Comparison of different Gaussian fitting and phase contrast retrieval methods

Nargiza Djurabekova (ucabndj@ucl.ac.uk)
University College London

Ian Buchanan
University College London

Adam Doherty
University College London

Marco Endrizzi
University College London

Joel Enwald
University College London

Peter Munro
University College London

Alessandro Olivo
University College London

Savvas Savvidis
University College London

Simon Arridge
University College London

Article

Keywords:

Posted Date: October 27th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2190429/v1

License: © This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
Comparison of different Gaussian fitting and phase contrast retrieval methods

Nargiza Djurabekova1,*, Ian Buchanan2, Adam Doherty2, Marco Endrizzi2, Joel Enwald1, Peter Munro2, Alessandro Olivo2, Savvas Savvidis2, and Simon Arridge1

1University College London, Department of Computer Science, London, WC1V 6LJ, UK
2University College London, Department of Medical Physics and Biomedical engineering, London, WC1E 7JE, UK
*ucabndj@ucl.ac.uk

ABSTRACT

Edge illumination is an X-ray phase contrast imaging technique that allows for the recovery of differential phase and dark-field (also known as scattering) contrast, in addition to conventional attenuation based contrast. These signals are retrieved by analysing the illumination profile of a series of beamlets across a detector. This can provide a wealth of information about the sample, especially biological tissues which produce little contrast using conventional attenuation radiography. To perform signal retrieval, Gaussian curves can be fit to the illumination profiles using a variety of methods. In this work, we compare the most prominent of these methods in terms of their contrast to noise ratio as well as computational times to investigate their effect on the retrieved projection data as well as the final reconstructions.

Introduction

X-ray imaging is a vast field of study which is essential in a wide range of fields from medicine, allowing fast diagnostic decisions, to quality inspection in manufacturing. For certain soft-tissue or similarly attenuating materials, however, conventional X-ray imaging may not be sufficient. In such cases, enhanced contrast can be achieved by studying the phase shift of rays as they pass through objects. This is commonly known as X-ray Phase Contrast imaging or XPCi. XPCi requires a specialized, phase sensitive setup and while there are many options, such as analyser based imaging, free space propagation, Bonse/Hart or grating interferometry1, in this work we focus on edge-illumination (EI)2,3.

Phase contrast imaging was previously only possible at highly specialized facilities with monochromatic parallel beams, such as synchrotrons. In the past couple of decades however, XPCi has been adapted to work with conventional laboratory X-ray systems, using EI. EI is a technique that uses a set of two masks with vertical slits - one just upstream of the sample (sample mask) and another in front of the detector (detector mask). A diagram of the system can be found in Fig. 1a. The sample mask splits the incident source beam into multiple beamlets with a period \( p \) such that there is no overlap of beams at the detector. Meanwhile the detector mask creates regions of reduced sensitivity around pixel edges. The two masks have pitches that are harmonically matched and a horizontal shift along the \( x \)-axis (transverse to the mask slits) of the sample mask results in a change to the integrated intensity registered at the detector. These variations follow a Gaussian curve at each pixel as the mask is shifted laterally through one period with perfectly aligned masks corresponding to the highest intensity. This intensity function \( C(x) \) is known as an illumination curve (IC)4. Placing an object between the two masks and measuring the illumination curve again gives us a different IC, \( O(x) \), for each pixel. Examples of two such curves can be observed in Fig. 1b.

We assume that \( O(x) \) and \( C(x) \) are well described by

\[
y = \frac{A}{\sqrt{2\pi \sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),
\]

where \( y \) is observed data, while \( A, \mu \) and \( \sigma^2 \) are the amplitude, mean and variance parameters of the distribution, respectively. After fitting an appropriate Gaussian function to both curves, three information channels can be recovered corresponding to the three parameters defining the curve. That is, we recover absorption data as \( A = -\log(A_O/A_C) \), lateral beam shift as \( \mu = \mu_O - \mu_C \) and beam scattering as \( \sigma^2 = \sigma_O^2 - \sigma_C^2 \). The subscript \( C \) here refers to the IC without the sample, also known as flat field IC, and \( O \) refers to the sample IC. More details can be found in5.

The EI system retrieves the gradient of the phase of a sample, which is proportional to the refraction angle caused by the sample. This angle, \( \theta \), can be recovered with \( \mu/|z_{sample} - z_{detector}| \), since \( \tan(\theta) \) can be approximated with \( \theta \) in case of small
(a) Schematic of the EI setup. The sample mask is translated along the X axis ([1]) to acquire an illumination curve. [2] indicates a path of a refracted photon. Detector pixels which attain the additional (refracted) photons appear brighter in the projection data, while photons refracted away from the detector apertures lead to darker pixels, thus providing refraction/phase sensitivity. Original diagram from 6.

(b) Gaussian distribution of integrated beam intensity as a function of mask misalignment for a single detector pixel with (O(x)) and without (C(x)) a sample. The images acquired without a sample are referred to as flat field images.

Figure 1. Summary of the edge illumination method

angles. The relationship between refraction and phase $\phi$ can be described by

$$
\phi(x) = \frac{2\pi}{\lambda} \int_{x_0}^{x} \theta(x')dx',
$$

where we assume, without loss of generality, that $\phi(x_0) = 0$ and $\lambda$ is the X-ray wavelength$^{1-2}$. In other words, with the EI system, we have access to the first derivative of the phase shift along the direction of phase sensitivity and so to get back the phase, we must perform integration. Our approach for processing refraction data in this instance focuses on the CT reconstruction of phase. When performing filtered backprojection (or FBP), instead of the more common high-pass filter, we can reconstruct phase from refraction data by using a Hilbert filter $H$,

$$
H = -i \text{sgn}(k) = \frac{|k|}{ik},
$$

where $\text{sgn}$ is the signum function$^7$. The reconstruction procedures for regular FBP and phase-specific FBP are summarized in Fig. 2.

We investigate the most prominent methods for performing Gaussian fitting in section 3 and apply them all to the same imaged sample to demonstrate the results in section 1. In doing this, our goal is to find the most reliable and computationally efficient method for retrieval of X-ray phase contrast data.
Methods: | GD | TGD | M | tM | Q | wQ | GN  
---|---|---|---|---|---|---|---
Times (s): | 249.6 | 324.5 | 1.105 | 8.800 | 5.294 | 7.344 | 268.1

Table 1. Table comparing computational performance between different fitting methods in seconds.

Figure 2. A schematic summarising FBP reconstruction steps for conventional X-ray CT (top) and phase-contrast X-ray CT (bottom). Here $\mathcal{R}$ refers to the X-ray forward operator and $\mathcal{R}^*$ its adjoint, while $\mathcal{F}$ is the Fourier transform and $\mathcal{F}^{-1}$ its inverse. In the first step, a sample is imaged to either get an attenuation function $f(s)$ (top) or a first order gradient of phase $\partial f$ (bottom) over a spatial parameter $s$. Then the functions are converted to the Fourier domain and an appropriate filter (ramp or Hilbert) is applied. Finally, inverse Fourier transform is applied and finally the data is backprojected to get the CT reconstruction of attenuation (top) and phase (bottom). More details can be found in §1.

1 Results

We tested all of the methods described in section 3 on a sample consisting of re-cellularised porcine oesophageal scaffolds. These scaffolds were placed inside a 5 mm cylindrical container and scanned with an EI system, using a source with a molybdenum target, located at UCL\(^8\). This sample was chosen as it contains challenging biological tissue, with small features which conventional X-ray imaging techniques struggle to resolve. More details about the design of the data acquisition experiment can be found in §9. The region of interest covered 413 × 143 pixels, with a 50 μm pixel size, for each projection angle, with 1200 such angles from $0^\circ$ to $359.7^\circ$. All of the data was acquired with 5 points on the illumination curve. Summary of computational times for every method can be found in Table 1. The algorithms were executed using MATLAB R2022a on a Linux workstation with an AMD EPYC 7543P 32-Core Processor.

Note that throughout this paper, the abbreviation GD refers to performing a gradient descent with a single Gaussian described in sec. 3.1, TGD to the gradient descent method which fits three Gaussians to three neighbouring pixels which we explain in sec. 3.2, M refers to using moments while tM is the symbol for moments computed with the trapezoidal rule, both detailed out in sec. 3.3, Q represents the quadratic fit and wQ its weighted counterpart, both described in sec. 3.4, and finally GN corresponds to the Gauss-Newton algorithm which we summarise in sec. 3.5.
Figure 3. An example of projections acquired by fitting Gaussians to illumination curve points using different methods. From left to right: gradient descent with a single Gaussian, gradient descent with three Gaussians, moments, moments using the trapezoid rule, quadratic fit, weighted quadratic fit and the Gauss-Newton algorithm. From top to bottom: retrieved attenuation, refraction and scattering.

The results of all the methods are displayed in Fig. 3 and 5, with the retrieved projections shown in Fig. 3 and the reconstructions of a single central slice in Fig. 5. To perform a quantitative comparison between the methods, we use the Conrast-to-Noise Ratio (CNR) which we compute as follows. We choose three regions on a reconstructed slice (oesophagus, parafilm and background, see Fig. 4) and compute the means of two of them, $\mu_{os}$ and $\mu_{pf}$, as well as the standard deviation of the background region $\sigma_{bg}$ to have an estimate of the noise. Then the CNR can be computed with

$$CNR = \frac{\mu_{os} - \mu_{pf}}{\sigma_{bg}}.$$  

(4)

Table 2 contains the CNRs for all three retrieved parameters for each investigated method.
Table 2. CNRs of the reconstructed slices, the higher the value, the clearer the image.

<table>
<thead>
<tr>
<th>Methods:</th>
<th>GD</th>
<th>TGD</th>
<th>M</th>
<th>tM</th>
<th>Q</th>
<th>wQ</th>
<th>GN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attenuation:</td>
<td>5.36</td>
<td>2.32</td>
<td>2.01</td>
<td>3.10</td>
<td>6.27</td>
<td>6.19</td>
<td>6.14</td>
</tr>
<tr>
<td>Phase:</td>
<td>27.9</td>
<td>26.3</td>
<td>24.4</td>
<td>26.0</td>
<td>22.9</td>
<td>28.0</td>
<td>27.9</td>
</tr>
<tr>
<td>Scattering:</td>
<td>8.66</td>
<td>8.95</td>
<td>8.54</td>
<td>8.26</td>
<td>9.08</td>
<td>8.44</td>
<td>8.39</td>
</tr>
</tbody>
</table>

2 Discussion

The purpose of this work has been to investigate the advantages and disadvantages of different Gaussian fitting methods to extract attenuation, scattering and refraction (and therefore phase) information from edge illumination phase contrast data. A frequently employed approach has so far involved computing moments and using them as initialization for a gradient descent scheme, particularly for three Gaussian fitting (sec. 3.2). This method’s popularity is derived from its ability to accurately account for pixel signals arising from adjacent beamlets associated with adjacent pixels. This should allow for more sensitive scatter retrieval, which seems to be fairly consistent with our findings. As can be seen in Table 2, there is some variation in performance depending on which channel’s reconstruction one prioritises. For attenuation, the best performing methods are quadratic, weighted quadratic and Gauss-Newton. For phase, it is weighted quadratic, single Gaussian gradient descent and Gauss-Newton. For the dark field, it is quadratic, and the single as well as triple Gaussian gradient descent schemes. Since visually all methods perform quite similarly (see Fig. 3 and Fig. 5), we create an overall score for each method based on the sum of its CNRs and subtracting the amount of time it takes with the current implementation (see Table 1). Based on this, the top three methods are weighted quadratic with the score of 35.3, moments with the score of 33.8 and the regular quadratic method with the score of 32.9. It is worthwhile to note that while fairly fast for small data, the weighted quadratic method is demanding in terms of memory capacity if one attempts to do a matrix inversion for a large number of pixels simultaneously, meaning its score could drop if it has to be repeated a few times for smaller pixel batches. This means that moments may be the

![Figure 5](image_url)
best choice for estimating all three parameters. Upon developing a phantom that has known attenuation and refraction as well as scattering which behaves in a predictable way, this experiment can be repeated and the different retrieval methods compared to the available ground truth.

3 Methods

We refer to the sample mask displacements as points on an illumination curve or IC points. As discussed earlier, we assume that this curve follows a Gaussian distribution. To study how the parameters of this Gaussian shape change when a sample is present, we perform retrieval, i.e., fitting an appropriate function to the measured IC points with and without the sample. As the two approaches are the same, we only demonstrate the fitting procedures for one illumination curve (corresponding to a single projection pixel). Note that for we assume that individual pixels all have their own ICs and we perform retrieval pixelwise.

3.1 Iteratively fitting a single Gaussian

In this section we describe the procedure for fitting a Gaussian to a few points (5 in our experiments) on the illumination curve using an iterative gradient descent method. This means the problem of fitting a gaussian to the IC data has to be first formulated as a loss and then the minimum of this loss is found by taking steps in the opposite direction of the gradient. As we want to perform the fitting for all IC curves in parallel, it helps for them to all have the same value range. Because of this, each pixel’s illumination curve is divided by its amplitude, so that the maximum of each one is 1.

To have an initial guess for a suitable Gaussian, we first need to initialize the different parameters as \( A_s, 0 \), \( \mu_s, 0 \), and \( \sigma_s, 0 \) and find an appropriate step size \( \tau \) for the steps to be taken in the direction of the negative gradient. Our current implementation performs initialization with basic moments (see more in section 3.3) and all the step sizes for the gradient descent scheme are chosen to be a constant value, which we manually tune. We currently have two stopping criteria: number of iterations and the difference between the fitted function and measured data falling below a pre-defined tolerance level. Let \( y \) be the measured IC points and

\[
g(x, A_s, \mu_s, \sigma_s) = \frac{A_s}{\sqrt{2\pi}\sigma_s^2} \exp\left(-\frac{(x - \mu_s)^2}{2\sigma_s^2}\right),
\]

be the function we try to fit to the data \( y \). Then the \( L_2 \) squared loss we try to minimize is:

\[
D = \|y - g(x, A_s, \mu_s, \sigma_s)\|_2^2.
\]

Each iteration of this gradient descent algorithm has the following steps. First, we find the gradient of (6) with respect to each individual parameter, \( A_s \), \( \mu_s \) and \( \sigma_s \) with

\[
\frac{\partial D}{\partial A_s} = \frac{1}{\sqrt{2\pi}\sigma_s^2} \exp\left(-\frac{(x - \mu_s)^2}{2\sigma_s^2}\right) \cdot (y_n - g(x, A_s, \mu_s, \sigma_s) / 2),
\]

\[
\frac{\partial D}{\partial \mu_s} = \frac{A_s(x - \mu_s)}{\sqrt{2\pi}\sigma_s^2} \exp\left(-\frac{(x - \mu_s)^2}{2\sigma_s^2}\right) \cdot (y_n - g(x, A_s, \mu_s, \sigma_s) / 2),
\]

\[
\frac{\partial D}{\partial \sigma_s} = -\frac{A_s(\mu_s + \sigma_s - x)(-\mu_s + \sigma_s + x)}{\sqrt{2\pi}\sigma_s^4} \cdot (y_n - g(x, A_s, \mu_s, \sigma_s) / 2).
\]

Then gradient descent is performed by iteratively updating the initial guess for the three parameters by taking a step \( \tau \) in the direction of the gradient. That is

\[
A_{s,n+1} = A_{s,n} + \tau \frac{\partial D}{\partial A_s}
\]

\[
\mu_{s,n+1} = \mu_{s,n} + \tau \frac{\partial D}{\partial \mu_s}
\]

\[
\sigma_{s,n+1} = \sigma_{s,n} + \tau \frac{\partial D}{\partial \sigma_s}
\]

We set \( \tau = 2 \times 10^{-5} \) manually to be small enough not to diverge for any given pixel and to match the step size used for the TGD method, described in sec. 3.2. The stopping criteria were defined as the relative residual dropping below a tolerance of \( 1 \times 10^3 \) or reaching the maximum number of iterations set to be 15. The method converged with the former stopping criterion only after around 65 iterations for the sample projections and 45 iterations for the flat field images but this took prohibitively long, so upon reaching the maximum number of iterations, the algorithm was interrupted. The converged result was comparable to that of the Gauss-Newton algorithm (see sec. 3.5).
3.2 Fitting three Gaussians

Due to the nature of the imaging system, there is a lot of crosstalk between neighbouring pixels, with up to 20% of photon overspill still reaching even the second nearest neighbour. The almost 60% of crosstalk from the nearest neighbour is handled by blocking every other column of the detector. Nevertheless, to ensure that we extract as much information about the distribution of the illumination curve as possible, not only do we fit a Gaussian curve to each pixel, but also to two of its secondary neighbours\(^{10}\). So the fitting function \(g\) is now a sum of three Gaussians

\[
g_t(x, A_t, \mu_t, \sigma_t) = \sum_{j=-1}^{1} A_{t,j} \exp \left( -\frac{(x + j p - \mu_{t,j})^2}{2\sigma_{t,j}^2} \right),
\]

where \(p\) is the sample mask period. The only parameters recorded for each pixel are \(A_{t,0}, \mu_{t,0}\) and \(\sigma_{t,0}^2\), but we perform gradient descent similarly to Eq. 8 to optimize for all 9 parameters

\[
A_{t,j,n+1} = A_{t,j,n} + \tau \frac{\partial D_t}{\partial A_{t,j}},
\]

\[
\mu_{t,j,n+1} = \mu_{t,j,n} + \tau \frac{\partial D_t}{\partial \mu_{t,j}},
\]

\[
\sigma_{t,j,n+1} = \sigma_{t,j,n} + \tau \frac{\partial D_t}{\partial \sigma_{t,j}}.
\]

As when fitting a single Gaussian (sec. 3.1), we use the same step size and stopping criteria. With this chosen step size of \(2 \times 10^5\) and even a hundred iterations, the method did not converge for all pixels with the relative residual as the stopping criterion. However, when choosing a larger step size of \(2 \times 10^4\), the method did converge in under 10 iterations for many of the pixels but since it diverged for some, we chose a smaller step size instead and interrupted the algorithm after just 15 iterations.

3.3 Moments

Moments can be used to describe the shape of a distribution, each moment describing a specific characteristic. Of particular interest to us are the first 3 moments, starting at the zeroth moment\(^{11}\). Assume that \(N_{IC}\) refers to the total number of points sampling the illumination curve.

1. The zeroth moment \(M_0 = E(X^0)\) approximates the area under the curve and corresponds to \(A\) from Eq. 1. It can be computed as

\[
A_m = \Delta x \sum_{n}^{N_{IC}} y_n,
\]

assuming the distance between the sampled points on the illumination curve are equally spaced. Alternatively, if we are certain that the distribution follows a Gaussian curve, \(A_m\) can be estimated as follows

\[
A_m = \max(y_n) \sqrt{2\pi \sigma_m^2},
\]

where \(\sigma_m^2\) can be estimated from the second moment. If the distribution is not Gaussian, the area can also be estimated using the trapezoidal rule (in MATLAB, \(\text{trapz}(x, y)\)).

2. The first moment \(M_1 = E(X^1)\) approximates the mean, i.e. the point around which the Gaussian curve is centred. This corresponds to \(\mu\) in Eq. 1 and is computed as

\[
\mu_m = \Delta x \sum_{n}^{N_{IC}} \frac{x_n \cdot y_n}{A_m}.
\]

This is equivalent to

\[
\mu_m = \frac{\sum_{n}^{N_{IC}} x_n y_n}{\sum_{n}^{N_{IC}} y_n},
\]

The trapezoidal method can be implemented in MATLAB with \(\text{trapz}(x, x \cdot y)/A_m\).
3. The second moment $M_2 = E(X^2)$ approximates variance (corresponding to $\sigma^2$ in Eq. 1). It can be computed either as

$$\sigma_m^2 = \Delta x \sum_{n=1}^{N_{IC}} (x_n - \mu_m)^2 \cdot y_n,$$

or as

$$\sigma_m^2 = \left| \frac{\sum_{n=1}^{N_{IC}} (x_n - \mu_m)^2 y_n}{\sum_{n=1}^{N_{IC}} y_n} \right|.$$

The trapezoidal rule gives the following MATLAB code: \texttt{trapz}(x, (x-\mu_m)^2 \cdot y)/\mu_m.

### 3.4 Quadratic fit

Given observed data $y$, a Gaussian curve with some amplitude $A_q$, mean $\mu_q$ and variance $\sigma_q^2$ can be fitted to the data as follows

$$y = \frac{A_q}{\sqrt{2\pi \sigma_q^2}} \exp \left( -\frac{(x - \mu_q)^2}{2\sigma_q^2} \right)$$  \hspace{1cm} (17a)

$$\log(y) = \log \left( \frac{A_q}{\sqrt{2\pi \sigma_q^2}} \exp \left( -\frac{(x - \mu_q)^2}{2\sigma_q^2} \right) \right)$$  \hspace{1cm} (17b)

$$\log(y) = \log \left( \frac{A_q}{\sqrt{2\pi \sigma_q^2}} \right) \frac{\mu_q^2}{2\sigma_q^2} + \frac{2\mu_q x}{2\sigma_q^2} - \frac{x^2}{2\sigma_q^2}$$  \hspace{1cm} (17c)

$$\log(y) = \xi_0 + \xi_1 x + \xi_2 x^2$$  \hspace{1cm} (17d)

$$\log(y) = Q\xi$$  \hspace{1cm} (17e)

$$\Rightarrow \xi = Q^{-1} \log(y).$$  \hspace{1cm} (17f)

Here, equality in 17e comes from setting $\xi_0 = \log \left( \frac{A_q}{\sqrt{2\pi \sigma_q^2}} \right) - \frac{\mu_q^2}{2\sigma_q^2}$, $\xi_1 = 2\mu_q / 2\sigma_q^2$ and $\xi_2 = -0.5 / \sigma_q^2$. At 17f, we simplify notation, with $\xi = [\xi_0, \xi_1, \xi_2]$ and $Q = [1, x, x^2]^T$, where $x$ is the vector of sample mask displacements. The parameters can be estimated as

$$\sigma_q^2 = -\frac{1}{2\xi_2}$$  \hspace{1cm} (18a)

$$\mu_q = -\frac{\xi_1}{2\xi_2}$$  \hspace{1cm} (18b)

$$A_q = \sqrt{2\pi \sigma_q^2} \exp \left( \xi_0 + \frac{\mu_q^2}{2\sigma_q^2} \right).$$  \hspace{1cm} (18c)

By giving more weight to larger values of $\log(y)$, we can achieve a lower mean squared error by using the data $y$, as a preconditioner\textsuperscript{12}. Starting from eq.(17e), we apply this preconditioner to both sides. Then, to be able to do pointwise multiplication between $Q$ and $y$, we repeat the matrix $Q$ as many times as there are pixels and also repeat $y$ 3 times to match the new dimensions of $Q$. Then the inversion and matrix multiplication at 19c is performed "pagewise", i.e. in parallel for every pixel.

$$\log(y) = Q\xi$$  \hspace{1cm} (19a)

$$y \odot \log(y) = y \odot Q\xi$$  \hspace{1cm} (19b)

$$\Rightarrow \xi = (y \odot Q)^{-1} (y \odot \log(y)).$$  \hspace{1cm} (19c)

where $\odot$ refers to pointwise multiplication.

Then $\sigma_q^2, \mu_q$ and $A_q$ are calculated as in eq.18. This weighted approach gives less importance to small-valued data, where the proportion of noise in the signal is higher.
3.5 The Gauss-Newton algorithm

To speed up the convergence of the iterative gradient descent method, we use a simplified version of the Newton algorithm, the Gauss-Newton algorithm. The Newton algorithm’s variable update has the form

\[ x_{n+1} = x_n + H^{-1} \nabla f, \]  

(20)

where \( H \) is the pre-computed Hessian. However, since Hessians are frequently slow and difficult to compute, it is instead far more common to use an approximation of the Hessian which utilizes the Jacobian \( J \). This approximation

\[ H \approx J^\top J, \]  

(21)

is derived from the Hessian. Let the residual for the data fidelity term \( D \) (as defined in Eq. 6) be \( r_D = (y - g) \), then the Hessian can be expanded as

\[ H_D = 2(J_D^\top J_D + \nabla^2 r_D). \]  

(22)

By ignoring the second order Laplacian term (second term of Eq. 22) we end up with the Gauss-Newton update

\[ x_{n+1} = x_n + (J^\top J)^{-1} \nabla f. \]  

(23)

We initialize the Gauss-Newton algorithm similarly to the gradient descent methods described above, with basic moments (see sec. 3.3) and each Gaussian only fitted to one pixel, without taking into account its neighbours. The benefit of using this algorithm is that there is no need to search for an appropriate step size and it is guaranteed to converge quite quickly. With the relative residual reaching a tolerance of \( 1 \times 10^{-3} \), the algorithm converged in under 10 iterations. However, due to the computationally demanding inversion of the Hessian approximation in Eq. 23, this algorithm actually took the longest to run, as can be seen in the Results section, Table 1.

Data availability

The datasets analysed during the current study are available from the corresponding author on reasonable request.

References

Author contributions statement

The different methods investigated in this work were implemented by ND, SA and JE. SS created the studied sample and performed the corresponding tomographic acquisition. The manuscript was largely written by ND with discussion and improvement suggestions from all authors.

Funding

This work was supported by the Engineering and Physical Sciences Research Council (EP/T005408/1) and by the National Research Facility for Lab X-ray CT (NXCT) through EPSRC grants EP/T02593X/1 and EP/V035932/1. AO was supported by the Royal Academy of Engineering under the Chairs in Emerging Technologies scheme (CiET1819/2/78). PM was supported by Royal Society University Research Fellowship URF\R\191036, ME by the Wellcome Trust 221367/Z/20/Z and SA by UK EPSRC EP/T000864/1 as well as UK EPSRC EP/V026259/1.

Competing interests

The authors declare no competing financial interests.