

An efficient system matrix factorization method for scanning diffraction based strain tensor tomography

Axel Henningsson* and Stephen A. Hall

Division of Solid Mechanics, Lund University, Ole Römersväg 1, Lund, Sweden. *Correspondence e-mail: axel.henningsson@solid.lth.se

Received 2 February 2023

Accepted 18 September 2023

Edited by D. A. Keen, STFC Rutherford Appleton Laboratory, United Kingdom

Keywords: X-ray diffraction; strain tensor; tomography; diffraction imaging.

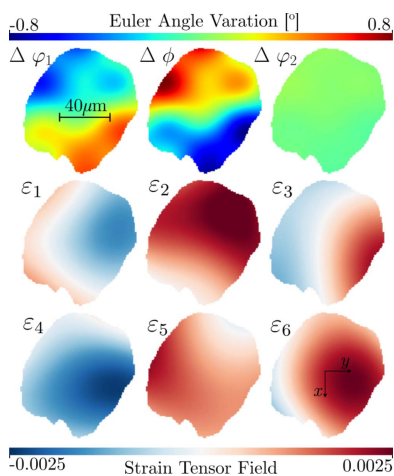
Diffraction-based tomographic strain tensor reconstruction problems in which a strain tensor field is determined from measurements made in different crystallographic directions are considered in the context of sparse matrix algebra. Previous work has shown that the estimation of the crystal elastic strain field can be cast as a linear regression problem featuring a computationally involved assembly of a system matrix forward operator. This operator models the perturbation in diffraction signal as a function of spatial strain tensor state. The structure of this system matrix is analysed and a block-partitioned factorization is derived that reveals the forward operator as a sum of weighted scalar projection operators. Moreover, the factorization method is generalized for another diffraction model in which strain and orientation are coupled and can be reconstructed jointly. The proposed block-partitioned factorization method provides a bridge to classical absorption tomography and allows exploitation of standard tomographic ray-tracing libraries for implementation of the forward operator and its adjoint. Consequently, RAM-efficient, GPU-accelerated, on-the-fly strain/orientation tensor reconstruction is made possible, paving the way for higher spatial resolution studies of intragranular deformation.

1. Introduction

Diffraction-based strain tomography is an experimental technique deployed for estimation of the six-component elastic strain tensor field, $\boldsymbol{\varepsilon}(\mathbf{x})$, within the bulk of polycrystalline aggregates. Whether used with X-rays (Hektor *et al.*, 2019; Korsunsky *et al.*, 2005; Lionheart & Withers, 2015) or neutrons (Hendriks *et al.*, 2020), the method offers a unique possibility to probe the internal heterogeneity of the strain in dense materials in a non-destructive way. In essence, the measured diffraction signal from the specimen can be reduced to average strains along line integral domains across a sample volume. Each of these scalar strain measures, γ_j , can be associated to a spatial sampling direction, $\boldsymbol{\kappa}_j$, which in general varies between measurements. Considering a set of $j = 1, \dots, m$ such measurements, it is possible to construct a global linear system of equations,

$$\mathbf{A}\mathbf{s} = \boldsymbol{\gamma}, \quad (1)$$

where \mathbf{s} holds the basis coefficients of a decomposed strain tensor field and $\boldsymbol{\gamma}$ is a vector with all measurements [the formation of $\boldsymbol{\gamma}$ from raw diffraction images and the decoupling of the crystal strain from orientation are discussed by Henningsson & Hendriks (2021)]. The rows of the system matrix, \mathbf{A} , are required to contain the integral weights of the strain tensor basis functions combined with non-linear combinations of the components of $\boldsymbol{\kappa}_j$. Using measurements



OPEN ACCESS

Published under a CC BY 4.0 licence

only from a single axis of rotation, there exist no known, closed-form, direct back-projection algorithms to recover the strain tensor field \mathbf{s} . This motivates the need for iterative solvers, which may seem to require the assembly of \mathbf{A} . Unfortunately, the storage of \mathbf{A} can be very RAM inefficient and the assembly routines needed to construct \mathbf{A} involve ray-tracing through the strain tensor volume. Indeed, direct storage of the forward operator is unfeasible for high-resolution scalar, absorption-based, tomography. Considering a fixed resolution, strain tomography of symmetric strain tensor fields offers no relaxation in this respect as the number of non-zeros in the system matrix, \mathbf{A} , is a factor six greater compared with scalar tomography. As a result, several existing reconstruction methods have been cast in settings with few strain tensor basis functions limiting the achievable reconstruction resolution (Henningsson *et al.*, 2020; Henningsson & Hendriks, 2021; Hendriks *et al.*, 2020). Similarly, in scanning 3D X-ray diffraction (scanning-3DXRD) microscopy applications, it is common to collapse the rich 2D pixel intensity distribution of the recorded diffraction peaks to a single centre of gravity prior to the pursuit of strain reconstruction (Hayashi *et al.*, 2015).

We present a system matrix factorization for strain tensor tomography in which the forward operator, \mathbf{A} , can be implemented as a weighted sum of scalar forward projections as

$$\mathbf{A}\mathbf{s} = \sum_{i=1}^{i=6} \mathbf{S}_i \mathcal{P}\mathbf{a}_i, \quad (2)$$

where \mathcal{P} is a scalar forward projection operator, $\mathbf{a}_1, \dots, \mathbf{a}_6$ are the six individual components of the strain tensor field and $\mathbf{S}_1, \dots, \mathbf{S}_6$ are diagonal weight matrices. This factorization allows for RAM-efficient, on-the-fly implementations to be easily achieved with existing tomographic libraries (for scalar projection). Additionally, our proposed factorization allows for access to GPU-accelerated implementations commonly deployed in scalar tomography to facilitate large sparse iterative solvers (Palenstijn *et al.*, 2011; van Aarle *et al.*, 2015, 2016).

For illustrative purposes we have selected to present our derivations in the context of strain reconstruction and for the experimental setup of scanning-3DXRD. The methodology is, however, also applicable for other neutron and X-ray scanning diffraction experiments given that a fixed axis of rotation is used and that the diffraction peak centre-of-mass positions can be accurately measured (typically in far-field geometry). The key ingredient in our derivation is the linearity of the diffraction model, which allows us to rearrange the order of the involved operators. In contrast to the far-field diffraction setting considered in this paper, near-field diffraction methods (Reischig & Ludwig, 2020) model the full detector intensity distribution of the diffraction peaks rather than the peak centroid positions. As a result, the forward operator in near-field diffraction models depends non-linearly on the intragranular strain and orientation. To highlight that our factorization method is applicable to multiple models, as long as they fall within the class of far-field diffraction, we derive and

demonstrate (in Appendix C) a factorization similar to that of equation (2) for a previously suggested diffraction model that features coupling between the intragranular strain and orientation. This factorization enables efficient reconstruction of the full intragranular deformation field, including both strain and orientations.

2. Per-ray factorization

Given an unknown, symmetric, second-order strain tensor field,

$$\boldsymbol{\varepsilon}(\mathbf{x}) = \begin{bmatrix} \varepsilon_1(\mathbf{x}) & \varepsilon_4(\mathbf{x}) & \varepsilon_5(\mathbf{x}) \\ \varepsilon_4(\mathbf{x}) & \varepsilon_2(\mathbf{x}) & \varepsilon_6(\mathbf{x}) \\ \varepsilon_5(\mathbf{x}) & \varepsilon_6(\mathbf{x}) & \varepsilon_3(\mathbf{x}) \end{bmatrix}, \quad (3)$$

defined on a 3D spatial domain, $\mathbf{x} = [x y z]^T$, we shall consider measurements of the average strain, γ_j , on the line integral domain \mathcal{R}_j as

$$\gamma_j = \frac{1}{L_j} \int_{\mathcal{R}_j} \boldsymbol{\kappa}_j^T \boldsymbol{\varepsilon}(\mathbf{x}) \boldsymbol{\kappa}_j \, d\mathbf{x}, \quad (4)$$

where $\boldsymbol{\kappa}_j = [\kappa_1 \kappa_2 \kappa_3]^T$ is a unit normal vector that describes the sampled strain direction and L_j is the ray intersection path length measured over the compact support of $\boldsymbol{\varepsilon}$. For scanning-3DXRD the formation of γ_j from the raw diffraction image data has been described elsewhere (Henningsson & Hendriks, 2021). Following a flattened vector format similar to that of Henningsson & Hendriks (2021) we find the alternative measurement model

$$\gamma_j = \int_{\mathcal{R}_j} \bar{\boldsymbol{\kappa}}_j^T \bar{\boldsymbol{\varepsilon}}(\mathbf{x}) \, d\mathbf{x}, \quad (5)$$

where

$$\bar{\boldsymbol{\varepsilon}}(\mathbf{x}) = \begin{bmatrix} \varepsilon_1(\mathbf{x}) \\ \varepsilon_3(\mathbf{x}) \\ \varepsilon_2(\mathbf{x}) \\ \varepsilon_4(\mathbf{x}) \\ \varepsilon_5(\mathbf{x}) \\ \varepsilon_6(\mathbf{x}) \end{bmatrix}, \quad \bar{\boldsymbol{\kappa}} = \frac{1}{L_j} \begin{bmatrix} \kappa_{1j}^2 \\ \kappa_{2j}^2 \\ \kappa_{3j}^2 \\ 2\kappa_{1j}\kappa_{2j} \\ 2\kappa_{1j}\kappa_{3j} \\ 2\kappa_{2j}\kappa_{3j} \end{bmatrix}. \quad (6)$$

Let $\bar{\boldsymbol{\varepsilon}}$ be decomposed on \mathbf{x} with n basis functions $\varphi_l(\mathbf{x})$ as

$$\bar{\boldsymbol{\varepsilon}}(\mathbf{x}) = \sum_{l=1}^n \varphi_l(\mathbf{x}) \mathbf{a}_l, \quad (7)$$

where the basis coefficients \mathbf{a}_l are defined as

$$\mathbf{a}_l = [\alpha_{1l} \quad \alpha_{2l} \quad \alpha_{3l} \quad \alpha_{4l} \quad \alpha_{5l} \quad \alpha_{6l}]^T. \quad (8)$$

In the following we select φ_l to represent an equidistant grid of pixels such that $\varphi_l(\mathbf{x}) = 1$ when \mathbf{x} is in pixel number l and $\varphi_l = 0$ otherwise. By insertion of (7) into (5) we have

$$\gamma_j = \int_{\mathcal{R}_j} \bar{\boldsymbol{\kappa}}_j^T \sum_{l=1}^n \varphi_l(\mathbf{x}) \mathbf{a}_l \, d\mathbf{x}. \quad (9)$$

Reordering the integral and sum we can write

$$\gamma_j = \bar{\kappa}_j^T \sum_{l=1}^n \alpha_l \int_{\mathcal{R}_j} \varphi_l(\mathbf{x}) \, d\mathbf{x}. \quad (10)$$

We now introduce the vector \mathbf{w}_j which contains the scalar weights of the ray integral with respect to basis functions,

$$\mathbf{w}_j = \begin{bmatrix} \int_{\mathcal{R}_j} \varphi_1(\mathbf{x}) \, d\mathbf{x} \\ \int_{\mathcal{R}_j} \varphi_2(\mathbf{x}) \, d\mathbf{x} \\ \vdots \\ \int_{\mathcal{R}_j} \varphi_n(\mathbf{x}) \, d\mathbf{x} \end{bmatrix}. \quad (11)$$

Using the weights, \mathbf{w}_j , we may form a matrix projection operator that projects the six components of the strain field along a single ray path as

$$\mathbf{R}_j = \underbrace{\begin{bmatrix} \mathbf{w}_j^T & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{w}_j^T & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{w}_j^T \end{bmatrix}}_{6 \times 6n}. \quad (12)$$

Additionally we introduce the vectors

$$\mathbf{a}_1 = \begin{bmatrix} \alpha_{11} \\ \alpha_{12} \\ \vdots \\ \alpha_{1n} \end{bmatrix}, \mathbf{a}_2 = \begin{bmatrix} \alpha_{21} \\ \alpha_{22} \\ \vdots \\ \alpha_{2n} \end{bmatrix} \dots \mathbf{a}_6 = \begin{bmatrix} \alpha_{61} \\ \alpha_{62} \\ \vdots \\ \alpha_{6n} \end{bmatrix}, \quad (13)$$

and stack the basis coefficients of the unknown strain tensor field in a single column vector as

$$\mathbf{s} = \underbrace{\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \\ \mathbf{a}_4 \\ \mathbf{a}_5 \\ \mathbf{a}_6 \end{bmatrix}}_{6n \times 1}. \quad (14)$$

We can now facilitate a fully vectorized and discretized format of the measurement model, equation (4), as

$$\gamma_j = \bar{\kappa}_j^T \mathbf{R}_j \mathbf{s}. \quad (15)$$

To arrive at a global format, in which several measurements, γ_j , are considered simultaneously, we introduce the vector

$$\boldsymbol{\gamma} = [\gamma_1 \quad \gamma_2 \quad \dots \quad \gamma_m]^T. \quad (16)$$

Stacking the matrices $\bar{\kappa}_j^T$ and \mathbf{R}_j in the same fashion,

$$\mathbf{K} = \underbrace{\begin{bmatrix} \bar{\kappa}_1^T & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \bar{\kappa}_2^T & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \bar{\kappa}_m^T \end{bmatrix}}_{m \times 6m}, \quad \mathbf{V} = \underbrace{\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_m \end{bmatrix}}_{6m \times 6n}, \quad (17)$$

we find the global matrix formulation as

$$\boldsymbol{\gamma} = \mathbf{K} \mathbf{V} \mathbf{s}. \quad (18)$$

We note that in equation (18) the matrix $\mathbf{A} = \mathbf{K} \mathbf{V}$ is factorized in two terms: \mathbf{K} , which contains information on the directional sampling of the strain field, and \mathbf{V} , which holds information on the projections of the sampled fields.

3. Hexa-block-diagonal form

In scalar tomography the forward projection operator, \mathcal{P} , is commonly block-partitioned over a series of projection views, \mathcal{P}_i , as

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_1 \\ \mathcal{P}_2 \\ \vdots \\ \mathcal{P}_k \end{bmatrix}, \quad (19)$$

where each projection view, \mathcal{P}_i , represents an ordered set of parallel line integrals defined over a single scalar field. In contrast, we note that the ray integrals contained in \mathbf{V} , as defined by equations (12) and (17), are neither ordered in complete views nor defined over a single scalar field. We therefore seek to reorder and partition the rays in \mathbf{V} in a way that will allow our projection operator to be easily implemented using standard tomographic libraries. To this end, we note that the set of rows in \mathbf{V} separated by a fixed multiple 6, with start at row number 1, forms the block-partitioned matrix

$$\underbrace{\begin{bmatrix} \mathbf{L} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{m \times 6n}, \quad (20)$$

where \mathbf{L} is now acting on the single scalar field \mathbf{a}_1 . If the measurements in $\boldsymbol{\gamma}$ are selected to be stacked in complete projection views, we find that $\mathbf{L} = \mathcal{P}$. Since the initial selected ordering of measurements in $\boldsymbol{\gamma}$ is arbitrary, we shall assume that this ordering has been selected. Now, by simply repeating the row shifting operation with increasing row starting index, 1, 2, ..., 6, it is possible to mutate \mathbf{V} into the block-diagonal matrix form,

$$\mathbf{P} = \underbrace{\begin{bmatrix} \mathcal{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathcal{P} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathcal{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathcal{P} \end{bmatrix}}_{6m \times 6n}, \quad (21)$$

which contains the reordered rows of \mathbf{V} . Naturally, to maintain the global formulation in equation (17), we are required to now also modify \mathbf{K} . The shifting of the rows of \mathbf{V} , therefore, requires a corresponding shifting of columns in \mathbf{K} , leading to the block-partitioned matrix

$$\mathbf{S} = \underbrace{\begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 & \mathbf{S}_3 & \mathbf{S}_4 & \mathbf{S}_5 & \mathbf{S}_6 \end{bmatrix}}_{m \times 6n}, \quad (22)$$

with diagonal blocks

$$\mathbf{S}_1 = \text{Diag}\left(\frac{\kappa_{1j}^2}{L_j}\right), \quad \mathbf{S}_2 = \text{Diag}\left(\frac{\kappa_{2j}^2}{L_j}\right), \quad (23)$$

$$\mathbf{S}_3 = \text{Diag}\left(\frac{\kappa_{3j}^2}{L_j}\right), \quad \mathbf{S}_4 = \text{Diag}\left(\frac{2\kappa_{1j}\kappa_{2j}}{L_j}\right), \quad (24)$$

$$\mathbf{S}_5 = \text{Diag}\left(\frac{2\kappa_{1j}\kappa_{3j}}{L_j}\right), \quad \mathbf{S}_6 = \text{Diag}\left(\frac{2\kappa_{2j}\kappa_{3j}}{L_j}\right). \quad (25)$$

It is now possible to write

$$\gamma = \mathbf{SPs} = \sum_{i=1}^{i=6} \mathbf{S}_i \mathcal{P} \mathbf{a}_i. \quad (26)$$

In this factorization the execution of the forward operator, $\mathbf{A} = \mathbf{SP}$, corresponds to six scalar forward projections followed by the application of \mathbf{S} , which, due to its diagonal form, presents a modest $6m$ multiplications and additions. The implementation of \mathcal{P} can be directly achieved by any ray-tracing library, e.g. the *ASTRA-toolbox* (Palenstijn *et al.*, 2011; van Aarle *et al.*, 2015, 2016). The implementation of \mathbf{S} is trivial and, owing to its diagonal form, there is no need to assemble the matrix, as it suffices to store the six vectors of diagonal weights. Since the six projections being executed in \mathbf{P} are independent, we note that the resulting arrays, $\mathbf{a}_1, \dots, \mathbf{a}_6$, may be stacked and projected in parallel on a GPU. To service the diffraction imaging community, and to illustrate how equation (26) can be put to use to achieve an easily implementable GPU-accelerated diffraction model, we provide an open-source demo Python code at <https://github.com/AxelHenningsson/flyxdm>.

4. Generalizations

For the sake of clarity, we derived equation (26) in the setting of strain reconstruction. This setting features scalar measurements, γ_j , which simplifies the exposition and allows us to focus on the core rearrangement of equations necessary to arrive at our block-partitioned factorized format. The same algebraic manipulations can be used to factorize a wider class of linear far-field diffraction models. We demonstrate the generality of our matrix factorization method in Appendix C where we have pursued an extended diffraction model originally suggested by Henningsson *et al.* (2020). In this alternative setting the intragranular orientation field is jointly reconstructed with the strain tensor field and the measurement associated to the ray integral is vector valued rather than scalar.

To reconstruct a target field, \mathbf{s} , in practice, it is often desirable to introduce a measurement weight matrix, \mathbf{W} , that describes the measurement precision. For instance, in the work of Henningsson *et al.* (2020) a diagonal weight matrix was used to reconstruct strain in a weighted least-squares sense. We note that our factorization is indifferent to the introduction of \mathbf{W} and the global equation system would in practical application be extended as

$$\mathbf{W}\gamma = \mathbf{WSPs}, \quad (27)$$

where $\mathbf{W}^2 = \Sigma_\gamma^{-1}$ is the covariance of the measurements, γ .

Another practical concern is the incorporation of constraints on the solution vector, \mathbf{s} . One popular approach is

to modify the basis of the unknown target field to encode the prior knowledge. To exploit our factorized format in these settings, we suggest introducing a rendering matrix, \mathbf{N} , that maps the basis coefficients, \mathbf{q} , from the constrained basis set back to the pixel basis coefficients, \mathbf{s} , as

$$\mathbf{s} = \mathbf{Nq}. \quad (28)$$

The resulting global equations now become

$$\mathbf{W}\gamma = \mathbf{WSPNq}, \quad (29)$$

where the columns of \mathbf{N} can be interpreted as pixel images of the selected basis set. As the forward operator, \mathbf{WSPN} , and the adjoint operator, $\mathbf{N}^T \mathbf{P}^T \mathbf{S}^T \mathbf{W}^T$, still feature the desired multiplicative block-partitioned split between \mathbf{P} and \mathbf{S} , we conclude that the results in equation (26) can be exploited in a wide range of applications.

As a final note, on the topic of generalizations, we would like to mention that, just as the tensor components of the target field can be stacked into a 3D volume and projected in parallel on a GPU card, one may instead consider stacking grain slices into a volume and projecting each tensor component separately. This modification, reconstructing a full grain volume rather than a grain slice, has no impact on the algebraic format of equation (26). The rendering matrix, \mathbf{N} , is then computing the coefficients of a set of voxels that are projected as a 3D volume by \mathbf{P} .

5. Demonstration

To demonstrate the memory benefits that can be achieved using equation (26) compared with assembling and storing the sparse matrix, \mathbf{A} , we consider a single-crystal diffraction simulation case study. The simulation is described in detail in Appendix A, together with illustrations of the reconstructions achieved when exploiting the format of equation (26) during regression (Appendices B and C). The supplementary code used to generate the simulation data as well as the recon-

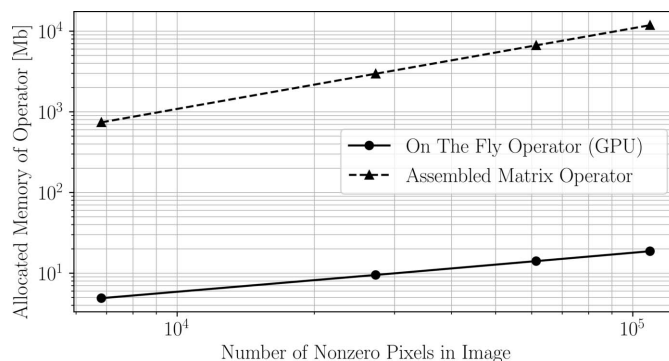


Figure 1 Number of megabytes of computer RAM necessary to compute $\gamma = \mathbf{As}$ using either an assembled sparse matrix (dashed line) or, alternatively, the discussed factorization in conjunction with an on-the-fly projection operator, \mathbf{P} (solid line). Note that the benchmarks correspond to a single-crystal grain slice and 500 diffraction peaks (projection views) as described in Appendix A.

structions is openly available at <https://github.com/AxelHenningson/flyxdm>.

In Fig. 1 we present the number of megabytes of computer RAM necessary to compute $\gamma = \mathbf{A}\mathbf{s}$ using either a fully assembled sparse matrix \mathbf{A} or, alternatively, the factorization $\mathbf{A} = \mathbf{S}\mathbf{P}$, where \mathbf{P} is represented using pre-existing, on-the-fly, projection operators, available in the *ASTRA-toolbox* (van Aarle *et al.*, 2015). Considering that the results presented in Fig. 1 represent the reconstruction of a single grain slice using 500 projection views (each corresponding to a diffraction event), it is evident that parallel, high-resolution, full volume/sample reconstructions are unfeasible using an assembled format of \mathbf{A} . For instance, reconstructing, in parallel, a single, cubic-shaped grain volume, with a cross-sectional resolution of 256×256 pixels from ~ 300 unique (with respect to Miller index) diffraction peaks would require ~ 1 TB of computer RAM storage.

6. Conclusion

We have presented a system matrix factorization for strain tensor tomography in which the directional sampling of the strain tensor field is separated from the tomographic projection operator. The proposed format allows for the exploitation of standard tomographic ray-tracing libraries in the implementation of the forward operator. We have also shown how our factorization method can be generalized for other diffraction models, for example one in which strain and orientation are coupled. We have provided an openly available GPU implementation of the approach and demonstrated the computational efficiency of our factorization method through application to a model example. By enabling RAM-efficient, GPU-accelerated, on-the-fly strain/orientation tensor reconstruction, our results facilitate higher spatial resolution studies of intragranular deformation.

APPENDIX A Demonstration example

To demonstrate the discussed matrix factorization in a practical application we have included a single-crystal X-ray diffraction simulation case study. Diffraction data were forward modelled from a 2D grain slice of α -quartz (SiO_2) subject to a spatially varying strain tensor field, $\boldsymbol{\varepsilon}(\mathbf{x})$, as well as a misorientation field, $\mathbf{U}(\mathbf{x})$, where $\mathbf{U}(\mathbf{x})$ is the local crystal orientation matrix. The synthetic strain tensor field can be viewed in Fig. 2 together with the Bunge Euler angle variation, which was defined as

$$\begin{aligned} \Delta\varphi_1 &= \varphi_1 - \langle \varphi_1 \rangle, \\ \Delta\phi &= \phi - \langle \phi \rangle, \\ \Delta\varphi_2 &= \varphi_2 - \langle \varphi_2 \rangle, \end{aligned} \tag{30}$$

where $\langle \cdot \rangle$ denotes volume average.

Using the space group of quartz ($P3221$) together with an X-ray wavelength of $\lambda = 0.2846 \text{ \AA}$, a total of 500 reflections

were randomly (uniformly) selected in the Bragg angle interval $\theta = [4^\circ, 13^\circ]$ to be included in the simulation. Using the Laue equations,

$$\mathbf{G} = \mathbf{U}\mathbf{B}\mathbf{G}_{hkl}, \tag{31}$$

where the matrix \mathbf{B} maps from the integer reciprocal-space Miller indices, $\mathbf{G}_{hkl} = [h \ k \ l]^T \in \mathbb{Z}^{3 \times 1}$, to crystal coordinates, diffraction vectors $\mathbf{G} = [G_1 \ G_2 \ G_3]^T$ were formed. The theoretical diffraction vectors were then corrupted with zero mean Gaussian noise as

$$\begin{aligned} G_{1j} &\leftarrow G_{1j} + n_{1j}, & n_{1j} &\sim \mathcal{N}(0, \sigma_{1j}^2) \\ G_{2j} &\leftarrow G_{2j} + n_{2j}, & n_{2j} &\sim \mathcal{N}(0, \sigma_{2j}^2) \\ G_{3j} &\leftarrow G_{3j} + n_{3j}, & n_{3j} &\sim \mathcal{N}(0, \sigma_{3j}^2), \end{aligned} \tag{32}$$

where $\sigma_{1j} = \sigma_{2j} = \sigma_{3j} = 10^{-3}$ for 90% of the measurements while $\sigma_{1j} = \sigma_{2j} = \sigma_{3j} = 10^{-2}$ for the remaining 10%, emulating the presence of outliers. The 10% selected for outlier noising was selected randomly uniformly from the full diffraction data set.

Using the *ASTRA-toolbox* we have implemented the matrix factorization derived in this paper in operator format. This means that we never need to form the explicit sparse matrices involved in the forward model, but, instead, implement an on-the-fly operator that operates on an input vector to produce the system matrix vector dot product (and the corresponding adjoint product). This operator implementation is only possible thanks to the algebraic results that constitute the contribution of this paper. The code used to implement our matrix factorization (on an NVIDIA GPU architecture), produce the simulated data and reconstruct the strain (and later orientation) field is openly available as a demo Python library <https://github.com/AxelHenningson/flyxdm>. In the same supplementary demo code, we provide

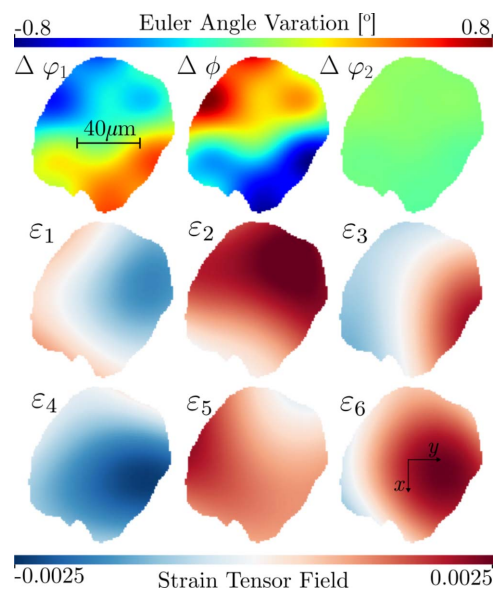


Figure 2 Strain (bottom) and mosaicity (top) in a simulated 2D grain slice of α -quartz (SiO_2). The Bunge Euler angles are displayed as variations around their respective mean values.

detailed comments on the simulation and reconstruction setup and provide the code used to generate all figures found in these Appendices (including the generalizations to orientation fields made in Appendix C).

APPENDIX B Strain reconstruction

In this Appendix, we use the simulation data in conjunction with the Taylor expansion described by Henningsson & Hendriks (2021) and convert the diffraction vectors, \mathbf{G} , into measurements of directional strain with the aim of reconstructing the intragranular strain field. In Appendix C we describe how the diffraction vectors can be used without transform to reconstruct strain and intragranular orientation variations jointly, again exploiting our matrix factorization method. Note that while the aim in Appendix B is to demonstrate our matrix factorization method for the case of strain reconstruction, the simulated data still originate from a grain that features both intragranular strain and orientation variation. As discussed elsewhere (Henningsson *et al.*, 2020; Henningsson & Hendriks, 2021), this is not a problem as long as the mosaicity of the grain is moderate.

In Fig. 3, the result of strain tensor reconstruction can be viewed. Here a radial basis expansion was used to construct \mathbf{N} [in equation (29)]. The true noise covariance of the diffraction vectors was propagated through the Taylor expansion to construct the directional strain covariance yielding \mathbf{W} . The small residuals and root-mean-squared errors (RMSEs) found in the bottom row of Fig. 3 are expected as a consequence of noise and model mismatch. For further details we refer the reader to the supplementary code.

APPENDIX C Generalization to coupled orientation–strain models

We shall now consider generalizing our matrix factorization method to a diffraction model discussed by Henningsson *et al.* (2020, Section 6 equation 10-16). Considering the Laue equations (31) in integral form, we find

$$\langle \mathbf{G}_j \rangle = \frac{1}{L_j} \int_{\mathcal{R}_j} \mathbf{U} \mathbf{B} \mathbf{G}_{hkl}^{(j)} d\mathbf{x}, \quad (33)$$

where $\langle \cdot \rangle$ denotes volume average. The target intragranular field of deformation is now generalized to include the local rigid-body rotations as

$$\mathbf{U} \mathbf{B} = \mathbf{U}(\mathbf{x}) \mathbf{B}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \in \mathbb{R}^{3 \times 3}. \quad (34)$$

In contrast to the small strain tensor, \mathbf{f} is not symmetric and the number of unknowns per point, \mathbf{x} , in the grain is now increased to 9 (compared with six strain tensor components). To reach a similar factorization as that of (26), we start by introducing a flattened format of (33) as

$$\langle \mathbf{G}_j \rangle = \mathbf{H}_j \int_{\mathcal{R}_j} \bar{\mathbf{f}}(\mathbf{x}) d\mathbf{x}, \quad (35)$$

where

$$\bar{\mathbf{f}} = \begin{bmatrix} f_{11}(\mathbf{x}) \\ f_{21}(\mathbf{x}) \\ f_{31}(\mathbf{x}) \\ f_{12}(\mathbf{x}) \\ f_{22}(\mathbf{x}) \\ f_{32}(\mathbf{x}) \\ f_{13}(\mathbf{x}) \\ f_{23}(\mathbf{x}) \\ f_{33}(\mathbf{x}) \end{bmatrix} \quad (36)$$

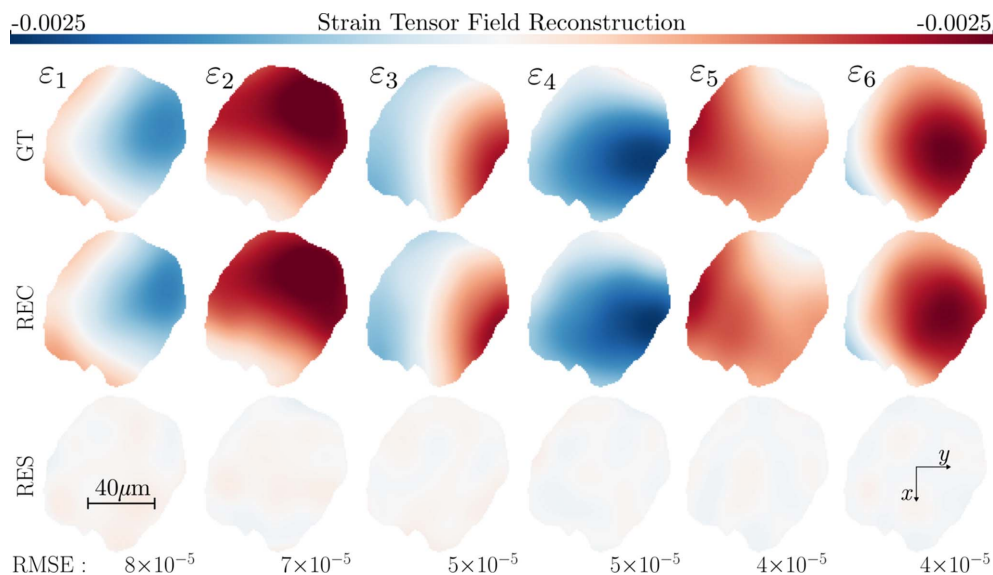


Figure 3 Strain tensor reconstruction (REC) of a single-crystal α -quartz grain slice. The matrix factorization in equation (29) has been used to reconstruct the strain field without the need to assemble the global system matrix. The top row ground-truth simulation input (GT) is to be compared with the middle row reconstructed strain field (REC). The bottom row shows the residual between reconstructed and true strain fields (RES).

and

$$\mathbf{H}_j = \frac{1}{L_j} [\mathbf{I}h_j \quad \mathbf{I}k_j \quad \mathbf{I}l_j], \quad (37)$$

and \mathbf{I} is the 3×3 identity matrix. Let us now decompose the target field, $\bar{\mathbf{f}}(\mathbf{x})$, on a finite basis as

$$\bar{\mathbf{f}}(\mathbf{x}) = \sum_{l=1}^n \beta_l \varphi_l(\mathbf{x}), \quad (38)$$

where $\varphi_l \in \mathbb{R}$ are the scalar (pixel/voxel) basis functions and

$$\beta_l = \begin{bmatrix} \beta_{1l} \\ \beta_{2l} \\ \beta_{3l} \\ \beta_{4l} \\ \beta_{5l} \\ \beta_{6l} \\ \beta_{7l} \\ \beta_{8l} \\ \beta_{9l} \end{bmatrix}. \quad (39)$$

Insertion of equation (38) into equation (35) yields after rearrangement that

$$\langle \mathbf{G}_j \rangle = \mathbf{H}_j \sum_{l=1}^n \beta_l \int_{\mathcal{R}_j} \varphi_l(\mathbf{x}) \, d\mathbf{x}, \quad (40)$$

where the linearity of the involved operators was used. We now note that equation (40) is a higher-dimensional copy of equation (10). Defining \mathbf{w}_j according to equation (11), we find in analogy with equation (12) that

$$\mathbf{Q}_j = \underbrace{\begin{bmatrix} \mathbf{w}_j^T & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{w}_j^T & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{w}_j^T \end{bmatrix}}_{9 \times 9n}. \quad (41)$$

We now introduce a set of coefficient vectors,

$$\mathbf{b}_1 = \begin{bmatrix} \beta_{11} \\ \beta_{12} \\ \vdots \\ \beta_{1n} \end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix} \beta_{21} \\ \beta_{22} \\ \vdots \\ \beta_{2n} \end{bmatrix}, \dots, \mathbf{b}_9 = \begin{bmatrix} \beta_{91} \\ \beta_{92} \\ \vdots \\ \beta_{9n} \end{bmatrix}, \quad (42)$$

and define a partitioned global coefficient vector as

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_9 \end{bmatrix}. \quad (43)$$

The ray integral equation (40) can now be cast as

$$\langle \mathbf{G}_j \rangle = \mathbf{H}_j \mathbf{Q}_j \boldsymbol{\tau}. \quad (44)$$

We now note that each row in equation (44) has an identical algebraic format compared with equation (15). Splitting the diffraction vector measurements into three separate vectors as

$$\mathbf{d}_1 = \begin{bmatrix} \langle \mathbf{G}_{11} \rangle \\ \langle \mathbf{G}_{12} \rangle \\ \vdots \\ \langle \mathbf{G}_{1m} \rangle \end{bmatrix}, \mathbf{d}_2 = \begin{bmatrix} \langle \mathbf{G}_{21} \rangle \\ \langle \mathbf{G}_{22} \rangle \\ \vdots \\ \langle \mathbf{G}_{2m} \rangle \end{bmatrix}, \mathbf{d}_3 = \begin{bmatrix} \langle \mathbf{G}_{31} \rangle \\ \langle \mathbf{G}_{32} \rangle \\ \vdots \\ \langle \mathbf{G}_{3m} \rangle \end{bmatrix}, \quad (45)$$

we are free to use the same arguments of row and column permutation as described in Section 3 to arrive at

$$\mathbf{d}_1 = \mathbf{M}_1 \mathbf{P}' \boldsymbol{\tau}, \quad (46)$$

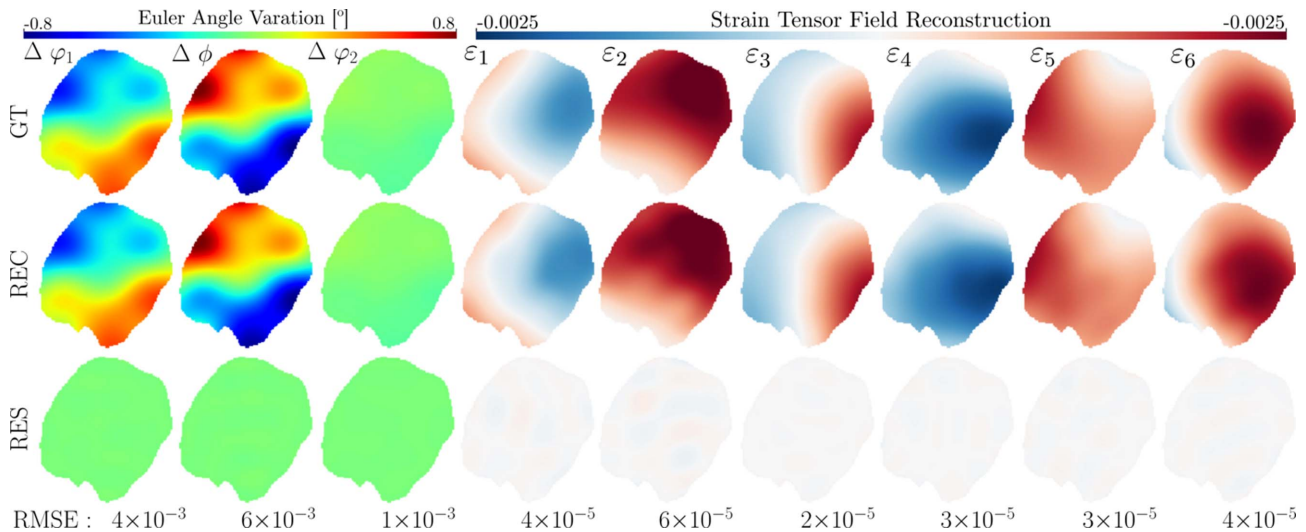


Figure 4 Coupled strain–orientation reconstruction (middle row) in a single slice of α -quartz. The simulated ground-truth (GT) field and corresponding data are described in Appendix A. The residual field (RES) can be viewed in the bottom row. Note that the Bunge Euler angles (left) are displayed as a deviation from their respective mean values, allowing for a shared colorbar.

where \mathbf{P}' now is a $9m \times 9n$ block-diagonal projection matrix in direct analogy to equation (21) and the block-diagonal matrix \mathbf{M}_1 holds the Miller indices weighted by path length as

$$\mathbf{M}_1 = \underbrace{\begin{bmatrix} \mathbf{h} & \mathbf{0} & \mathbf{0} & \mathbf{k} & \mathbf{0} & \mathbf{0} & \mathbf{l} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{m \times 9m}, \quad (47)$$

with

$$\begin{aligned} \mathbf{h} &= \text{Diag} \left(\frac{h_j}{L_j} \right), \\ \mathbf{k} &= \text{Diag} \left(\frac{k_j}{L_j} \right), \\ \mathbf{l} &= \text{Diag} \left(\frac{l_j}{L_j} \right). \end{aligned} \quad (48)$$

The global system of equations can now be written as

$$\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \mathbf{M}_3 \end{bmatrix} \mathbf{P}' \boldsymbol{\tau}, \quad (49)$$

with

$$\begin{aligned} \mathbf{M}_2 &= \begin{bmatrix} \mathbf{0} & \mathbf{h} & \mathbf{0} & \mathbf{0} & \mathbf{k} & \mathbf{0} & \mathbf{0} & \mathbf{l} & \mathbf{0} \end{bmatrix}, \\ \mathbf{M}_3 &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{h} & \mathbf{0} & \mathbf{0} & \mathbf{k} & \mathbf{0} & \mathbf{0} & \mathbf{l} \end{bmatrix}. \end{aligned} \quad (50)$$

Alternatively, denoting the measurement vector as

$$\mathbf{d} = \underbrace{\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \end{bmatrix}}_{3m \times 1} \quad (51)$$

and the Miller sampling matrix as

$$\mathbf{M} = \underbrace{\begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \mathbf{M}_3 \end{bmatrix}}_{3m \times 9m}, \quad (52)$$

we arrive at our final factorized diffraction model,

$$\mathbf{d} = \mathbf{M} \mathbf{P}' \boldsymbol{\tau}. \quad (53)$$

The forward pass in equation (53) is defined by nine separate (scalar) projection operations followed by nine multiplications with the diagonal blocks (\mathbf{h} , \mathbf{k} , \mathbf{l}) of \mathbf{M} . This factorization therefore admits the same computational benefits that are discussed for the decoupled strain model in the main paper.

The discussion on generalizations held in Section 4, introducing a weight matrix \mathbf{W} and a change of basis matrix \mathbf{N} , is likewise applicable to equation (53). To verify that our derivations are correct we dedicate the following section to applying equation (53) to our demonstration example presented in Appendix A.

C1. Application to demonstration example

For completeness we have implemented and solved equation (53) in our demo supplementary code for the same demonstration example that was considered in Appendices A and B. The same radial basis expansion as described in Appendix B was used during regression. Likewise, the true noise covariance matrix was used to solve for the unknown radial basis coefficients in a weighted least-squares sense. The resulting maximum likelihood reconstruction (REC) can be viewed in the middle row of Fig. 4 together with the ground truth (GT) and residual (RES).

Funding information

This work was funded by Vetenskapsrådet – Röntgen Ångström Cluster, project No. 2017-06719.

References

- Aarle, W. van, Palenstijn, W. J., Cant, J., Janssens, E., Bleichrodt, F., Dabrovolski, A., De Beenhouwer, J., Joost Batenburg, K. & Sijbers, J. (2016). *Opt. Express*, **24**, 25129–25147.
- Aarle, W. van, Palenstijn, W. J., De Beenhouwer, J., Altantzis, T., Bals, S., Batenburg, K. J. & Sijbers, J. (2015). *Ultramicroscopy*, **157**, 35–47.
- Hayashi, Y., Hirose, Y. & Seno, Y. (2015). *J. Appl. Cryst.* **48**, 1094–1101.
- Hektor, J., Hall, S. A., Henningsson, N. A., Engqvist, J., Ristinmaa, M., Lenrick, F. & Wright, J. P. (2019). *Materials*, **12**, 446.
- Hendriks, J. N., Wensrich, C. M. & Wills, A. (2020). *Strain*, **56**, e12341.
- Henningsson, A. & Hendriks, J. (2021). *J. Appl. Cryst.* **54**, 1057–1070.
- Henningsson, N. A., Hall, S. A., Wright, J. P. & Hektor, J. (2020). *J. Appl. Cryst.* **53**, 314–325.
- Korsunsky, A. M., Vorster, W. J. J., Zhang, S. Y., Dini, D., Latham, D., Golshan, M., Liu, J., Kyriakoglou, Y. & Walsh, M. J. (2006). *Acta Mater.* **54**, 2101–2108.
- Lionheart, W. R. B. & Withers, P. J. (2015). *Inverse Probl.* **31**, 045005.
- Palenstijn, W. J., Batenburg, K. J. & Sijbers, J. (2011). *J. Struct. Biol.* **176**, 250–253.
- Reischig, P. & Ludwig, W. (2020). *Curr. Opin. Solid State Mater. Sci.* **24**, 100851.