Research paper

Micromagnetic tomography: Numerical libraries

David Cortés-Ortuño [a], Frenk Out [b], Martha E. Kosters [b], Karl Fabian [c], Lennart V. de Groot [b,*]

[a] *Departamento de Física, Universidad Técnica Federico Santa María, Avenida España 1680, Valparaíso, Chile*
[b] *Paleomagnetic laboratory Fort Hoofddijk, Department of Earth Sciences, Utrecht University, Budapestlaan 17, 3584 CD Utrecht, The Netherlands*
[c] *Department of Geoscience and Petroleum, Norwegian University of Science and Technology (NTNU), S. P. Andersens veg 15a, 7031 Trondheim, Norway*

ARTICLE INFO

ABSTRACT

Micromagnetic tomography (MMT) is an emerging technique in rock and paleomagnetism to determine individual magnetic moments of tomographically defined magnetic source regions within a natural sample by means of surface scans of the magnetic field above the sample. MMT relies on combining large high-resolution data sets from X-ray tomography and magnetic scanning devices, like quantum diamond magnetometers, together with advanced inversion algorithms potentially capable to solve for millions of individual magnetic moment vectors. We here provide an overview of existing algorithms that have been developed to tackle different aspects of MMT-related problems and discuss recent advances and future challenges of MMT.

1. Introduction

1.1. Micromagnetic tomography

Micromagnetic Tomography (MMT) is a novel technique to extract the remanent magnetization signal in rocks. Natural remanent magnetization (NRM) is initially imprinted in magnetic rocks at the time of their formation. Above an individual threshold temperature, the magnetic moments of magnetic grains in a rock statistically align with the direction and strength of the local Earth's magnetic field. On cooling, this orientation freezes in and forms a statistical magnetic signal recorded in the rock. In paleomagnetic studies, this signal is the main source of information to estimate direction and strength of the Earth's magnetic field from ancient to modern times. Standard paleomagnetic methods measure the bulk average over all magnetic grains in a rock sample. Hence, it is advantageous to select samples with narrow magnetic grain-size distributions with a majority of grains having optimal recording properties. Nevertheless, stepwise demagnetization of bulk magnetic measurements cannot always reliably separate primary from secondary, or stable from unstable remanence carriers (Dunlop and Özdemir, 1997). The MMT method solves this problem by analyzing the magnetic signal of a rock sample at the individual grain level. In particular, reliable magnetic carriers with similar properties, such as grain size or composition can be selected after the measurement to obtain optimal statistical ensembles for refined estimations of the Earth's magnetic field.

The MMT technique is founded in an experimental routine, a theoretical model and a computational implementation (de Groot et al.,

2018a, 2021). The experimental side of MMT requires measuring the locations and geometries of the magnetic grains in a rock sample using high-resolution computed tomography (Sakellariou et al., 2004). The current limit is about 50 nm voxel size in synchrotron facilities (Pattammattel et al., 2020). The second experimental step is to measure the magnetic signal generated by these grains on a surrounding surface using a highly sensitive magnetometer, such as a Quantum Diamond Microscope (QDM) (Glenn et al., 2017; Levine et al., 2019). The theoretical part of the MMT method is based on a forward model of the experimental setup, in which the magnetic flux, detected at the sensors of the magnetometer, is calculated from the magnetic potential of the grain sources in the measured magnetic sample. The corresponding inversion problem uses both, the defining forward matrix that depends on the locations of the magnetic sources, and the data from the magnetic signal of the sources. The solution of the inversion problem provides the magnetic dipole moments of the magnetic sources, which can be used to calculate their magnetization. The precondition that the inverse problem is well-posed is substantiated by a unique-source-assignment theorem (Fabian and de Groot, 2019) which requires that the source regions are topologically separated. In the MMT technique, the magnetic moment solutions of every magnetic source are obtained numerically from the theoretical model. This requires optimized inversion algorithms and routines to generate the forward matrix, in particular for large samples on length-scales that are relevant for paleomagnetic applications.

In this work, we discuss the theoretical and computational aspects of the MMT technique. Specifically, we focus on the numerical implementation of the forward model and compare the results using two

---

different formulations of the problem: a cuboid aggregation model of the grains assuming uniform magnetizations per grain, which means dipole order magnetic moment solutions; and a multipole expansion of the magnetic potential of the grains by modeling them as magnetic point sources (Cortés-Ortuño et al., 2021). On the one hand, this latter model produces more accurate magnetization solutions for grains hosting complex magnetic configurations, since higher order magnetic moments improve the fitting of their magnetic signal. On the other hand, this requires a larger number of fitting parameters, making the numerical solutions more challenging to calculate.

Finally, we discuss the implementation of a newly developed micromagnetic modeling technique of the magnetic grains (Cortés-Ortuño et al., 2022a). This method makes it possible to infer the magnetic configurations hosted in the grains by comparing multipole magnetic moment solutions of computational grain models, with the magnetic moments obtained from multipole inversion of experimental data.

2. Micromagnetic tomography and the inversion problem

MMT exploits knowledge about magnetic source locations and geometries acquired by computed tomography to define a unique inversion problem using a suitable forward model. Namely, a surface magnetic anomaly from the sources, obtained via magnetometry, uniquely determines the individual potentials of the sources if their location is known, and they are spatially disjointed, which is assured by the tomographic data. As a result, individual magnetic moments can be uniquely determined, with the limitation that a distribution of moments within a source cannot be resolved (Fabian and de Groot, 2019). There are two approaches to the forward model: (1) the homogeneously magnetized particle (HMP) approach uses an exact voxel or cuboid model of the source and assumes homogeneous magnetization; (2) the multipole expansion (MPE) approach uses only topological information about the source regions and represents the magnetic potential of each separate source region by a spherical harmonic or multipole expansion with a center inside the source region. In both approaches the forward calculation of the magnetic field on a planar or even curved surface above the source region is relatively straightforward. An important difference is that the MPE approach requires that all measurement points lie outside the near-field expansion regime of the spherical harmonic expansions.

In both forward models, the experimental setup is described by a scanning surface on which the magnetic anomalies produced by the grains are measured. This scanning surface here is devised as a regular grid, and the sensor systems detects the out-of-plane magnetic field component integrated over the sensor. For the HMP model the sensor is a rectangular area, and for the MPE model one can choose between point, rectangular and cuboid sensor systems.

Forward modeling can optionally account for imperfect co-registration between the two coordinate systems used for the tomography and the magnetic anomaly measurements. Altogether, there are three coordinate systems that need to be considered: (a) The tomographic voxel system VOX, (b) the magnetometer scanning system SCAN, and (c) the spherical harmonic expansion system EXP. The co-registration defines a generally nonlinear coordinate transform VOX→SCAN, $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3, (v_x, v_y, v_z) \mapsto (x, y, z)$. For instance, if the SCAN system is considered fixed and the VOX system is simply rotated around the out-of-plane SCAN axis by an angle θ_s, with respect to the $\mathbf{r}_{\text{rot}}$ reference point in SCAN, the systems can be co-registered by a transformation $T(\mathbf{r}_{\text{vox}}) = R(\theta_s) \cdot (\mathbf{r}_{\text{vox}} - \mathbf{r}_{\text{rot}})$, with $R(\theta_s)$ as a two-dimensional rotation matrix, applied to the coordinates of the voxel data. In the case of the MPE model, each multipole expansion requires a coordinate transform from the EXP system relative to the expansion center C to the SCAN system, $S : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3, (c_x, c_y, c_z, r, \theta, \phi) \mapsto (x, y, z)$. To retain linearity, it may be ideal to use an independent pre-processing step to optimize co-registration. For this work we do not focus on the details of the co-registration. We assume that the VOX

and SCAN coordinate systems are aligned. For the MPE system, the expansion is performed from the grain centers in the VOX system, and the spherical harmonic formalism is expressed in Cartesian coordinates, which is adequate to the rectangular geometry of the scan surface. Hence, the expansion depends only in the distance from the expansion centers to the sensor positions in the VOX system.

3. The homogeneously magnetized particle model

The first proof of concept of the MMT technique was applied to a synthetic sample containing a few magnetic particles (de Groot et al., 2018a). Here, computed X-ray micro-tomography was employed to compute the grain locations and grain profiles. Additionally, SQUID magnetometry was used to scan the magnetic field signal of the particles. By modeling the grains as homogeneously magnetized, the magnetic signal was numerically inverted into the grains to calculate their magnetizations. The HMP model and algorithms to process and invert this data have been developed in the `mmt_dipole_cuboid_inversion` library, and are detailed in the following section.

3.1. Theoretical model

Computed tomography data produces three-dimensional contrast images that can be used to distinguish different material compositions within a sample. By extracting the sections with magnetic material, individual magnetic grains can be distinguished via segmentation algorithms. The algorithms produce both the locations and three-dimensional shape profiles of the particles by converting single-pixel data into voxels. In order to optimize the inversions, the voxelated grain profiles can be processed with an aggregation algorithm, known as cuboid decomposition, that finds the largest possible cuboids within a geometry, substantially reducing the number of elements to describe a single grain.

Using the grains decomposed into cuboids, a forward model is formulated starting from the magnetic potential of the cuboids. With the assumption that cuboids belonging to a single grain are uniformly magnetized, the potential is computed only from the surface magnetic charges at the cuboid faces $\partial\Omega$. For an individual cuboid, the potential reads

$$\phi(\mathbf{r}) = \frac{1}{4\pi} \int_{\partial\Omega} \frac{\mathbf{M} \cdot d\mathbf{S}'}{|\mathbf{r} - \mathbf{r}'|}. \tag{1}$$
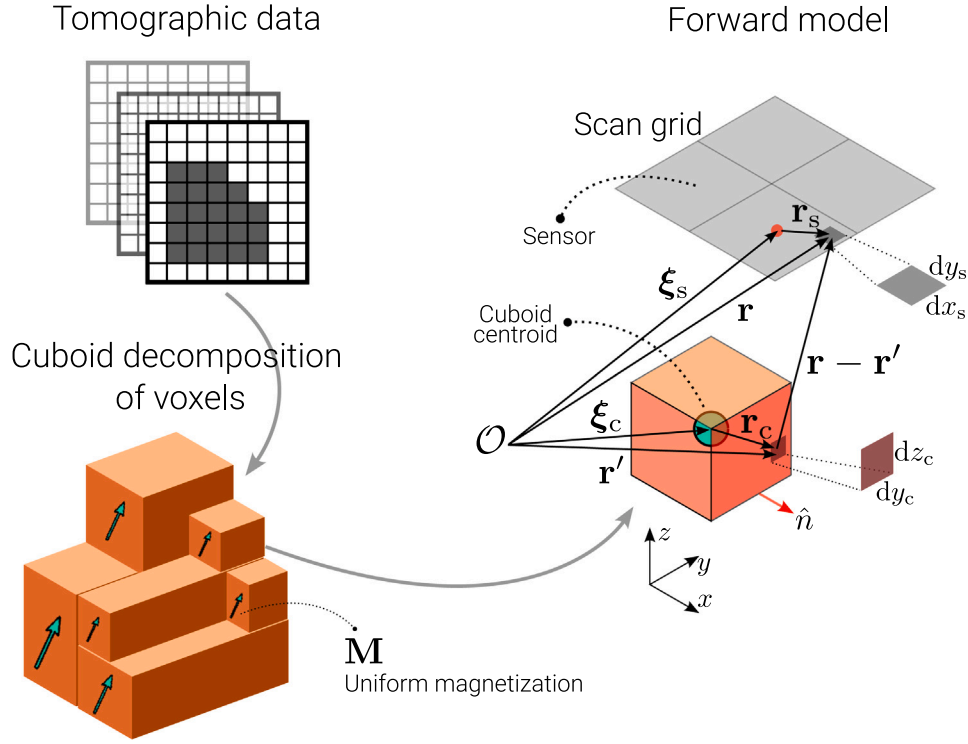
Here $d\mathbf{S}' = \hat{n} \, dS'$ is the surface element, with $\hat{n}$ as the unit vector normal to the cuboid faces, $\mathbf{M}$ is the cuboid magnetization, $\mathbf{r}$ is the location of the reference point, and the vectors $\mathbf{r}'$ point to the locations of the magnetic sources, which in this case means an infinitesimal element from a cuboid face. The magnetic field is defined as

$$\mathbf{B}(\mathbf{r}) = -\mu_0 \nabla\phi(\mathbf{r}). \tag{2}$$

A scanning magnetometer registers the magnetic signal from the magnetic particles in a two-dimensional grid made of individual sensors. The theoretical model for these sensors can be either a point source, a two-dimensional rectangle or a cuboid volume, which record the magnetic flux from the magnetic field of the grains. For the `mmt_dipole_cuboid_inversion` code, the sensor grid is defined in the xy-plane and the sensors are modeled as rectangles with area $\partial\Omega_s$ that register the magnetic flux normal to the sensor plane, i.e. the z-component of the magnetic field. This model reproduces the experimental setup used for the SQUID measurements in de Groot et al. (2018a). According to this, by defining $\gamma_B = \mu_0/(4\pi)$, and an infinitesimal sensor area vector $d\mathbf{S} = dx \, dy \, \hat{z}$, the field flux through a single sensor is calculated as

$$\Phi_{\text{flux}} = -\mu_0 \iint_{\partial\Omega_s} d\mathbf{S} \cdot \nabla\phi(\mathbf{r}) \tag{3}$$

$$= -\mu_0 \iint_{\partial\Omega_s} dx \, dy \, \frac{\partial}{\partial z} \left(\frac{1}{4\pi} \int_{\partial\Omega} \frac{\mathbf{M} \cdot d\mathbf{S}'}{|\mathbf{r} - \mathbf{r}'|} \right) \tag{4}$$

Fig. 1. Overview of the modeling of magnetic grains as aggregations of cuboids (HMP method) and the forward model used for a dipole order inversion. In the picture, grain data is obtained from tomographic images by converting two-dimensional pixel data into three-dimensional voxel data. This information is optimized by decomposing the voxels from a single grain into the largest possible cuboids that represent the grain geometry. For a single grain, the model assumes that all its cuboids have the same uniform magnetization. Finally, it is shown the coordinate system used to formulate the forward model for a single cuboid. In this model the magnetic field of an infinitesimal area of one face of the cuboid is detected by an infinitesimal area of a sensor unit in the scan surface of the magnetometer. An integration of the associated equations allows computing the total flux through the scan surface and to invert the magnetic signal into the magnetization of the grain. Because of the assumption of a uniform magnetization for a single grain, the magnetic moments from the inversion are only of dipole order.

$$= -\gamma_B \iint_{\partial\Omega_s} \mathrm{d}x\,\mathrm{d}y \int_{\partial\Omega} \mathbf{M} \cdot \mathrm{d}\mathbf{S}' \frac{\partial}{\partial z}\left(\frac{1}{|\mathbf{r}-\mathbf{r}'|}\right) \tag{5}$$

$$= \gamma_B \iint_{\partial\Omega_s} \mathrm{d}x\,\mathrm{d}y \int_{\partial\Omega} \mathbf{M} \cdot \mathrm{d}\mathbf{S}' \frac{\partial}{\partial z'}\left(\frac{1}{|\mathbf{r}-\mathbf{r}'|}\right) \tag{6}$$

In Eq. (6) the derivative variable was changed, which is more convenient for two of the cuboid faces. The integral limits can be simplified if the integration variable is changed to the coordinate system of the cuboid center $\boldsymbol{\xi}_c = (\xi_c, \eta_c, \zeta_c)$ by noticing that $\mathbf{r}' = \boldsymbol{\xi}_c + \mathbf{r}_c$, as shown in Fig. 1. In this case $\mathrm{d}\mathbf{S}' = \mathrm{d}\mathbf{S}_c$ and $\frac{\partial f}{\partial z'} = \frac{\partial f}{\partial z_c}$ for an arbitrary function f, hence the field can be expressed as

$$\Phi_{\text{flux}} = \gamma_B \iint_{\partial\Omega_s} \mathrm{d}x\,\mathrm{d}y \int_{\partial\Omega} \mathbf{M} \cdot \mathrm{d}\mathbf{S}_c \frac{\partial}{\partial z_c}\left(\frac{1}{|\mathbf{r}-(\boldsymbol{\xi}_c+\mathbf{r}_c)|}\right) \tag{7}$$

$$= \gamma_B \iint_{\partial\Omega_s} \mathrm{d}x\,\mathrm{d}y \int_{\partial\Omega} \mathbf{M} \cdot \mathrm{d}\mathbf{S}_c \frac{\partial}{\partial z_c}\left(\frac{1}{|\mathbf{X}-\mathbf{r}_c|}\right) \tag{8}$$

with $\mathbf{X} = \mathbf{r} - \boldsymbol{\xi}_c$. Alternatively, the field can also be written as

$$\Phi_{\text{flux}} = -\gamma_B \frac{\partial}{\partial z} \iint_{\partial\Omega_s} \mathrm{d}x\,\mathrm{d}y \int_{\partial\Omega} \mathbf{M} \cdot \mathrm{d}\mathbf{S}_c \left(\frac{1}{|\mathbf{X}-\mathbf{r}_c|}\right) \tag{9}$$

To perform the integrations with respect to the sensor area we follow a similar procedure as before, by changing the integration variables to the reference system at the sensor center. This means using the fact that $\mathbf{r} = \boldsymbol{\xi}_s + \mathbf{r}_s$ (see Fig. 1), with $\mathbf{r}_s = (x_s, y_s, 0)$, and integrating with respect to $\mathrm{d}x_s$ and $\mathrm{d}y_s$ within the sensor area limits. The fourfold integration can be calculated by referring to Hubert and Schäfer (1998, p. 122), based on systematic integrations of the source function $F_{000}(r) = 1/r$. Details about the solutions have been published

in the Supporting Information of de Groot et al. (2018a) and Cortés-Ortuño et al. (2022a). Notice that the magnetization in the flux term can be separated from the integration, for instance, the cuboid face plane normal to the x-direction contains the M_x term. For this case $\mathbf{r}_c = (\Delta x_c, y_c, z_c)$ and the flux is computed as

$$\Phi_{\text{flux}}^{\hat{n}=+\hat{x}} = -\gamma_B M_x F_{12-1}\left(\boldsymbol{\xi}_s + \mathbf{r}_s - \boldsymbol{\xi}_c - \mathbf{r}_c\right)\Big|_{x_s=-\Delta x_s}^{x_s=+\Delta x_s}\Big|_{y_s=-\Delta y_s}^{y_s=+\Delta y_s}\Big|_{z_c=-\Delta z_c}^{z_c=+\Delta z_c}\Big|_{y_c=-\Delta y_c}^{y_c=+\Delta y_c}$$

$$= -\gamma_B M_x F_{12-1}\left(\xi_s + x_s - \xi_c - \Delta x_c, \eta_s + y_s - \eta_c - y_c, \zeta_s - \zeta_c - z_c\right)$$

$$\times \Big|_{x_s=-\Delta x_s}^{x_s=+\Delta x_s}\Big|_{y_s=-\Delta y_s}^{y_s=+\Delta y_s}\Big|_{z_c=-\Delta z_c}^{z_c=+\Delta z_c}\Big|_{y_c=-\Delta y_c}^{y_c=+\Delta y_c}$$

We have defined here that the cuboid has dimensions $2\Delta x_c$, $2\Delta y_c$ and $2\Delta z_c$, and similarly for the two-dimensional sensor dimensions. The result for the face plane normal to the $-x$-direction follows the same structure. If we group the flux components from surface normals with opposite sign, with the corresponding magnetization terms in the same direction, M_x, M_y, and M_z, we can write, in general,

$$\Phi_{\text{flux}} = \sum_{\alpha \in \{x,y,z\}} \Phi_{\text{flux}}^\alpha M_\alpha, \tag{10}$$

where $\Phi_{\text{flux}}^\alpha = \Phi_{\text{flux}}^{\hat{n}=+\hat{\alpha}} + \Phi_{\text{flux}}^{\hat{n}=-\hat{\alpha}}$. For a magnetic particle decomposed into $\mathscr{C}$ cuboids having the same magnetization, the magnetic flux can be written as the sum of their contributions

$$\Phi_{\text{flux}} = \sum_{\alpha \in \{x,y,z\}} \left(\sum_{i=1}^{\mathscr{C}} \Phi_{\text{flux}}^{\alpha,i}\right) M_\alpha \tag{11}$$

For $\mathcal{N}$ magnetic grains, the flux contribution can be written as a matrix multiplication,

$$\boldsymbol{\Phi}_{\text{flux}} = \left[\sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_1^{x,i} \quad \sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_1^{y,i} \quad \sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_1^{z,i} \quad \sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_2^{x,i} \quad \cdots \quad \sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_{\mathcal{N}}^{z,i} \right] \cdot \overline{\mathbf{M}}, \quad (12)$$

with the column vector of $3\mathcal{N}$ magnetization components,

$$\overline{\mathbf{M}}^T = \left[M_{x,1} \, M_{y,1} \, \ldots \, M_{z,\mathcal{N}} \right]. \quad (13)$$

Since the surface measurement is made of $\mathscr{P}$ sensors, the total flux can be written as a forward matrix $\overline{\boldsymbol{\Phi}}$ of size $(\mathscr{P} \times 3\mathcal{N})$, with components $\sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_{p,q}^{\alpha,i}$ representing the flux contribution from the faces normal to the α-direction, from grain q, to the sensor p. Hence, every row of this matrix contains the contribution from all the magnetic sources, detected by a specific sensor p. Using the forward matrix, the inverse problem is expressed as

$$\overline{\boldsymbol{\Phi}} \cdot \overline{\mathbf{M}} = \begin{pmatrix} \sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_{1,1}^{x,i} & \sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_{1,1}^{y,i} & \cdots & \sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_{1,\mathcal{N}}^{z,i} \\ \sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_{2,1}^{x,i} & \ddots & & \vdots \\ \vdots & & & \\ \sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_{\mathscr{P},1}^{x,i} & \cdots & & \sum_{i=1}^{\mathscr{C}} \boldsymbol{\Phi}_{\mathscr{P},\mathcal{N}}^{z,i} \end{pmatrix} \begin{pmatrix} M_{x,1} \\ M_{y,1} \\ \vdots \\ \vdots \\ M_{z,\mathcal{N}} \end{pmatrix}$$

$$= \begin{pmatrix} B_1 \\ B2 \\ \vdots \\ \vdots \\ B_{\mathscr{P}} \end{pmatrix} = \overline{\mathbf{B}}, \quad (14)$$

with $\overline{\mathbf{B}}$ as a column vector of size $\mathscr{P}$ which has the scanning surface measurements for every sensor. If the number $\mathscr{P}$ of scanning sensor measurements is larger than the number of independent magnetization components $3\mathcal{N}$, a least squares minimization results in a linear inversion equation. Accordingly, the forward matrix $\overline{\boldsymbol{\Phi}}$ has a pseudo-inverse that can be used to calculate the magnetization vector. A standard method to compute the inverted matrix is the Moore–Penrose pseudo-inverse.

The HMP formalism allows the calculation of the magnetization of all the grains in the sample. Notice that this model only provides solutions at the dipole order for the magnetic potential, owing to the single-domain particle assumption. Therefore, the residual of the inverted forward field (Zhdanov, 2015)

$$\overline{\mathbf{B}}^{\text{res}} = \overline{\mathbf{B}}^{\text{inv}} - \overline{\mathbf{B}} \quad (15)$$

will exhibit strong anomalies of multipole character if the grains are not uniformly magnetized, which occurs frequently for grains above the pseudo-single-domain size range (Cortés-Ortuño et al., 2021). The residual error of the magnetic field data can be quantified using the relative error, B_{err}, by computing the Frobenius norm $\lVert \cdot \rVert_F$ of the field matrices as

$$B_{\text{err}} = \frac{\left\lVert \overline{\mathbf{B}}^{\text{res}} \right\rVert_F}{\left\lVert \overline{\mathbf{B}} \right\rVert_F}. \quad (16)$$

3.2. Code

The mmt_dipole_cuboid_inversion library (Out et al., 2023) is written in Python and contains a main class named DipoleCuboidInversion to load both tomographic data and magnetometry data. The former, ideally, has to be pre-processed with a cuboid decomposition algorithm to speed up the inversion. The mmt_dipole_cuboid_inversion library contains a module to decompose a grain from raw pixel format to coarser cuboids, for a single grain, as shown in Code 1. For a full sample, either the raw or decomposed-cuboid data from all the grains has to be compiled into a single file with seven columns: x y z dx dy dz index, which are the cuboid center

```
1 from mmt_dipole_cuboid_inversion.tools
      import CuboidDecomposition
2
3 # Process a single grain in a text file
4 my_data = 'grain_voxel_data.txt'
5 CuboidDecomposition(my_data,
      'cuboid_decomposed_data.txt',
      format_output=True)
```

Code 1: Code snippet for the decomposition of the voxel representation of a grain into the largest possible cuboids that represent the same grain geometry.

coordinates, the half lengths in the three spatial directions and the index of the grain where the cuboid belongs to.

In the case of the magnetometry data, it must be formatted as a text file or two-dimensional Numpy array that can be passed to the read_files method of the DipoleCuboidInversion class. In addition to the data files, the following properties of the scanning magnetometer are required when instantiating the class: the scan domain (either from the sensor center positions or the scan surface limits), the distances between two adjacent sensors from their centers, the sensor area and the magnetometer height from the top of the sample, which is defined with position $z = 0$.

A diagram with the structure of the mmt_dipole_cuboid_inversion library is shown in Fig. 2. The main module, dipole_cuboid_inversion, contains the inversion class DipoleCuboidInversion which can call three different methods to populate the forward matrix. Two of these methods are parallelized using OpenMP and CUDA (if available) and are compiled using the Cython compiler (Behnel et al., 2011). The third method is compiled from C to Python using the Numba compiler (Lam et al., 2015). For the inversion, the main class contains Moore–Penrose pseudo inverse algorithms from the Numpy and Scipy libraries, and a least squares solution via LAPACK libraries. These methods can be extended in the future using other singular value decomposition algorithms, such as dgesvj based on Jacobi plane rotations, from the LAPACK library. The tools modules contain the cuboid decomposition code and several functions to plot the sample and the inversion results. The inversion_tools module was developed by Out et al. (2022) to analyze the accuracy of the inversion solutions via a covariance matrix, when the noise level of the field signal is known.

In Code 2 we show an example of using the DipoleCuboidInversion class to invert a 102 μm × 102 μm scanning surface measuring the signal of two single dipole point sources. The surface domain, in this case, is specified using the coordinates of the center of the lower left and upper right sensors. The library also allows specifying the scan domain using the corners of the full surface. The first dipole is located at a depth of $z = -10$ μm, has a volume of 8 μm³ and is oriented in the $(0, 1, 1)$ direction, with a saturation magnetization of $M_{\text{s}} = 0.48$ MA/m. The second dipole is below the first one at a depth of $z = -20$ μm, with a volume of 1 μm³ and oriented in the $(-1, -1, 0)$ direction. The magnetic field of the dipoles is calculated analytically at every sensor center position of the scanning surface located at $z = 2$ μm. The inversion considers the average signal over a sensor area of 2 μm × 2 μm, thus a small error between the inverted magnetizations is expected. For the inversion, two cubes are specified centered at the dipole positions with corresponding side lengths to match the dipole volumes. This information is stored in the files ending in _cuboids.txt. The population of the forward matrix that is inverted, is specified with the OpenMP optimization, and the inversion is done using the Scipy library.

For the case of Code 2, the relative error for the inverted magnetizations, compared to the true value, is 1.25% for dipole 1 and 12.45% for dipole 2. The larger error of the second dipole is caused by the dipole on
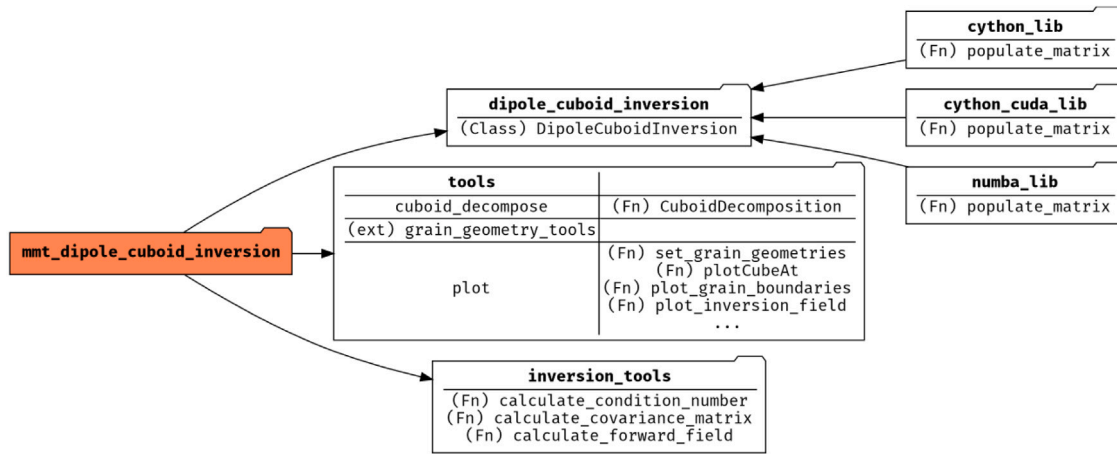
Fig. 2. Diagram for the code structure of the `mmt_dipole_cuboid_inversion` Python library. Every node represents a module (which is either a directory or a file) named in bold letters. Each module includes: submodules, functions specified with the (Fn) label, classes specified with the (Class) label, and external modules specified with the (ext) label. For this library, three different optimization modules are available for the `DipoleCuboidInversion` class, which can be called from one of its methods.
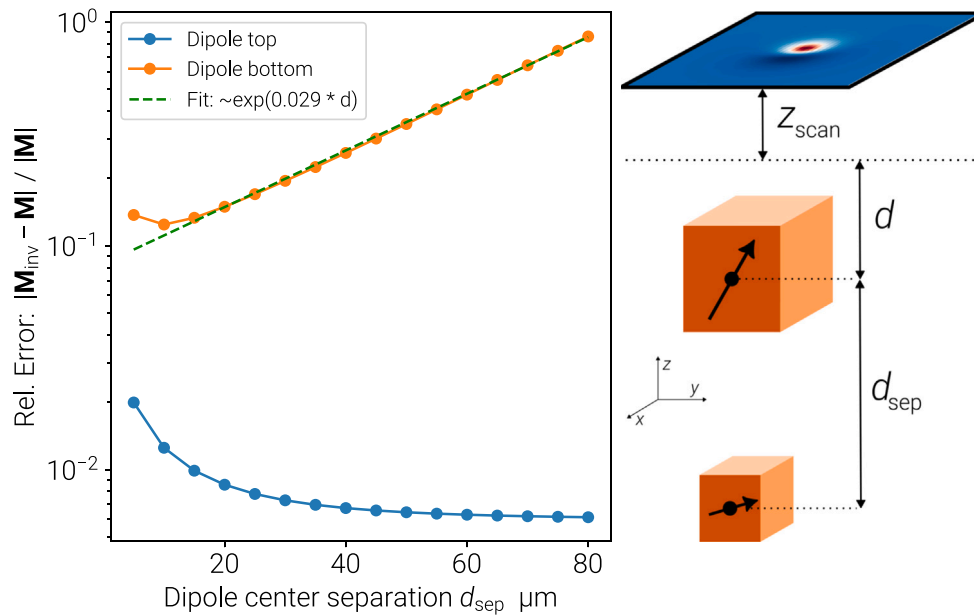

Fig. 3. Relative error of the norm of the inverted magnetization of two single dipoles, as a function of their separation. In the model, one dipole is located directly above the other. The dipole at the top is kept fixed at $d = 10$ μm, while the dipole below it is moved deeper into the sample. The two dipoles have a magnetization of 0.485 MA/m with volumes equivalent to cuboids of 1 μm³ for the bottom dipole, and 8 μm³ for the top dipole. The scan surface is located at a height of $z_{scan} = 2$ μm. The magnetic signal is computed analytically from the dipole field of the magnetic sources into a square surface of side length 102 μm divided into a grid of point sensors spaced by 2 μm distance. The performance of the `DipoleCuboidInversion` class is tested by numerically inverting the surface signal to obtain the magnetizations of the dipoles. The bottom dipole signal exhibits an exponential error due to an ill-conditioned forward matrix caused by its weak signal at the scan surface.

top of it, which has a larger volume and shadows its signal, generating an ill-conditioned matrix. Code 2 has been utilized to observe the error tendency as the distance of dipole 2 from dipole 1, increases. The result is illustrated in Fig. 3, where we observe that the relative error for dipole 2 increases exponentially, while it decreases for dipole 1 in a similar exponential fashion for depths above 10 μm. An analysis of the condition number of the forward matrix reveals that it also increases exponentially with the depth of dipole 2.

4. The multipole expansion method

To improve the accuracy of the inverted magnetization solutions, particularly from complex magnetic anomalies, Cortés-Ortuño et al. (2021) developed a multipole expansion (MPE) of the magnetic potential by considering that the scanning surface measurements are sufficiently far away from the magnetic sources. The MPE uses fully orthogonal (see the Supporting Information of Cortés-Ortuño et al. (2021)) real spherical harmonic polynomials. These polynomials are obtained by changing the non-orthogonal Maxwell-Cartesian polynomial basis that arises from the far-field expansion of the magnetic potential, and which might lead to numerical instabilities. Using the orthogonal basis, a numerical inversion computes quadrupole, octupole and higher order magnetic moments per grain. For the MPE the magnetic grains are modeled as point sources, thus only the location of the grains from the tomographic data is necessary. On the other hand, to compute the magnetizations the grain volumes are still required. In contrast to the HMP method, the MPE method introduces more free parameters (magnetic moments) that need to be fitted during the inversion. Nonetheless, it can reconstruct complex magnetic signals more accurately under certain conditions, such as grains being sufficiently close to the measurement sample surface having a strong signal. Recent experimental setups have set the scanning surface 2 μm (SQUID, de

```
 1 import numpy as np
 2 import mmt_dipole_cuboid_inversion as dci
 3 from pathlib import Path
 4
 5 # Data generated by the generate_two_dipoles.py script:
 6 datadir = Path("./two_dipoles_sep_data")
 7 dipole_sep = 10   # micro-m
 8 scanfile = datadir / f'two_dipoles_sep_{dipole_sep:02d}_Bzgrid.txt'
 9 cuboidfile = datadir / f'two_dipoles_sep_{dipole_sep:02d}_cuboids.txt'
10
11 # Use the coords of lower-left and upper-right sensors to define scan surface:
12 sensor_domain = np.array([[0, 0], [100., 100.]]) * 1e-6
13 (scan_spacing, scan_deltax, scan_deltay,
14  scan_area, scan_height) = (2e-6, 1e-6, 1e-6, 4e-12, 2e-6)
15 # First parameter contains the surface scan corner coords, but here we
16 # use the sensor positions instead:
17 dip_inversion = dci.DipoleCuboidInversion(None, sensor_domain, scan_spacing,
18                                           scan_deltax, scan_deltay,
19                                           scan_area, scan_height)
20
21 dip_inversion.read_files(scanfile, cuboidfile, cuboid_scaling_factor=1e-6)
22 dip_inversion.set_scan_domain(gen_sd_mesh_from='sensor_center_domain')
23 dip_inversion.prepare_matrix(method='cython')
24 dip_inversion.calculate_inverse(method='scipy_pinv')
```

Code 2: Code snippet for the inversion of the two dipoles system of Fig. 3, using the `mmt_dipole_cuboid_inversion` library.
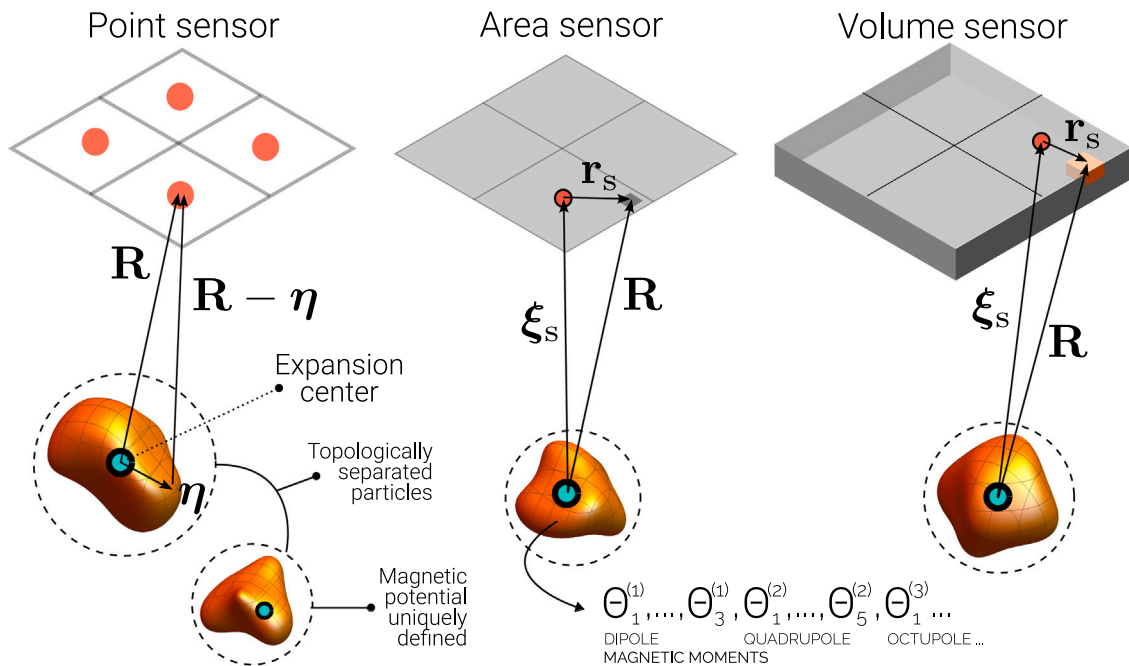

Fig. 4. Overview of the forward model for the MPE formalism. The coordinate system is shown with the origin located at an expansion center specified at the centroid of the grains. Three different models for the scan surface sensors are depicted, based on the sensor dimensions. For a point sensor, a distance **R** from the source to the sensor center is used. For larger sensor dimensions, this distance points to an infinitesimal sensor area or volume, and it is easier to integrate the magnetic flux equations with respect to the distance to the sensor center ξ_s and sensor center to sensor unit $\mathbf{r}_s$. Numerical inversions of the scan surface signal, using the `mmt_multipole_inversion` code, are used to calculate dipole and higher order magnetic moments for every grain in a sample. Unique individual magnetic moment solutions for every grain are guaranteed by topologically separated grain volumes, *i.e.* with non-overlapping grain bounding spheres.

Groot et al. (2018a)) or 6 μm (QDM, Kosters et al. (2023)) above the sample surface, with grains located from 6 to 50 μm below the sample surface. For these cases grains are sufficiently separated from the scan surface and a far-field multipole expansion can be applied. Although shallow grains usually exhibit a strong signal that is useful for reliable inversions, if they are located too close to the surface, such that a sphere enclosing their volume reaches the scan surface, a near-field multipole expansion might be more accurate.

The forward model for the MPE method starts from the magnetic potential of Eq. (1), considering a distribution of volume charges from a magnetic source rather than the magnetic surface charges. A multipole expansion is more effectively described with respect to a reference point within the magnetic source, for instance, its geometric center, as shown in Fig. 4. Based on this, it is possible to write the denominator of Eq. (1) as $|\mathbf{R} - \boldsymbol{\eta}|$, with $\mathbf{R} = (x, y, z)$ and $\boldsymbol{\eta}$ as the positions of the sensor center and an infinitesimal volume charge, respectively, measured

$$\mathbf{P}_{z|i,j} = \left(\begin{array}{cccc} \frac{3xz}{R^5}, & \frac{3yz}{R^5}, & \frac{3z^2-R^2}{R^5}, \vdots & \sqrt{\frac{3}{2}}z\frac{5z^2-3R^2}{R^7}, \\[2mm] -\sqrt{2}x\left(\frac{R^2-5z^2}{R^7}\right), & -\sqrt{2}y\left(\frac{R^2-5z^2}{R^7}\right), & \frac{5}{\sqrt{2}}\frac{z(x^2-y^2)}{R^7}, & 5\sqrt{2}\frac{xyz}{R^7}, \vdots \\[2mm] \frac{1}{\sqrt{10}}\frac{(3R^4+35z^4-30R^2z^2)}{R^9}, & \frac{\sqrt{15}xz}{2}\frac{(7z^2-3R^2)}{R^9}, & \frac{\sqrt{15}yz}{2}\frac{(7z^2-3R^2)}{R^9}, & -\sqrt{\frac{3}{2}}\frac{(R^2-7z^2)}{R^9}(x^2-y^2), \\[2mm] -\sqrt{6}xy\frac{(R^2-7z^2)}{R^9}, & \frac{7}{2}xz\frac{(x^2-3y^2)}{R^9}, & -\frac{7}{2}yz\frac{(y^2-3x^2)}{R^9} & \end{array}\right) \tag{20}$$

Box I.

from the expansion center. For this case, we are considering that the sensors are point detectors. A far-field expansion of the potential of a single magnetic grain, when $|\mathbf{R}| = R \gg \eta = |\boldsymbol{\eta}|$, results in

$$\phi(\mathbf{R}) = \frac{1}{4\pi}\sum_{n=1}^{3}\sum_{\alpha=1}^{2n+1}\Theta_\alpha^{t(n)}Q^{\alpha(n)}(\mathbf{R}) + \mathcal{O}\left(\eta^4\right). \tag{17}$$

The sum shows the expansion terms up to octupole order. The term $\Theta_\alpha^{t(n)}$ denotes the components of the traceless magnetic multipole tensor of rank n, which contains the magnetic moment information about the magnetic source. The functions $Q^{\alpha(n)}(\mathbf{R})$ represent the orthogonal spherical harmonic polynomials proportional to $R^{-(n+1)}$. Notice that $\boldsymbol{\Theta}^{(1)} = \left(\Theta_1^{t(1)}, \Theta_2^{t(1)}, \Theta_3^{t(1)}\right)$ is the magnetic dipole moment vector that defines the net grain magnetization $\mathbf{M} = \boldsymbol{\Theta}^{(1)}/V$, where V is the grain volume.

The magnetic field is, therefore, computed from the gradient of the potential and contains nth-order multipole terms that decay as $R^{-(n+2)}$. If the magnetic sensor is modeled as a point sensor detecting the field component normal to the scanning surface in the xy-plane, the field is obtained by the z-derivative of Eq. (17). According to this, for a sample with $\mathcal{N}$ magnetic grains and $\mathcal{P}$ sensors, the forward model in block matrix notation reads

$$\overline{\boldsymbol{\Phi}} \cdot \overline{\boldsymbol{\Theta}} = \gamma_B \begin{pmatrix} \mathbf{P}_{z|1,1} & \mathbf{P}_{z|1,2} & \cdots & \mathbf{P}_{z|1,\mathcal{N}} \\ \mathbf{P}_{z|2,1} & \mathbf{P}_{z|2,2} & & \vdots \\ \vdots & & \ddots & \\ \mathbf{P}_{z|\mathcal{P},1} & \cdots & & \mathbf{P}_{z|\mathcal{P},\mathcal{N}} \end{pmatrix}\begin{pmatrix} \boldsymbol{\Theta}_1 \\ \boldsymbol{\Theta}_2 \\ \vdots \\ \boldsymbol{\Theta}_{\mathcal{N}} \end{pmatrix} = \begin{pmatrix} B_1 \\ B2 \\ \vdots \\ B_{\mathcal{P}} \end{pmatrix} = \overline{\mathbf{B}}. \tag{18}$$

In Eq. (18), each magnetic moment tensor $\boldsymbol{\Theta}_i$, corresponding to grain i, in the column vector $\overline{\boldsymbol{\Theta}}$ contains $n(n+2)$ components, depending on the maximal multipole order n. The $\overline{\mathbf{B}}$ column vector contains the scanning signal at every sensor. And $\mathbf{P}_{z|i,j}$ is the row vector containing the z-component of the gradient of the spherical harmonic polynomials, at sensor i from grain j, and which has $n(n+2)$ components. Specifically, it is defined as

$$\overline{\boldsymbol{\Phi}}_{i,j} = \gamma_B \mathbf{P}_{z|i,j} = -\gamma_B \frac{\partial \mathbf{Q}_{i,j}}{\partial z}. \tag{19}$$

For an octupole order inversion, i.e. $n = 3$, the gradient of the polynomial row vector has 15 elements and reads (see Eq. (20) which is given in Box I).

Up to octupole order, $n = 3$, the multiplication of the sub-matrices in Eq. (18), using Eq. (19), means

$$\mathbf{P}_{z|i,j} \cdot \boldsymbol{\Theta}_j = -\sum_{n=1}^{3}\sum_{\alpha=1}^{2n+1}\frac{\partial Q_{i,j}^{\alpha(n)}}{\partial z}\Theta_{\alpha|j}^{t(n)}. \tag{21}$$

In this model the $\overline{\boldsymbol{\Phi}}$ matrix has size $(\mathcal{P}, \mathcal{N} \times n(n+2))$. Equivalently, in block notation it has size $(\mathcal{P}, \mathcal{N})$.

In the case that the sensor is modeled as a two-dimensional rectangle of area $\partial\Omega_s$, detecting the out of plane magnetic signal from the grains, the $\mathbf{R}$ vector will point to an infinitesimal unit of sensor area (see Fig. 4). In this model, the flux matrix block components change to

$$\overline{\boldsymbol{\Phi}}_{i,j} = \gamma_B \iint_{\partial\Omega_s}\mathrm{d}x\,\mathrm{d}y\,\mathbf{P}_{z|i,j} = -\gamma_B \iint_{\partial\Omega_s}\mathrm{d}x\,\mathrm{d}y\,\frac{\partial\mathbf{Q}_{i,j}}{\partial z}$$

$$= -\gamma_B \frac{\partial}{\partial z}\iint_{\partial\Omega_s}\mathrm{d}x\,\mathrm{d}y\,\mathbf{Q}_{i,j}. \tag{22}$$

As in the case of the cuboid aggregation model, it is easier to switch to a coordinate system with respect to the sensor center by defining $\mathbf{R} = \boldsymbol{\xi}_s + \mathbf{r}_s$, with $\mathbf{r}_s = (x_s, y_s, 0)$, and integrating within the sensor limits. The flux now has units of Tesla per meter squared, hence the average flux within the sensor can be computed by averaging by the sensor area.

Similarly, the sensor can be modeled as a three-dimensional cuboid. Again, we change to the coordinate system of the sensor center, where an infinitesimal unit of sensor volume is described with $\mathbf{r}_s = (x_s, y_s, z_s)$. The simplest case is where only the z-component of the field is detected within the cuboid volume defining the sensor, Ω_s. As a result, the flux matrix components become

$$\overline{\boldsymbol{\Phi}}_{i,j} = \gamma_B \iiint_{\Omega_s}\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z\,\mathbf{P}_{z|i,j} = -\gamma_B \iiint_{\Omega_s}\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z\,\frac{\partial\mathbf{Q}_{i,j}}{\partial z}. \tag{23}$$

Note that the z-integral can be simplified if the spherical harmonic polynomials are used instead of their gradient. The integrals in Eqs. (22) and (23) are not straightforward and could only be computed up to quadrupole order using a symbolic computation library, such as Wolfram Mathematica.

4.1. Code

The MPE formalism has been implemented in the Python library `mmt_multipole_inversion` (Cortés-Ortuño et al., 2022c), which contains two main classes. One of them is the `MagneticSample` class, which generates samples of point magnetic sources with given or random locations and magnetic moments, and which specifies the properties of the scanning surface. This information can be saved in Numpy arrays and `json` files to be inputted into the second main class of the library called `MultipoleInversion`. This class is the core of the `mmt_multipole_inversion` library because it calculates the magnetic moments of all the grains in a sample by computing the forward flux matrix and solving the inversion problem numerically. The `MultipoleInversion` module can both accept the files from the `MagneticSample` class or work independently. The latter means specifying the scan and sample properties directly from the user. The population of the forward flux matrix can be done using a vectorized code based on the Numba library (Lam et al., 2015), or a highly parallel GPU code written in the CUDA language.

A diagram of the `mmt_multipole_inversion` code is shown in Fig. 5. The `susceptibility_modules` contain the functions to populate the forward matrix using different polynomial basis and scan sensor models. The latter are currently implemented for the spherical harmonics polynomial basis only, since it is fully orthogonal. A similar design is implemented for the field functions of the `magnetic_sample` modules. A `multipole_field` module includes functions to calculate the radial component of the magnetic field at different multipole orders if the corresponding multipole moments are known. These functions have been used by Cortés-Ortuño et al. (2021) to show the radial field around a bounding sphere containing the grains, since this field contains all the information about the magnetic sources within the
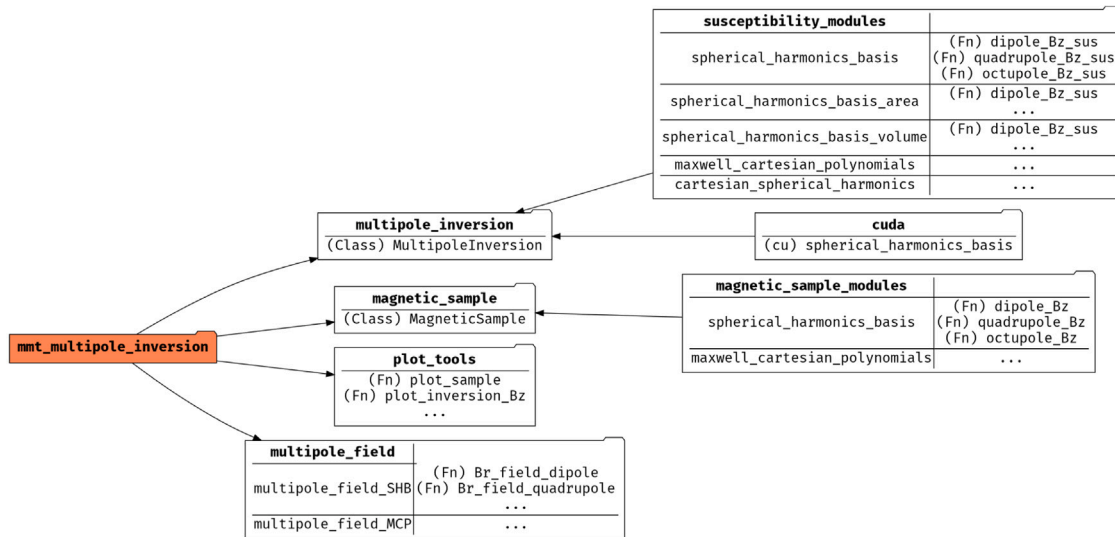
Fig. 5. Diagram for the code structure of the `mmt_multipole_inversion` Python library. Every node represents a module (which is either a directory or a file) named in bold letters. Each module includes: submodules, functions specified with the (Fn) label and classes specified with the (Class) label. For the multipole inversion, the susceptibility matrix can be computed via functions optimized with the Numba library, which are in the `susceptibility_modules` module, or via a function optimized with CUDA. The former implements three different models for the sensors: point, area and volume; and three different polynomial basis: fully orthogonal spherical harmonics, Maxwell Cartesian polynomials, and spherical harmonics directly transformed to the Cartesian coordinate system. Functions to compute the radial field using different polynomial basis are contained in the `multipole_field` module.

sphere and uniquely determines the spherical harmonic expansion of the potential. Plotting functions for the inversion results are provided in the `plot_tools` module.

In Code 3 we show the inversion of a quasi-quadrupole from a source defined at the location $(10, 10, -10)$ in micrometer units, with a scanning surface of $20\,\mu\text{m} \times 20\,\mu\text{m}$ located at $z = 1\,\mu\text{m}$. The quadrupole field is generated using the `MagneticSample` class to define two dipole sources at the x-axis separated by a $2\,\mu\text{m}$ distance. These dipoles are oriented in the $\pm\hat{y}$ directions and have magnetic moments of $|\Theta^{(1)}| = 4.8 \times 10^5 \times 10^{-18}\,\text{A m}^2$ magnitude, which are associated to particles of $1\,\mu\text{m}^3$ volume. Then, for the inversion, we define a single magnetic source of $1\,\mu\text{m}^3$ volume, with an expansion center at the middle of the sample, instead of the two dipoles, with the idea of obtaining quadrupole moment solutions (see Code 3). The data is saved in two files: a `json` file containing a dictionary with the scanning surface properties; and a Numpy `npz` file containing arrays with the particle positions, the magnetic signal and, optionally, other information about the system that is not required for the inversion. These arrays can also be assigned manually in a class instance.

To perform the inversion we load the data saved previously and use a quadrupole order expansion. Notice that the inversion is specified first using the function argument `'spherical_harmonics_basis'`, which means the expansion of the magnetic potential is done with the orthogonal spherical harmonic polynomials. For this option the sensors are modeled as point detectors, thus the signals are measured at every sensor center. The resulting inverted magnetic moments are given by the array `[-4.00e-17, -8.20e-17, 6.61e-19, -2.98e-24, 1.57e-22, 3.21e-22, -1.23e-24, -1.95e-18]`. The first three values are the dipole magnetic moments. Using the source's volume, its magnetization has magnitude of $|\mathbf{m}| = 91.23\,\text{A/m}$. Compared to the magnetization of each of the original two dipole sources, which were defined with a magnetization of $4.8 \times 10^5\,\text{A/m}$, the inverted magnetization of the single source is negligible. The same applies for its dipole moments since the single source was define with the same volume of each of the original sources. The next five values in the array are the quadrupole tensor components, where the latest has the largest magnitude, and which is the prefactor of the field polynomial $\partial Q^{5(2)}/\partial z = 5\sqrt{2}xyz/R^7$ (see Eq. (20)). We assess the error of the inverted field, compared to the measurement, using its

relative error given by Eq. (16), which results in a small value of $B_{\text{err}} = 4.52\%$ for both polynomial basis. The quadrupolar field is given primarily by a single quadrupole moment, however a physical interpretation of its associated polynomial term is not straightforward. In the code we have implemented other polynomial basis, for instance the Maxwell Cartesian polynomials, that are specified with the argument `'maxwell_cartesian_basis'`. On the one hand, these polynomials are not orthogonal (for quadrupole and higher orders) and, thus, they are not numerically stable (Cortés-Ortuño et al., 2021). On the other hand, they can be used to interpret simple models. In this example, they generate the inverted magnetic moment values `[-4.00e-17, -8.20e-17, 6.61e-19, 3.46e-25, -1.38e-18, 1.11e-22, 2.09e-24, 2.27e-22]`. The second quadrupole component, $\Theta_{xy}^{(2)} = -1.38\text{e}{-18}\,\text{A m}^3$ (in the non-orthogonal basis; see Cortés-Ortuño et al. (2021)), is the largest and corresponds to a pure quadrupole oriented in the xy-direction, which is the case here, and thus proves that the inversion is producing the right components of the multipole tensor.

5. Benchmark problem: synthetic sample

We define a benchmark problem to compare the performance of our two inversion codes using the open data of a synthetic sample investigated by de Groot et al. (2018a) and Cortés-Ortuño et al. (2021), and which is hosted in de Groot et al. (2018b). We set this problem as a reference for the verification of the performance of new inversion codes, and alternative inversion algorithms.

The synthetic sample contains magnetite grains with diameters ranging from $5\,\mu\text{m}$ up to $35\,\mu\text{m}$. The data includes both the locations and sizes of the aggregations of cuboids composing the magnetic grains, which were obtained via microCT. The cuboid data was processed with the cuboid decomposition algorithm in order to optimize the number of elements describing a single grain. The magnetic signal image data was obtained using a SQUID magnetometer and co-registered with a bird's-eye view of an image of the sample depicting the grains. Three areas of the magnetic signal image were selected and studied in de Groot et al. (2018a), and here we focus on the so-called Area 1 for the case of unknown magnetic states. An overview of the sample and chosen scanning surface area are shown in Fig. 6.

```python
 1 from mmt_multipole_inversion import MagneticSample
 2 from mmt_multipole_inversion import MultipoleInversion
 3 import numpy as np
 4
 5 # Scan height, Scan area x and y, sensor half-legth x and y (meter)
 6 Hz, Sx, Sy, Sdx, Sdy = 2e-6, 20e-6, 20e-6, 0.1e-6, 0.1e-6
 7 Lx, Ly, Lz = Sx * 0.9, Sy * 0.9, 5e-6  # Sample lengths (meter)
 8 hLx, hLy, hLz = sample.Lx * 0.5, sample.Ly * 0.5, sample.Lz * 0.5
 9
10 # Initialise class to create a sample (with user-defined or random particles)
11 sample = MagneticSample(Hz, Sx, Sy, Sdx, Sdy, Lx, Ly, Lz)
12
13 Ms = 4.8e5
14 # Set the positions and dipole magnetic moments of two dipoles
15 # Put them at the middle of the sample, separated in the x-axis
16 dipole_positions = np.array([[hLx - 1e-6, hLy, -hLz],
17                              [hLx + 1e-6, hLy, -hLz]])
18 # Magnetic moments
19 mu_s = Ms * (1 * 1e-18) * np.array([[0., 1., 0], [0., -1, 0]])
20 volumes = np.array([1e-18, 1e-18])
21
22 sample.generate_particles_from_array(dipole_positions, mu_s, volumes)
23 # Generate the dip field: the Bz field flux through the measurement surface
24 sample.generate_measurement_mesh()
25
26 # Redefine positions to make a single particle at the centre (ideal quadrupole)
27 sample.dipole_positions = np.array(
28     [[sample.Lx * 0.5, sample.Ly * 0.5, -sample.Lz * 0.5]])
29 # Update the N of particles in the internal dict
30 sample.N_particles = 1
31
32 sample.save_data(filename='quadrupole_y-orientation')
33
34 # Inversions -----------------------------------------------------------
35
36 shinv = MultipoleInversion('./MetaDict_quadrupole_y-orientation.json',
37                            './MagneticSample_quadrupole_y-orientation.npz',
38                            expansion_limit='quadrupole',
39                            sus_functions_module='spherical_harmonics_basis')
40 shinv.generate_measurement_mesh()
41 shinv.compute_inversion(method='sp_pinv')
42
43 mcinv = MultipoleInversion('./MetaDict_quadrupole_y-orientation.json',
44                            './MagneticSample_quadrupole_y-orientation.npz',
45                            expansion_limit='quadrupole',
46                            sus_functions_module='maxwell_cartesian_polynomials')
47 mcinv.generate_measurement_mesh()
48 mcinv.compute_inversion(method='sp_pinv')
```

Code 3: Code for the inversion of the magnetic signal of an ideal quadrupole point source using the `mmt_multipole_inversion` library.

We show in Code 4 the script to perform the inversion of the chosen scanning surface area using the `mmt_dipole_cuboid_inversion` library. The sensors of the SQUID measurement have a square geometry with 1 μm side length, and are separated by a distance of the same dimension. The scan surface has a size of 350 μm × 200 μm and is located at 2 μm above the sample. The cuboid and scanning surface data are already formatted for the `DipoleCuboidInversion` class, thus the piece of code for the inversion requires only a few lines. In this case, we have used Scipy's pseudo-inverse function.

For the MPE inversion code we have also modeled the SQUID magnetometer sensors as squares by specifying the argument `'spherical_harmonics_basis_area'`, in order to make the results comparable to those of the cuboid aggregation model. The scanning surface and sensor dimensions are defined in a Python dictionary containing the properties of the scanning surface and sensors, which is stored in a `json` file. For the code we use dipole and quadrupole inversions. For the MPE inversions, a small cutoff for the singular values is particularly important because of the small magnitude of

higher order multipole moments. In the `'scipy_pinv'` method, a relative threshold can be specified by the `rtol` parameter, which is proportional to the maximum singular value of the forward matrix.

A comparison of the results of the two codes, together with the original results published by de Groot et al. (2018a), are shown in Code output 6. Both the magnetization strength and spherical angles of the magnetization are computed. For a dipole order inversion, the three codes are in agreement. In Cortés-Ortuño et al. (2021) it was shown that large residuals are observed in the inverted magnetic signal when using a dipole approximation for the forward model. Although, in principle, a quadrupole expansion should improve the inverted magnetization values, this optimization is affected by multiple factors. Firstly, both grain depth and size, which affect the grain signal. Shallow grains usually have a stronger signal but the far-field approximation fails if they are too close to the measurement surface. Larger grains might have stronger signal, although it might be weaker if they host magnetic structures with a flux-closure pattern, such as three-dimensional vortices. Secondly, grain superposition, i.e. one grain above another,

```python
1  from pathlib import Path
2  import mmt_dipole_cuboid_inversion as dci
3  import numpy as np
4
5  # Load data from the synthetic sample of de Groot et al. (2018)
6  data_dir = Path('deGroot2018_data/PDI-16803')
7
8  # Location and name of SQUID scan data, and tomog. cuboid file
9  ScanFile = data_dir / 'Area1-90-fig2MMT.txt'
10 CuboidFile = data_dir / 'FWInput-FineCuboids-A1.txt'
11 # Define sensor area using sensor center positions
12 SQUID_sensor_domain = np.array([[0, 0], [350, 200]]) * 1e-6
13 SQUID_spacing = 1e-6
14 SQUID_deltax, SQUID_deltay, SQUID_area = 0.5e-6, 0.5e-6, 1e-12
15 SQUID_height = 2e-6
16 # Use lower-left and upper-right sensors to define the scan domain
17 mag_inv = dci.DipoleCuboidInversion(
18     None, SQUID_sensor_domain, SQUID_spacing,
19     SQUID_deltax, SQUID_deltay, SQUID_area, SQUID_height)
20 # Read files and define scan area
21 mag_inv.read_files(ScanFile, CuboidFile, cuboid_scaling_factor=1e-6)
22 mag_inv.set_scan_domain(gen_sd_mesh_from='sensor_center_domain')
23 # We then compute the forward (Green's) matrix to be inverted
24 mag_inv.prepare_matrix(method='cython')
25 # And we do the inversion
26 mag_inv.calculate_inverse(method='scipy_pinv', rtol=1e-20)
```

Code 4: Code for the inversion of Area 1 of the synthetic sample of de Groot et al. (2018a), in the case of unknown magnetic states, using the mmt_dipole_cuboid_inversion library.

```python
1  import numpy as np
2  from pathlib import Path
3  import mmt_multipole_inversion as minv
4
5  # ----------------------------------------------------------------------------
6  # INVERSION
7
8  data_dir = Path('deGroot2018_data/PDI-16803')
9  SaveDir = Path('SyntheticSampleFiles')
10 SaveDir.mkdir(exist_ok=True)
11
12 # The area sensor formalism is implemented up to quadrupole order only
13 # The Area1_UMS_NPZ_ARRAYS.npz numpy file contains: grain centers and volumes
14 # The AREA1_UMS_METADICT.json contains: scanning surface properties
15 inv_area1_ums = minv.MultipoleInversion(
16     SaveDir / "AREA1_UMS_METADICT.json",
17     SaveDir / 'Area1_UMS_NPZ_ARRAYS.npz',
18     expansion_limit='quadrupole',
19     sus_functions_module='spherical_harmonics_basis_area')
20 # Load the scanning array manually:
21 inv_area1_ums.Bz_array = np.loadtxt(data_dir / 'Area1-90-fig2MMT.txt')
22 # Compute the inversion with a small relative tolerance
23 inv_area1_ums.compute_inversion(method='sp_pinv', rtol=1e-20)
24
25 # Compute magnetizations - each row has the magnetic moments of a single grain
26 # We get the first 3 columns for every row
27 mag_area1_ums = inv_area1_ums.inv_multipole_moments[:, :3]
28 mag_area1_ums /= inv_area1_ums.volumes[:, None]
29 mag_area1_ums = np.sqrt(np.sum(mag_area1_ums[:, :3] ** 2, axis=1))
```

Code 5: Code for the inversion of Area 1 of the synthetic sample of de Groot et al. (2018a), in the case of unknown magnetic states, using the mmt_multipole_inversion library.

can lead to shadowing of the signal of one of the grains due to an ill-conditioned matrix. Thirdly, the scan signal quality, which directly affects the accuracy of the numerical inversion. In the case of Area 1, grain 5 was found to be one of the most optimal solutions because of its strong signal-to-noise ratio (Cortés-Ortuño et al., 2021).
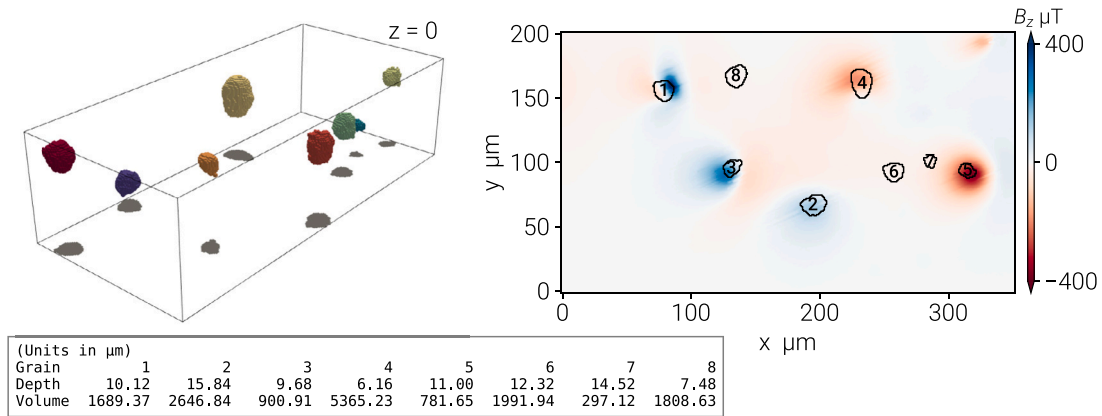
6. Micromagnetic modeling

Computational models of individual grains using the micromagnetic formalism can be used together with the inverted magnetic moments of the grains, to infer their internal magnetic state. This was demonstrated

```
 1  Magnetization in units of A / m
 2  ----------------------------------------------------------------------------------
 3  Grain        MultInv[dipole]       MultInv[quadrup]         DipCubInv        de Groot (2018)*
 4  ----------------------------------------------------------------------------------
 5   1                 3384.149              4612.276             3467.090              3544.300
 6   2                 4384.657              6318.841             3876.450              3923.700
 7   3                15265.621             19826.988            15131.965             15346.800
 8   4                 4190.415              4273.237             3740.296              3770.700
 9   5                28120.792             39414.447            27753.220             28147.800
10   6                 2119.661             15929.647             3059.004              2845.900
11   7                88682.365             75612.761            92531.810             92191.200
12   8                 6796.900             14666.921             7430.479              7154.400
13
14  Pairs of (azimuthal,polar) magnetization angles in degrees.
15  Polar angles positive above the equator
16  ----------------------------------------------------------------------------------
17  Grain        MultInv[dipole]       MultInv[quadrup]         DipCubInv        de Groot (2018)*
18  ----------------------------------------------------------------------------------
19   1           ( 16.43,   21.74)     (352.35,   51.27)     ( 14.49,   23.71)     ( 14.30,   24.11)
20   2           (248.23,   49.06)     (237.77,   40.34)     (251.59,   53.90)     (251.38,   53.80)
21   3           (194.18,   28.63)     (172.70,   25.99)     (194.99,   30.48)     (194.79,   30.46)
22   4           (337.27,  -51.26)     ( 11.23,  -44.09)     (333.52,  -50.79)     (333.69,  -50.68)
23   5           ( 64.34,  -67.41)     (  2.05,  -50.60)     ( 64.00,  -67.16)     ( 63.35,  -67.14)
24   6           (305.30,   22.81)     (259.86,  -24.03)     (313.24,   20.86)     (311.68,   21.32)
25   7           (343.95,  -50.71)     ( 96.51,   62.14)     (350.26,  -53.11)     (348.09,  -50.41)
26   8           (210.85,  -53.10)     (181.80,  -31.04)     (200.50,  -48.71)     (200.76,  -51.84)
27
28  * Positive z-axis in de Groot (2018) point towards depth.
29    Polar values here are inverted. Azimuthal values are re-computed.
```

Code 6: Results of the inversions of Area 1 of the synthetic sample of de Groot et al. (2018a), in the case of unknown magnetic states. Sensors are defined as two-dimensional square areas of 1 μm side length.

(Units in μm)								
Grain	1	2	3	4	5	6	7	8
Depth	10.12	15.84	9.68	6.16	11.00	12.32	14.52	7.48
Volume	1689.37	2646.84	900.91	5365.23	781.65	1991.94	297.12	1808.63

Fig. 6. Overview of the benchmark problem defined by a synthetic sample containing magnetite grains. To the left is shown a three-dimensional view of the location and size of the grains using the tomographic data. Depths and volumes of the grains are detailed in the table at the bottom. To the right of the figure it is shown the magnetometry data with the out-of-plane magnetic signal of the grains at a surface located at $z = 2$ μm. A bird's eye view of the grain profiles are shown here.

by Cortés-Ortuño et al. (2022a) by modeling both synthetic and natural grains, and it was shown that for the grain sizes used in the study, three-dimensional vortex structures, with left- or right-handed chirality, are commonly found as the stable internal configurations that minimize the magnetic energy locally.

Within the set of MMT numerical libraries, we have implemented the `mmt_micromagnetic_demag_signature` code (Cortés-Ortuño et al., 2022b) to generate the magnetic signal from the micromagnetic model of a grain obtained with the micromagnetic code MERRILL (Ó Conbhuí et al., 2018). The inversion of the magnetic signal of multiple stable states from the computational model can be corresponded to those obtained from a natural sample and unveil what is the internal magnetic configuration of natural grains.

6.1. Method

Micromagnetism is the approximation of the magnetization field in the continuum and to model a system numerically it is discretized using, for instance, finite elements (FEs). The FE method consists of partitioning the material into a mesh of tetrahedral units wherein unit
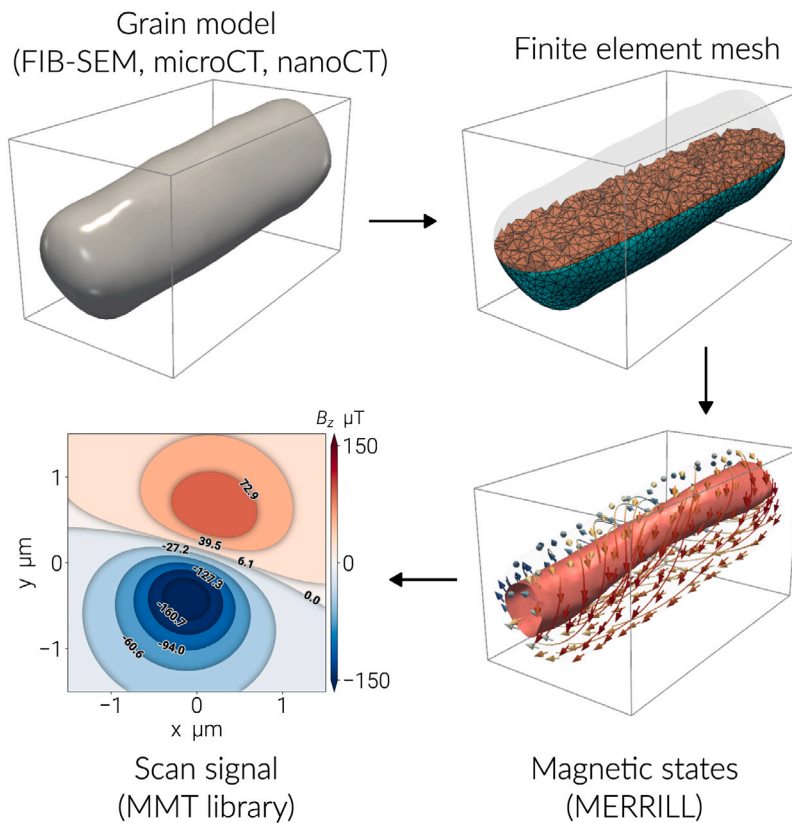
Fig. 7. Overview of the micromagnetic modeling technique applied to a natural magnetite grain with its shape characterized using a tomographic method. The grain profile is converted to a finite element mesh and then simulated using the micromagnetic code MERRILL in order to find states of local or global energy minimum. For every magnetic state, it is possible to calculate its stray field, or magnetic signal, at a surface above the sample. This signal can be numerically inverted to obtain the magnetic moments of the grain.

```
 1 import mmt_micromagnetic_demag_signature as mds
 2 import numpy as np
 3 nm, mum = 1e-9, 1e-6
 4
 5 # Load the MERRILL files from random state 2 of Cortes et al. (2022):
 6 FILE_energy = './grain_OPX042_rnd2.log'  # To read the magnetization
 7 SAVENAME = 'scan_signal_rnd2_scan-height_1000nm.npy'
 8 VBOXFILE = './mag_vol_rnd2.vbox'  # data of FE mesh and M field
 9 # Scan limits are given by lower-left and upper-right sensor positions
10 scan_spacing = (10 * nm, 10 * nm)
11 scan_limits = np.array([[-1.5, -1.5], [1.5, 1.5]]) * mum
12 scan_height = 1000 * nm
13 demag_signal = mds.MicroDemagSignature(scan_limits, scan_spacing, scan_height,
14                                        VBOXFILE, FILE_energy)
15 # Shift the geom center of the grain to the origin
16 demag_signal.read_input_files(origin_to_geom_center=True)
17 # Compute stray field signal
18 demag_signal.compute_scan_signal(method='cython')
19 np.save(SAVENAME, demag_signal.Bz_grid)
```

Code 7: Code for computing the stray field signal of a natural magnetite grain, using a micromagnetic model of the system obtained with the MERRILL code. The grain is labeled OPX042 in Nikolaisen et al. (2020), and its internal state is named random state 2 in Cortés-Ortuño et al. (2022a).

magnetization vectors $\mathbf{m}_i$, with $\|\mathbf{m}_i\| = 1$, are located at the tetrahedra vertices i. The discretization allows the numerical calculation of the total energy from the magnetic interactions of the magnetization, which include the exchange, anisotropy, stray field, external field and magnetoelastic interactions. The total energy can be minimized using a suitable algorithm such as gradient descent methods, in order to find local, and possibly the global, states of minimum energy. For details

of the formalism, refer to Cortés-Ortuño et al. (2022a) and Ó Conbhuí et al. (2018).

The code MERRILL uses the FE micromagnetic formalism to create models of rock magnetic grains, and implements energy minimization algorithms to find stable magnetic configurations. The mmt_micromagnetic_demag_signature library uses both the magnetization at the mesh nodes $\mathbf{m}_i$, and the associated volume at the nodes

V_i, of a finite element mesh to compute the nodal dipole moments, $\Theta^{(1)} = M_s V_i \mathbf{m}_i$. The two properties can be extracted from MERRILL's `vbox` output files, which contain the location, magnetization and volume of every mesh node. Using the dipole moments, the dipole-order magnetic field of the system is computed at a scanning surface whose dimensions and location, above or below the sample, are defined by the user.

The workflow of micromagnetic modeling applied to a single grain is shown in Fig. 7, using the data from the public repository of Nikolaisen (2020). This database contains an extensive record of FE meshes, in `stl` format, of natural magnetite grains from a pyroxene volume that was milled and imaged by Nikolaisen et al. (2020) using focused-ion-beam nanotomography (FIB-nt). In the case of Fig. 7, we chose the grain labeled OPX042, which was analyzed by Cortés-Ortuño et al. (2022a). Alternatively, it is possible to obtain the grain profile from micro- or nano-tomography and apply a numerical surface reconstruction method to generate a FE mesh, as detailed in Cortés-Ortuño et al. (2022a). The grain model is converted from `stl` to `tec` format using the `meshio` Python library (Schlömer, 2021), and then inputted into MERRILL to find states of minimum energy, such as the single vortex configuration of Fig. 7. Finally, the stray field signal of the stable states is computed using the MMT library. For the case of Fig. 7, a scan surface is specified at a height of $z = 1$ μm. The grain is centered at the origin with its top surface reaching $z = 0.11$ μm. The magnetic configuration shown in the Figure corresponds to the state labeled 2 in Cortés-Ortuño et al. (2022a). This configuration was simulated at room temperature and its energy was minimized by initializing the system with randomly oriented magnetization vectors. From the simulation the expected magnetization of this magnetic state is $|\mathbf{M}| = 3.5048 \times 10^5$ A/m. The MPE library was applied to numerically invert the magnetic signal of this state using an octupole order expansion. As a result, the dipole moments obtained are [4.96e-16, 1.60e-15, -4.36e-16] in units of A m^2, with a magnetization of magnitude $|\mathbf{M}_{\text{inv}}| = 3.5029 \times 10^5$ A/m. The simulated dipole and higher order magnetic moments can be compared to the moments obtained from inversion of experimental data.

In the work of Cortés-Ortuño et al. (2022a) the micromagnetic modeling technique was theoretically proved by obtaining unique magnetic moment solutions for different magnetic states in a magnetite sphere model and in grains modeled from tomographic data of natural and synthetic samples. A direct comparison of the modeled magnetic moments with magnetometry scan signals was not feasible because, for example, the natural magnetite grains were destroyed by FIB-nt. Additional challenges to test the micromagnetic modeling with magnetometry scans, include measuring grains with sizes that can be computed with our current computational equipment, analyzing the correct anisotropy axes direction from individual grains, and studying how the magnetometry noise levels affect the inverted moment solutions. With the rapid progress in magnetometry and tomography technology, particularly in increasing measurement resolution, *e.g.* a QDM, it will be possible to fully prove the micromagnetic modeling with experimental data.

The `mmt_micromagnetic_demag_signature` library can be extended in the future to support simulations of finite-difference micromagnetic codes. In the case of FEs, alternative methods for the calculation of the stray field in a measurement surface, include (i) using the magnetic scalar potential of the system, which is computed by the micromagnetic code when solving the demagnetizing field boundary problem, and (ii) the fast multipole method. The search of local minimum states can also be optimized, by adding a boundary condition to the energy minimization, related to the spherical harmonics obtained from the inversion of the experimental data.

7. Conclusions

An essential prerequisite of the MMT technique is the implementation of numerical libraries that allow to obtain the magnetizations,

and higher order magnetic moments, of individual magnetic grains. Here we described the theoretical formulation of two different forward models (HPE and MPE) to retrieve magnetic moments via numerical inversions. Moreover, we presented open source Python libraries based on these models, which process the MMT experimental data. These libraries have been applied to obtain solutions in both synthetic and natural samples.

Refined magnetization solutions are obtained by considering higher order magnetic moments in the inversions with the multipole inversion method (MPE). Because these MPE solutions are more versatile and provide a better description of magnetic moments in the near field, future works will likely focus on the application of this technique to large natural samples, for example, the volcanic sample data studied by Kosters et al. (2023), which is publicly available (Kosters et al., 2023). Such multipole inversions, however, are challenging because of the large number of fitting multipole moment parameters required, making the calculations limited by computer memory. Inverting a larger number of multipole parameters, or increasing sample sizes, may be achieved by applying alternative, memory optimized, numerical inversion methods, such as iterative methods, in the future (Zhdanov, 2015).

It was recently shown that MMT inversion methods combined with micromagnetic modeling enable unveiling the internal magnetic configurations hosted by magnetic grains (Cortés-Ortuño et al., 2022a). To obtain the magnetic moments associated to these stable states from computational modeling, we have developed a code that computes the magnetic signal of the grains in a surface using the stray field of the grain. This emulates the data obtained through magnetometry. A direct comparison of the inversion using the micromagnetic model and the magnetometry data, makes it possible to infer the most likely internal magnetic configuration in the grain.

Our results illustrate that MMT and micromagnetic modeling are promising techniques to improve measurements of paleomagnetic data. These methods are now underpinned by a series of robust Python libraries that facilitate the process of obtaining unique magnetization solutions and also internal magnetic configurations.

CRediT authorship contribution statement

David Cortés-Ortuño: Conceptualization, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing, Supervision, Validation. **Frenk Out:** Conceptualization, Formal analysis, Investigation, Software, Validation. **Martha E. Kosters:** Conceptualization, Formal analysis, Validation. **Karl Fabian:** Conceptualization, Formal analysis, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Lennart V. de Groot:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data is available in repositories and is cited in the manuscript.

Acknowledgments

Code availability section

All the codes are publicly available and hosted in the Github repository at the Micromagnetic Tomography organization website https://github.com/Micromagnetic-Tomography. Every code has a Zenodo DOI that can be cited according to the library version. The codes are written in the Python, Cython, C, and CUDA programming languages. The codes utilize robust Python numerical libraries which include Numpy (Harris et al., 2020), Scipy (Virtanen et al., 2020), Pyvista (Sullivan and Kaszynski, 2019), Meshio (Schlömer, 2021), matplotlib (Hunter, 2007), Jupyter notebooks (Kluyver et al., 2016), Cython (Behnel et al., 2011) and Numba (Lam et al., 2015).

All the codes and results from this study are published in the public GitHub repository (Cortés-Ortuño et al., 2022d).

References

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D.S., Smith, K., 2011. Cython: The best of both worlds. Comput. Sci. Eng. 13 (2), 31–39.

Cortés-Ortuño, D., Fabian, K., de Groot, L.V., 2021. Single particle multipole expansions from micromagnetic tomography. Geochem. Geophys. Geosyst. 22 (4), e2021GC009663. http://dx.doi.org/10.1029/2021GC009663.

Cortés-Ortuño, D., Fabian, K., de Groot, L.V., 2022a. Mapping magnetic signals of individual magnetite grains to their internal magnetic configurations using micromagnetic models. J. Geophys. Res.: Solid Earth 127 (5), e2022JB024234. http://dx.doi.org/10.1029/2022JB024234.

Cortés-Ortuño, D., Fabian, K., de Groot, L.V., 2022b. MMT Numerical Libraries: Micromagnetic Demag Signature. http://dx.doi.org/10.5281/zenodo.6211355.

Cortés-Ortuño, D., Fabian, K., de Groot, L.V., 2022c. MMT Numerical Libraries: Multipole Inversion. http://dx.doi.org/10.5281/zenodo.6473257.

Cortés-Ortuño, D., Out, F., Kosters, M., Fabian, K., de Groot, L.V., 2022d. Dataset for: "Micromagnetic Tomography: Numerical Libraries". http://dx.doi.org/10.5281/zenodo.7838786.

de Groot, L.V., Fabian, K., Béguin, A., Kosters, M.E., Cortés-Ortuño, D., Fu, R.R., Jansen, C.M.L., Harrison, R.J., van Leeuwen, T., Barnhoorn, A., 2021. Micromagnetic tomography for paleomagnetism and rock-magnetism. J. Geophys. Res.: Solid Earth 126 (10), e2021JB022364. http://dx.doi.org/10.1029/2021JB022364.

de Groot, L.V., Fabian, K., Béguin, A., Reith, P., Barnhoorn, A., Hilgenkamp, H., 2018a. Determining individual particle magnetizations in assemblages of micrograins. Geophys. Res. Lett. 45 (7), 2995–3000. http://dx.doi.org/10.1002/2017GL076634.

Dunlop, D.J., Özdemir, Ö., 1997. Rock Magnetism: Fundamentals and Frontiers. vol. 3, Cambridge University Press.

Fabian, K., de Groot, L.V., 2019. A uniqueness theorem for tomography-assisted potential-field inversion. Geophys. J. Int. 216 (2), 760–766. http://dx.doi.org/10.1093/gji/ggy455.

Glenn, D.R., Fu, R.R., Kehayias, P., Le Sage, D., Lima, E.A., Weiss, B.P., Walsworth, R.L., 2017. Micrometer-scale magnetic imaging of geological samples using a quantum diamond microscope. Geochem. Geophys. Geosyst. 18 (8), 3254–3267. http://dx.doi.org/10.1002/2017GC006946.

de Groot, L.V., Fabian, K., Béguin, A., Reith, P., Barnhoorn, A., Hilgenkamp, H., 2018b. List of grains and results of the scanning SQUID magnetometer (SSM) scan. http://dx.doi.org/10.1594/PANGAEA.886724.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585 (7825), 357–362. http://dx.doi.org/10.1038/s41586-020-2649-2.

Hubert, A., Schäfer, R., 1998. Magnetic Domains: The Analysis of Magnetic Microstructures. Springer, Berlin, Heidelberg.

Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. Comput. Sci. Eng. 9 (3), 90–95. http://dx.doi.org/10.1109/MCSE.2007.55.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., development team, J., 2016. Jupyter notebooks - a publishing format for reproducible computational workflows. In: Loizides, F., Schmidt, B. (Eds.), Positioning and Power in Academic Publishing: Players, Agents and Agendas. IOS Press, Netherlands, pp. 87–90. http://dx.doi.org/10.3233/978-1-61499-649-1-87.

Kosters, M.E., de Boer, R.A., Out, F., Cortés-Ortuño, D.I., de Groot, L.V., 2023. Unraveling the magnetic signal of individual grains in a hawaiian lava using micromagnetic tomography. Geochem. Geophys. Geosyst. 24 (4), e2022GC010462. http://dx.doi.org/10.1029/2022GC010462.

Kosters, M.E., de Boer, R.A., Out, F., Cortés-Ortuño, D.I., de Groot, L.V., 2023. Magnetic surface flux data and spatial information of magnetic grains from a hawaiian lava and results of using this data for micromagnetic tomography. http://dx.doi.org/10.1594/PANGAEA.955725.

Lam, S.K., Pitrou, A., Seibert, S., 2015. Numba: A LLVM-based python JIT compiler. In: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. LLVM '15, Association for Computing Machinery, New York, NY, USA, http://dx.doi.org/10.1145/2833157.2833162.

Levine, E.V., Turner, M.J., Kehayias, P., Hart, C.A., Langellier, N., Trubko, R., Glenn, D.R., Fu, R.R., Walsworth, R.L., 2019. Principles and techniques of the quantum diamond microscope. Nanophotonics 8 (11), 1945–1973. http://dx.doi.org/10.1515/nanoph-2019-0209.

Nikolaisen, E.S., 2020. Replication data and appendix for: Hysteresis of natural magnetite ensembles: Micromagnetics of silicate-hosted magnetite inclusions based on focused-ion beam nanotomography. http://dx.doi.org/10.18710/Q0F8PA.

Nikolaisen, E.S., Harrison, R.J., Fabian, K., McEnroe, S.A., 2020. Hysteresis of natural magnetite ensembles: Micromagnetics of silicate-hosted magnetite inclusions based on focused-ion-beam nanotomography. Geochem. Geophys. Geosyst. 21 (11), e2020GC009389. http://dx.doi.org/10.1029/2020GC009389.

Ó Conbhuí, P., Williams, W., Fabian, K., Ridley, P., Nagy, L., Muxworthy, A.R., 2018. MERRILL: Micromagnetic earth related robust interpreted language laboratory. Geochem. Geophys. Geosyst. 19 (4), 1080–1106. http://dx.doi.org/10.1002/2017GC007279.

Out, F., Cortés-Ortuño, D., Fabian, K., van Leeuwen, T., de Groot, L.V., 2022. A first-order statistical exploration of the mathematical limits of micromagnetic tomography. Geochem. Geophys. Geosyst. 23 (4), e2021GC010184. http://dx.doi.org/10.1029/2021GC010184.

Out, F., Cortés-Ortuño, D., Kosters, M., Fabian, K., de Groot, L.V., 2023. MMT numerical libraries: Dipole cuboid inversion. http://dx.doi.org/10.5281/zenodo.7823450.

Pattammattel, A., Tappero, R., Ge, M., Chu, Y.S., Huang, X., Gao, Y., Yan, H., 2020. High-sensitivity nanoscale chemical imaging with hard x-ray nano-XANES. Sci. Adv. 6 (37), http://dx.doi.org/10.1126/sciadv.abb3615.

Sakellariou, A., Sawkins, T.J., Senden, T.J., Limaye, A., 2004. X-ray tomography for mesoscale physics applications. Mech. Appl. 339 (1–2), 152–158. http://dx.doi.org/10.1016/j.physa.2004.03.055.

Schlömer, N., 2021. Meshio: Tools for mesh files. http://dx.doi.org/10.5281/zenodo.5541160.

Sullivan, C.B., Kaszynski, A., 2019. PyVista: 3D plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK). J. Open Sour. Softw. 4 (37), 1450. http://dx.doi.org/10.21105/joss.01450.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental algorithms for scientific computing in python. Nature Methods 17, 261–272. http://dx.doi.org/10.1038/s41592-019-0686-2.

Zhdanov, M.S., 2015. Inverse Theory and Applications in Geophysics, second ed. Elsevier, Amsterdam.