Figure 3.2.

With the structure prepared, the electrostatic potential of the unit cell can be created. An IAM potential was set up using the `abtem.Potential` function. The Lobato parametrization and finite projection discussed in chapter 2.1 were used in conjunction with a value of 0.05 Å for the real-space sampling of the potential in x-y plane. The `slice_thickness` parameter of value 1 Å amounts to slicing the 4 Å tall potential into four slices of 1 Å in z, the electron beam direction.

```
potential = abtem.Potential(
    graphene,
    sampling=0.05,            #spacing of the xy-samples of the potential
    parametrization="lobato",
    slice_thickness=1,
    projection="finite",      #type of integrals used for multislice
    plane="xy"
)
```

Next, a probe wave function needs to be set up. It represents the energy and shape of the STEM's electron beam. A probe object is set up using `probe = abtem.Probe`. Throughout the different stages of the project, different probe parameters were used. In early simulations, the parameters given in the *ab*TEM tutorial of 80 keV, a semi-angle cutoff of 20 mrad and some aberrations in C_{10} and C_s were used, despite not representative of the planned

experimental setup. They were later changed to 60 keV, an increased cutoff at 30 mrad and no aberrations, making it an idealized version of the STEM setup used in experiments.

Inspecting the probe profile is made easy using by calling `probe.profiles().show()`, which displays a line profile of the probe intensity through its center. For a sense of probe size, the FWHM of this profile can also be displayed by typing `.width(height=0.5)` instead of `.show()` and printing the value in Å. The FWHM of initial 80 keV simulations amounts to 1.11 Å, whereas later aberration-less 60 keV simulations exhibited a less realistic 0.74 Å.

Two more adjustments needed to be made to prepare the probe wave function and the potential for simulations. First, a larger, tiled version of the unit-cell potential was set up using `tiled_potential = abtem.CrystalPotential(potential, tiling)`. Here, `potential` is an object containing our previous unit-cell potential and tiling is a tuple that states how often the cell should be repeated in (x, y, z) directions. This is done in order to minimize artifacts that can occur in simulations, stemming from a probe function that self-interacts due to a unit-cell smaller than the probe. In 80 keV simulations this was set to $(5, 4, 1)$, whereas $(3, 2, 1)$ was chosen for 60 keV simulations. Second, `probe.grid.match(tiled_potential)` was used to match probe wave function parameters such as extent and sampling to the tiled potential.

To perform 4D-STEM simulations, a so-called pixelated detector is initiated with `abtem.PixelatedDetector`. A maximum scattering angle can be provided as a parameter, to emulate a specific, finite detector size. For this project, the parameter was set to 200 mrad for 80 keV and 100 mrad for 60 keV simulations. In both cases the detector records angles much greater than the semi-angle cutoff of 30 mrad.

The last prerequisite for a simulation is a scan object specifying the real-space positions of the probe. The scan object can be set up as either a 2D grid scan, a 1D line scan or a 0D point scan. A grid scan is object is created using `abtem.GridScan`. The start and end corners of the area to be scanned may be provided in coordinates of Å, or in fractional coordinates relative to a given potential object. For simulations in this work, the original potential of a single unit cell is chosen and the fractional end point is set to $(1, 1)$ resulting in a grid scan encompassing the entire cell. The default real-space sampling of this scan is set to the Nyquist sampling property of the probe during simulation to minimize computation time without introducing aliasing artifacts.

Table 3.1: *ab*TEM simulation probe parameters used throughout the project and corresponding FWHM of the probe at sample plane.

	version 1	version 2	version 3	single C
electron energy [keV]	80	60	—"—	—"—
aberrations C_{10}, C_s [Å]	50, -50e4	-	-	-
semiangle cutoff [mrad]	20	30	34	30
extent [Å]	20	-	-	-
sampling [Å]	0.05	—"—	—"—	—"—
defocus [Å]	-	2	—"—	—"—
maximum angle [mrad]	200	—"—	100	200
Gaussian source size [Å]	-	-	0.3	-
FWHM [Å]	1.11	0.74	—"—	—"—

Line scan objects are created using `abtem.LineScan`. Start and end points can be assigned in the same way as for grid scan objects. Alternatively to providing a sampling value, one can set the grid-points parameter `gpts` to the number of scan positions. For line scan simulations in this work, `gpts` is set to 101 or 1001, combined with setting `endpoint = True`. This allows for a smoother representation of the center of mass and aligns the zero values with the positions of atoms, as can be seen later in Figure (3.6) b).

For the purpose of testing individual positions without simulating a larger portion of the unit cell, a point scan object is created using `abtem.scan.CustomScan` with coordinates in Å given as parameters.

With everything set up, a multi-slice simulation can be started by calling `probe.multislice`. The parameters used for this function consist of the tiled potential object, a scan object and the pixelated detector object. The output of this function is an object containing a set of diffraction patterns, arranged in a 4D dataset. The actual computation of this dataset can be launched by appending `.compute()` to the simulation call or the resulting object. Gaussian noise is then added by appending `.gaussian_source_size` to the pattern object. *ab*TEM did not support this for line scans, however.

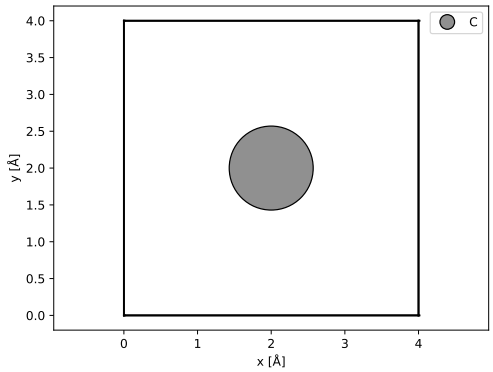

Figure 3.3: a) Single carbon atom "cell" used in simulation.

Single carbon atom simulation

Besides graphene, STEM simulations of a single carbon atom were performed. In this case, a carbon atom is placed into the center of a cubic cell with side lengths of 4 Å, with periodic boundary conditions turned on.

```
C = ase.Atoms("C",
              positions=[(2.0,2.0,2.0)],
              cell=[4,4,4],
              pbc=True)
```

The rest of the simulation setup is identical to version 2 of the graphene simulations.

3.1.1 Graphene's center of mass

The previous chapter detailed how the simulation of a 4D dataset of graphene was set up and executed using *ab*TEM. To calculate the center of mass for all diffraction patterns in the resulting dataset, *ab*TEM provides a method belonging to the object containing the patterns, called `.center_of_mass`. Calling it produces a new *ab*TEM image object, containing a complex 2D array. Like most other *ab*TEM objects, it can be quickly visualised using the `.show` method, as shown in Figure (3.4) for three different real-space sampling parameters.

Another more intuitive way to visualize the center of mass is as a vector field by interpreting the real parts as x components and imaginary parts as y components of a vector at each probe position. Figure (3.5) a) shows a plot of such a vector field at 0.123 Å sampling. Considering the physical interpretation discussed in chapter 1.2.2, these vectors are negatively

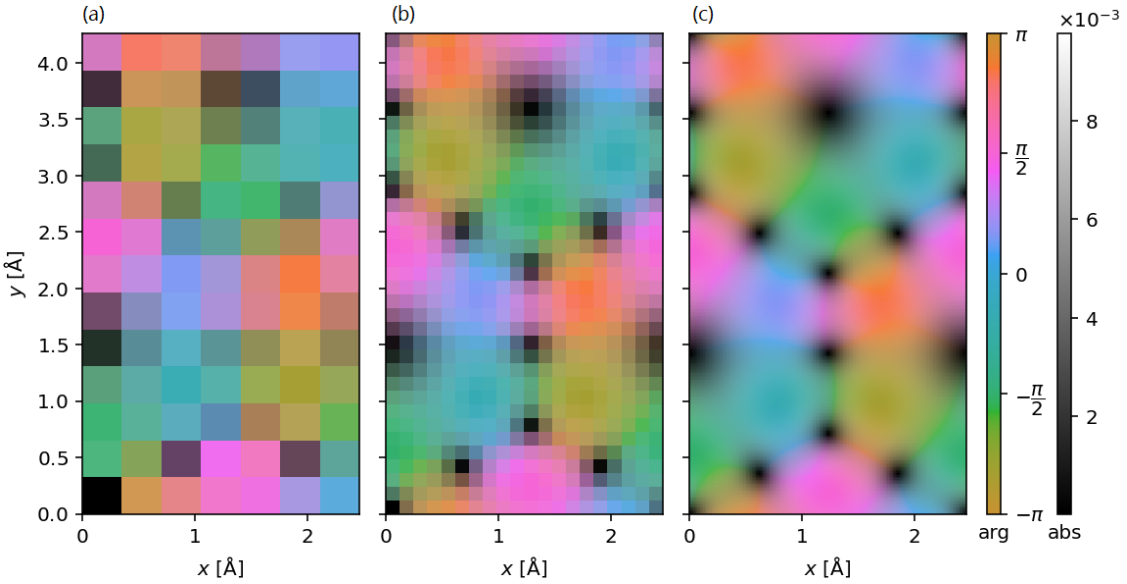

Figure 3.4: *ab*TEM's visualisation of graphene's center of mass at (a) Nyquist, (b) 0.123 Å and (c) 0.0123 Å scan sampling. The colorbars pertaining to the argument and absolute of the complex datapoints can be enabled using the `cbar` parameter.

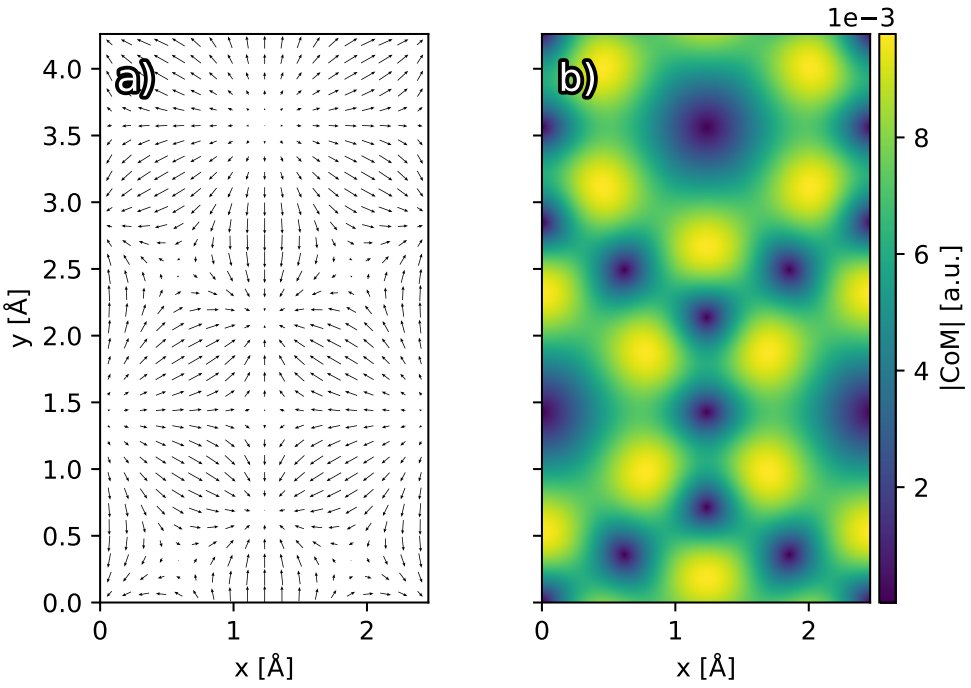

Figure 3.5: Visualisation of graphene's center of mass as a) a vector field, and b) magnitudes of CoM using the viridis colormap.

proportional to vectors representing the projected electric field in the sample, meaning they point towards the nearest positive charge.

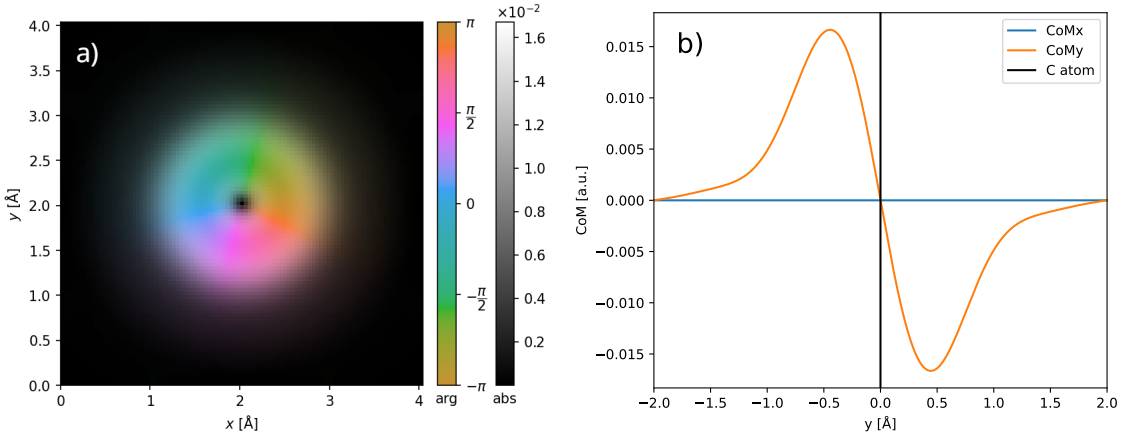

Figure 3.6: a) Center of mass of a free-standing carbon atom, b) center of mass of a line scan across the atom.

Single carbon atom center of mass

The center of mass in the single carbon atom case is depicted in Figure (3.6). Graph a) depicts magnitude as brightness and angle as color. Graph b) shows the center of mass of a line-scan over the atomic site along y, starting from 0 in simulation coordinates or -2.0 Å relative to the atom. Here, an anti-symmetric relation between position and the center of mass magnitude, equivalent to the CoM's y-component, becomes evident.

The maximum center of mass magnitude, without any Gaussian blur, is found to be ≈ 0.0167 at 0.444 Å distance from the carbon atom, in the shape of a ring around it. For comparison, a line scan through two neighboring atoms in graphene exhibited two different magnitude maxima: one of ≈ 0.0198 at 0.412 Å towards a hexagon center and one of ≈ 0.0176 at 0.355 Å towards a nearest neighbor.

It is apparent that the circular and anti-symmetric center of mass, reminiscent of an atom's electric field, could in principle be approximated by a one-dimensional radial fit, or a scalar function.

3.1.2 Fitting the center of mass

In order to translate the direction and magnitude of a singular CoM-vector into a distance or displacement vector towards the nearest atom in real time, a viable, yet computationally efficient approximation to the inverse of the relation (1.19) from chapter 1.2.2 must be found.

Instead of deriving an explicit expression, for this project a function is fitted to the simulated center of mass line profile of a carbon atom. There are differences in CoM to graphene, with the latter having steeper gradients between neighboring atoms and no circular symmetry.

The circular symmetry of a single C atom, however, allows for a simple one-dimensional fit of the CoM to real-space distances. Using a transformation to polar coordinates, CoM-vectors can be represented by a magnitude CoM_r and an orientation CoM_φ:

$$\mathrm{CoM}_r = \sqrt{\mathrm{CoM}_x^2 + \mathrm{CoM}_y^2}\,, \tag{3.1}$$

$$\mathrm{CoM}_\varphi = \arctan\left(\frac{\mathrm{CoM}_y}{\mathrm{CoM}_x}\right)\,, \tag{3.2}$$

with CoM_x and CoM_y being components of a CoM vector. In the plugin's code, the arctangent is implemented using numpy's `arctan2` function.

The reference data for all fits consists of the CoM_y values and corresponding y coordinates that were obtained from the single C line-scan simulation. This $1001\mathbf{x}2$ long list is first shifted by -2 Å in y coordinates, to position the carbon atom at the real space origin. Then, a section spanning from the CoM_y maximum at -0.444 Å to the origin is selected to allow for a bijective function to be fit, at the cost of ignoring the decreasing values further away from the atom. Lastly, because the real space coordinates of this section are negative as the coordinates progress towards the atom, they are inverted for the fitting process. This way, the fitted function becomes strictly positive and mappable to the magnitude of the CoM's polar representation, CoM_r.

Two different methods were used to fit functions in this thesis, numpy's `polynomial.polynomial.Polynomial.fit` and scipy's `optimize.curve_fit`. At their core they perform a linear and a non-linear least-squares parameter estimation respectively. Notable utilized options are `Polynomial.fit`'s `domain` parameter, which was set to $[-1, 1]$ due to more desirable outcomes at the time, and `curve_fit`'s initial guess vector `p0`.

The model functions and their parameters are set up as follows. Polynomial model function of order 6:

$$a + b \cdot x + c \cdot x^2 + d \cdot x^3 + e \cdot x^4 + f \cdot x^5 + g \cdot x^6 \,, \tag{3.3}$$

Square root model function:

$$a + b \cdot \sqrt{\left| \frac{c + x}{d} \right|} \,, \tag{3.4}$$

Arbitrary root model function:

$$a + b \cdot \left| \frac{c + x}{d} \right|^e \,, \tag{3.5}$$

Hyperbolic arc-tangent model function:

$$a + b \cdot \operatorname{arctanh} \left[(x + c) \cdot d \right] \,. \tag{3.6}$$

During first attempts at fitting, the input data was defined as a function mapping real space to radial Center of Mass $x \mapsto \mathrm{CoM}_r$. Fitting polynomials of 2nd and 3rd degree this way produces good agreement with CoM near the atom. The goal however is to find a function $g(\mathrm{CoM}_r)$ that maps CoM to displacement relative to the closest atom $\mathrm{CoM}_r \mapsto x$. Subsequent attempts of 2nd and 3rd degree polynomial fits using both fitting methods but with an inverted input function (CoM and x swapped), showed much worse results. Thus, polynomial fitting attempts continued up to the 6th degree.

A few other model functions were selected for `curve_fit`-method fitting attempts, such as the square root, an arbitrary root with the exponent itself being a fitting parameter, and a hyperbolic arc-tangent. A full list of starting and fitting parameters can be seen in Table 3.2.

To compare the accuracy of all fits against each other, two sets of plots showing the absolute and relative errors with respect to the reference function $\mathrm{CoM}_r(x)$ were drawn. The absolute errors $\Delta\mathrm{CoM}$ and Δx are calculated as follows:

$$\Delta\mathrm{CoM} = f(x) - \mathrm{CoM}_r(x) \,, \tag{3.7}$$

$$\Delta x = g\left(\mathrm{CoM}_r(x)\right) - x \,. \tag{3.8}$$

Here $f(x)$ and $g(\mathrm{CoM}_r(x))$ are functions fitted to the reference function $\mathrm{CoM}_r(x)$, whereas $f(x)$ maps $x \mapsto \mathrm{CoM}_r$ while $g(\mathrm{CoM}_r(x))$ maps $\mathrm{CoM}_r \mapsto x$. Dividing the absolute error by the original data results in the relative errors $\delta\mathrm{CoM}$ and δx of the fitted functions f and g:

$$\delta\mathrm{CoM} = \frac{\Delta\mathrm{CoM}}{\mathrm{CoM}_r(x)} = \frac{f(x) - \mathrm{CoM}_r(x)}{\mathrm{CoM}_r(x)} = \frac{f(x)}{\mathrm{CoM}_r(x)} - 1 \,, \tag{3.9}$$

Table 3.2: Obtained fitting parameters

model function	Mapping direction	fitting method	starting parameters	a	b	c	d	e	f	g
polyn. 2nd deg.	$x \rightarrow \mathrm{CoM}_r$	Polynomial.fit	-	-3.753E-04	7.309E-02	-7.679E-02				
polyn. 3rd deg.	$x \rightarrow \mathrm{CoM}_r$	Polynomial.fit	-	-7.008E-05	6.465E-02	-2.907E-02	-7.164E-02			
polyn. 2nd deg.	$\mathrm{CoM}_r \rightarrow x$	curve_fit	(1,1,1)	2.378E-02	3.889E+00	1.093E+03				
polyn. 3rd deg.	$\mathrm{CoM}_r \rightarrow x$	Polynomial.fit	-	-1.385E-02	2.942E+01	-2.447E+03	1.314E+05			
polyn. 6th deg.	$\mathrm{CoM}_r \rightarrow x$	Polynomial.fit	-	4.609E-03	6.031E-01	1.095E+04	2.938E+06	3.660E+08	2.113E+10	4.616E+11
polyn. 6th deg.	$\mathrm{CoM}_r \rightarrow x$	curve_fit	(0,0,0,0,0,4.616e11)	-5.852E-03	2.910E+01	-5.805E+03	9.316E+05	-5.134E+07	-2.931E+02	5.597E+10
Square root	$\mathrm{CoM}_r \rightarrow x$	curve_fit	(0,0,-1.671e-2,1)	4.512E-01	-3.277E+00	-1.673E-02	9.224E-01			
Arbitrary root	$\mathrm{CoM}_r \rightarrow x$	curve_fit	(0,0,-1.671e-2,1,2)	4.243E-01	-1.503E+01	-1.671E-02	6.028E+00	6.039E-01		
arctanh	$\mathrm{CoM}_r \rightarrow x$	curve_fit	(0.1,1,-0.0001,1/0.02)	9.866E-02	1.988E-01	-5.740E-03	8.457E+01			

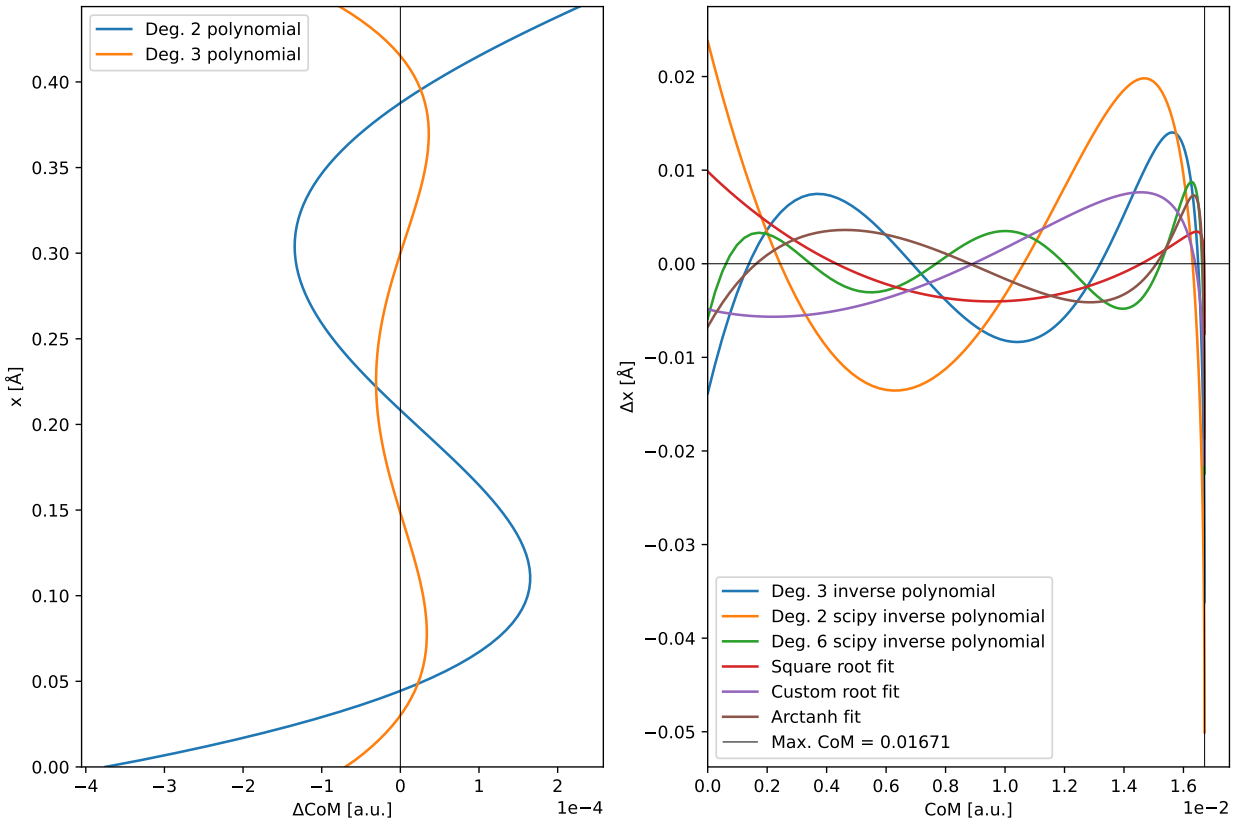

Figure 3.7: Absolute fitting error with respect to reference dataset.

$$\delta x = \frac{\Delta x}{x} = \frac{g\left(\mathrm{CoM}_r(x)\right) - x}{x} = \frac{g\left(\mathrm{CoM}_r(x)\right)}{x} - 1\,. \tag{3.10}$$

Based on the absolute (Figure 3.7) and relative (Figure 3.8) errors of various fitting functions, the 6th degree inverse polynomial was chosen to be used for drift tracking.

Figure (3.9) shows the selected fitting function and the fitting area in red, defined by the fitting boundaries: $x \in [0, 0.444]$Å, $|\mathrm{CoM}_r| \in [0, 0.01671]$. It also shows the CoM magnitude of a simulated single carbon atom, close to the atomic site. It is important to note that applying this polynomial to a CoM magnitude beyond the fitting area results in asymptotic behavior, strongly deviating from the rest of the simulation. This makes computational safe-guards necessary.

Implementation of the chosen fit into code was done by hard-coding the fitting parameters into a function `poly6inv(com)` that calculates a 6th degree polynomial when called. At the time of writing, the safe-guard consists of a magnitude normalisation to 1, as the decision seen in flowchart (3.21), and an additional normalization to the upper input boundary of 0.01671 inside of the polynomial function.

Thanks to the approximation of radial symmetry used for this fit, the center of mass angle

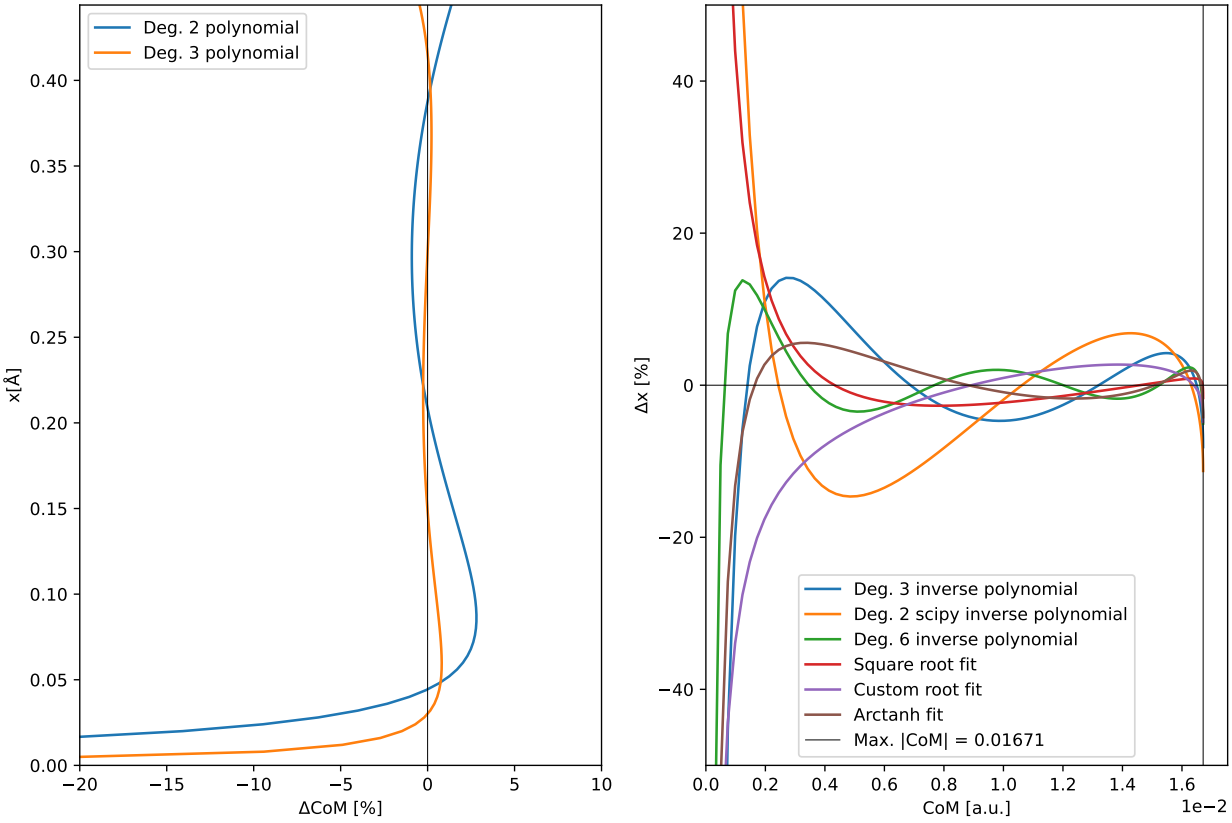

Figure 3.8: Relative fitting error with respect to reference dataset.

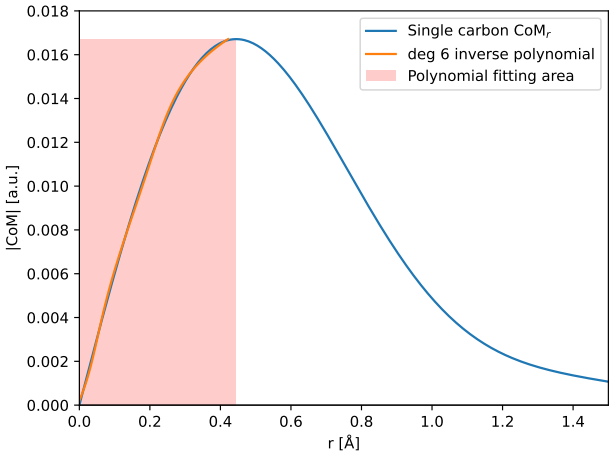

Figure 3.9: Single carbon radial CoM and fitted 6th degree inverse polynomial with indicated fitting area.

is left unchanged and mapped directly to a real-space orientation $\mathrm{CoM}_\varphi = \varphi$. This is valid as long as there is no QR-rotation present (see Chapter 2.1.4). The distance x and the orientation angle φ are then transformed into a cartesian displacement vector:

$$\vec{s} = \begin{pmatrix} x \sin \varphi \\ x \cos \varphi \end{pmatrix} . \tag{3.11}$$

The fitted function can now be used to perform an algorithm that seeks out the closest atomic site based on its projected electric field.

3.1.3 Tracking procedure

The previous chapter outlined the process of fitting a function $\mathrm{CoM}_r \mapsto x$ and combining it with the untouched orientation CoM_φ to a displacement vector $\vec{s}$. This allows us to approximate the position of a carbon atom relative to our STEM's electron probe if it were a single atom fixed in space.

Obviously, graphene's projection of the electric field is determined by the lattice and it does not exhibit a circular symmetry. Additionally, the stage drift to be corrected is constantly moving the goal post, so to speak, making an iterative approach necessary.

However, given that atomic sites in a 4D dataset can be found by following the direction of CoM vectors to a minimum position (see chapter 3.1.1), our fitted function may still be sufficient to reach a nearby site after a reasonable amount of probe displacements. The retention of the CoM orientation is essential to this approach. It ensures that any applied change to the probe position progresses towards a lattice site.

The iterative nature of the drift correction method proposed here can best be described as a loop. Every iteration starts by acquiring a CBED pattern at the current probe position $\vec{r_p}$. The resulting pattern object contains the 2D intensity data and its reciprocal space calibration, which consists of scale, offset and units. The latter are then used to span the coordinate vectors $\vec{k_x}$ and $\vec{k_y}$.

Next, the data is fed into the center of mass equation (1.18). The result of this calculation is the 2D CoM vector $\vec{I_{com}}$. When used with the total intensity normalisation $\frac{1}{I_{tot}}$, its units coincide with the input pattern's reciprocal-space units.

The fitted displacement function from chapter 3.1.2 is then used to assign a real space 2D

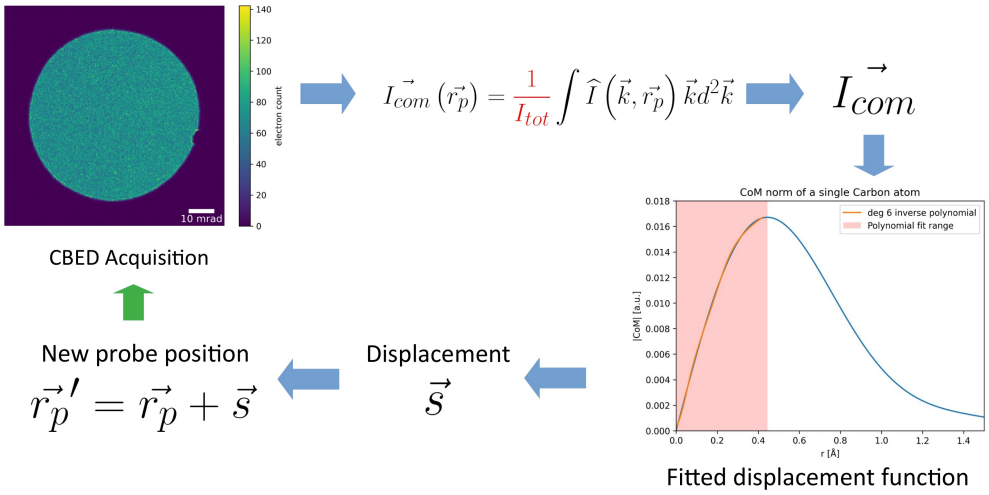

Figure 3.10: The iterative drift correction scheme.

displacement vector $\vec{s}$ to the CoM vector. Note that doing so requires transforming the center of mass to a polar coordinate system and back to Cartesian after applying the function to the radial component.

The last step is then to apply this displacement. This is done by calculating the new probe position $\vec{r_p}'$ by adding $\vec{s}$ to $\vec{r_p}$ and instructing the microscope probe to move there using its beam-shifter control values.

This approach is reminiscent of the gradient descent method, in which the gradient of a scalar field is calculated to follow it and find a minimum. There are two key differences however. First, the gradient in question being the projected electric field, it is represented by the center of mass of diffraction patterns instead of an explicit gradient operation. Secondly, the step size is determined though an educated guess, thanks to the center of mass fit from the previous chapter.

3.2 Testing on simulated data

To verify that the tracking procedure introduced in the previous chapter works as intended, it is first tested using a simulation of graphene and its center of mass, as described in chapters 3.1 and 3.1.1.

A Python function was written to test the proposed scheme in Figure (3.10) on simulated data. Figure (3.11) shows a flowchart of this function. The flowchart is colored based on the type of information being processed at each step. Red indicates a manual input values,

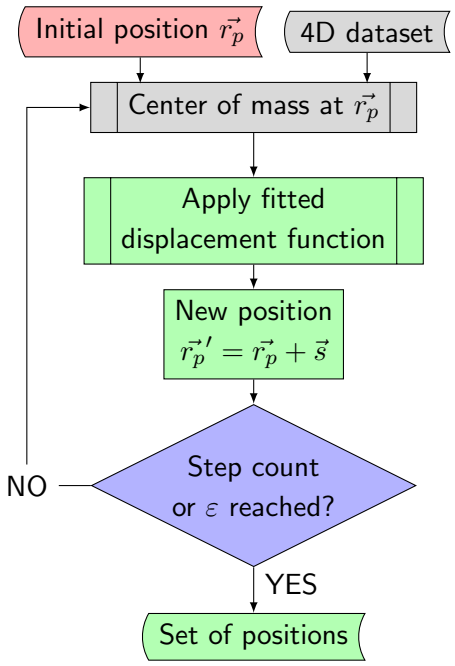

Figure 3.11: Flowchart of the drift tracking simulation function.

gray stands for input or center of mass data and green describes displacement vectors, positions and their generation using the established fitted displacement function. Finally, blue elements describe a decision or actions in connection with said decision.

The tracking simulation function requires an existing 4D dataset or a corresponding CoM-vector array, in this case generated using the aforementioned *ab*TEM graphene simulations, and an initial position within the real-space field of view of this dataset. It utilizes a loop that ends after a set amount of iterations. Alternatively, setting a threshold value ε for the displacement magnitude $|\vec{s}|$ has also been also tried, with the loop ending once $|\vec{s}| < \varepsilon$. Practicality of this convergence threshold was limited, however, due to the discrete nature of the data.

Inside the loop, first the center of mass is either calculated or selected at the dataset indices closest to the position $\vec{r_p}$. Then the displacement $\vec{s}$ is found by applying the displacement function from chapter 3.1.2 according to the procedure established in chapter 3.1.3. The new position $\vec{r_p}'$ is fed back to the start of the loop, if the loop's condition has not been met. Once the loop ends, a list of all simulated probe positions is returned for plotting and further analysis.

During these tests, a step-size multiplier was introduced to potentially speed up tracking convergence and counteract the diminishing step-size for probe positions that are further away from an atom site than the fitting function's boundary of 0.44 Å (see chapter 3.1.2).

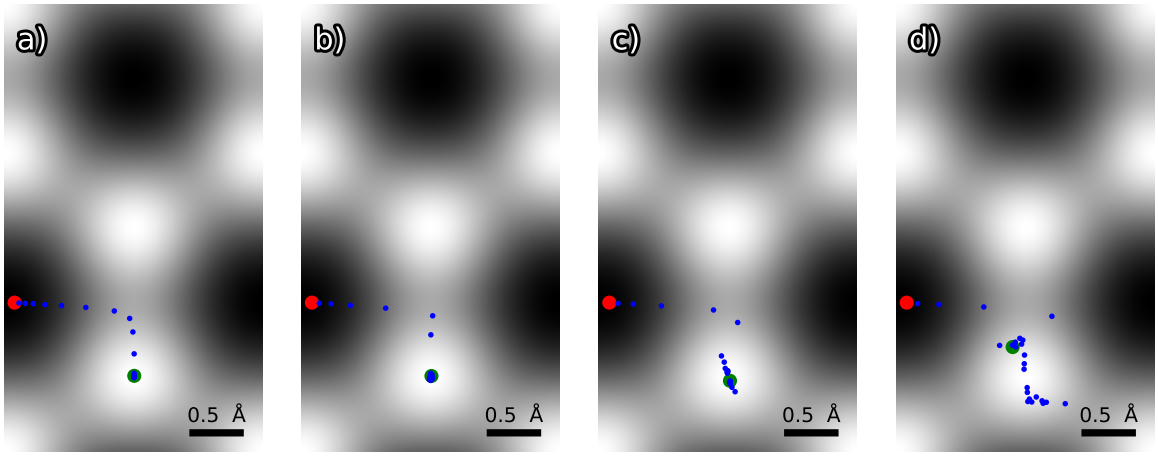

Figure 3.12: Tracking simulations using version 3 CoM, interpolated to a scale of 0.0123 Å, with step-size multipliers of a) 1.2, b) 2.0, c) 2.4, d) 3.0 .

This Multiplier is a scalar that applies to the approximated displacement $\vec{s}$:

$$\vec{s}\,' = \text{mult} \cdot \vec{s}. \tag{3.12}$$

Figure (3.12) shows tracking simulations based on CoM data of a version 3 *ab*TEM graphene simulation (see chapter 3.1), interpolated by a factor of 10. The contrast is formed by the integrated center of mass, whereas starting positions are drawn in red, the last position in green, and steps in between in blue. The starting position was chosen to be $(0.1, 1.42)$ Å, near a hexagon center, as it represents one of the worst cases in terms of amount of required steps to reach an atomic site. The tracking simulation was limited to 25 steps: higher amounts only negligibly affected the plots.

The first panel a) shows a tracking simulation using a multiplier of 1.2, reaching the atomic site after 11 steps. In b) the multiplier was increased to 2.0, allowing the loop to reach the site after just 7 steps. We start to see signs of slight under-damping however, based on the step between two atomic sites landing a bit past the atomic site along x. With a further increase to 2.4, the tracking loop starts oscillating around the site and is now clearly under-damped. At a multiplier of 3.0 the oscillation has started to draw a pattern that depends on the scan sampling of the underlying CoM simulation. Increasing step count to 50 does not alter this pattern, it only shifts the final position.

At the time of writing, a conservative multiplier of 1.2 is implemented as default, and has been used for all in-situ tracking experiments.

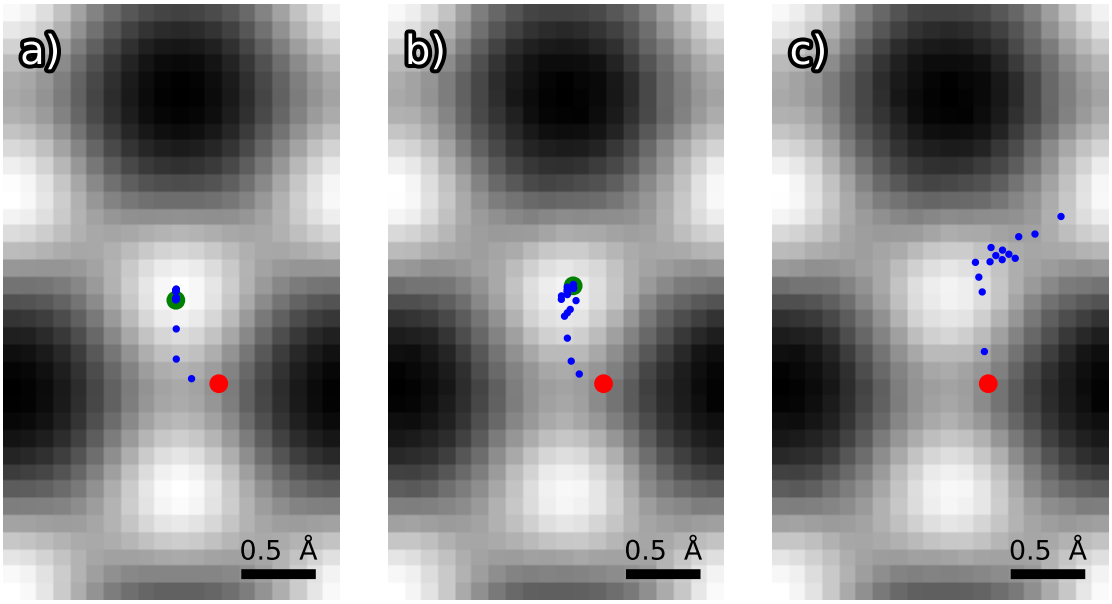

Figure 3.13: Tracking simulations using version 3 simulated CBED patters: a) unaltered, b) noisy CBED patterns, c) noisy and displaced by $(0.25, 0.25)$ mrad.

Testing with noise and a reciprocal-space offset

Drift tracking simulations were also attempted for more realistic *ab*TEM simulations featuring noise, aberrations or an offset CBED pattern. Overall the tracking algorithm seems to be resilient to uniform noise, but sensitive to factors that change a pattern's intensity distribution asymmetrically. The tracking simulations in Figure (3.13) also use a version 3 graphene simulation, but without interpolation. The contrast shown is thus the iCoM at a scan sampling of 0.123 Å. The input for the tracking loops this time are the simulated CBED patterns at each probe position, as opposed to pre-calculated CoM. The loops are run for 25 iterations, with a multiplier of 1.2, starting from the $(1.5, 1.5)$ Å position located between two atomic sites, but closer to the one in the center of the plot. A calibration routine from chapter 3.3.1 is used for a) and b) simulations, whereas c) uses b)'s calibration. This includes normalisation of CBED patterns to their total intensity I_{tot}.

Figure (3.13) a) shows expected tracking behaviour, converging on the closest atomic site. Perceived overshoot in this case is an artifact due to low scan sampling and the resulting discrete center of mass values. For the tracking simulation in b) Poisson noise is applied to the CBED patterns using the *ab*TEM `poisson_noise` method, with parameter `total_dose=1e3` chosen to make plots of the center of mass look close to that of recorded microscopy data (Figure 3.18). The iCoM contrast is visually unaffected by the addition of noise.

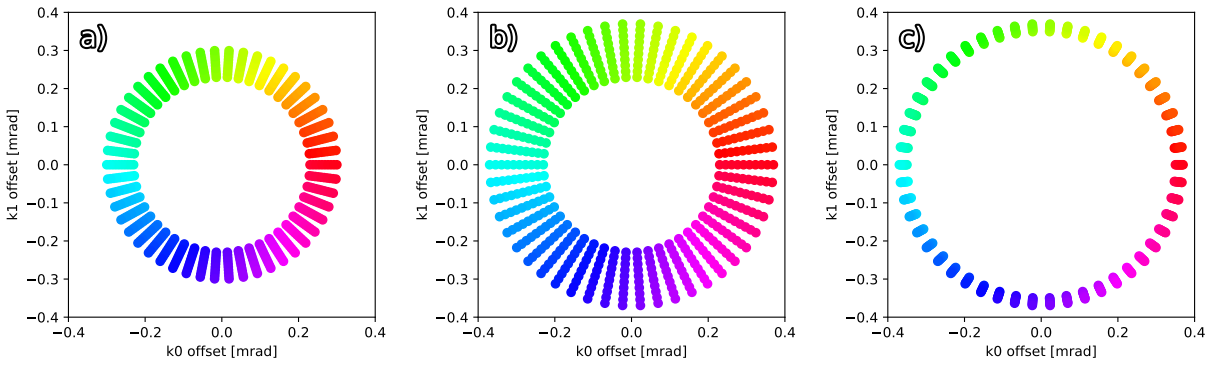

Figure 3.14: Color-coded offsets to be added to CoM vectors, representing a shift in BF disk position with different magnitude ranges: a) 0.23 to 0.30 mrad, b) 0.23 to 0.37 mrad, c) 0.35 to 0.37 mrad.

Tracking steps look more erratic with Poisson noise added, but still manage to reasonably converge on the atomic site.

For the tracking simulation in c), the elements of reciprocal coordinate vectors $\vec{k}_x$ and $\vec{k}_y$ used for calculating CoM from CBED patterns were increased by 0.25 mrad, equivalent to shifting the pattern by an offset $j_0 = (+0.25, +0.25)$ mrad. The result in tracking behaviour is a clear bias in direction of the offset. A denser group of positions to the upper right of the targeted atomic site becomes visible, indicating a dislocated CoM minimum by the pattern offset. At this offset magnitude, the majority of CoM vectors point in direction of the offset, and the roughly circular symmetry required for tracking is drowned out.

To further test how the tracking loop behaves when an offset $\vec{j_0}$ is present, three sets of offsets were chosen and color-coded, as visualised in Figure (3.14). Each set spans 50 equidistant angles and 10 magnitudes for a total of 500 offset vectors. Set a) contains CoM magnitudes from 0.23 mrad to 0.30 mrad, set b) from 0.23 mrad to 0.37 mrad and set c) from 0.35 mrad to 0.37 mrad.

The simulated 4D datasets and iCoM without Poisson noise, used for the loops in Figure (3.13) a), were then tiled by a factor of 3 along x and 2 along y to increase the field of view. A tracking loop was run for 40 steps for each of the offset vectors. Instead of plotting each step, only the final step is shown as a dot in the color assigned to the corresponding offset (Fig. 3.15). The starting position remains $(1.5, 1.5)$ Å, marked by a black triangle and the original cell extent is marked by a red rectangle. Final positions that lie in the tiled areas outside of the original cell are counted.

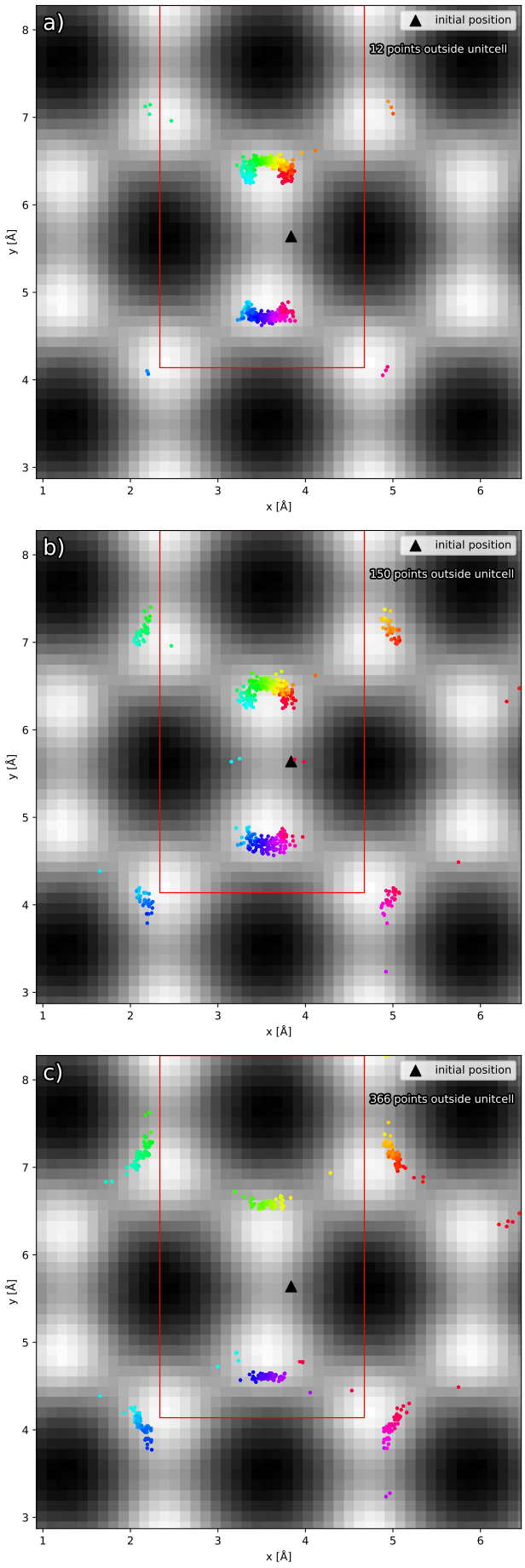

Figure 3.15: Final positions of tracking simulations after 40 steps with color coded offsets added to CoM vectors with offset magnitude ranges: a) 0.23 to 0.30 mrad, c) 0.23 to 0.37 mrad, e) 0.35 to 0.37 mrad.

Figure (3.15) a) shows the final position of loops using the first set of offset vectors. They are predominantly distributed around the two atoms closest to the starting point in a slightly rectangular shape due to scan sampling. While most of the loops can be considered converged after 40 steps, about 2.5% of loops have wandered off to neighboring atoms. Increasing the upper disk shift magnitude from 0.30 mrad to 0.37 mrad in Figure (3.15) b) leads to 30% of final positions missing the designated cell. Additionally, a few loops have progressed to even further away atoms. Raising the lower disk shift magnitude from 0.23 mrad to 0.35 mrad in c) results in most loops missing the target atoms, about 73% of all loops.

It can be concluded that a shift of the CBED pattern in reciprocal space introduces an offset $\vec{j_0}$ that translates to a bias direction reflected by displacement vectors generated from these patterns. For conditions as seen in version 3 of the above *ab*TEM graphene simulations, offset magnitudes need to stay below 0.3 mrad for the tracking loop to remain stable. Thanks to the normalisation used on diffraction patterns, this offset tolerance should be reasonably representative of real in-situ experiments with a setup similar to the *ab*TEM simulation.

3.3 Testing on recorded 4D-STEM data

Simulated diffraction patterns are idealised approximations of their real counterparts. And while additional Poisson noise might make it look realistic, there may still be various other factors affecting a CBED recorded on a microscope that differentiate it from a simulation.

To test for such differences and eventually come up with a calibration routine, one previously recorded graphene 4D dataset from the Nion UltraSTEM 100 dated to 29th of March 2023 (Fig. 3.16), was first provided by the supervisor. A second dataset dated to the 22nd of October 2023 (Fig. 3.17) was provided shortly after the first few in-situ tests. Both datasets are recorded using the Orca CCD camera. The former is enhanced using multi-frame 4D data acquisition [61], meaning it was made from multiple 4D datasets, increasing signal to noise ratio and alleviating scanning artifacts as well as correcting for drift. Additional datasets were recorded later, as part of in-situ testing.

The testing procedure is almost identical to testing on simulated data. A recorded 4D dataset is loaded and a tracking loop is run for a specific or random real-space starting point. The steps are then plotted on top of an iCoM phase reconstruction image of the dataset to check tracking behaviour.

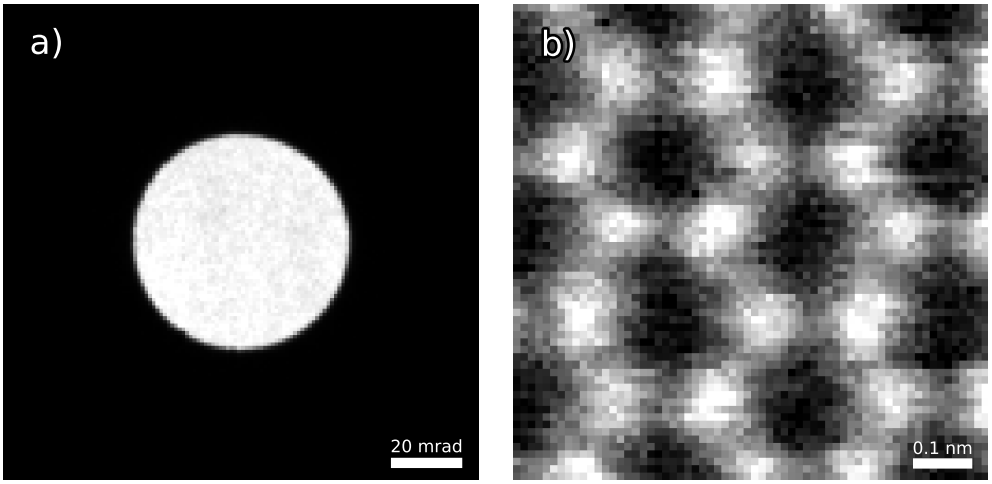

Figure 3.16: a) CBED pattern and b) ADF image using a roughly 32 to 61 mrad virtual annular detector of the enhanced graphene 4D dataset recorded on 29.3.2023.

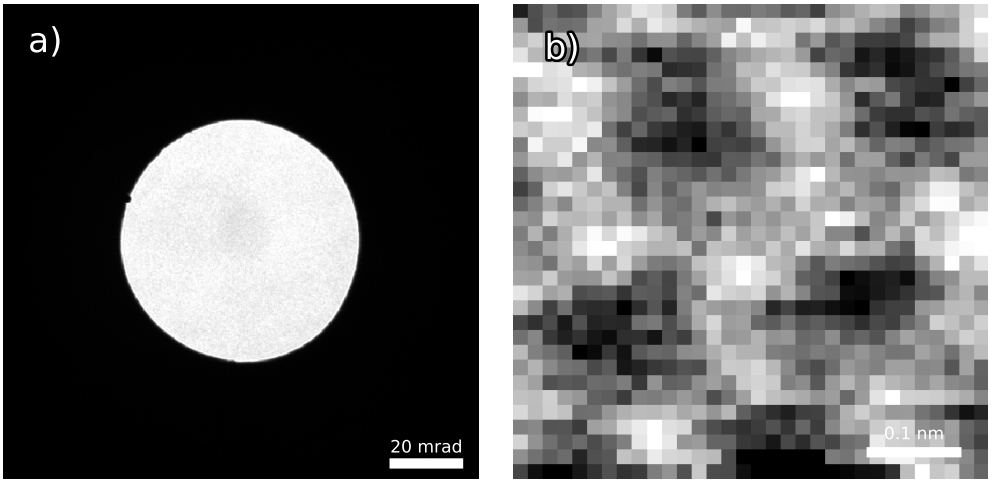

Figure 3.17: a) CBED pattern and b) ADF image using a roughly 32 to 61 mrad virtual annular detector of the graphene 4D dataset recorded on 29.10.2023.

New however is the requirement of a calibration step before the tracking loop. This is done on one hand to make the data compatible with the tracking algorithm, and on the other to calculate a valid phase reconstruction using the iCoM method. This process will be outlined in the next chapter (3.3.1).

3.3.1 Center of mass and integrated center of mass

Up until now, most center of mass calculations were conducted without the total intensity normalisation from equation (1.18). Proceeding that way with realistic datasets results in "raw" CoM images, as seen in Figure 3.18. Its magnitude a) is on the order of 10^4 to 10^5

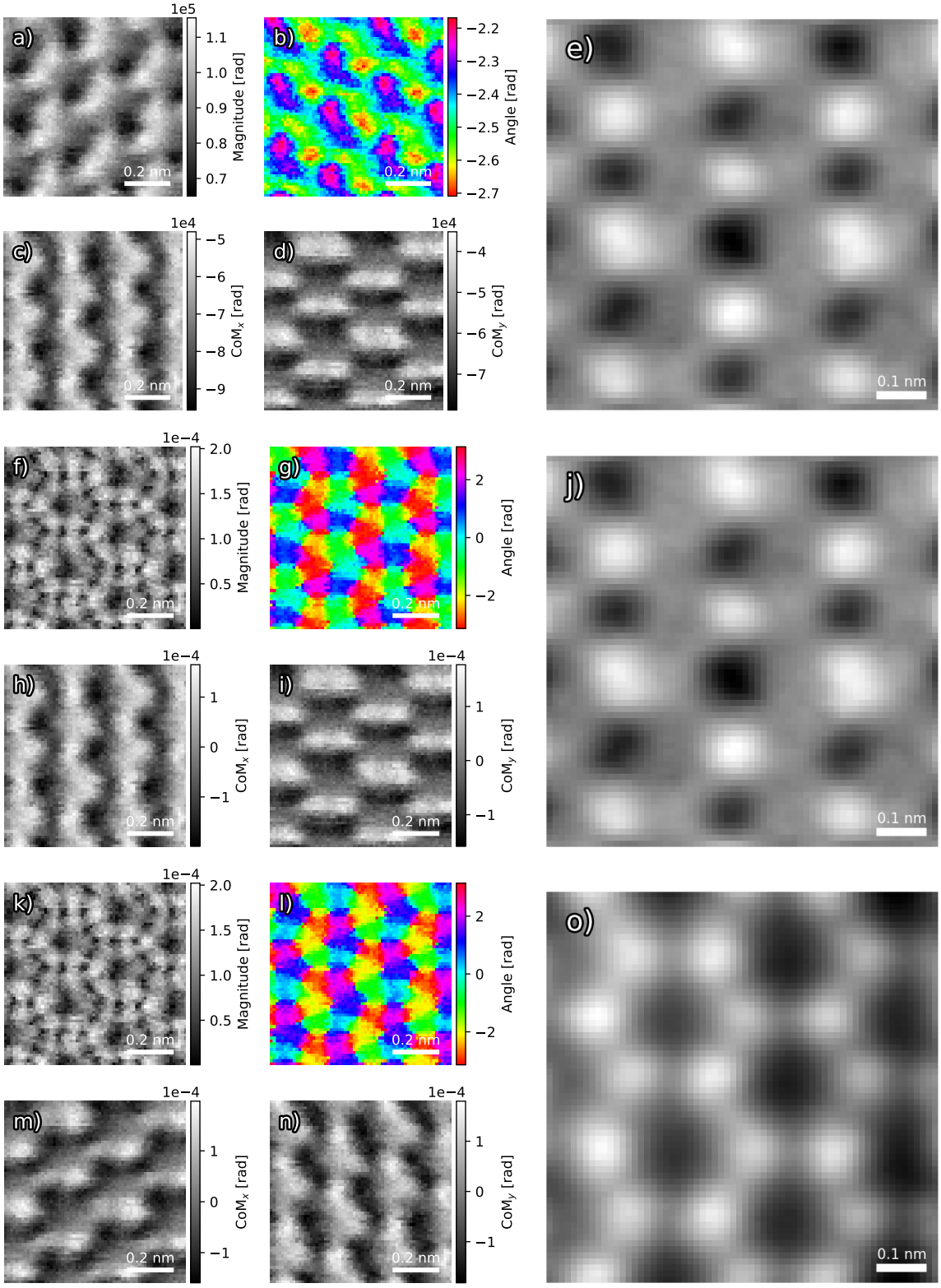

Figure 3.18: a)-d) Untreated center of mass of the enhanced graphene 4D dataset dated to 29th of March 2023, represented by its magnitude (upper left), angle (upper right) and its vector components (bottom). e) iCoM reconstruction attempt of the corresponding phase shift. f)-i) Center of mass centered, normalised to maximum magnitude and sum of CBED intensities. j) iCoM reconstruction attempt using normalised center of mass. k)-n) Center of mass with QR-rotation correction applied in addition to normalisations. o) Successful phase shift reconstruction due to QR-rotation correction.

in this particular case, due to the dependence on measured CBED pattern intensities. The argument or CoM angle b) exhibits a narrow range of roughly -2.7 to -2.2 radians and the components CoM_x c) and CoM_y d) only have negative values. There is no obvious way to interpret these values as a vector field pointing to atomic sites. The iCoM phase reconstruction in e) also fails.

To prepare the dataset for a tracking simulation, first the center of mass is normalised according to eq. (1.18) with:

$$I_{tot}(\vec{r_p}) = \sum_{i=1,j=1}^{n,m} I_{i,j}(\vec{r_p}) \quad , \tag{3.13}$$

where $I_{i,j}(\vec{r_p})$ are the intensities of a CBED pattern at probe position $\vec{r_p}$ and $I_{tot}(\vec{r_p})$ the total CBED pattern intensity. A properly normalised center of mass is no longer affected by total intensity variations between different diffraction patterns and can be written in units of wave vectors k, typically milliradians.

Next, to approximate a usable center of mass proportional to a projected electric field, the offset $\vec{j_0}$ from equation (1.19) is estimated. This is done by subtracting the arithmetic mean value of the components CoM_x and CoM_y from themselves. I will refer to this stage as "centering", as it has the effect of setting the new mean to zero:

$$\mathrm{CoM}_{x,i,j}^{\mathrm{centered}} = \mathrm{CoM}_{x,i,j} - j_{0,x} \approx \mathrm{CoM}_{x,i,j} - \frac{1}{n \cdot m} \sum_{i=1,j=1}^{n,m} \mathrm{CoM}_{x,i,j} \quad , \tag{3.14}$$

with $\mathrm{CoM}_{x,i,j}$ being values of the x-component of a 4D dataset's center of mass, $j_{0,x}$ the x-component of the real offset, n, m the dimensions of array, with $n = m$ for quadratic grid-scans and $\mathrm{CoM}_{x,i,j}^{\mathrm{centered}}$ the new values, with 0 as their new mean value.

Figure (3.18) f)-i) shows the result of these two operations. The magnitude f) now shows a clover-leaf pattern reminiscent of graphene's electric field and the CoM from simulations (Fig. 3.4). And the angle g) spans values from $-\pi$ to π radians, allowing for all CoM vector directions. Looking at the components h) CoM_x and i) CoM_y however, we can see that their value ranges have a middle close to, but not quite at 0. This hints at a non-symmetrical distribution of values, which can be visualised using histograms.

The last step is to perform an QR-rotation correction as described in chapter (2.1.4) to align the center of mass vector orientations with the real space structure. Doing so will also allow iCoM phase reconstruction to produce a correct image (Fig. 3.18) o) and thus an

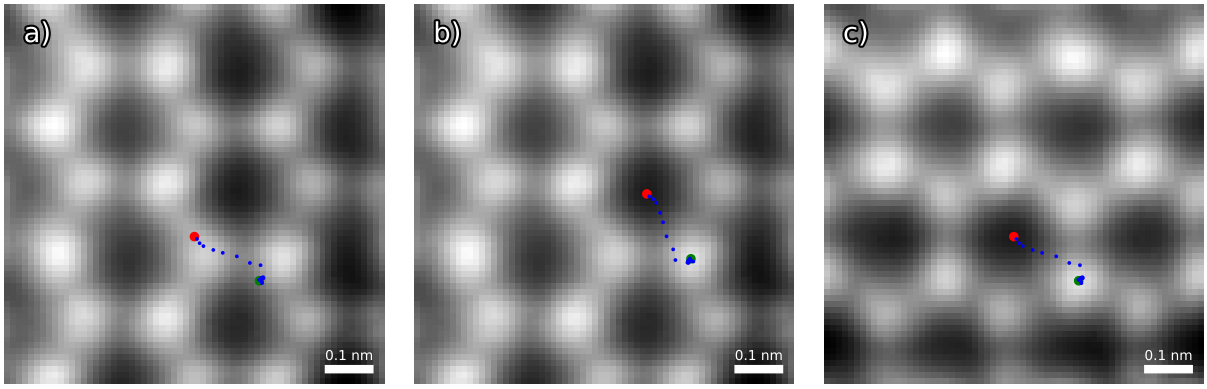

Figure 3.19: Drift tracking simulation on enhanced recorded graphene data iCoM. a) Positions and image transposed relative to each other, b) position indices swapped, c) image transposed.

easy way to check if CoM is aligned. The rotation angle for this dataset was found to be $\theta_{min} \approx -112.3°$.

Because the QR-rotation angle using the contrast method has two degenerate solutions 180° apart, the phase reconstruction may appear inverted, with dark atomic sites and bright voids. In that case, both CoM and iCoM arrays are multiplied with a factor of -1, as was the case here. The QR-rotation angle also changes accordingly, with $\theta_{min,inv} = \theta_{min} + 180° \approx -112.3° + 180° \approx 67.7°$.

3.3.2 Tracking behaviour

With a recorded 4D dataset's center of mass properly calibrated, it is now possible to choose a starting position apply the tracking loop from the previous chapter (Fig. 3.11). As a starting position let us again choose a worst case scenario, close to a hexagon center. The real space coordinates of this microscopy dataset have an origin at the center of the iCoM image. So to start inside the middle-right hexagon the initial position is chosen as $(0, 0.9)$ Å. Looking at the result of a 20-step tracking loop in Figure (3.19) a) reveals a seemingly erroneous path, starting from an atomic site below the middle of the image, proceeding to a neighboring site and converging at its edge. Still, the path's shape is somewhat reminiscent of the tracking simulations from chapter (3.2).

Pixel coordinate system

Unlike in *ab*TEM, images recorded on the microscope are saved in and processed using the top-left origin pixel coordinate convention. This may be counter-intuitive to most people, as they are taught to use cartesian coordinates with a bottom-left origin during math class. In essence, the y-axis of the pixel coordinate system is flipped, now upside down, while the x-axis stays the same, but is often pictured at the upper border of a graph. Printing a numpy array containing the image, on the other hand, results in a correctly displayed matrix.

To select individual pixels stored in a 2D numpy array, you have to provide the row and column indices, in that order, e.g. `array(row,column)`. If we were to rename the indices according to the above axis convention, we would get `array(y,x)`. This also seems counter-intuitive, because in Cartesian geometry, coordinates of points are usually written using (x, y), as is the case for the Python plotting library matplotlib.

In short, all data recorded on the microscope is firstly arranged in the pixel coordinate system with a top-origin, inverted y-axis. And secondly, individual data points are addressable in a transposed manner $(y, x)_{\text{pixel}} = (x, y)^T_{\text{cart}}$ with respect to the cartesian coordinate system.

Back to the tracking simulation, there are now two options to correct the tracking path. Either the indices of the tracking step positions are swapped (Fig. 3.19 b), or the underlying image is transposed (Fig. 3.19 c). The former option has the advantage of the image being displayed as intended, and is thus preferred.

The drift-tracking plugin applies a swap of y and x indices (marked in code using `#[IS]` comments, meaning "index-swap") whenever real-space data points, such as calculated displacements, are logged, displayed and when new beam-shifter values are applied.

3.4 Application

The program written for this thesis takes the form of a Python plugin for the Nion Swift image-processing and electron microscope control software (see chapter 2.3). It is also contained within a Git-distributed version control system repository called CoMTracker' and is synchronised to the PNM group's forgejo server as it's origin repository. The plugin can be installed like a normal Python package, by pointing pip or conda install commands to its git

repository. At the microscope's User PC, it was installed using the pip's `-e` or `--editable` parameter, which places a symbolic link to the local repository instead of copying files into the destination folder.

Much of the code is written in a "learn as you go" approach, or taken from other pre-existing plugins written by people from the research group. As a side effect, a majority of the plugin's code, particularly the control code, is regrettably contained within a single class of the `gui_comtracking.py` file. Calibration and calculation routines on the other hand are usually found in `calib_standalone.py`. An older version of this file called `calib.py` used to contain these functions but required the object calling it to be passed as a parameter in order to access its variables. For ease of use, the functions in the most recent version ask for these variables explicitly. This way, `calib_standalone.py` can be imported into a Python notebook and its functions can be accessed directly for analysis after microscope sessions.

At the time of writing, the plugin consists of nine Python script files. The remaining six files are `acquisition.py` for an experimental multi-threaded image acquisition approach, `arithmetic.py` for arithmetic operations inside of GUI-textboxes, the mandatory `__init__.py` for nionswift plugins, `lib_comtracking_old.py` which is an unfinished attempt at separating control from GUI code, `lib_graph.py` for plotting functions and `lib_widgets.py`, a collection of nionswift GUI element functions, written by Jacob Madsen.

The pieces of code most integral to this thesis are `calibrate` and `com2disp`, both part of `calib_standalone.py`.

The most recent implementations of the calibration function `calibrate()` and displacement step function `com2disp()` are laid out in their respective flowcharts (Fig. 3.20 and Fig. 3.21). Orange-colored elements describe measurement and application of calibration values. The order in which calibration values are measured and applied was now set to be QR-rotation correction, charge inversion, centering and then a new normalisation step. This additional step normalises all CoM vectors to the maximum CoM magnitude encountered during calibration. Moreover, during in-situ experiments, this maximum magnitude is updated if a new measured CoM vector exceeds the established reference magnitude. This safe-guard, previously mentioned at the end of chapter (3.1.2), is intended to prevent magnitudes larger than 1 from entering the displacement function and operating outside of fitting boundaries.

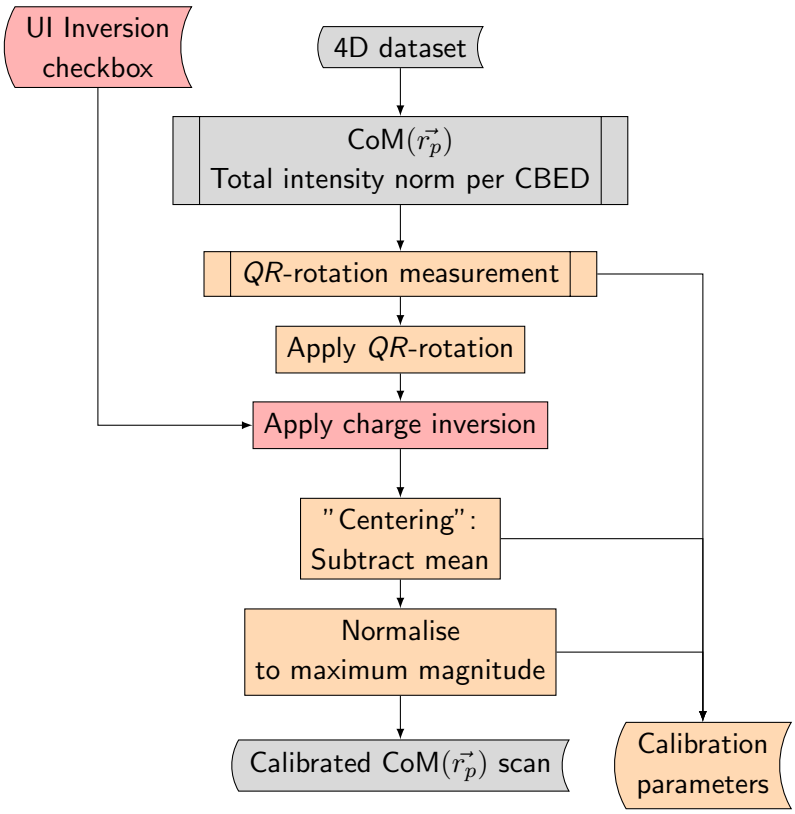

Figure 3.20: Flowchart of the calibration function **calibrate()**.

3.4.1 Plugin usage

The main way of interfacing with this nionswift plugin is through its own graphical user interface (GUI) panel, which can be toggled on in nionswift's `Window` tab under the name `CoM Tracking`. The plugin's graphical user interface (GUI) is using nionswift's UI, which in turn has a backend based on the Qt framework. All GUI elements of this plugin are defined in the `gui_comtracking.py` file.

A common tracking procedure using this plugin consists of the following steps. First, record a 4D scan of a graphene sample. Select the 4D data item and press the "Calibrate from 4D-Data" button. After a short delay, check the new "Calibration iCoM" item for validity. An example of an inverted and a successful iCoM image can be seen in Figure (3.23). If the iCoM image is inverted, toggle the "Invert calibration" check-box. If the image shows a wrong or distorted lattice, select a different *QR*-rotation method and try calibrating again.

The last step is to pick a starting point, stop the ADF scan and quickly park the probe at the desired spot using the scan control panel, followed by pressing "Start tracking". The tracking routine can be halted at any time using the same button, now showing "Stop tracking". The plugin stops tracking automatically if the shifter values exceed a norm of 10 nanometers.

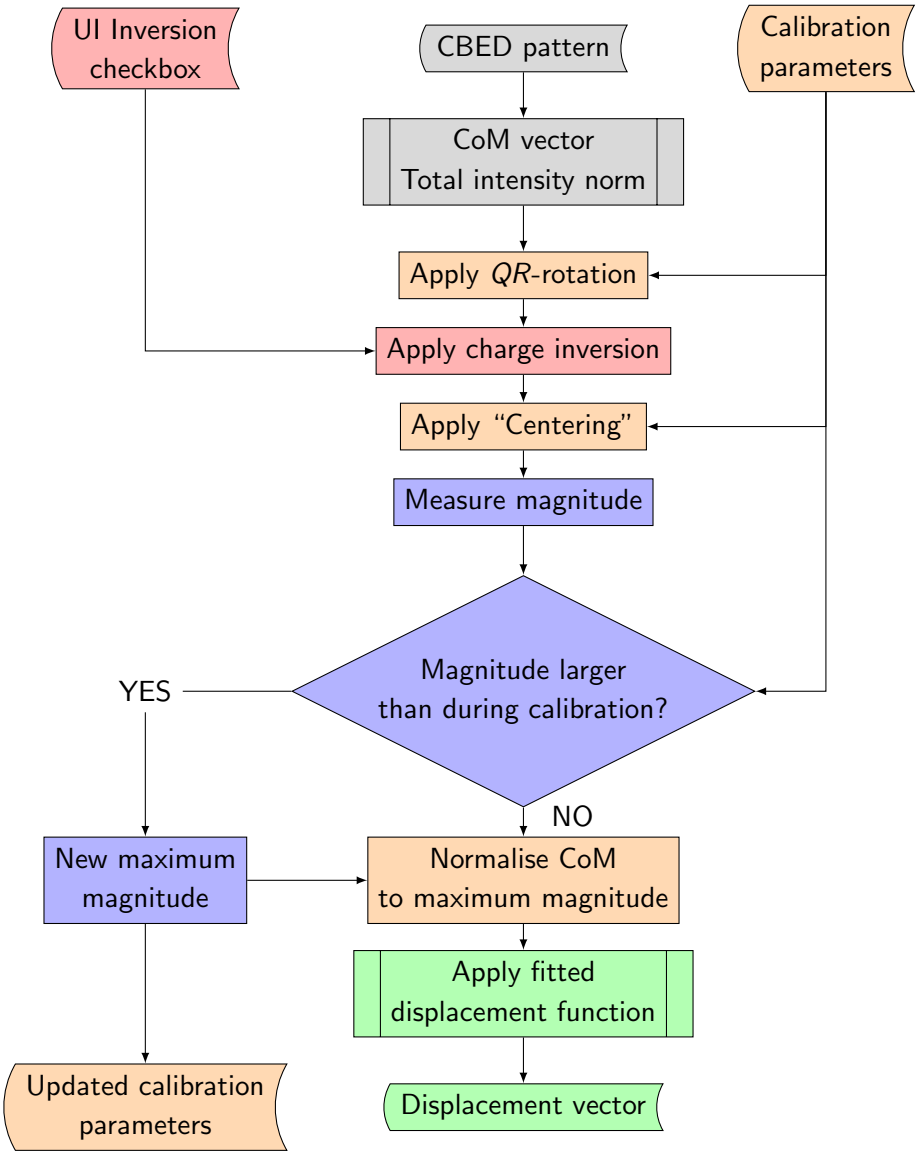

Figure 3.21: Flowchart of the displacement step function **com2disp()**.

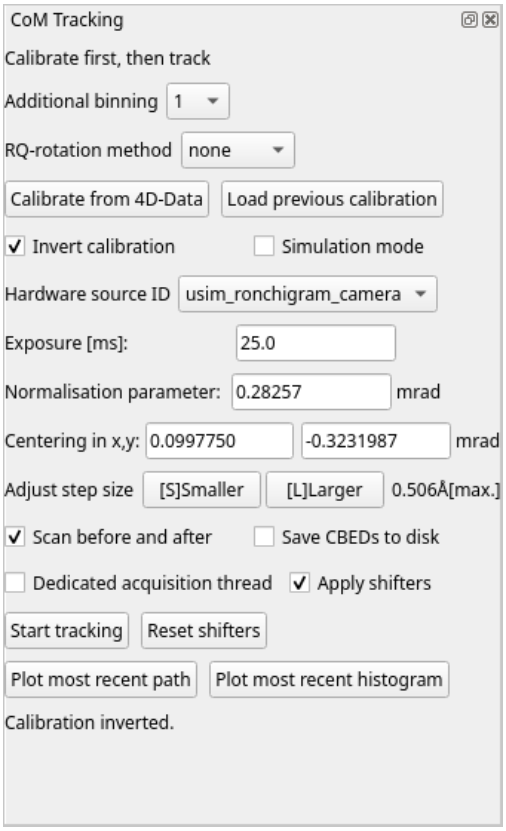

Figure 3.22: CoM tracking plugin GUI in nionswift.

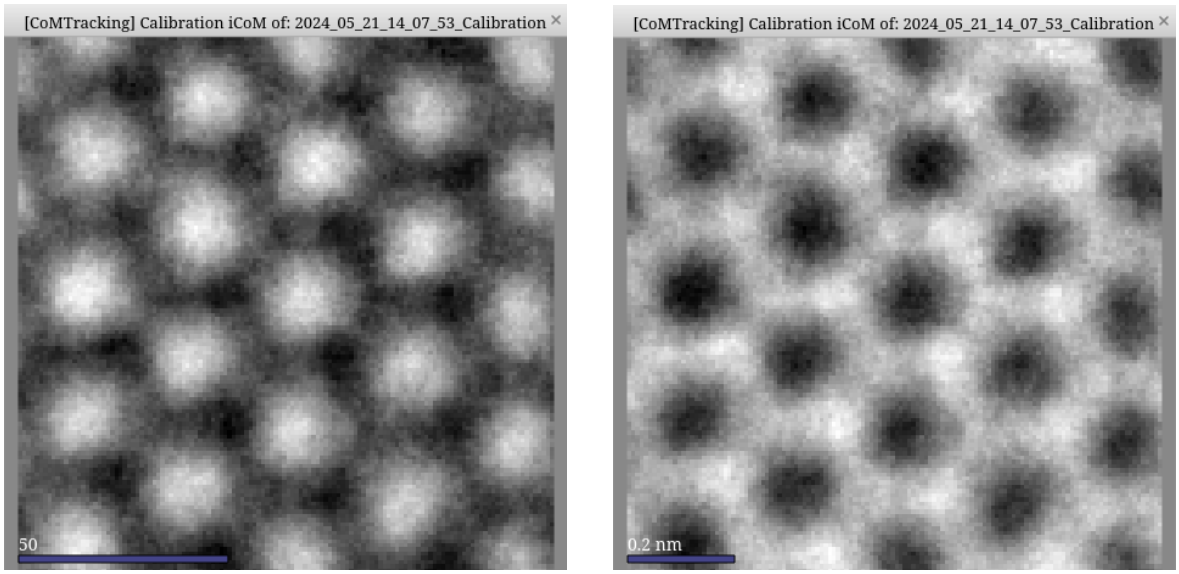

Figure 3.23: iCoM images generated by the plugin during calibration. Left: inverted iCoM, right: correct iCoM.

There are some additional features present on the GUI (Fig. 3.22), providing the following functionality:

- **Additional binning (optional)**: Bins down diffraction patters by a factor selectable from a list containing $1, 2, 4, 8, 16$ in order to improve performance. A specific use-case was improving calibration wait time when calibrating from Hamamatsu Orca 4D datasets with an QR-rotation method selected.
- ***QR*-rotation method (optional)**: Select a method of QR-rotation detection from a list containing "none", "curl" and "contrast" according to chapter 2.1.4.
- **Calibration buttons**: Either select a 4D data item from the nionswift UI and click "Calibrate from 4D-Data" to begin calibration or click "Load previous calibration" to load the last saved calibration parameters.
- **Invert calibration checkbox**: Check or uncheck in case the iCoM image calculated by calibration looks inverted. Atoms are supposed to look bright to indicate iCoM-maxima.
- **Simulation mode checkbox**: Check to perform simulated tracking steps, visualised by a moving colored pixel on the iCoM-data item in nionswift.
- **Hardware source ID**: Select the instrument id of the detector to be used for CBEDs during tracking. Must be the same detector that was used for recording the dataset used in calibration.
- **Exposure [ms]**: If found, presents the exposure parameter of the selected detector in milliseconds. Changing this value attempts to apply a new exposure value to the selected detector.
- **Normalisation parameter (diagnostic)**: Initially shows the longest CoM-vector length found during calibration in mrad. If larger values are encountered during tracking, the plugin will overwrite this value as to not exceed input limits of the fitted polynomial (see chapter 3.1.2). If this value deviates from the original calibrated value considerably, consider re-calibrating as it may indicate that the BF disk has drifted, introducing a bias-vector to CoM and drowning out the sample's signal.
- **Centering in x,y (diagnostic)**: Shows the components of the bias-CoM-vector measured during calibration in mrad. This is should ideally be equal to the bias-vector $\vec{j_0}$

that the BF disk introduces by being off-center. In the case that the BF disk drifts to a new position, it is advised to recalibrate rather than changing these values manually (though that has been tried before, with little success).

- **Adjust step size**: Pressing these buttons decreases and increases the default step size multiplier of 1.2 during tracking by a factor of 0.9. The maximum step size label to the right assumes a distance of 0.422 between the center of a carbon atom and its radial CoM-maxima, based on graphene simulations in chapter 3.1.1.

- **Scan before and after checkbox (diagnostic)**: Checking this will record appropriately named ADF-images before the start and after the end of tracking.

- **Save CBEDs to disk checkbox (diagnostic)**: Caches diffraction patterns during tracking and writes them to disks in stacks of 100 by default for later analysis.

- **Dedicated acquisition thread (experimental)**: Performs CBED acquisition during tracking using its own thread and two alternating buffers during tracking. This was originally meant to stabilize and decrease overall waiting times when consecutively acquiring new diffraction patterns. As this turned out to be ineffective, the issue was deemed to be caused by a matter outside the plugin.

- **Apply shifters checkbox**: Uncheck this to de-couple the tracking loop from the beam-shifters. The status labels at the bottom will still display CoM-vector values at the current probe position. These include CoM-vector components normalised to the above normalisation parameter, as well as the length and orientation of the CoM-vector after subtracting the above bias vector, in millirad and degrees respectively.

- **Start/Stop tracking button**: Starts and stops the tracking loop.

- **Reset shifters button**: Sets beam-shifter values to 0 in both dimensions, effectively returning to the original probe position. Pressing this button can be used to reveal the stage drift that was compensated during tracking.

- **Plot most recent path button**: Plots and shows the path drawn by the beam shifter positions during the most recent tracking loop. The output will be an XY plot with a red starting point and a green end point.

- **Plot most recent histogram button**: Plots and shows the 2D histogram of CoM vectors minus the bias vector from calibration.

3.4.2 Tracking testing

The nionswift plugin was developed over the course of over ten testing sessions at the University of Vienna's Nion UltraSTEM 100 microscope. A total of over 35 4D datasets were recorded of five graphene samples. The first four sessions used the Hamamatsu Orca CCD camera (chap. 2.2.1) and the last six sessions utilized the Dectris ARINA hybrid pixel detector (chap. 2.2.2) for CBED acquisition.

Table 3.3: Graphene specimens used in experiments.

Sample number	Description
1346	Graphenea easy transfer graphene on silicon nitride
1533	hematene on Graphenea on Quantifoil grid
1847	AB-stacked CVD-grown graphene on silicon/silicon oxide
1699	Graphenea on Quantifoil grid
932	Graphenea easy transfer graphene on silicon nitride

No new samples were prepared as part of this thesis. Instead, existing graphene samples kept in UHV garages of the CANVAS [53] environment and the Microscope's sample holder magazine were used to test the plugin. The samples used are listed in Table (3.3) in order of their frequency of use.

In-situ testing starts by recording a 4D dataset of a pristine monolayer area of the graphene sample. Based on this dataset, the plugin establishes calibration values and displays a iCoM phase reconstruction item in nionswift. Checking "invert calibration" may be necessary if it appears inverted.

Next, a monolayer area near static landmarks such as immovable dirt or stable impurity atoms is chosen. The intention is to record an HAADF scan image right before and after drift tracking experiments, to check whether the beam's position has successfully kept up with the specimen's drift. Without valid landmarks, all that is visible on these scans is graphene's periodic lattice and little can be said about the plugin's effectiveness. A HAADF detector is used for this purpose, as it can operate without obscuring the BF detectors. Anecdotally, switching between HAADF and MAADF detectors had caused considerable movement of the field of view, for at least one of the specimens.

Before starting the tracking experiment, any active ADF scan is stopped and the beam promptly positioned on top of a target atom. The tracking loop is then started via the plugin interface and run until either the plugin stops the loop, or it is stopped manually using the same button. The most recent version of the code implements a travel distance limit of 10 nm, equating to about 5 to 10 minutes of tracking time, before the loop stops by itself, depending on drift velocity. During tracking, labels on the bottom of the plugin UI display current values for CoM and displacement vectors. These values are logged as well. Pressing the plot path button returns a plots of the steps taken during ongoing tracking loop. This can be helpful in determining whether the loop should be stopped manually.

Once the tracking loop is stopped and an ADF-image is recorded, the field of view around the new area can be examined and compared to the one before tracking started. Additionally, the "Reset shifters" button can be used to move the field of view back to original position, and show where it would have ended up without any drift correction.

3.4.3 Experiment results

4D-STEM datasets

The following is a selection of 4D datasets that were used for calibration before drift tracking experiments. Figure (3.24) shows a) a CBED pattern, b) virtual ADF image, c)-f) center of mass and g) iCoM phase reconstruction of a dataset recorded of sample 1533 at a field of view of 0.5 nm, a real-space resolution of 32×32 and reciprocal-space resolution of 512×512 after $4\times$ binning, on the 8th of November 2023 at 13:25. Binning had a negligible impact on tracking accuracy, but sped up calibration significantly. The exposure of the ORCA camera was set to 50 ms for this dataset and the subsequent tracking experiments.

The CoM tracking plugin at the time did not normalise for total diffraction pattern intensities I_{tot}. It only normalised CoM vectors to a maximum magnitude, and only during calibration. This meant that the measured CoM contained a background contribution of up to 3% of its maximum magnitude after normalisation (Fig. 3.25). This type of accumulated noise may destabilise tracking, especially near atomic sites, where the CoM magnitude is at its lowest.

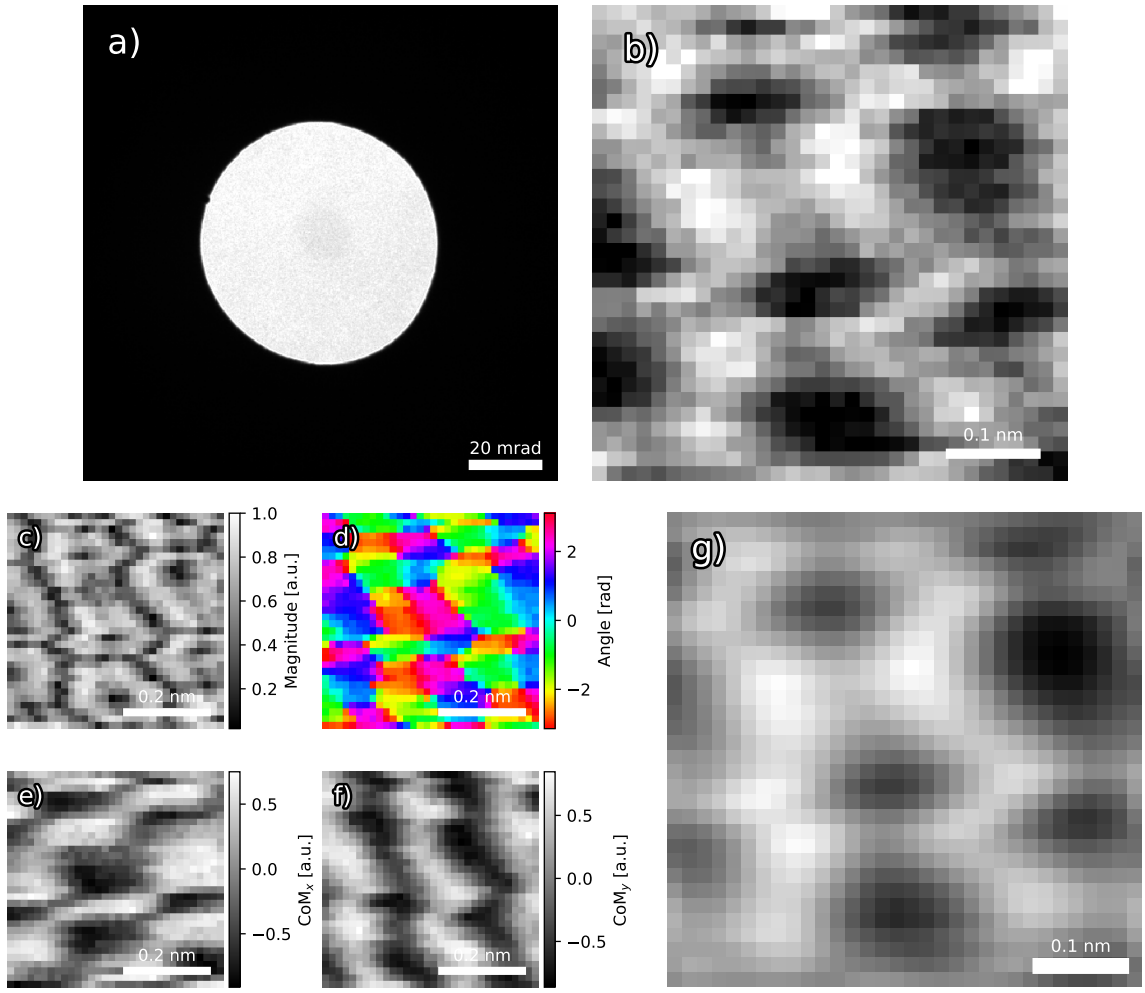

Figure 3.24: a) CBED pattern, b) ADF image using a roughly 34 to 63 mrad virtual annular detector, c)-f) CoM and g) iCoM images of the graphene 4D dataset recorded on 08.11.2023.

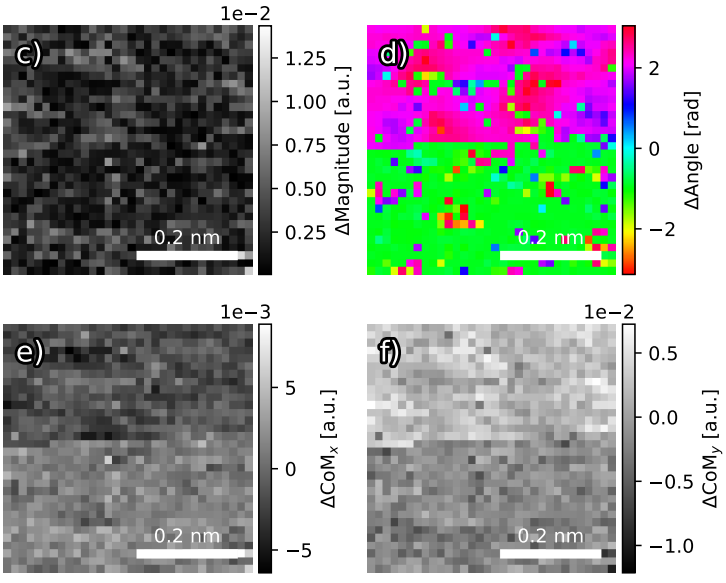

Figure 3.25: Residual CoM after subtracting the center of mass as calibrated using the I_{tot} normalised code, from the one calibrated for tracking on 08.11.2023.

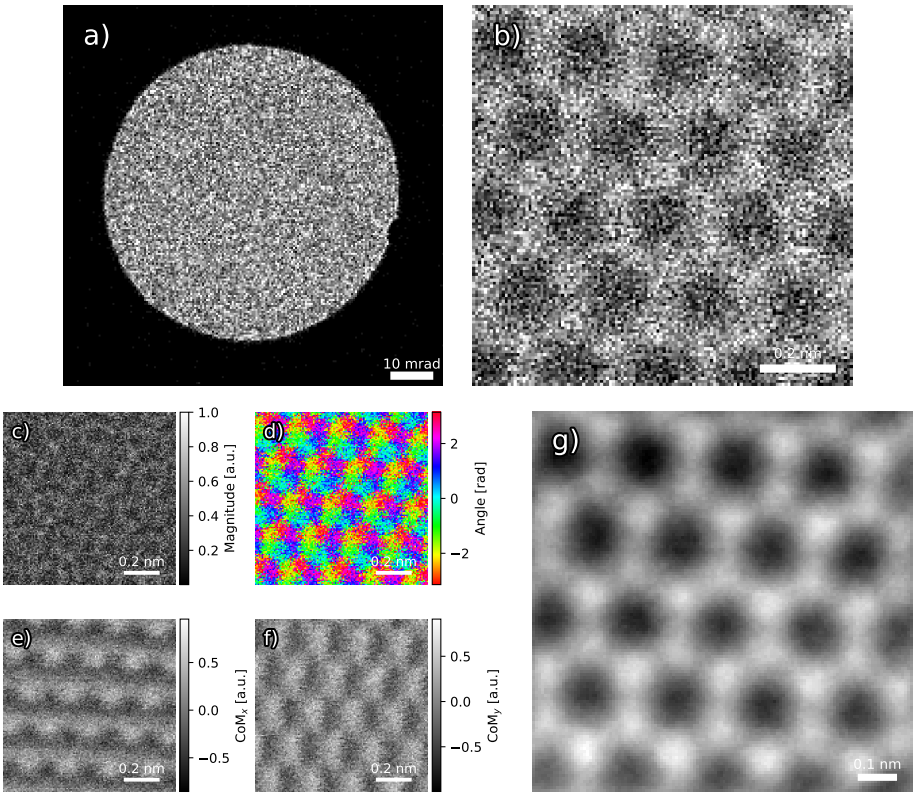

Figure 3.26: a) CBED pattern, b) virtual ADF image (roughly 36 to 62 mrad), c)-f) CoM and g) iCoM images of the 17:19 graphene 4D dataset recorded on 19.3.2024.

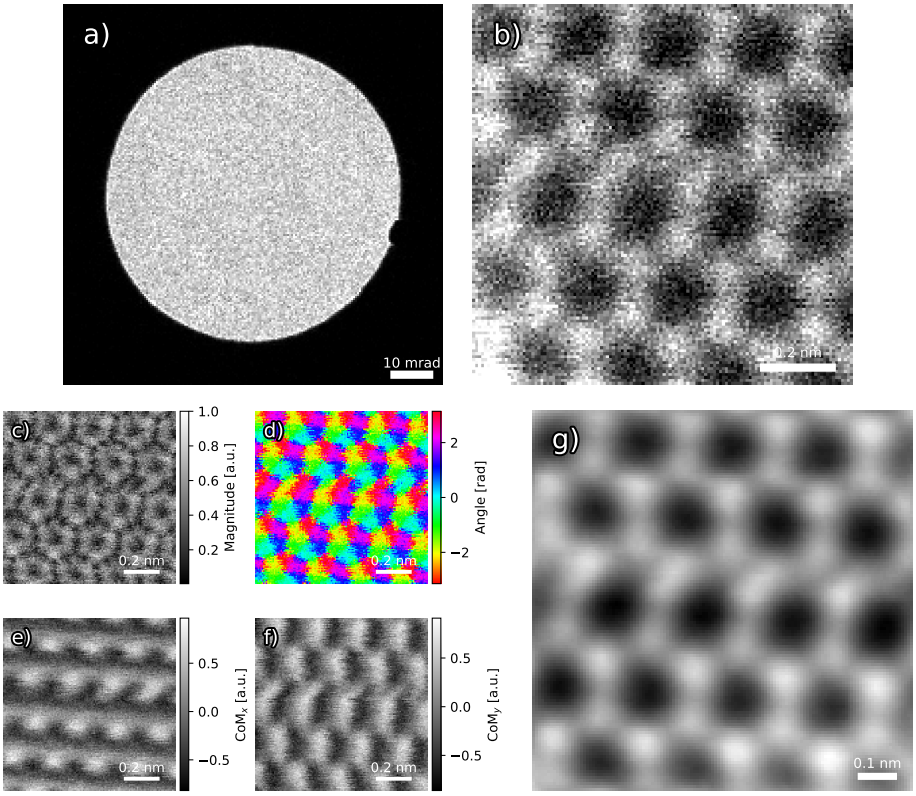

Figure 3.27: a) CBED pattern, b) virtual ADF image (roughly 36 to 62 mrad), c)-f) CoM and g) iCoM images of the 18:28 graphene 4D dataset recorded on 19.3.2024.

Figures (3.26) and (3.27) depict CBED, virtual ADF and iCoM of two 4D datasets at a field of view of 1 nm, a real-space resolution of 128×128 and reciprocal-space resolution of 192×192, recorded of sample 1346 on March 19th 2024. The former dataset records patterns at 0.1 ms of exposure time, wheras the latter increases exposure to 1 ms. Both datasets exhibit oscillations in their center of mass graphs.

Recorded paths of tests

The results of in-situ drift tracking experiments will be presented using a selection of logged successful drift tracking experiments, and some notable unsuccessful ones.

The first two somewhat successful tracking loop experiments were conducted during the 3rd session on the 8th of November 2023. Figures (3.28) and (3.29) depict beam shifter positions sent to the microscope during tracking loops using the calibration from Figure (3.24), connected in sequential order. The starting point is drawn in red color at position $(0, 0)$ and the final position is drawn in green color. The tracking behaviour can be described as quick initial movements to a potential atomic site, followed by periods consisting of short steps, presumed to follow the specimen's drift. These periods of tracking are occasionally interrupted by fast-paced movement to presumably neighboring atomic sites. One possible explanation for this, specific to November 8th and previous sessions, is the previously mentioned lack of CBED intensity normalisation. The first tracking loop (Fig. 3.28) lasted for 4 minutes and 55 seconds, covering a distance of 0.595 nm from the starting point, for an average beam shift velocity of 1.21 Å/min. The second loop (Fig. 3.29) lasted for 11 minutes and 22 seconds, covered a distance of 1.198 nm with an average velocity of 1.05 Å/min. Before- and after-scans of these two experiments, with a Gaussian filter of $\sigma = 1.5$ applied, lack substitutional impurities in the lattice. They only show narrow areas of dirt, making it difficult to determine actual values for corrected and residual drift.

The two other reasonably successful tracking experiments were conducted using calibrations of the 17:19 graphene 4D dataset recorded on March 19th 2024 (Fig. 3.26). Here, an area featuring an impurity was selected, as depicted on the HAADF image in Figure (3.30) a), with a Gaussian filter of $\sigma = 2.0$. The impurity is marked using an orange dot, and the position of the parked beam, saved to the image a)'s metadata by the plugin, is shown using a red triangle.

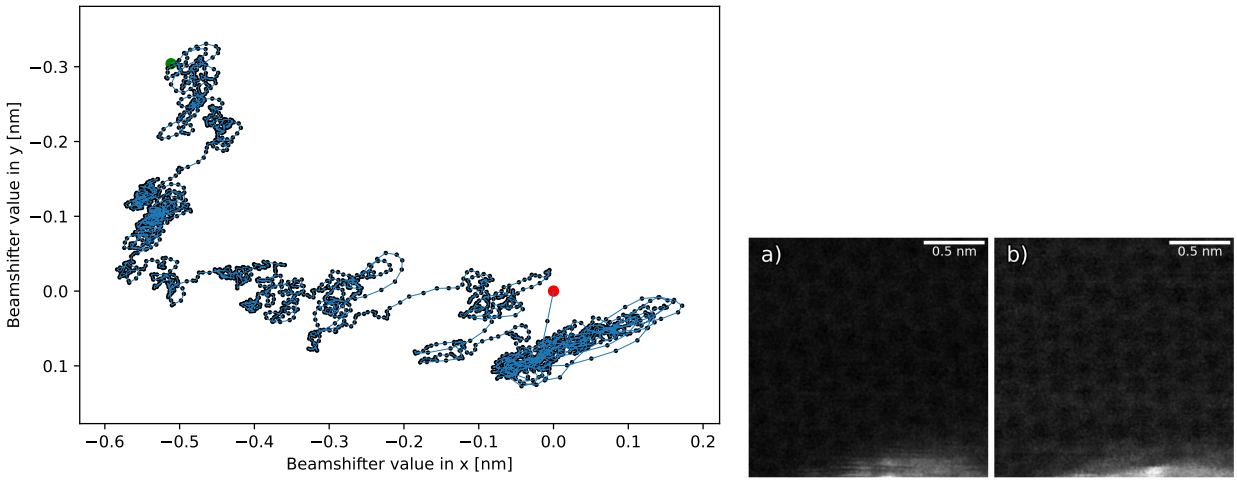

Figure 3.28: Path drawn by beam shifter control commands on 08.11.23 from 13:27 and before a) and after b) HAADF images.

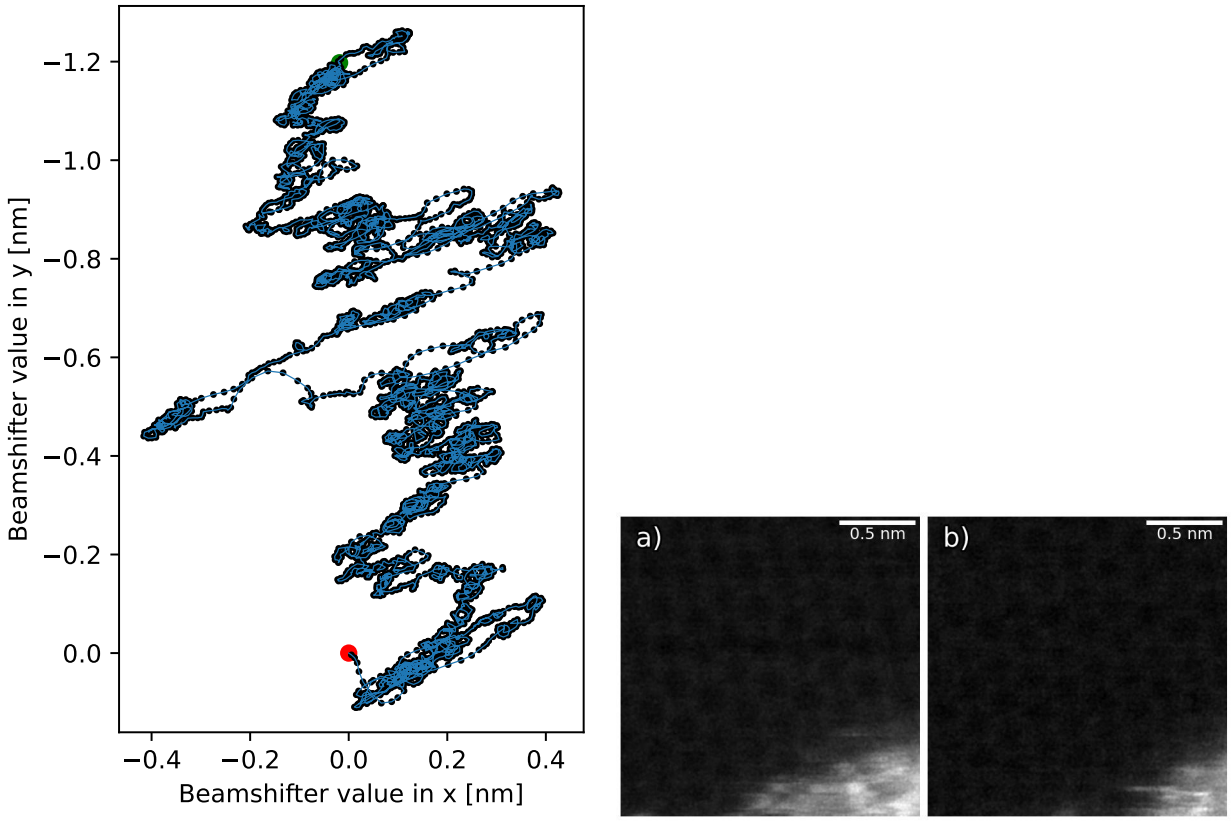

Figure 3.29: Path drawn by beam shifter control commands on 08.11.2023 from 13:32 and before a) and after b) HAADF images.

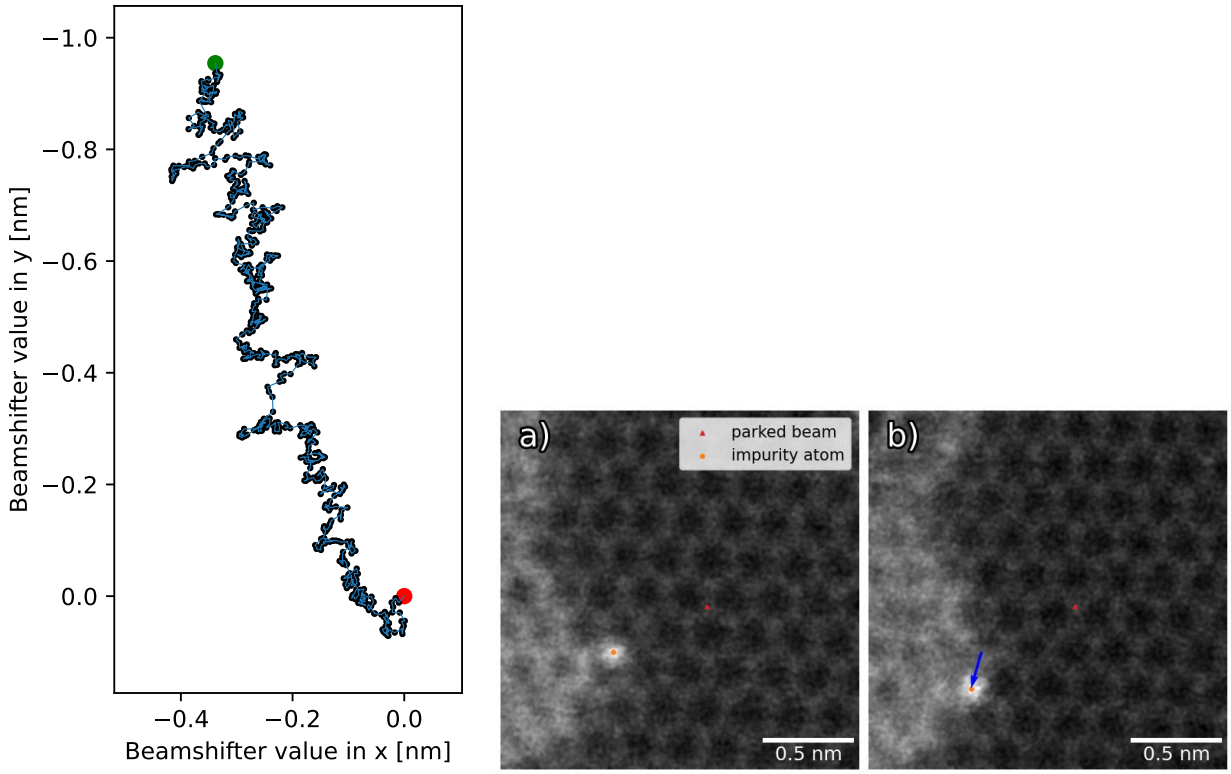

Figure 3.30: Path drawn by beam shifter control commands on 19.03.24 from 17:27 and before a) and after b) HAADF images.

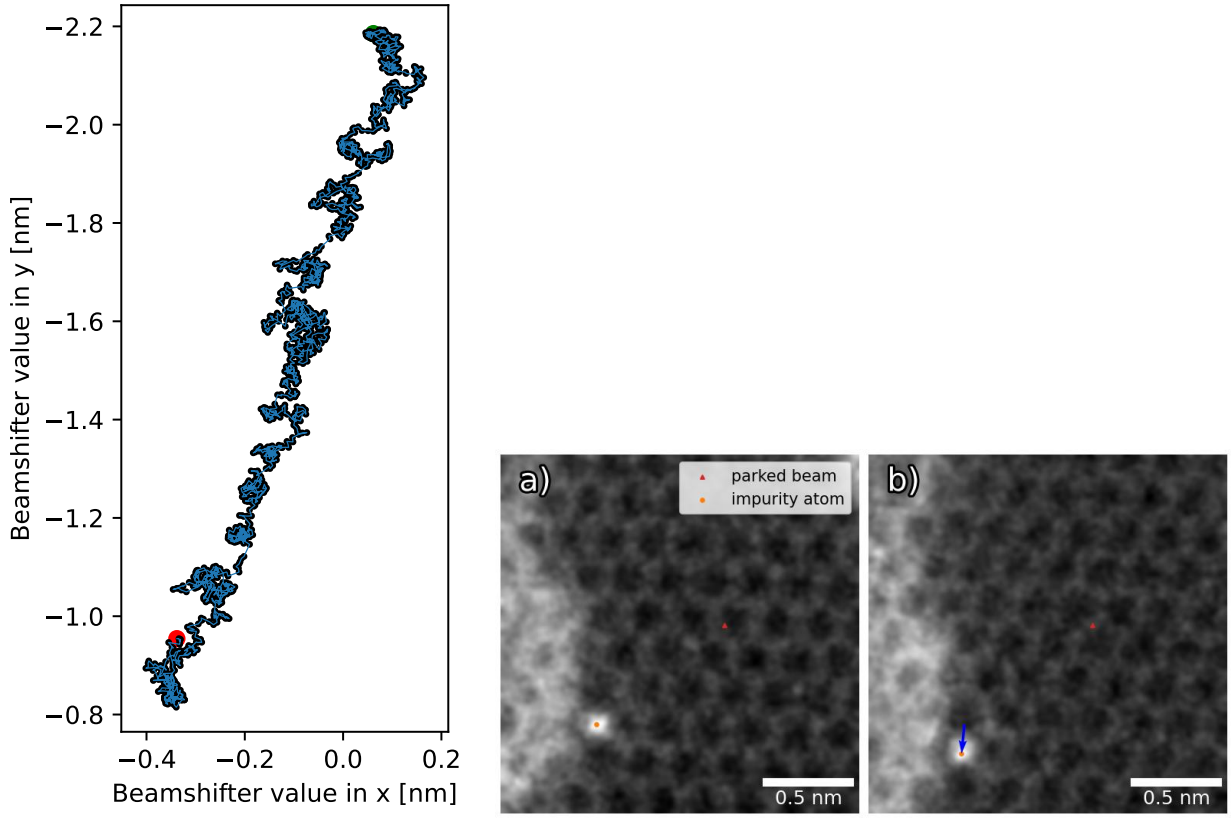

Figure 3.31: Path drawn by beam shifter control commands on 19.03.24 from 17:33 and before a) and after b) HAADF images.

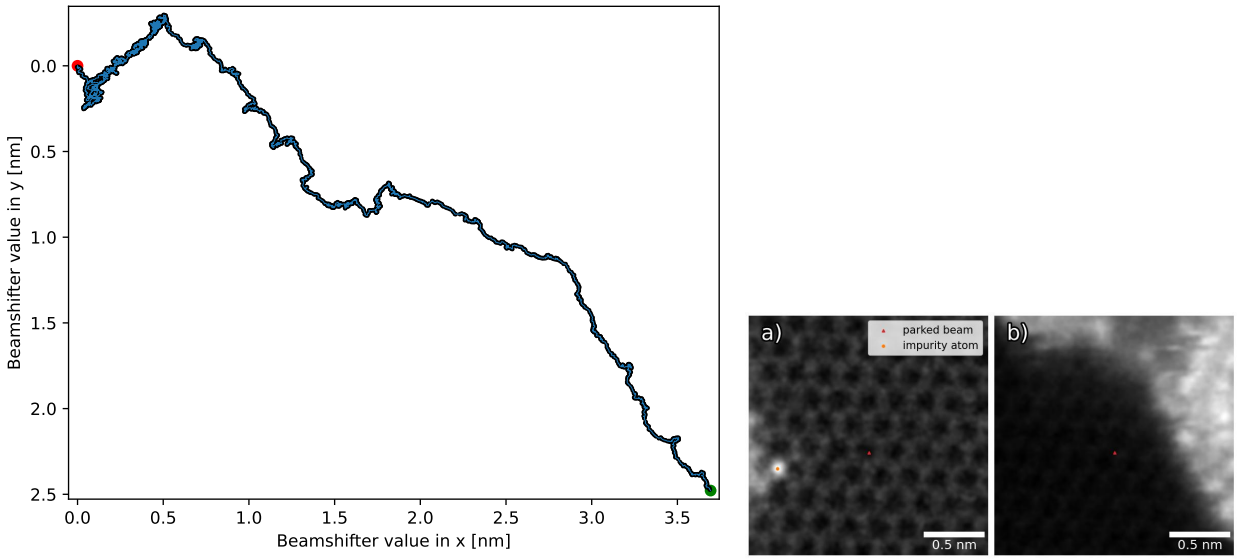

Figure 3.32: Path drawn by beam shifter control commands on 19.03.24 from 17:51 and before a) and after b) HAADF images.

The tracking loop in Figure (3.30) lasted 2 minutes and 57 seconds, with a distance of 1.01 nm covered by the beam shifters, which amounts to a velocity of 3.43 Å/min. Figure (3.30) b) shows an image of the area after the experiment, with a blue arrow indicating a displacement of the impurity atom of about 2.11 Å.

Assuming the impurity atom has not moved, the total drifted displacement $\vec{s}_{\text{drift}}$ of the sample from a) to b), can be approximated using vector subtraction of the total beam shifter displacement $\vec{s}_{\text{shift}}$ from the residual impurity displacement $\vec{s}_{\text{res}}$:

$$\vec{s}_{\text{drift}} = \vec{s}_{\text{res}} - \vec{s}_{\text{shift}} \,, \tag{3.15}$$

with the resulting drifted distance being $|\vec{s}_{\text{drift}}| = 1.19$ nm. Divided by the duration, the approximated drift velocity is thus 4.03 Å/min.

The second experiment (Fig. 3.31) was performed 3 minutes later and lasted 7 minutes and 6 seconds. It moved the beam by a distance of 1.29 nm with an average velocity of 1.83 Å/min. The residual impurity displacement this time is 1.65 Å, resulting in an approximated total drifted distance of 1.46 nm and a drift velocity of 2.05 Å/min.

The above calculations neglect initial beam shifting steps towards an atomic site, which are not representative of the actual drift of the sample. It is possible that all or part of the residual impurity displacement $\vec{s}_{\text{res}}$ is caused by these initial tracking steps. Other sources of error include the possibility of the tracking loop intermittently switching its “target” to a neighboring atom, as well as movement of the impurity atom through the lattice.

Another tracking experiment from the same day (Fig. 3.32) uses calibration values from the later dataset depicted in Figure (3.27). The logged beam shifter values start off following an atomic site for 5 minutes and 41 seconds, up to a beam shifter distance of 5.76 Å from the starting point. The recorded path then proceeds along a seemingly biased direction with increasingly minor disturbances by presumably other atomic sites along the way. This goes on for 6 minutes and 48 seconds, over a distance of 4.19 Å, until tracking is stopped manually. The average beam shifter velocities during these two sections are 1.01 Å/min and 6.16 Å/min respectively.

HAADF images in Figure (3.32) a) and b) support the notion that the field of view has moved considerably and tracking has failed. A likely explanation for this behavior is a bias introduced by a change in offset position of the bright-field disk compared to its position at the time of calibration (see Chapter 3.2). Comparing the centering parameter $\vec{j_0}$, established during calibration 32 minutes prior, to the parameter values of a calibration 37 minutes later, results in a difference with a magnitude of 0.576 mrad.

Reciprocal-space offset during experiments

To investigate the diffraction pattern offset contribution to the center of mass, the position of the bright-field disk $\vec{q}_{\text{probe}}$ was determined using the py4DSTEM library's `get_probe_size` function and compared to the centering parameter $\vec{j_0}$ obtained using the plugin's up-to-date calibration function, for 24 recorded 4D datasets used in tracking experiments. In principle, both functions calculate the mean, I_{tot}–normalised center of mass of a 4D dataset. `get_probe_size` performs an additional step of approximating the radius of a circular probe using a set of binary masks within a range of threshold parameters before applying the center of mass. This step eliminates outlier dark field contributions to the CoM and allows for visual inspection and manual adjustments to probe size fitting. For this comparison, parameters were set to `thresh_lower=0.01` and `thresh_upper=0.5`. An example can be seen in Figure (3.33) a), where a circle is drawn using py4DSTEM's probe size and position $\vec{q}_{\text{probe}}$ on top of the averaged diffraction pattern of the April 24th 2024, 18:58 dataset. The mean CoM magnitude differed between these two methods on average by (0.016 ± 0.002) mrad, up to 0.036 mrad and on average by (3.1 ± 0.3)%, up to 5.3% of the magnitude. This high similarity supports the findings of Chapter (3.2) that the majority of $\vec{j_0}$ values encountered during experiments can be attributed to an offset of the bright-field

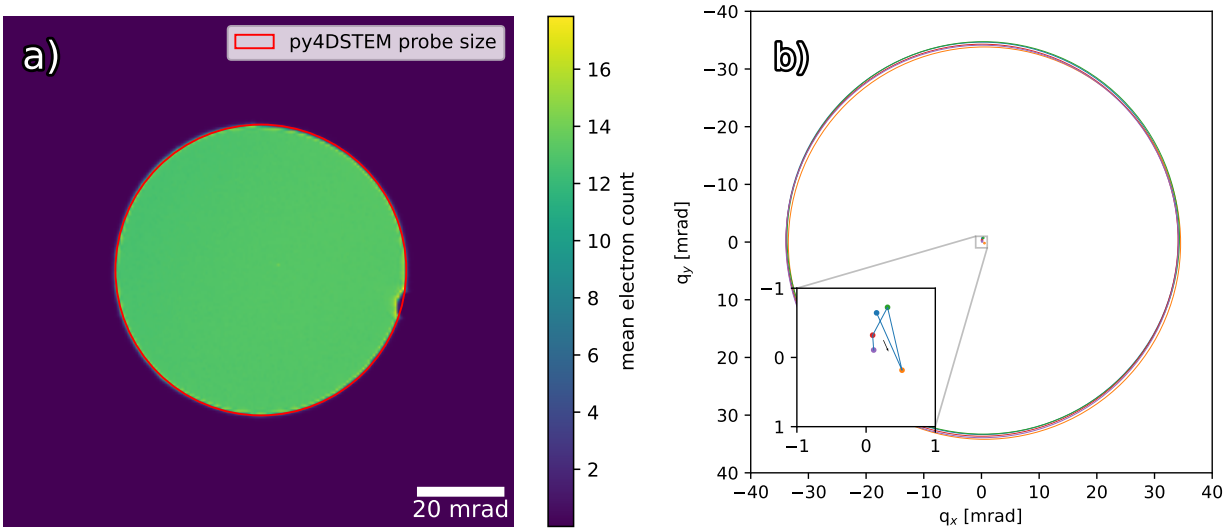

Figure 3.33: a) Circle representing probe size radius and BF disk position determined using py4DSTEM on top of mean diffraction pattern from the April 24th 2024 18:58 dataset, and b) BF disk comparison of five datasets from April 24th, 2024.

disk in reciprocal space, rather than dark field and aberration contributions. Next, to investigate the change of this offset over time, $\vec{j_0}$ and $\vec{q}_{\text{probe}}$ values of 4D datasets recorded during the same testing session were compared to each other and weighted by the time between two consecutive recordings. Figure (3.33) b) shows bright-field disk positions as dots and probe sizes as circles, as calculated using `get_probe_size`, belonging to five 4D datasets recorded as part of tracking experiments on April 24th, 2024. The datasets were recorded at intervals of 45, 24, 37 and 5 minutes. An arrow indicates the sequence order of the positions. In this particular example, $\vec{q}_{\text{probe}}$ exhibited changes in magnitude by over 0.9 mrad in two of four cases. This equates to an offset three times larger than the stability limit of $\sim$ 0.3 mrad discussed in Chapter (3.2).

From the 24 assessed datasets, 16 comparisons could be made. On average, the offset magnitude changed at a rate of (0.015 ± 0.004) mrad/min, with a maximum observed rate of change at about 0.06 mrad/min. This rate of change may well depend on difficult-to-control experimental conditions and factors such as lens drift. In a worst-case scenario at 0.06 mrad/min, the plugin may lose tracking after about five minutes due to an uncompensated offset $\vec{j_0}$ growing to a magnitude of 0.3 mrad.

The calibration routine was originally introduced to compensate for a reciprocal-space offset $\vec{j_0}$ under the assumption that it would not change significantly over time. Due to the sensitivity of the center of mass to this change, however, it is not sufficient to keep a fixed calibration value for $\vec{j_0}$. A possible solution would be to measure the BF disk offset of single diffraction patterns recorded during tracking. Such a measurement may be possible using CoM, but only if the intensity distribution inside the bright field disk is flattened, to remove all contributions of aberrations and projected electric fields.

One alternative not based on CoM would be measurement of the BF disk offset by performing edge detection using a gradient operator, such as the ones described by Sobel [62] or Canny [63], then fitting a circle or ellipse to this BF disk edge, or even performing a circle Hough Transform (CHT) [64] to find the circle position. Another method could consist of training a neural network to recognize and measure the position of BF disks [65].

Beam-shifter coordinates

Depending on how a segmented or pixelated detector is oriented with respect to the microscope's stage, an additional rotation matrix operation may be required to undo part of the QR-rotation correction. While the detector's misalignment angle is a component of the QR-rotation angle, this component persists as a rotation between the corrected CoM data and the beam shifters. As such, displacement vectors need to be rotated by this misalignment angle before being applied to the shifters.

This misalignment angle was empirically determined to be roughly 90° for the Hamamatsu Orca sCCD camera, and 0° for the newly installed Dectris ARINA direct electron detector and the usim virtual ronchigram camera. These values have been hard-coded into the `assess_hardware_sources` function and are not actively measured by the plugin.

Drift measurement

To assess the drift behaviour of the microscope, a MAADF image series was recorded of an unverified four-fold Si impurity (Si-C_4) in the graphene sample 1346, at the end of the 5th testing session on the 5th of February 2024. The series consists of 813 images at 128×128 pixels resolution, with a dwell time of 32 µs per pixel for a total of roughly 524 ms per image and 426 seconds or 7 minutes and 6 seconds for the whole series.

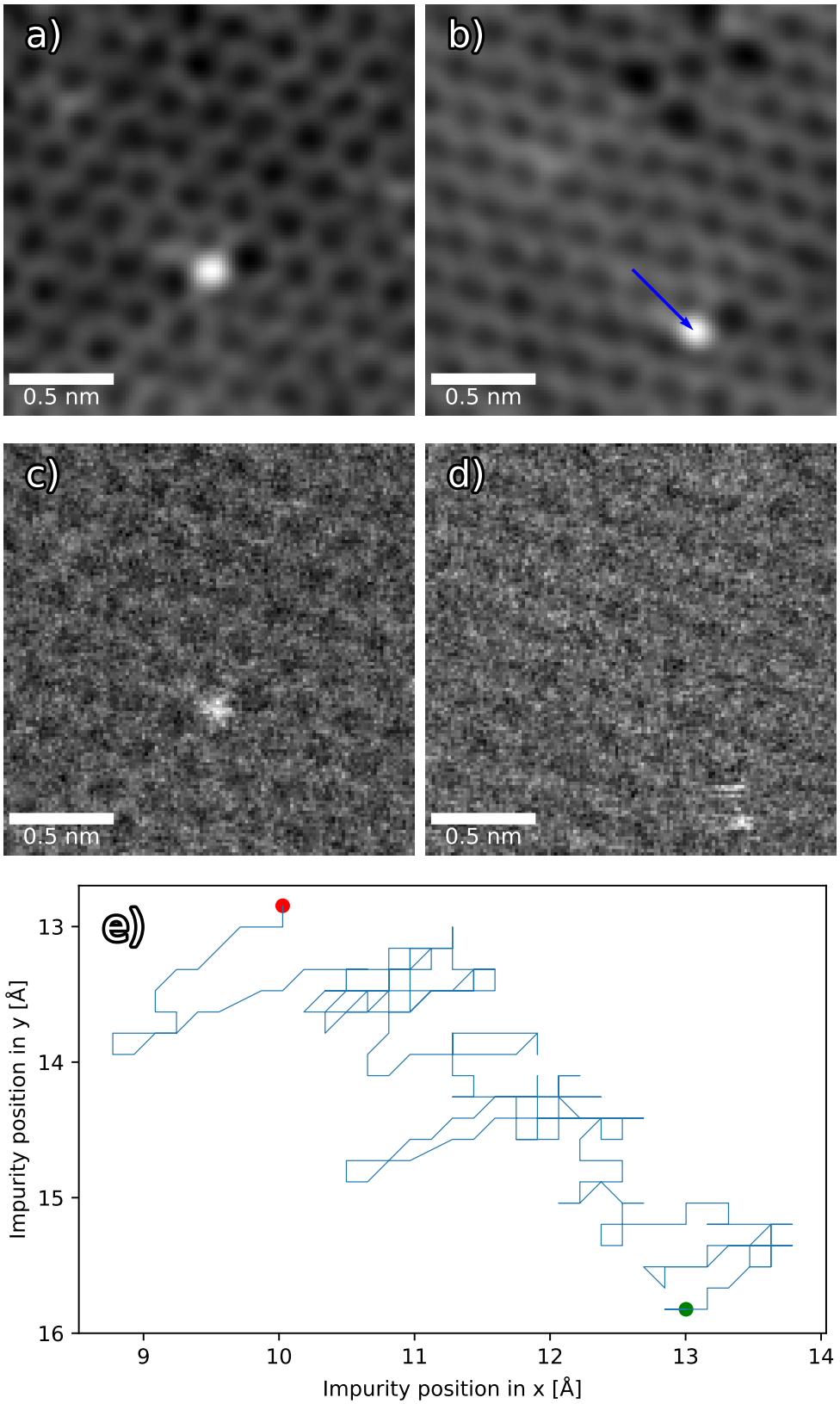

Figure 3.34: MAADF images of drift measurement: a) first image in series, b) 700th image in series, c) first image with $\sigma = 0.5$, d) 813th image in series, showing an unstable impurity. e) Approximate path drawn by impurity.

For impurity position analysis, a `gaussian_filter` function from the SciPy module, with a parameter of $\sigma = 2.4$, was applied to the whole image series array, resulting in more stable images than applying it individually. Figure (3.34) a) shows the first image of the set, displaying a prominent foreign atom after application of the Gaussian filter. Displayed in b) is the 700th image of the series, after which the impurity's position in the lattice started to change. As before, an arrow shows the foreign atom's position relative to the first image. The drifted distance amounts to 4.21 Å, resulting in a drift velocity of 0.69 Å/min. Figure (3.34) c) is based on the same image as a), but with a lower filter parameter of $\sigma = 0.5$. Using the same filter parameter, Figure (3.34) d) shows the last image of the series in order to visualise scan distortion due to a moving impurity.

Figure (3.34) e) shows a 2D plot of the first 700 impurity positions, with the red dot marking the first position. Although crude, this path shows mostly erratic, presumably thermal movement, with a bias in $(+x, +y)$ direction, as well as two jerky movements in $(-x, +y)$ direction. The impurity's position was approximated by selecting the highest intensity pixel indices from each $\sigma = 2.4$ filtered image. Calculating displacements from these positions allows us to determine an average "micro drift" velocity of (7.08 ± 0.36) Å/min. Displacements during the two jerky movements reached the highest velocities at roughly 40 Å/min.

Time logs of drift tracking experiments revealed that the time required for a drift correction step can rarely exceed 500 ms, while execution times between 200 and 300 ms are more common. Furthermore, the majority of this delay was found to originate from the acquisition of individual diffraction patterns through nionswift, despite pattern exposure times being orders of magnitude lower. With a default maximum step size of about 0.5 Å as described in Chapter (3.4.1), the tracking loop may fail to compensate the highest encountered displacements found during this drift measurement, being ~ 0.4 Å at a 524 ms timestep, especially with noise, aberrations or a reciprocal-space offset (see chapter 3.2) present.

Chapter 4

Conclusions

In this thesis, we attempted to compensate for stage drift in a scanning transmission electron microscope in real time, when working with a stationary electron beam and graphene.

First, simulations of graphene and free carbon atoms were run to fit a function that maps the center of mass (CoM) of convergent beam electron diffraction (CBED) patterns to a displacement relative to an atomic site, at a given microscope voltage of 60 keV. Next, pre-recorded 4D datasets of graphene were used to test for discrepancies between simulated and realistic microscopy data, as well as prepare a tracking algorithm for in-situ testing. Finally, real-time compensation was attempted via a custom plugin for the microscope control user interface (UI) Nion Swift, over the course of ten sessions. The plugin interacts with the microscope by translating the CoM signal of the most recent Ronchigram or CBED pattern into a displacement, then shifting the probe accordingly.

Tracking of atomic sites in graphene using static data such as simulations and recorded datasets was successfully demonstrated. And while there have been seemingly successful tracking sessions, consistent real-time drift compensation was not achieved. There are many limiting factors, such as Poisson noise, aberrations and acquisition speed; however, the main point of failure seems to be drift of the probe on the camera. This pertains to the fact that the center of mass contains an offset component that is largely determined by the reciprocal-space position of the BF disk, relative to the middle of the diffraction pattern. While the calibration process accounts for a constant disk offset, movement of the disk during in-situ tracking experiments, introduces a bias that may grow until it overshadows the CoM signal of the specimen.

The reciprocal-space offset thus needs to be monitored for changes during tracking. One solution to this could be CoM-based position measurements of CBED patterns with the specimen signal filtered out. Possible solutions to this problem without using CoM include detection of the BF disk position by edge detection algorithms, such as the circle Hough transform (CHT) [64] or disk detection by a trained neural network [65].

Another area of improvement is the type of fit used for the tracking algorithm. A two-dimensional center of mass fit of the graphene lattice under consideration of its symmetries could improve tracking accuracy, but may require additional context, such as lattice orientation during calibration.

The approach described in this thesis may serve as a useful stepping stone for future attempts at automation of atom manipulation experiments and other drift-sensitive STEM experiments using two-dimensional materials. It may also contribute to existing automation software, such as the Atom Manipulator plugin [66].

Bibliography

[1] D. M. Eigler and E. K. Schweizer, "Positioning single atoms with a scanning tunnelling microscope," vol. 344, no. 6266, pp. 524–526. https://www.nature.com/articles/344524a0

[2] N. Dellby, O. L. Krivanek, P. D. Nellist, P. E. Batson, and A. R. Lupini, "Progress in aberration-corrected scanning transmission electron microscopy," vol. 50, no. 3, pp. 177–185. https://academic.oup.com/jmicro/article-lookup/doi/10.1093/jmicro/50.3.177

[3] T. Susi, J. Kotakoski, D. Kepaptsoglou, C. Mangler, T. C. Lovejoy, O. L. Krivanek, R. Zan, U. Bangert, P. Ayala, J. C. Meyer, and Q. Ramasse, "Silicon–Carbon Bond Inversions Driven by 60-keV Electrons in Graphene," *Physical Review Letters*, vol. 113, no. 11, p. 115501, Sep. 2014. https://link.aps.org/doi/10.1103/PhysRevLett.113.115501

[4] G. Zagler, M. Stecher, A. Trentino, F. Kraft, C. Su, A. Postl, M. Längle, N. Pesenhofer, C. Mangler, E. Harriet Åhlgren, A. Markevich, A. Zettl, J. Kotakoski, T. Susi, and K. Mustonen, "Beam-driven dynamics of aluminium dopants in graphene," vol. 9, no. 3, p. 035009. https://iopscience.iop.org/article/10.1088/2053-1583/ac6c30

[5] C. Su, M. Tripathi, Q.-B. Yan, Z. Wang, Z. Zhang, C. Hofer, H. Wang, L. Basile, G. Su, M. Dong, J. C. Meyer, J. Kotakoski, J. Kong, J.-C. Idrobo, T. Susi, and J. Li, "Engineering single-atom dynamics with electron irradiation," vol. 5, no. 5. https://www.science.org/doi/10.1126/sciadv.aav2252

[6] D. B. Williams and C. B. Carter, *Transmission electron microscopy: a textbook for materials science*, 2nd ed., ser. Cambridge library collection. New York: Springer, 2008.

[7] C. Ophus, S. M. Ribet, G. Varnavides, J. Madsen, and T. Susi, "A practical guide to scanning and transmission electron microscopy simulations." https://doi.curvenote.com/10.69761/aghr7357

[8] E. J. Kirkland, *Advanced Computing in Electron Microscopy*, 3rd ed. Springer International Publishing AG.

[9] E. Zeitler and H. Olsen, "Screening effects in elastic electron scattering," vol. 136, no. 6, pp. A1546–A1552. https://link.aps.org/doi/10.1103/PhysRev.136.A1546

[10] E. Zeitler and H. Olsen, "Complex scattering amplitudes in elastic electron scattering," vol. 162, no. 5, pp. 1439–1447. https://link.aps.org/doi/10.1103/PhysRev.162.1439

[11] R. Glauber and V. Schomaker, "The theory of electron diffraction," vol. 89, no. 4, pp. 667–671. https://link.aps.org/doi/10.1103/PhysRev.89.667

[12] R. McWeeny, "X-ray scattering by aggregates of bonded atoms. i. analytical approximations in single-atom scattering," vol. 4, no. 6, pp. 513–519. https://journals.iucr.org/paper?S0365110X51001732

[13] A. J. Freeman, "Atomic scattering factors for spherical and aspherical charge distributions," vol. 12, no. 4, pp. 261–271. https://journals.iucr.org/paper?S0365110X59000834

[14] J. C. Meyer, S. Kurasch, H. J. Park, V. Skakalova, D. Künzel, A. Groß, A. Chuvilin, G. Algara-Siller, S. Roth, T. Iwasaki, U. Starke, J. H. Smet, and U. Kaiser, "Experimental analysis of charge redistribution due to chemical bonding by high-resolution transmission electron microscopy," vol. 10, no. 3, pp. 209–215. https://www.nature.com/articles/nmat2941

[15] T. Susi, J. Madsen, U. Ludacka, J. J. Mortensen, T. J. Pennycook, Z. Lee, J. Kotakoski, U. Kaiser, and J. C. Meyer, "Efficient first principles simulation of electron scattering factors for transmission electron microscopy," vol. 197, pp. 16–22. https://linkinghub.elsevier.com/retrieve/pii/S0304399118302481

[16] P. W. Hawkes, *Electron optics and electron microscopy*. Taylor and Francis.

[17] M. Knoll and E. Ruska, "Das elektronenmikroskop," vol. 78, no. 5, pp. 318–339. http://link.springer.com/10.1007/BF01342199

[18] M. Von Ardenne, "Das elektronen-rastermikroskop: Theoretische grundlagen," vol. 109, no. 9, pp. 553–572. http://link.springer.com/10.1007/BF01341584

[19] M. K. Tripathi, "Modifying low-dimensional materials using energetic charged particles." https://doi.org/10.25365/thesis.56691

[20] A. V. Crewe, J. Wall, and J. Langmore, "Visibility of single atoms," vol. 168, no. 3937, pp. 1338–1340. https://www.science.org/doi/10.1126/science.168.3937.1338

[21] D. H. Shin, E. J. Kirkland, and J. Silcox, "Annular dark field electron microscope images with better than 2 Å resolution at 100 kV," vol. 55, no. 23, pp. 2456–2458. https://pubs.aip.org/apl/article/55/23/2456/55951/Annular-dark-field-electron-microscope-images-with

[22] H. Yang, R. N. Rutte, L. Jones, M. Simson, R. Sagawa, H. Ryll, M. Huth, T. J. Pennycook, M. Green, H. Soltau, Y. Kondo, B. G. Davis, and P. D. Nellist, "Simultaneous atomic-resolution electron ptychography and z-contrast imaging of light and heavy elements in complex nanostructures," vol. 7, no. 1. https://www.nature.com/articles/ncomms12532

[23] T. J. Pennycook, A. R. Lupini, H. Yang, M. F. Murfitt, L. Jones, and P. D. Nellist, "Efficient phase contrast imaging in STEM using a pixelated detector. part 1: Experimental demonstration at atomic resolution," vol. 151, pp. 160–167. https://linkinghub.elsevier.com/retrieve/pii/S0304399114001934

[24] M. C. Cao, Y. Han, Z. Chen, Y. Jiang, K. X. Nguyen, E. Turgut, G. D. Fuchs, and D. A. Muller, "Theory and practice of electron diffraction from single atoms and extended objects using an EMPAD," *Microscopy*, vol. 67, no. suppl_1, pp. i150–i161, Mar. 2018. https://academic.oup.com/jmicro/article/67/suppl_1/i150/4835603

[25] I. Lazić, E. G. Bosch, and S. Lazar, "Phase contrast STEM for thin samples: Integrated differential phase contrast," *Ultramicroscopy*, vol. 160, pp. 265–280, Jan. 2016. https://linkinghub.elsevier.com/retrieve/pii/S0304399115300449

[26] E. Waddell, J. Chapman, and R. Ferrier, "LINEAR IMAGING OF STRONG PHASE OBJECTS BY DIFFERENTIAL PHASE CONTRAST." no. 36, pp. 267 – 270. https://www.scopus.com/inward/record.uri?eid=2-s2.0-0017696212&partnerID=40&md5=3582a5e41c53e37639088dd3f6e62a3b

[27] E. Waddell and J. Chapman, "LINEAR IMAGING OF STRONG PHASE OBJECTS USING ASYMMETRICAL DETECTORS IN STEM." vol. 54, no. 2, pp. 83

– 96. https://www.scopus.com/inward/record.uri?eid=2-s2.0-0018534682&partnerID=40&md5=f9ad511b3973cb77d2dbb8271f8f1284

[28] R. Peierls, "Quelques propriétés typiques des corps solides," *Ann. Inst. Henri Poincaré*, vol. 5, pp. 177–222, 1935. https://eudml.org/doc/78996

[29] L. D. Landau, "Zur theorie der phasenumwandlungen ii," *Phys. Z. Sowjetunion*, vol. 11, no. 545, pp. 26–35, 1937.

[30] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zhang, S. V. Dubonos, I. V. Grigorieva, and A. A. Firsov, "Electric field effect in atomically thin carbon films," vol. 306, no. 5696, pp. 666–669. https://www.science.org/doi/10.1126/science.1102896

[31] K. Cao, S. Feng, Y. Han, L. Gao, T. Hue Ly, Z. Xu, and Y. Lu, "Elastic straining of free-standing monolayer graphene," vol. 11, no. 1. https://www.nature.com/articles/s41467-019-14130-0

[32] D. R. Cooper, B. D'Anjou, N. Ghattamaneni, B. Harack, M. Hilke, A. Horth, N. Majlis, M. Massicotte, L. Vandsburger, E. Whiteway, and V. Yu, "Experimental review of graphene," vol. 2012, pp. 1–56. https://www.hindawi.com/journals/isrn/2012/501686/

[33] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, M. I. Katsnelson, I. V. Grigorieva, S. V. Dubonos, and A. A. Firsov, "Two-dimensional gas of massless dirac fermions in graphene," vol. 438, no. 7065, pp. 197–200. https://www.nature.com/articles/nature04233

[34] F. Banhart, J. Kotakoski, and A. V. Krasheninnikov, "Structural defects in graphene," vol. 5, no. 1, pp. 26–41. https://pubs.acs.org/doi/10.1021/nn102598m

[35] Y.-B. Tang, L.-C. Yin, Y. Yang, X.-H. Bo, Y.-L. Cao, H.-E. Wang, W.-J. Zhang, I. Bello, S.-T. Lee, H.-M. Cheng, and C.-S. Lee, "Tunable band gaps and p-type transport properties of boron-doped graphenes by controllable ion doping using reactive microwave plasma," vol. 6, no. 3, pp. 1970–1978. https://pubs.acs.org/doi/10.1021/nn3005262

[36] D. Usachov, O. Vilkov, A. Grüneis, D. Haberer, A. Fedorov, V. K. Adamchuk, A. B. Preobrajenski, P. Dudin, A. Barinov, M. Oehzelt, C. Laubschat, and D. V. Vyalikh, "Nitrogen-doped graphene: Efficient growth, structure, and electronic properties," vol. 11, no. 12, pp. 5401–5407. https://pubs.acs.org/doi/10.1021/nl2031037

[37] W. Zhou, J. Lee, J. Nanda, S. T. Pantelides, S. J. Pennycook, and J.-C. Idrobo, "Atomically localized plasmon enhancement in monolayer graphene," vol. 7, no. 3, pp. 161–165. https://www.nature.com/articles/nnano.2011.252

[38] M. Tripathi, A. Mittelberger, N. A. Pike, C. Mangler, J. C. Meyer, M. J. Verstraete, J. Kotakoski, and T. Susi, "Electron-beam manipulation of silicon dopants in graphene," vol. 18, no. 8, pp. 5319–5323. https://pubs.acs.org/doi/10.1021/acs.nanolett.8b02406

[39] D. B. Lingerfelt, T. Yu, A. Yoshimura, P. Ganesh, J. Jakowski, and B. G. Sumpter, "Nonadiabatic effects on defect diffusion in silicon-doped nanographenes," vol. 21, no. 1, pp. 236–242. https://pubs.acs.org/doi/10.1021/acs.nanolett.0c03587

[40] K. R. Fiedler, M. J. Olszta, K. H. Yano, C. Doty, D. Hopkins, S. Akers, and S. R. Spurgeon, "Evaluating stage motion for automated electron microscopy," p. ozad108. https://academic.oup.com/mam/advance-article/doi/10.1093/micmic/ozad108/7313517

[41] F. E. Kalff, M. P. Rebergen, E. Fahrenfort, J. Girovsky, R. Toskovic, J. L. Lado, J. Fernández-Rossier, and A. F. Otte, "A kilobyte rewritable atomic memory," vol. 11, no. 11, pp. 926–929. https://www.nature.com/articles/nnano.2016.131

[42] R. Drost, T. Ojanen, A. Harju, and P. Liljeroth, "Topological states in engineered atomic lattices," vol. 13, no. 7, pp. 668–671. https://www.nature.com/articles/nphys4080

[43] N. Pavliček, A. Mistry, Z. Majzik, N. Moll, G. Meyer, D. J. Fox, and L. Gross, "Synthesis and characterization of triangulene," vol. 12, no. 4, pp. 308–311. https://www.nature.com/articles/nnano.2016.305

[44] J. Madsen and T. Susi, "The abTEM code: transmission electron microscopy from first principles," vol. 1, p. 24. https://open-research-europe.ec.europa.eu/articles/1-24/v2

[45] A. Hjorth Larsen, J. Jørgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment—a python library for working with atoms," vol. 29, no. 27, p. 273002. https://iopscience.iop.org/article/10.1088/1361-648X/aa680e

[46] J. Madsen and T. Susi, "abTEM walkthrough." https://abtem.github.io/doc/user_guide/walkthrough/walkthrough.html

[47] I. Lobato and D. Van Dyck, "An accurate parameterization for scattering factors, electron densities and electrostatic potentials for neutral atoms that obey all physical constraints," vol. 70, no. 6, pp. 636–649. https://journals.iucr.org/paper?S205327331401643X

[48] L.-M. Peng, "Electron atomic scattering factors and scattering potentials of crystals," vol. 30, no. 6, pp. 625–648. https://linkinghub.elsevier.com/retrieve/pii/S0968432899000335

[49] J. M. Cowley and A. F. Moodie, "The scattering of electrons by atoms and crystals. i. a new theoretical approach," vol. 10, no. 10, pp. 609–619. https://journals.iucr.org/paper?S0365110X57002194

[50] W. Van Den Broek, X. Jiang, and C. Koch, "FDES, a GPU-based multislice algorithm with increased efficiency of the computation of the projected potential," vol. 158, pp. 89–97. https://linkinghub.elsevier.com/retrieve/pii/S0304399115300127

[51] py4DSTEM. py4DSTEM 0.13.7 DPC Module functions for differential phase contrast imaging. https://github.com/py4dstem/py4DSTEM/blob/1ee24863ec8835dfa81a3dab3969e774b5ead027/py4DSTEM/process/dpc/dpc.py

[52] O. Krivanek, G. Corbin, N. Dellby, B. Elston, R. Keyse, M. Murfitt, C. Own, Z. Szilagyi, and J. Woodruff, "An electron microscope for the aberration-corrected era," vol. 108, no. 3, pp. 179–195. https://linkinghub.elsevier.com/retrieve/pii/S0304399107002331

[53] C. Mangler, J. Meyer, A. Mittelberger, K. Mustonen, T. Susi, and J. Kotakoski, "A materials scientist's CANVAS: A system for controlled alteration of nanomaterials in vacuum down to the atomic scale," vol. 28, pp. 2940–2942. https://academic.oup.com/mam/article/28/S1/2940/6997028

[54] M. Hotz, G. Corbin, N. Dellby, O. Krivanek, C. Mangier, and J. Meyer, "Ultra-high vacuum aberration-corrected STEM for in-situ studies," vol. 22, pp. 34–35. https://academic.oup.com/mam/article/22/S3/34/6898837

[55] Hamamatsu. ORCA-Flash4.0 V3 digital CMOS camera C13440-20CU. https://web.archive.org/web/20250514083738/https://www.hamamatsu.com/eu/en/product/cameras/cmos-cameras/C13440-20CU.html

[56] Dectris. Dectris ARINA hybrid-pixel detector for 4D STEM applications. https://web.archive.org/web/20250617151932/https://www.dectris.com/en/detectors/electron-detectors/for-materials-science/arina/

[57] P. Zambon, J. Vávra, G. Montemurro, S. Bottinelli, A. Dudina, R. Schnyder, C. Hörmann, M. Meffert, C. Schulze-Briese, D. Stroppa, N. Lehmann, and L. Piazza, "High-frame rate and high-count rate hybrid pixel detector for 4d STEM applications," vol. 11. https://www.frontiersin.org/articles/10.3389/fphy.2023.1308321/full

[58] J. J. P. Peters, T. Mullarkey, E. Hedley, K. H. Müller, A. Porter, A. Mostaed, and L. Jones, "Electron counting detectors in scanning transmission electron microscopy via hardware signal processing," vol. 14, no. 1. https://www.nature.com/articles/s41467-023-40875-w

[59] C. Meyer and Nion. ORCA-Flash4.0 V3 digital CMOS camera C13440-20CU. github.com/nion-software/nionswift-plugin-sample/

[60] J. Madsen and T. Susi, "abTEM advanced tutorial - orthogonal supercells." https://abtem.github.io/doc/user_guide/tutorials/advanced_atomic_models.html#orthogonal-and-periodic-supercells

[61] C. M. O'Leary, B. Haas, C. T. Koch, P. D. Nellist, and L. Jones, "Increasing spatial fidelity and SNR of 4d-STEM using multi-frame data fusion," vol. 28, no. 4, pp. 1417–1427. https://academic.oup.com/mam/article/28/4/1417/6995511

[62] I. Sobel, "An isotropic 3x3 image gradient operator," 02 2014.

[63] J. Canny, "A computational approach to edge detection," vol. PAMI-8, no. 6, pp. 679–698. https://ieeexplore.ieee.org/document/4767851

[64] P. Hough, "Method and means for recognizing complex patterns," Patent, 12, 1962.

[65] L. Huang, Q. Wu, X. Wang, Z. Zhuo, and Z. Zhang, "Circle detection using a spiking neural network." IEEE, pp. 1442–1446. http://ieeexplore.ieee.org/document/6743901/

[66] A. Postl, "Towards automated atom manipulation in the scanning transmission electron microscope." https://utheses.univie.ac.at/detail/64452