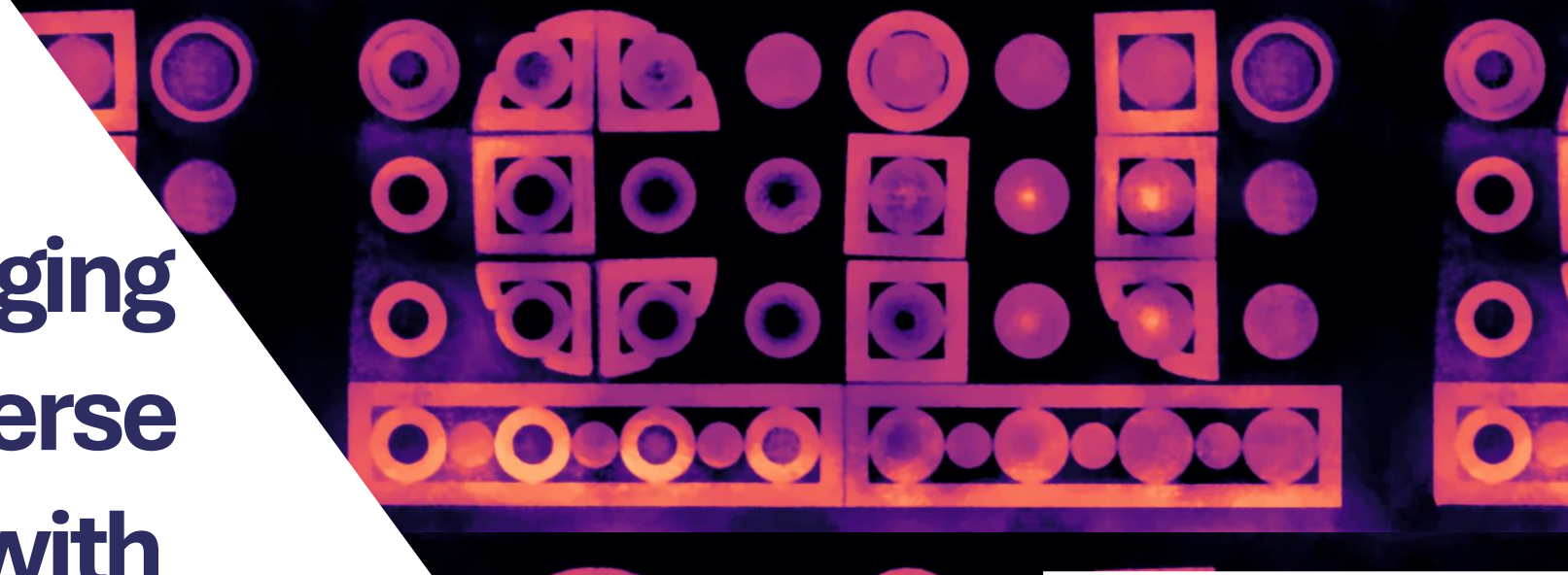
Using the Core Imaging Library (CIL) for inverse imaging problems with applications in CT



Margaret Duff, Edoardo Pasca and Letizia Protopapa

Scientific Computing,
Science and Technology Facilities Council, UK

Science and Technology Facilities Council
Scientific Computing

CIL
CORE IMAGING LIBRARY

CCPi
TOMOGRAPHIC IMAGING

ada lovelace centre

Science and Technology Facilities Council

- Run facilities at the Rutherford Appleton Laboratory, UK
- Scientific Computing provides computing infrastructure, software, support and expertise
- CIL team also at DTU and Finden

STFC Tomography and Imaging

- CCPi: Bring together expertise in computational research for tomography and imaging
  - Open-source software development, maintenance and distribution
  - Methods development
  - Community building
  - Training and user support network

Plan for this workshop

- Introduce tomography and the Core Imaging Library (CIL)
  - Standard reconstruction methods for CT
  - The importance of data (pre) processing

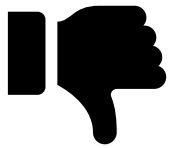
- Introduction to imaging inverse problems
  - Variational regularisation
  - CIL optimisation toolkit

- Combining mathematical approaches for deep learning
  - Ideas from the literature
  - Example in CIL
  - Over to you!

Confidence level check

- Python and juypter notebooks

- Inverse imaging problems
- Computed tomography

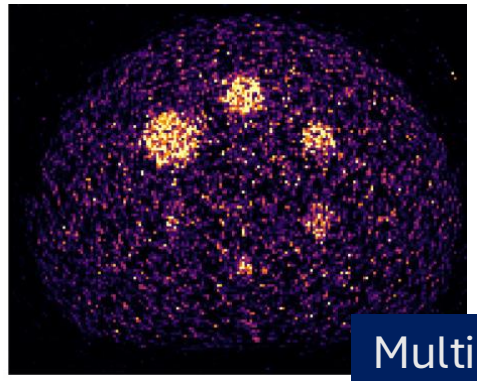


- Deep learning inverse problem approaches

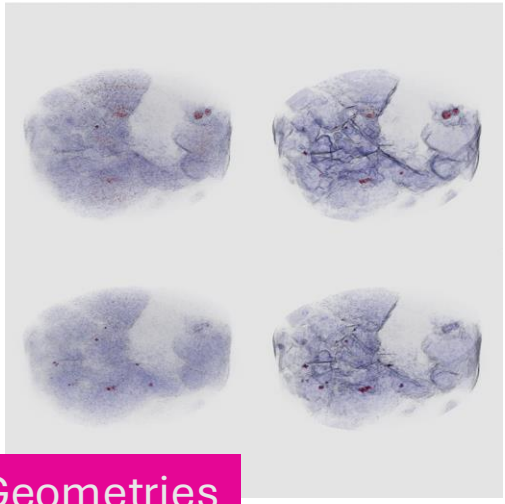
Core Imaging Library (CIL)

- Open-source python framework for CT and other inverse problems
- Methods for cone and parallel beam geometries for CT
- Tools for reading, processing, reconstructing and visualising CT data
- Optimisation tools for iterative reconstruction methods with a focus on challenging data
- Apache v2 license.
- Actively developed on GitHub: https://github.com/TomographicImaging/CIL

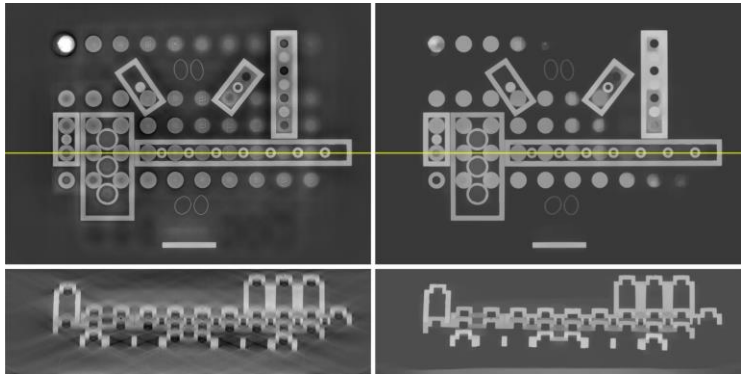
Noisy



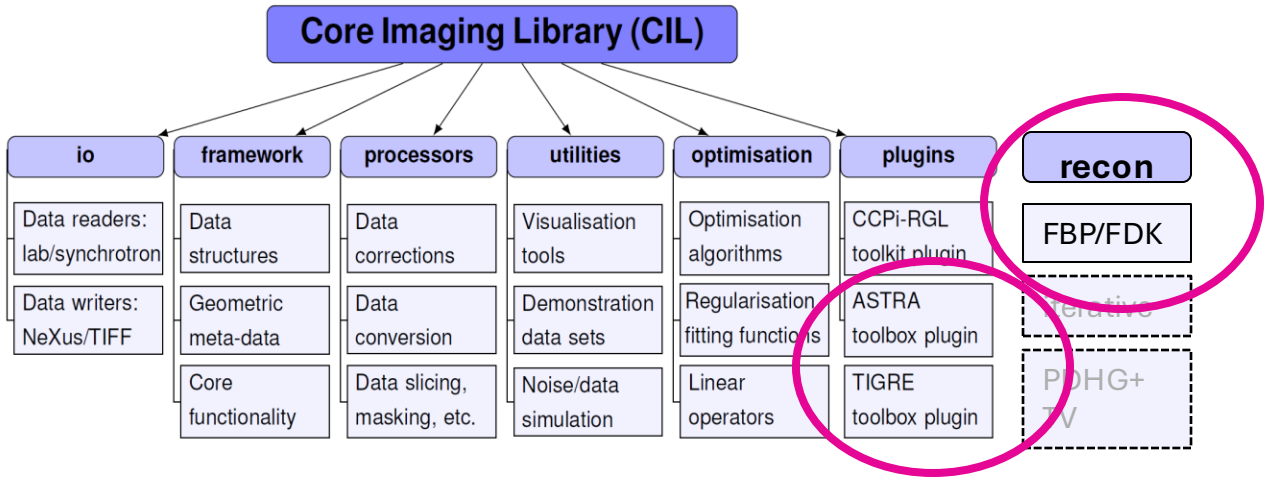
Multi-channel



Non-standard Geometries



Intro to CT and FBP

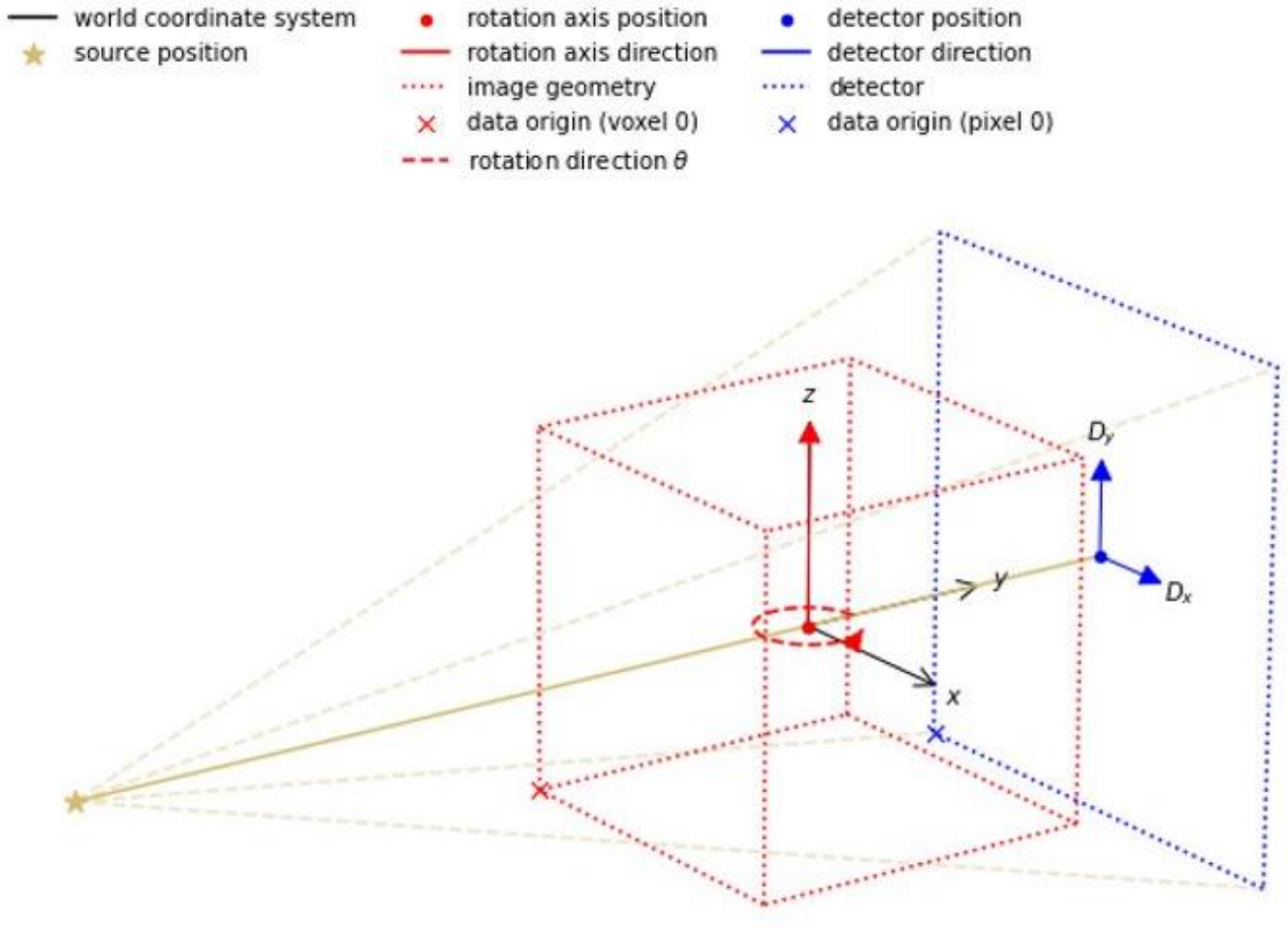


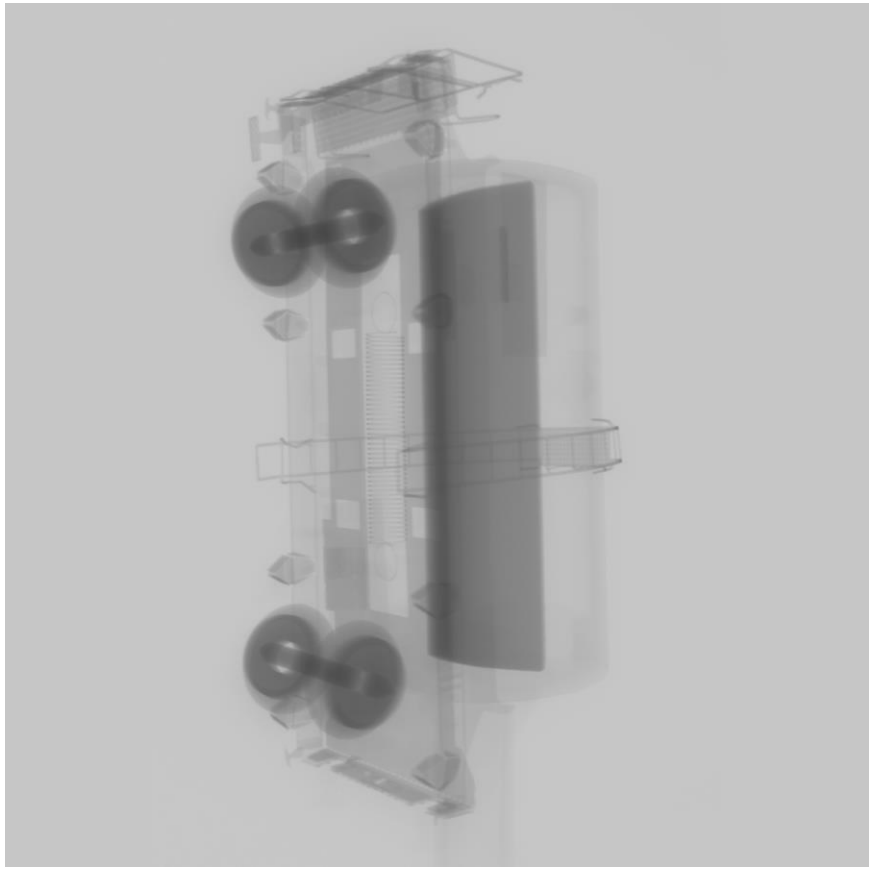
Science and Technology Facilities Council
Scientific Computing

CIL CORE IMAGING LIBRARY

CCPi TOMOGRAPHIC IMAGING

Tomography


world coordinate system
source position
rotation axis position
rotation axis direction
image geometry
data origin (voxel 0)
rotation direction θ
detector position
detector direction
detector
data origin (pixel 0)
z
y
x
D_y
D_x



Beer-Lambert Law

$$\frac{I}{I_0} = \exp \int_{L_i} -\mu(s)ds$$

$$b_i = -\log \frac{I_i}{I_0} = \int_{L_i} \mu(s)ds$$

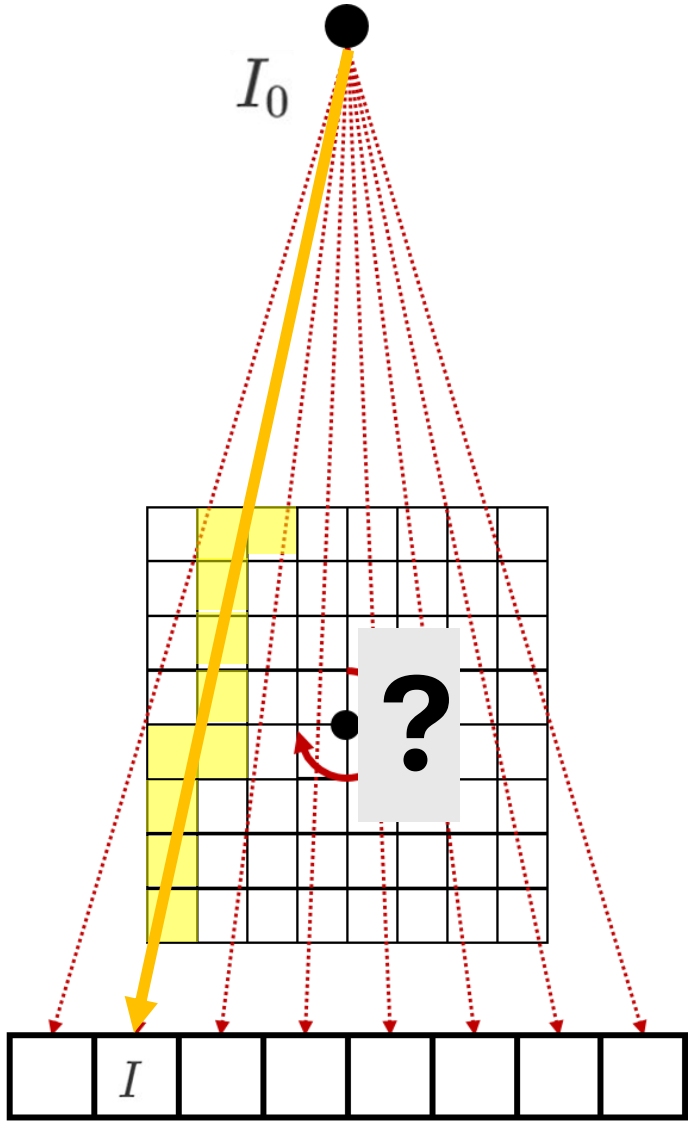
X-ray source

Measurement volume

Reconstructed volume:

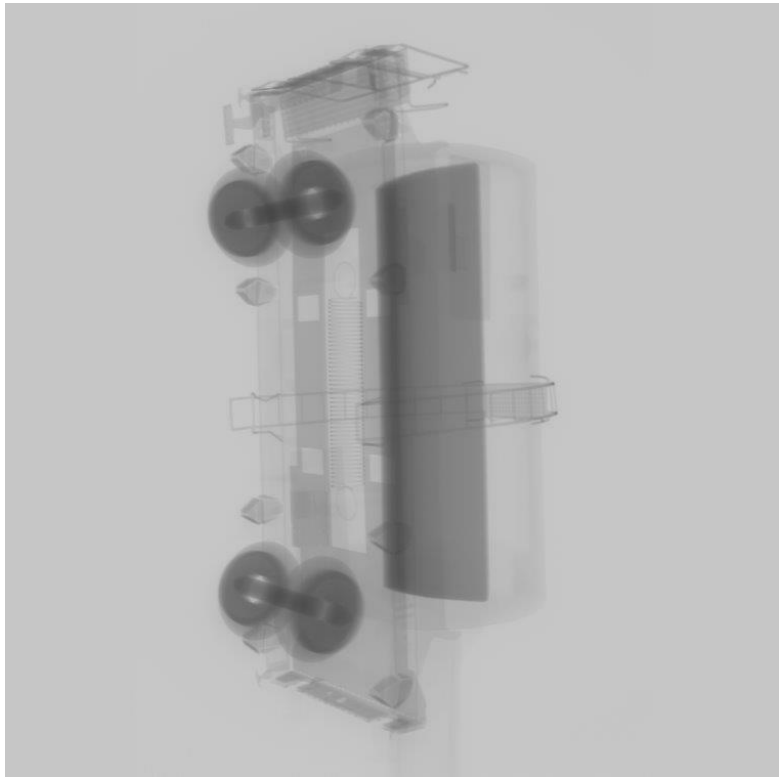
- map of linear attenuation coefficients (μ)
- often expressed in cm^{-1}

X-ray detector

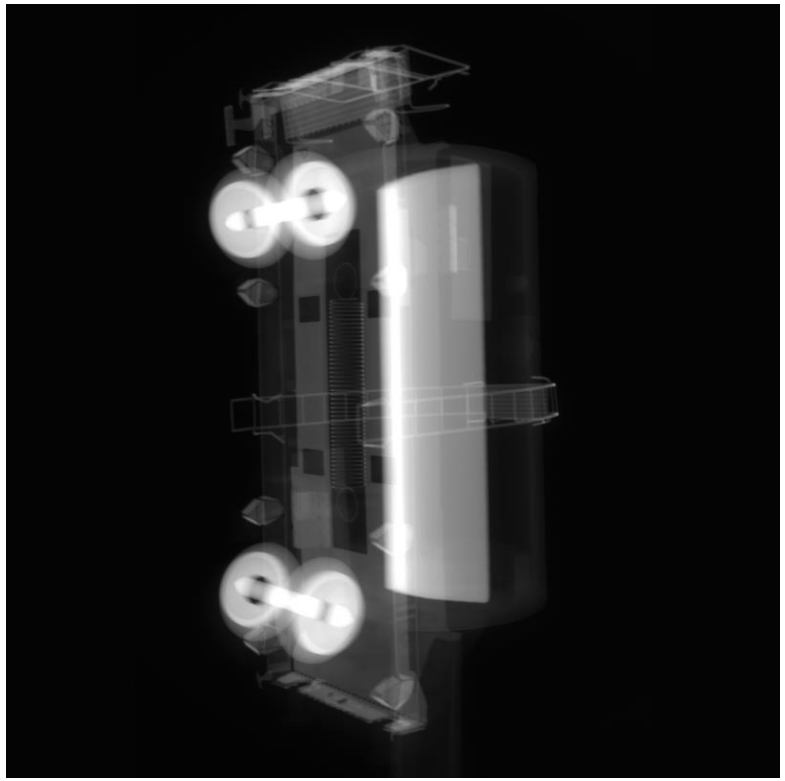


Minus log

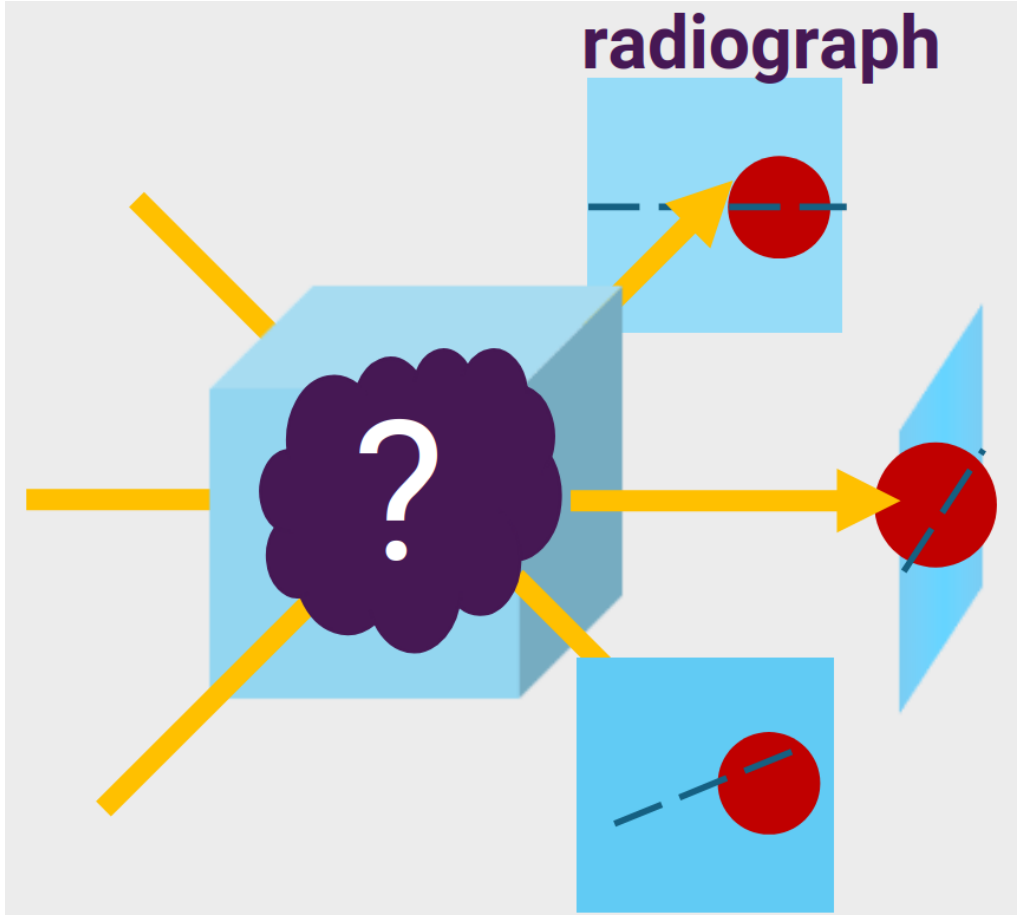
$$\frac{I}{I_0} = \exp \int_{L_i} -\mu(s)ds$$

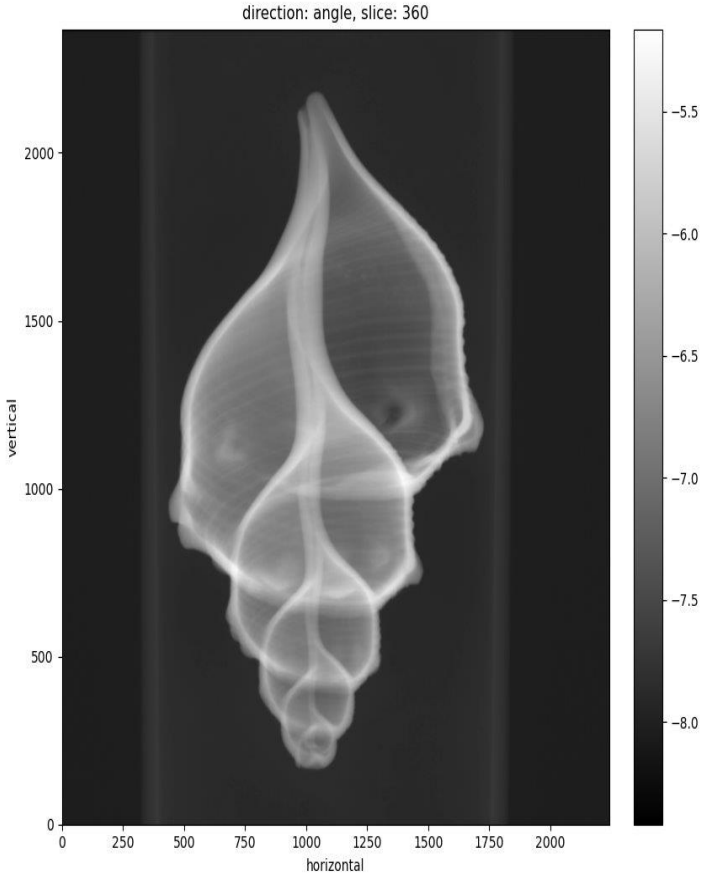


$$b_i = -\log \frac{I_i}{I_0} = \int_{L_i} \mu(s)ds$$



Take many projections

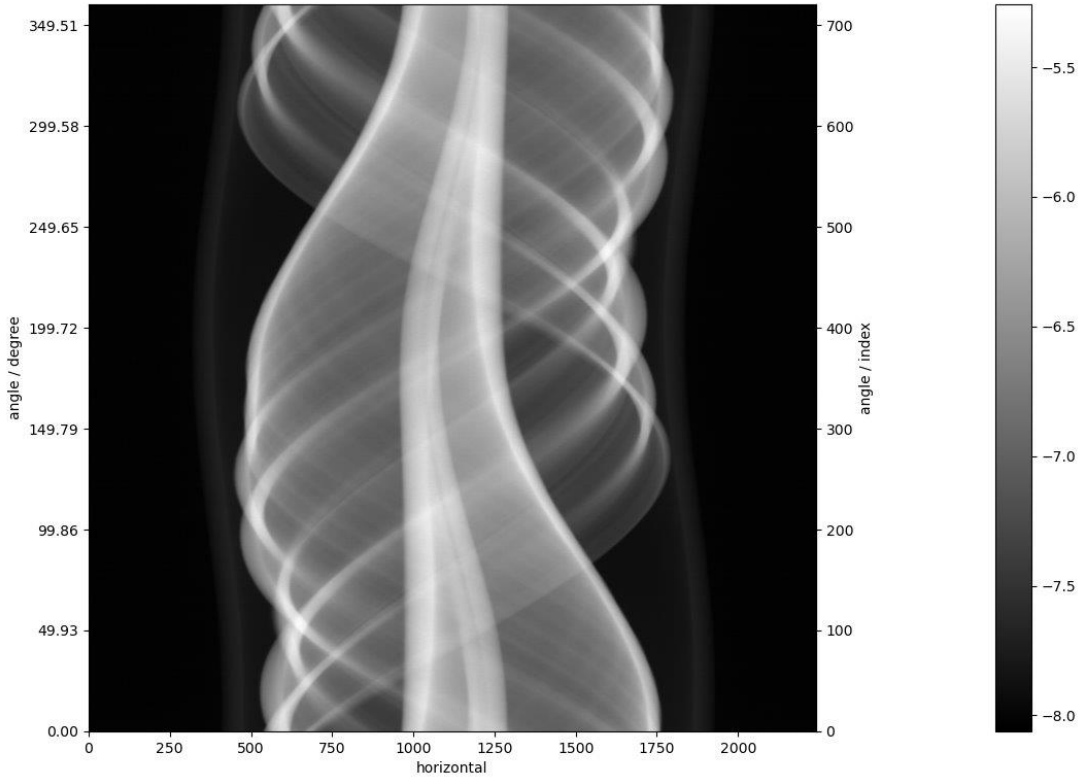




An x-ray radiograph of a shell measured at the University of Helsinki [1]

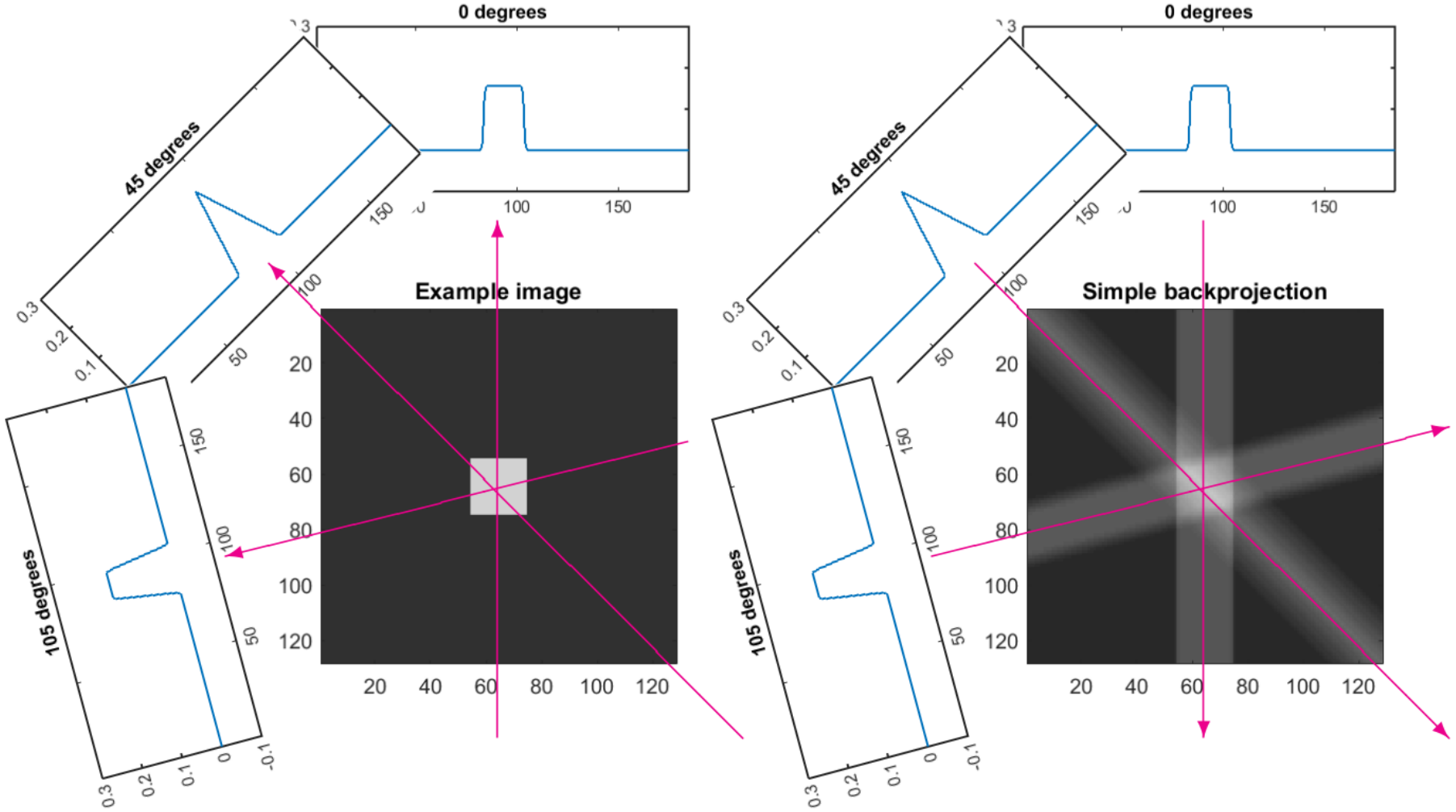
[1] Kamutta, E., Mäkinen, S., & Meaney, A. (2022). Cone-Beam Computed Tomography Dataset of a Seashell (1.1.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.6983008

Sinogram: How the attenuation changes as we rotate the object

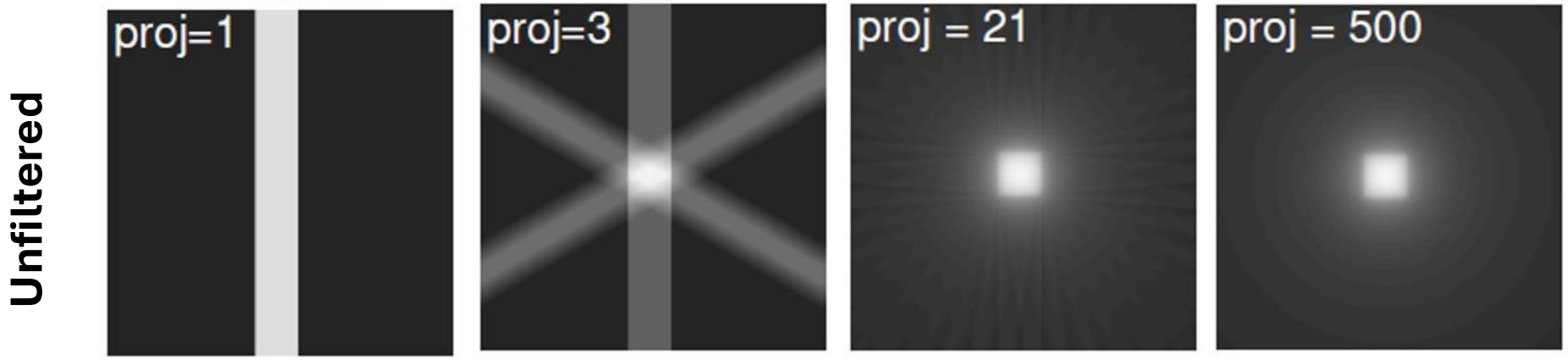


The shell has a much more complex sinogram

Filtered Back Projection (FBP)

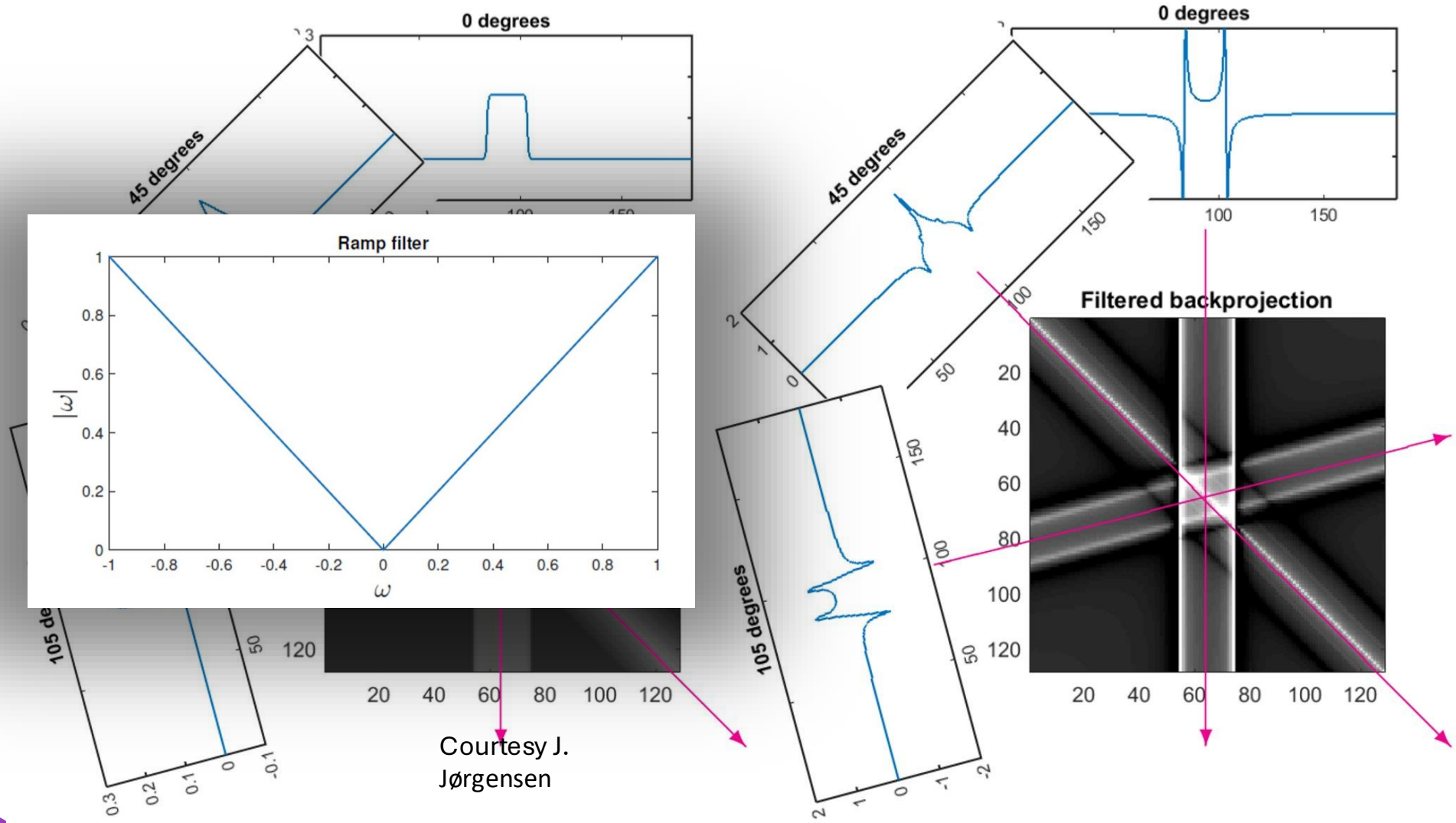


Filtered Back Projection (FBP)



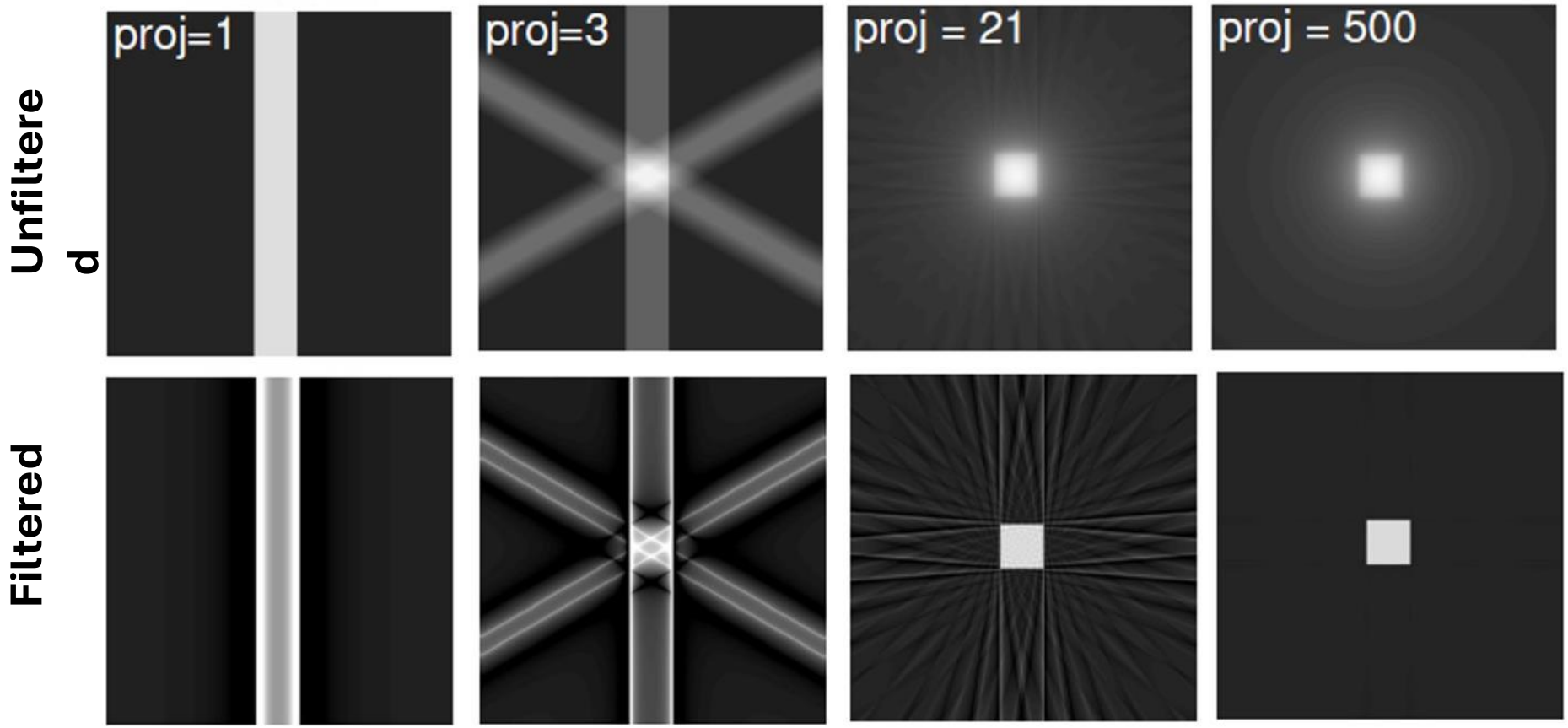
Courtesy J. Jørgensen

Filtered Back Projection (FBP)



Courtesy J. Jørgensen

Filtered Back Projection (FBP)



Courtesy J. Jørgensen

Filtered Back Projection (FBP)

Pros

- Fast as based on FFT and backprojection
- Few parameters
- Typically works very well
- Reconstruction behaviour well understood

Filtered Back Projection (FBP)

Pros

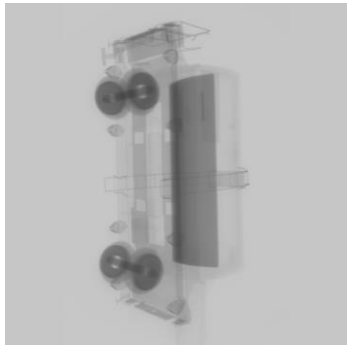
- Fast as based on FFT and backprojection
- Few parameters
- Typically works very well
- Reconstruction behaviour well understood

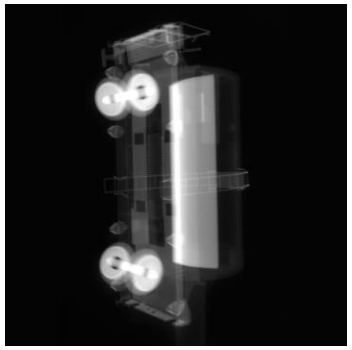
Cons

- Number of projections needed proportional to acquisition panel size
- Full angular range required (**limited angle** problem)
- Modest amount of noise tolerated
- Fixed scan geometries
- Cannot make use of prior knowledge such as non-negativity

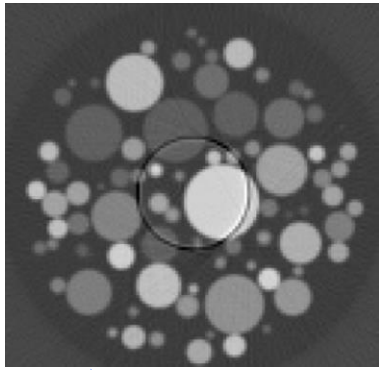
Data processing methods

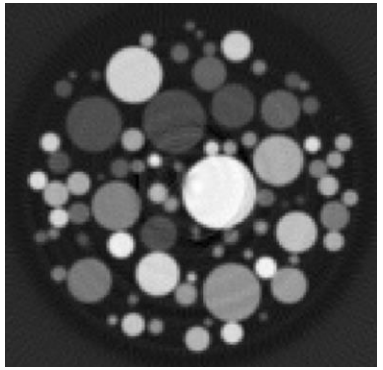
Convert to absorption



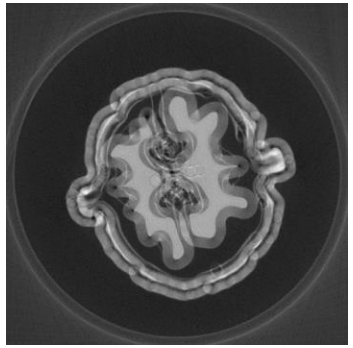


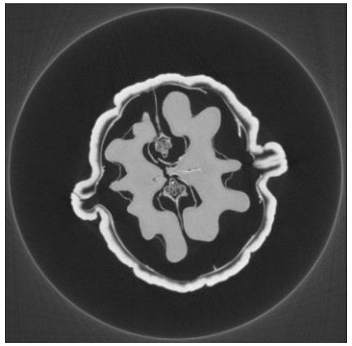
Ring removal

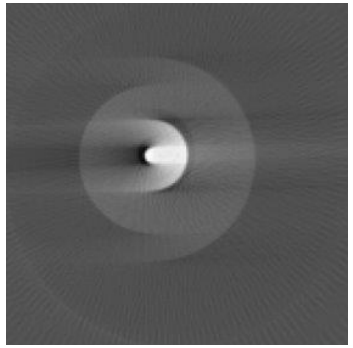


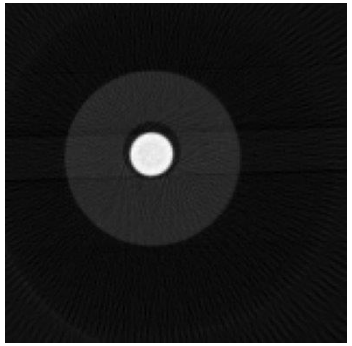


Centre of Rotation correction

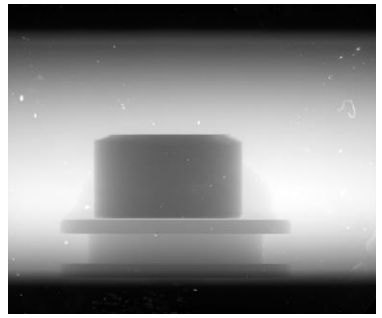


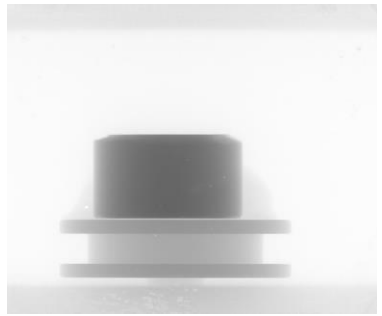




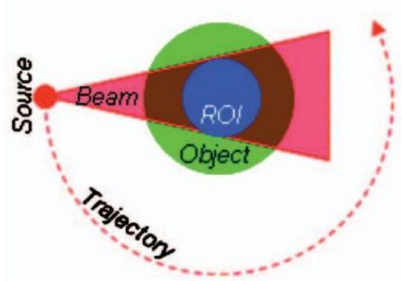


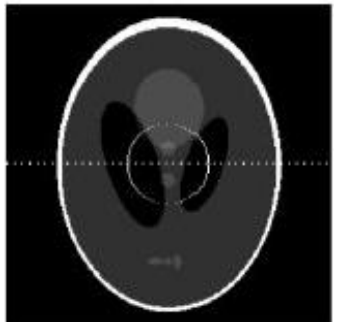
Flat-field correction

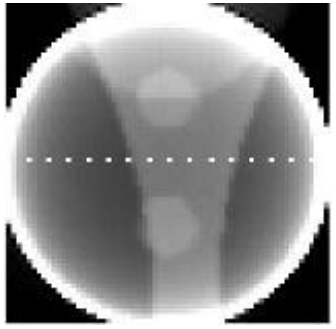


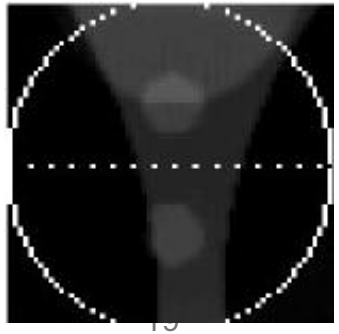


Region of interest scans



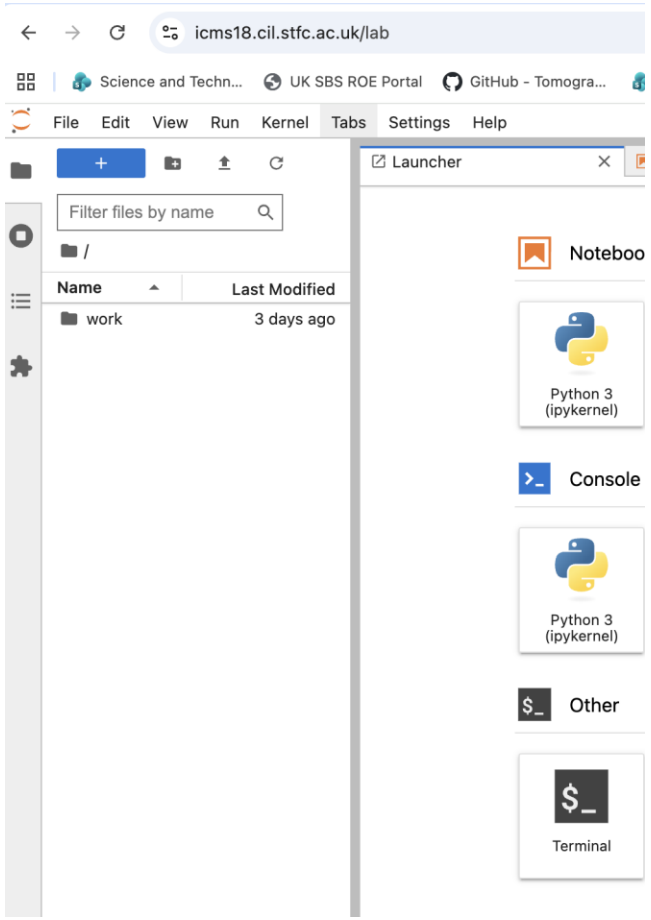






Accessing the jupyter system

1) Get into pairs or threes (we have 17 compute set-ups)
2) We will give you a link and a password
3) Follow the link and type in your password
4) (Hopefully) you will see something like this:



Example reconstruction - walkthrough

CIL-Demos/demos/1_introduction/**01_intro_walnut_conebeam.ipynb**

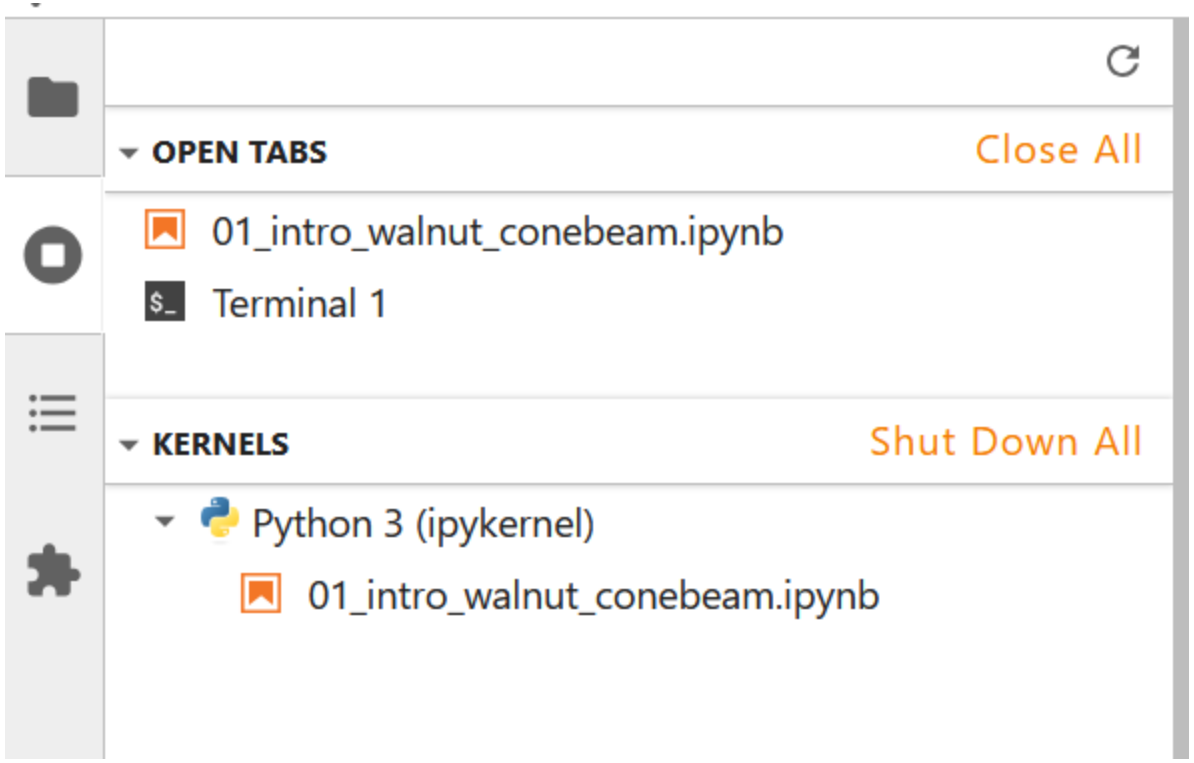
- Load and investigate a dataset
- Convert from Transmission to Absorption
- Compute a FDK reconstruction

Example reconstruction

CIL-Demos/demos/1_introduction/**01_intro_sandstone_parallel_roi.ipynb**

- Load and investigate a dataset
- Determine geometric information and set up data structures
- Apply basic pre-processors
- Compute a FBP reconstruction

Shut Down Notebooks

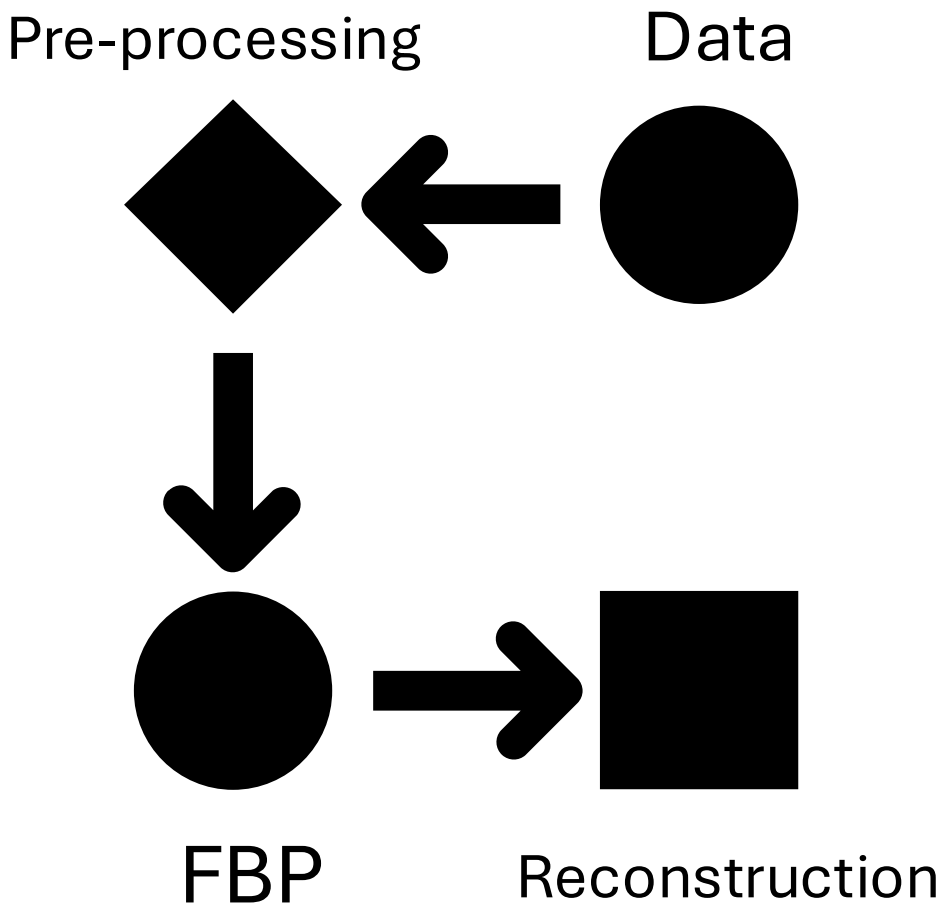


Wrap up

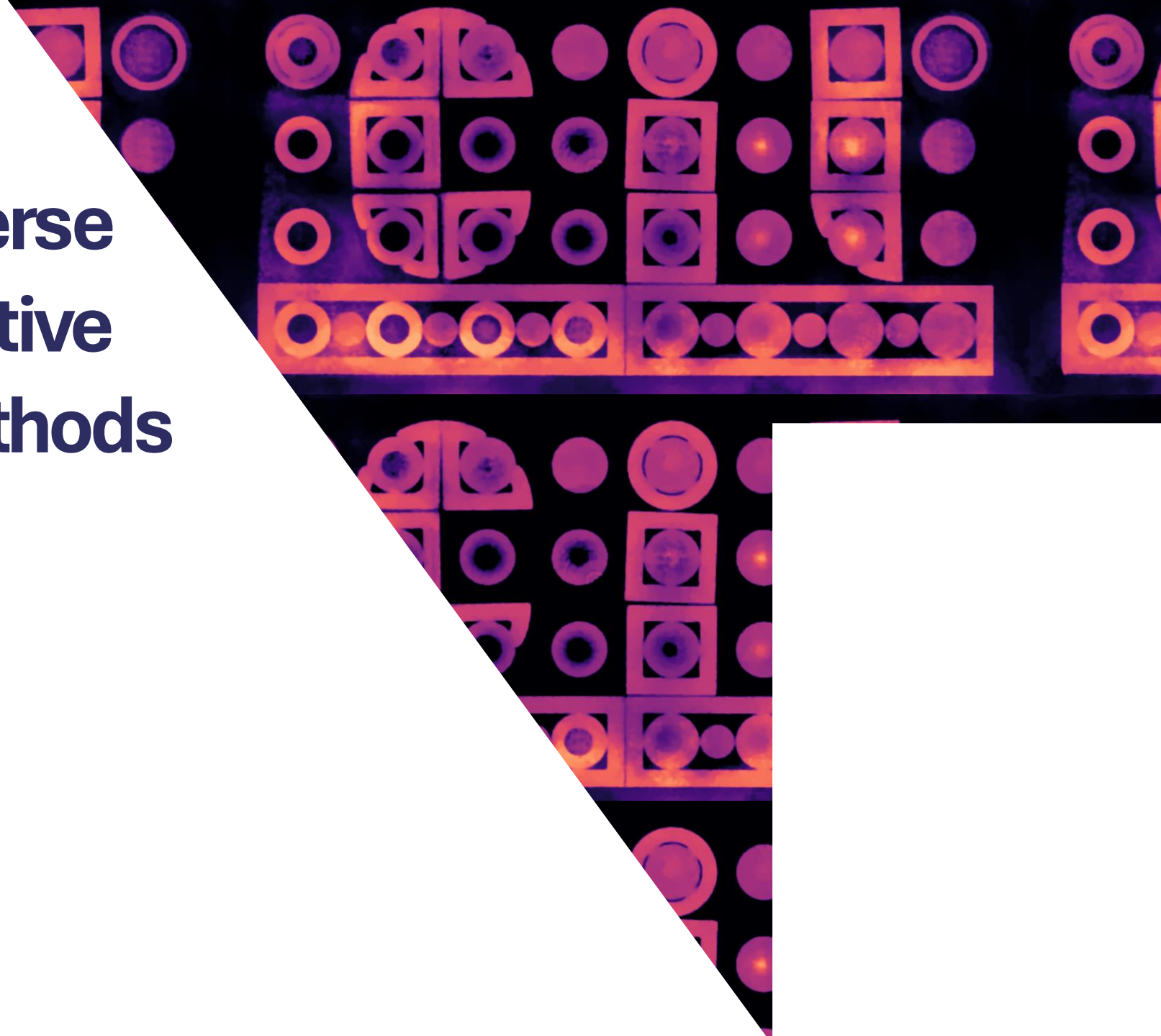
Filtered back-projection is very good!

- If your data is good it should work well
- Do any necessary pre-processing

If your data is not good… consider other methods



Introduction to inverse problems and iterative reconstruction methods



Science and Technology Facilities Council
Scientific Computing

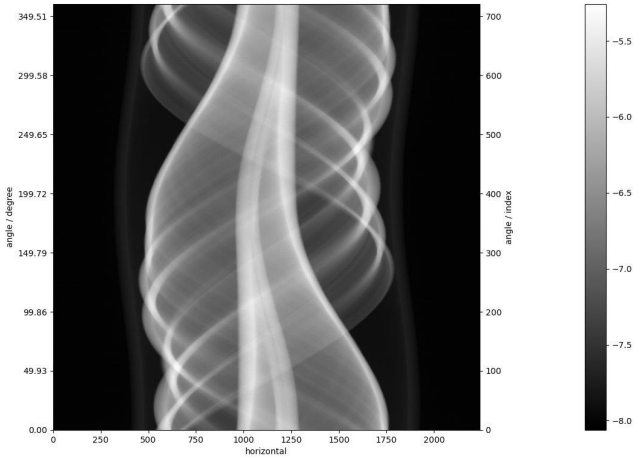
CIL
CORE IMAGING LIBRARY

CCPi
TOMOGRAPHIC IMAGING

Inverse problems

Forward A

Inverse



Ill posed problem

$$Au = b$$

Forward model **Image to reconstruct** **Data**

In case either:

1. No solution
2. Not unique solution
3. Solution sensitive to noise
4. Modelling errors in **A**

Inverse problem example - CT

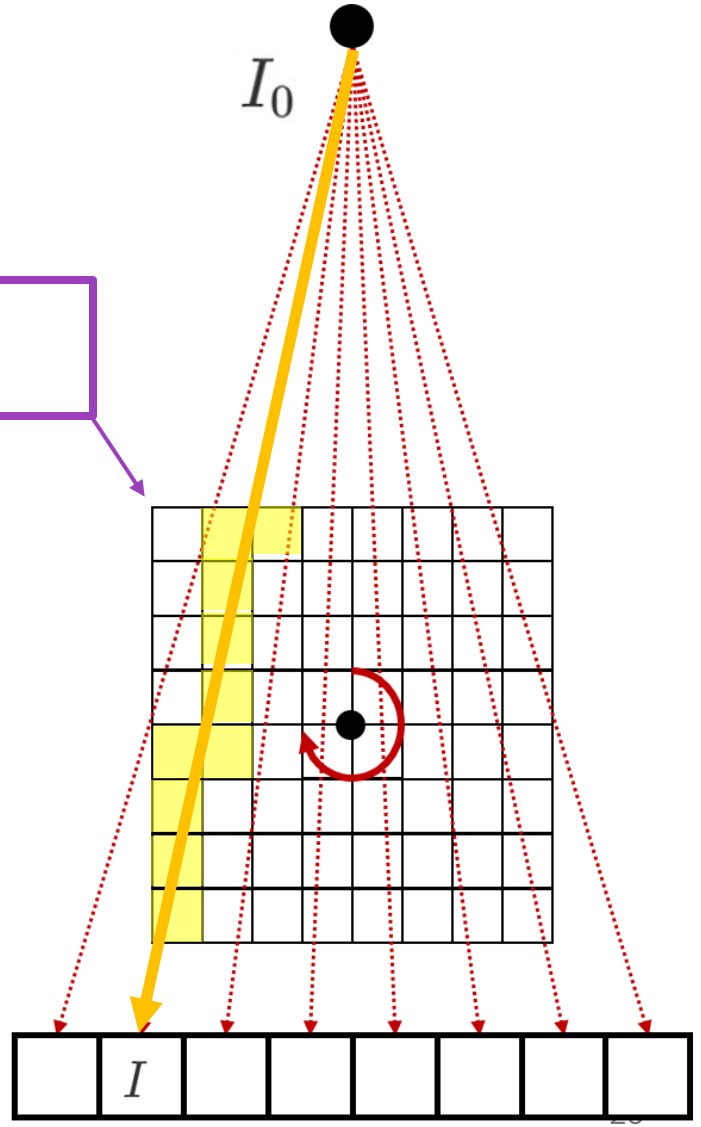
$$\frac{I}{I_0} = \exp \int_{L_i} -\mu(s)ds$$

$$b_i = -\log \frac{I_i}{I_0} = \int_{L_i} \mu(s)ds$$

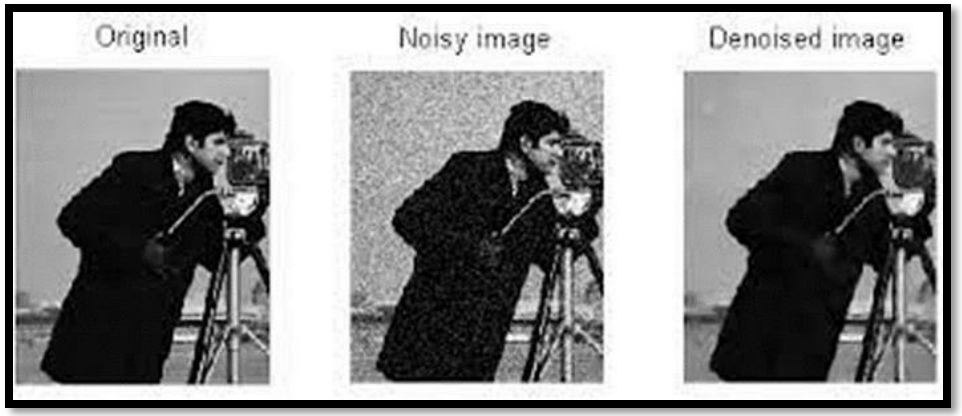
$$b_i = \sum_j a_{ij} u_j \rightarrow Au = b$$

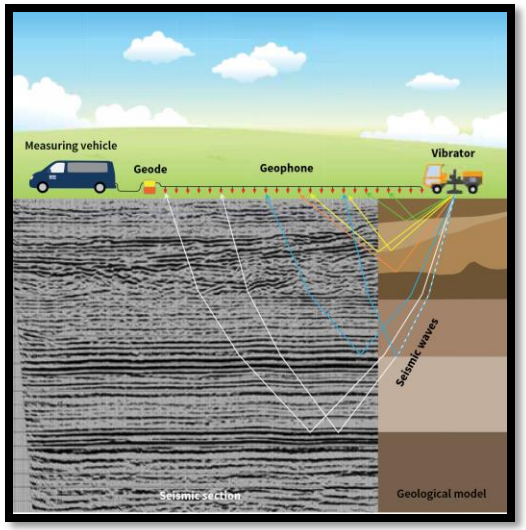
Extremely large set of linear equations!

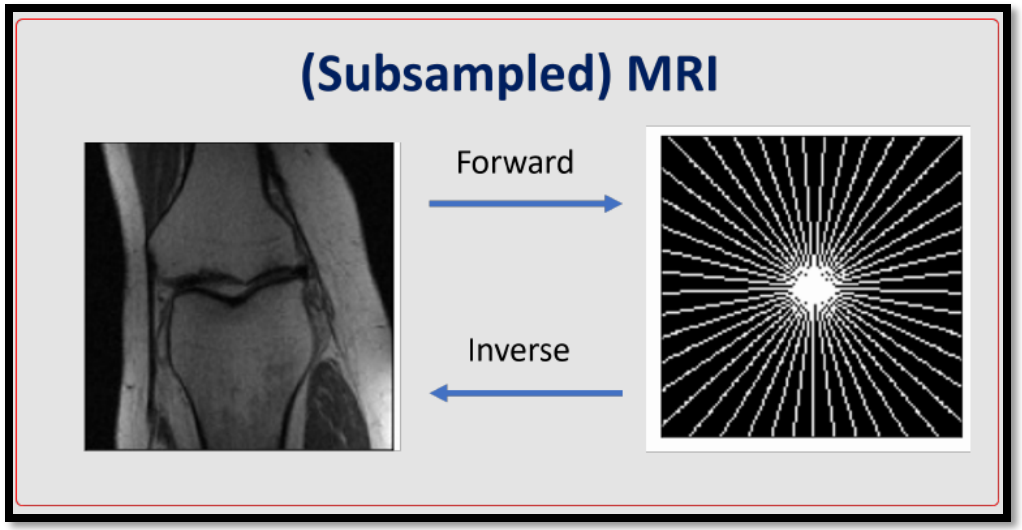
- Assume object is constant in each pixel
- u_j value of pixel j
- a_{ij} path length of ray i in pixel j

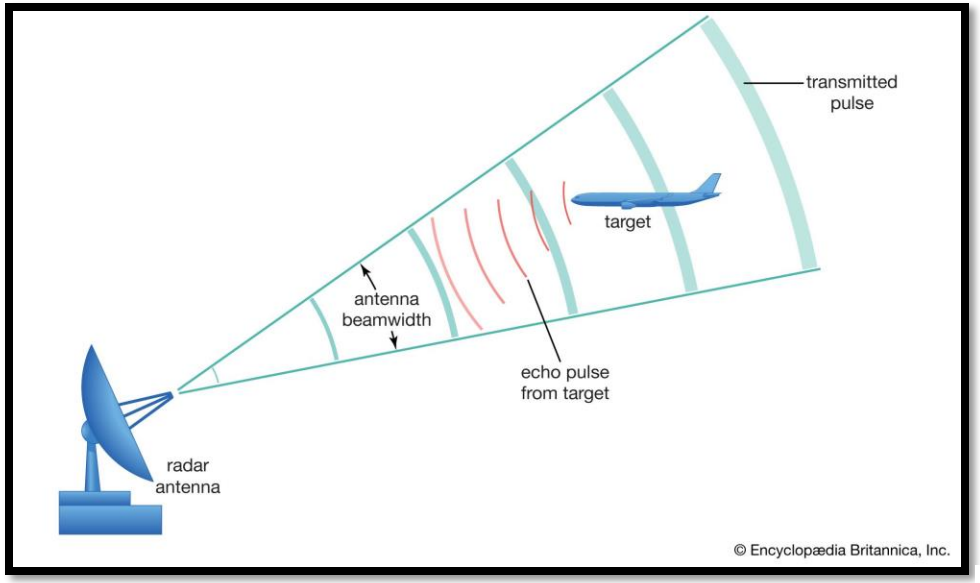


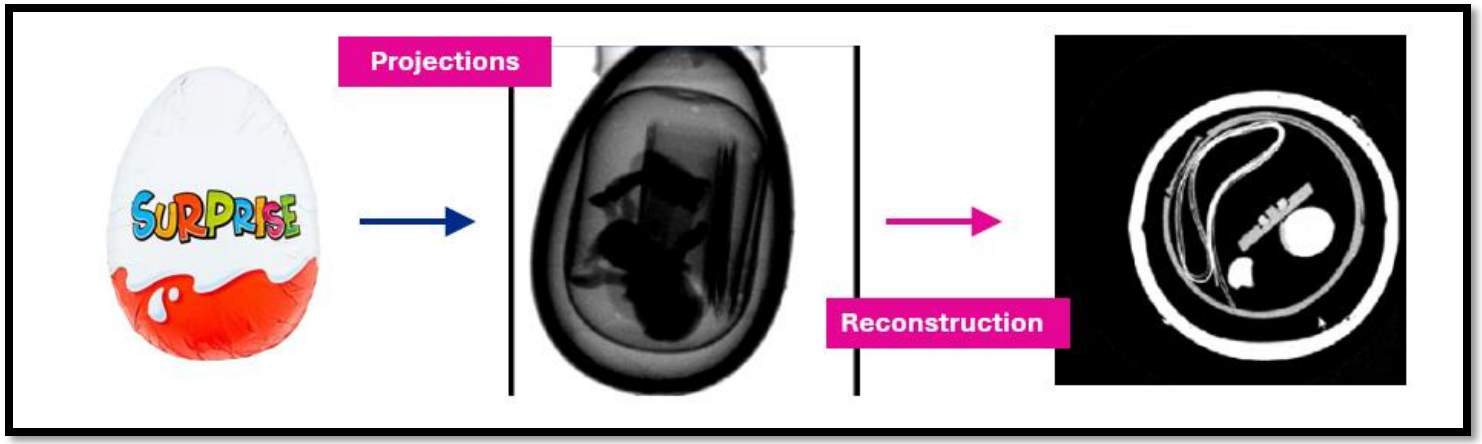
Other examples

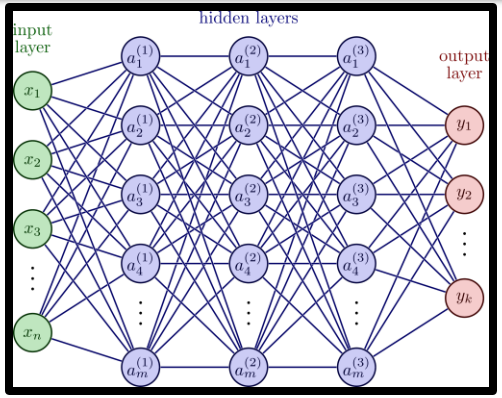












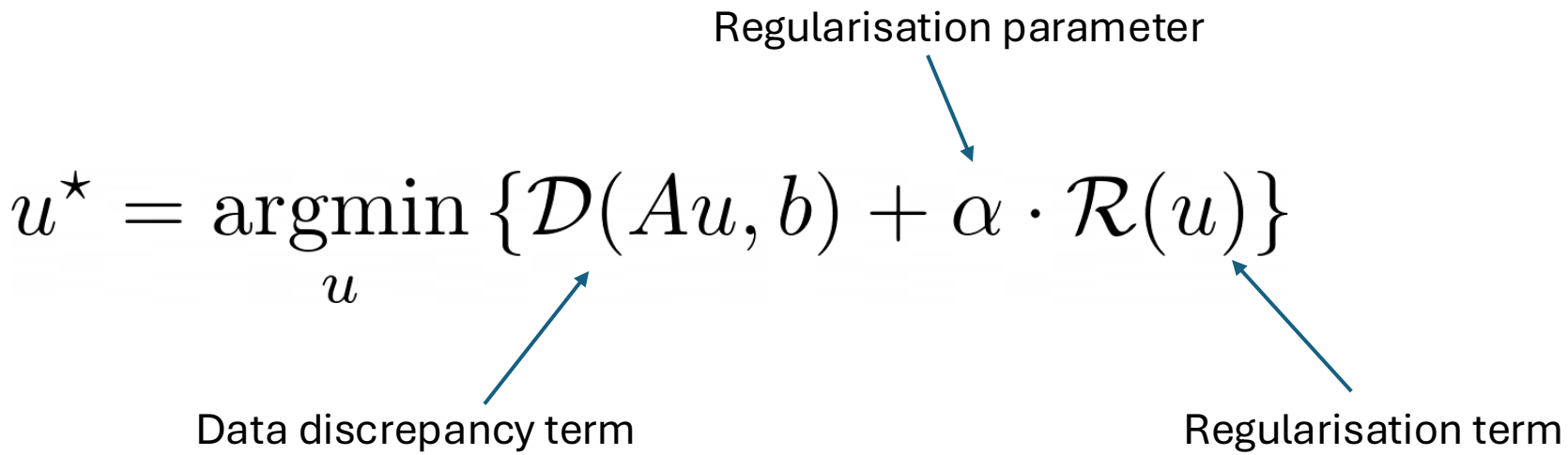
Science and Technology Facilities Council
Scientific Computing

CIL CORE IMAGING LIBRARY

CCPi TOMOGRAPHIC IMAGING

Why iterative methods?

- Direct inversion (e.g., FBP) subject to noise and incomplete data
- Iterative methods allow to incorporate noise models, priors etc.
- Most CT iterative reconstructions use regularization



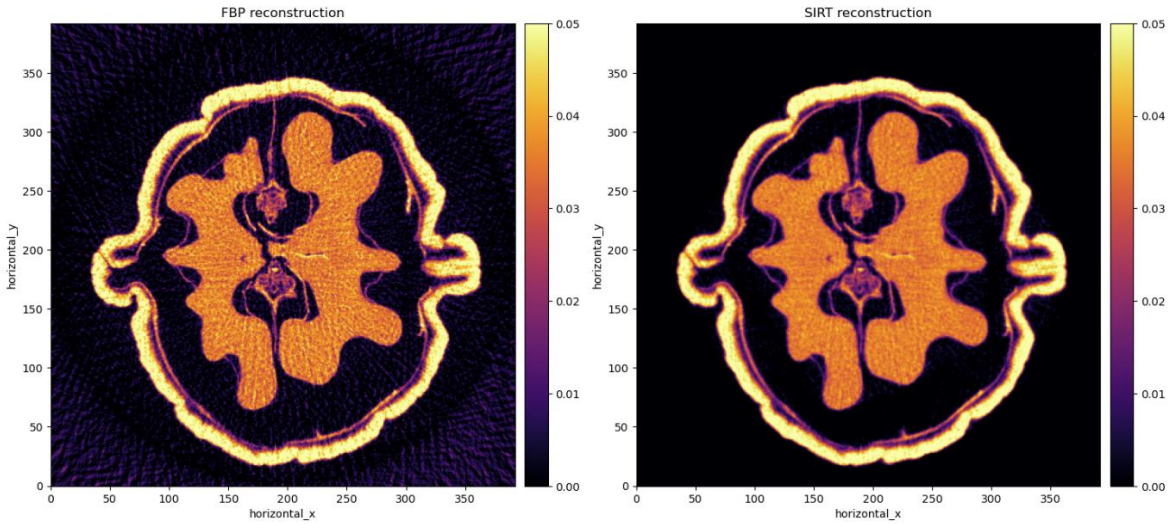
Example – no regularisation

$$Au = b$$

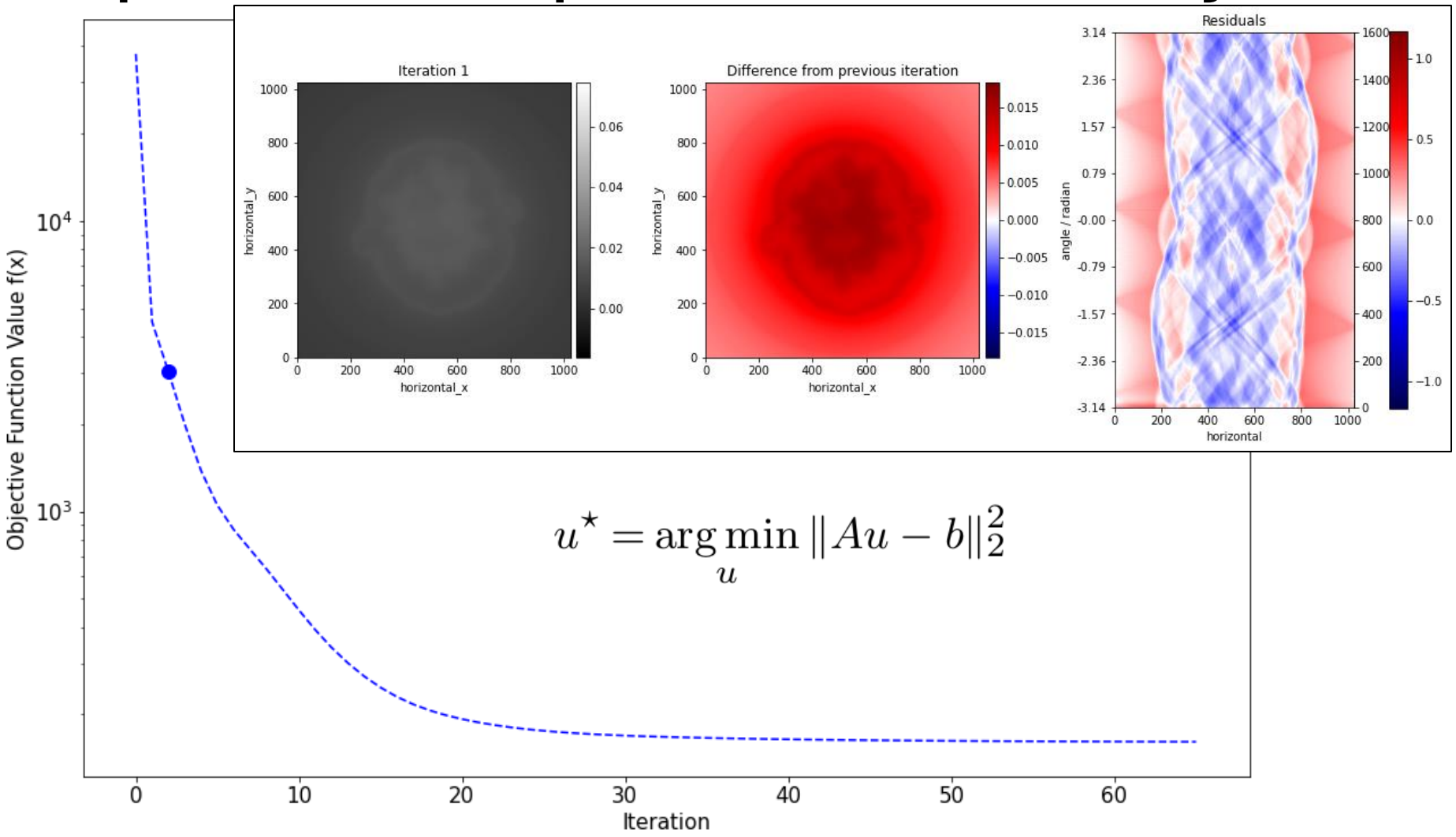
Construct the iterative reconstruction method based on optimisation algorithms and objective functions

$$u^\star = \underset{u}{\operatorname{argmin}} \{\mathcal{D}(Au, b)\}$$

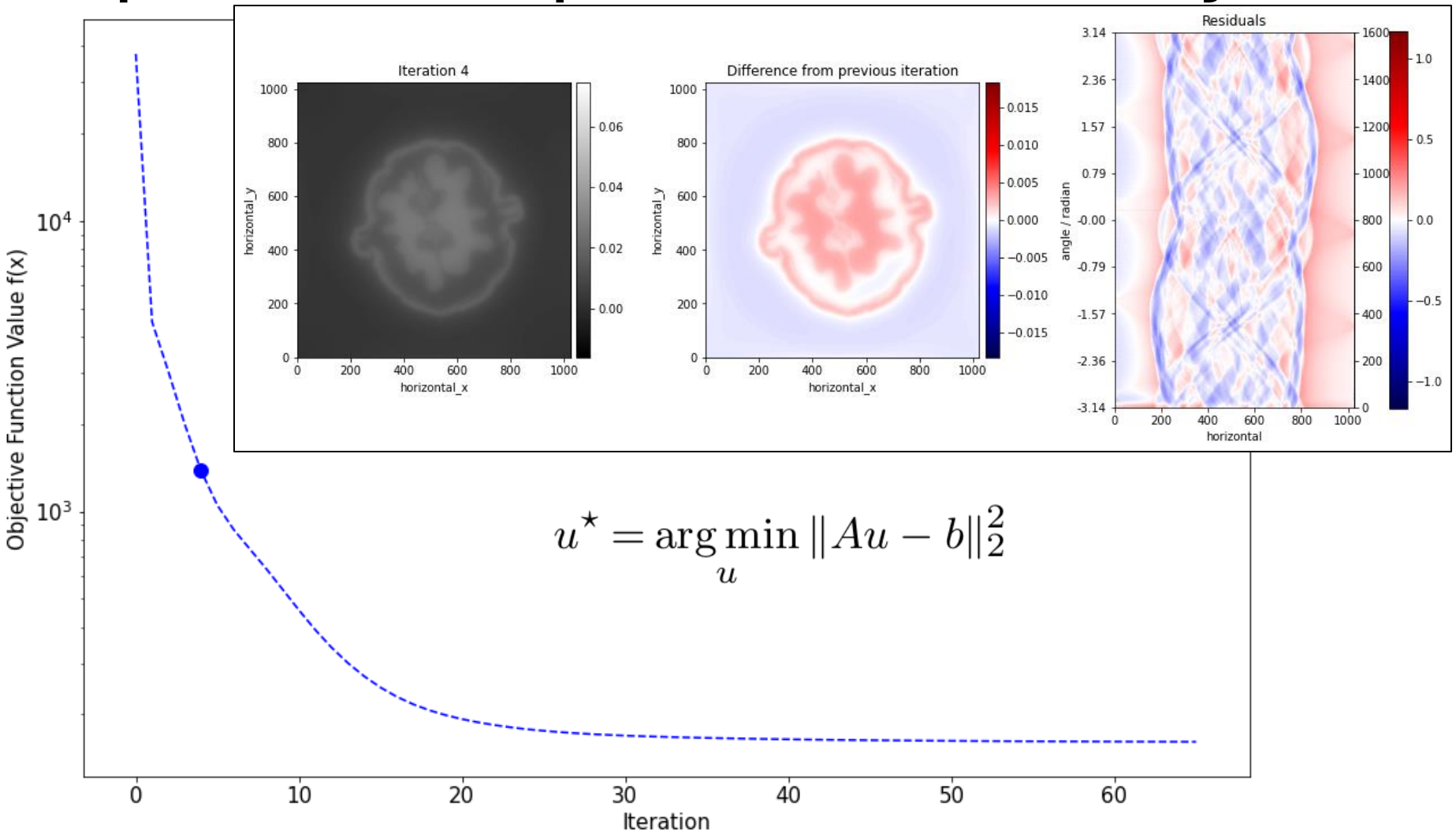
```
A = ProjectionOperator(ig, ag)
init = ig2D.allocate(0)
sirt = SIRT(initial = x_init, operator = A , data = b)
sirt.run(300, verbose=1)
sirt_recon = sirt.solution
show2D([fbp_recon,sirt_recon],
        title = ['FBP reconstruction','SIRT reconstruction'],
        cmap = 'inferno', fix_range=(0,0.05))
```



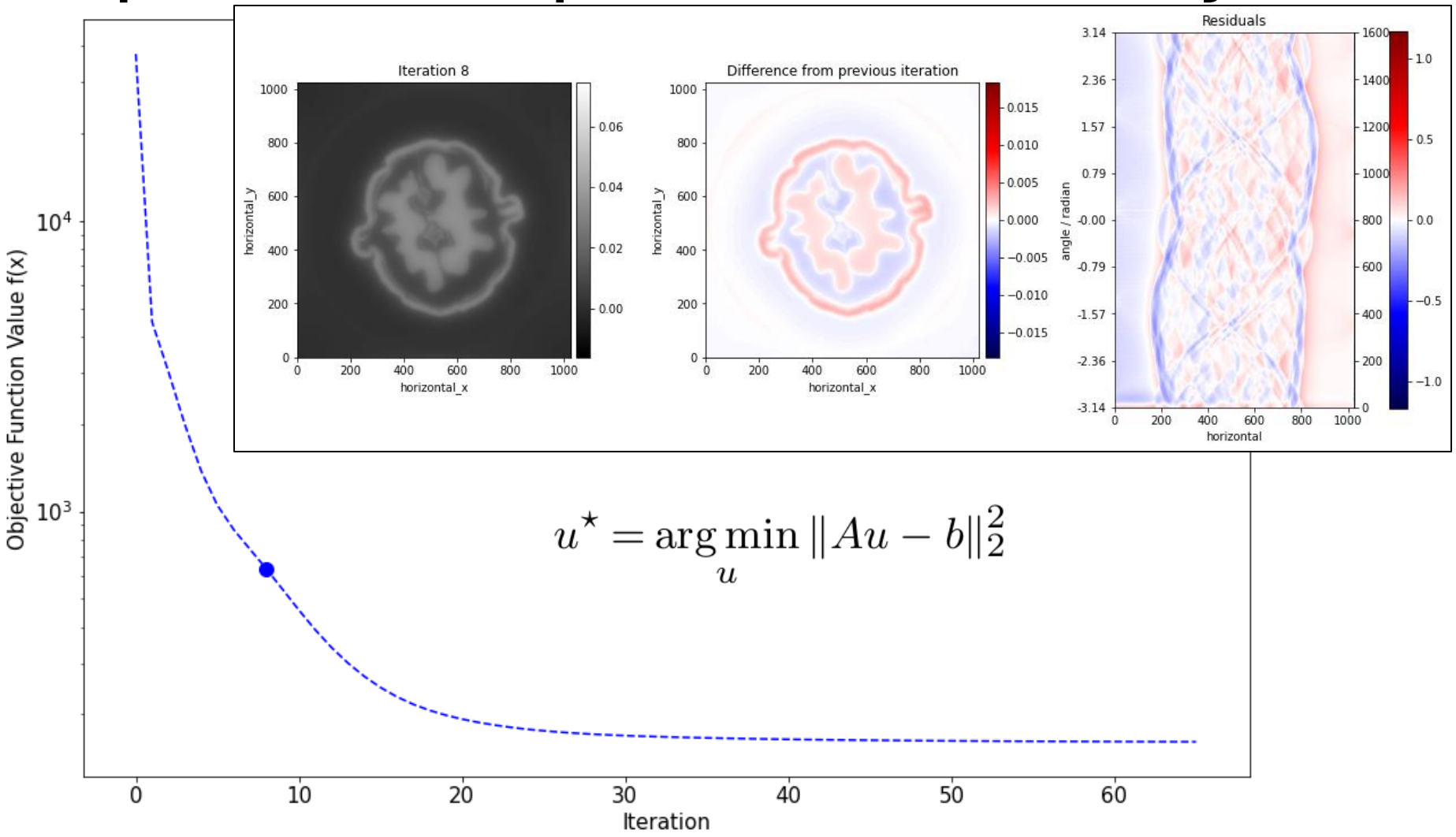
Solve optimisation problem iteratively



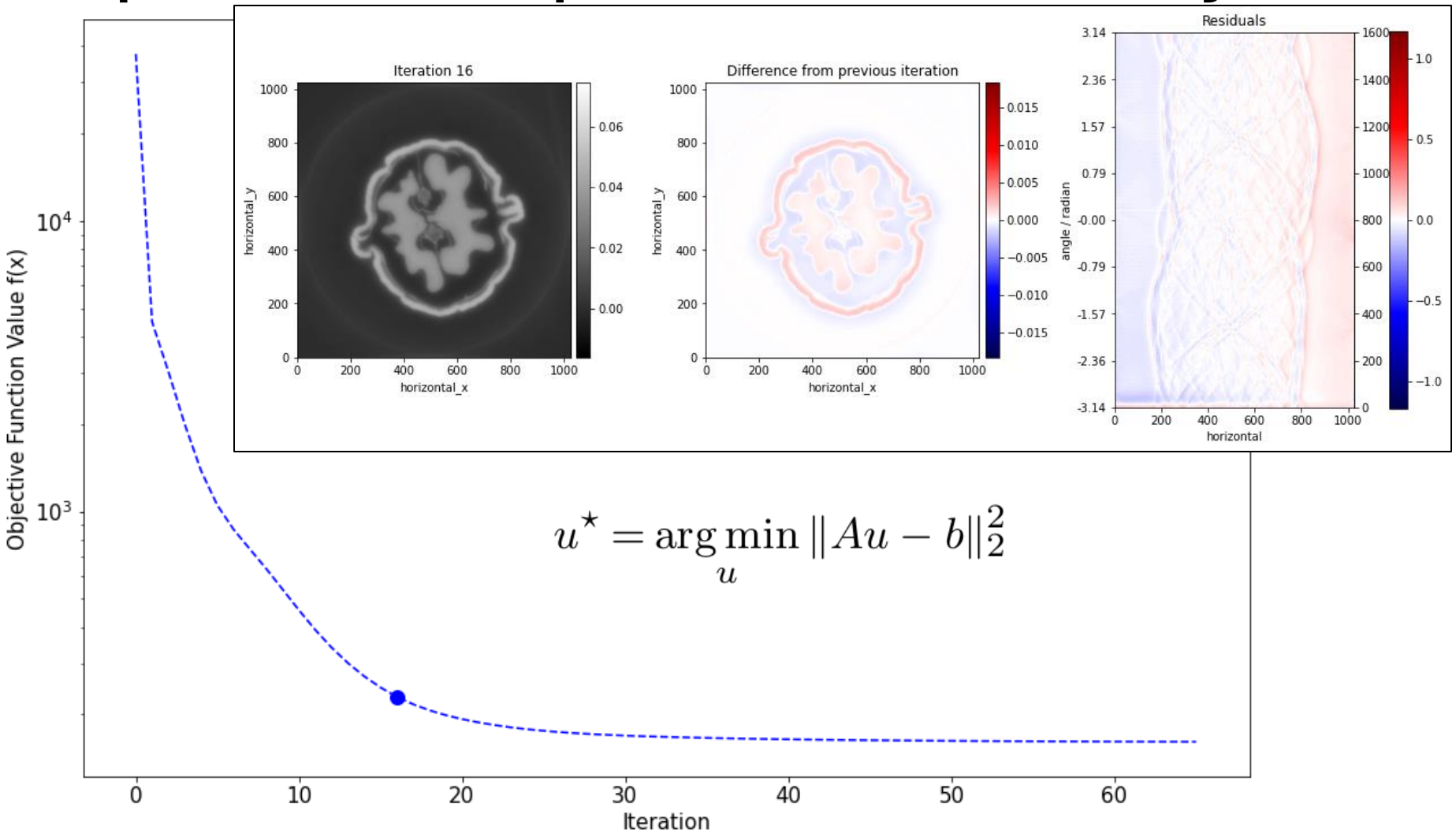
Solve optimisation problem iteratively



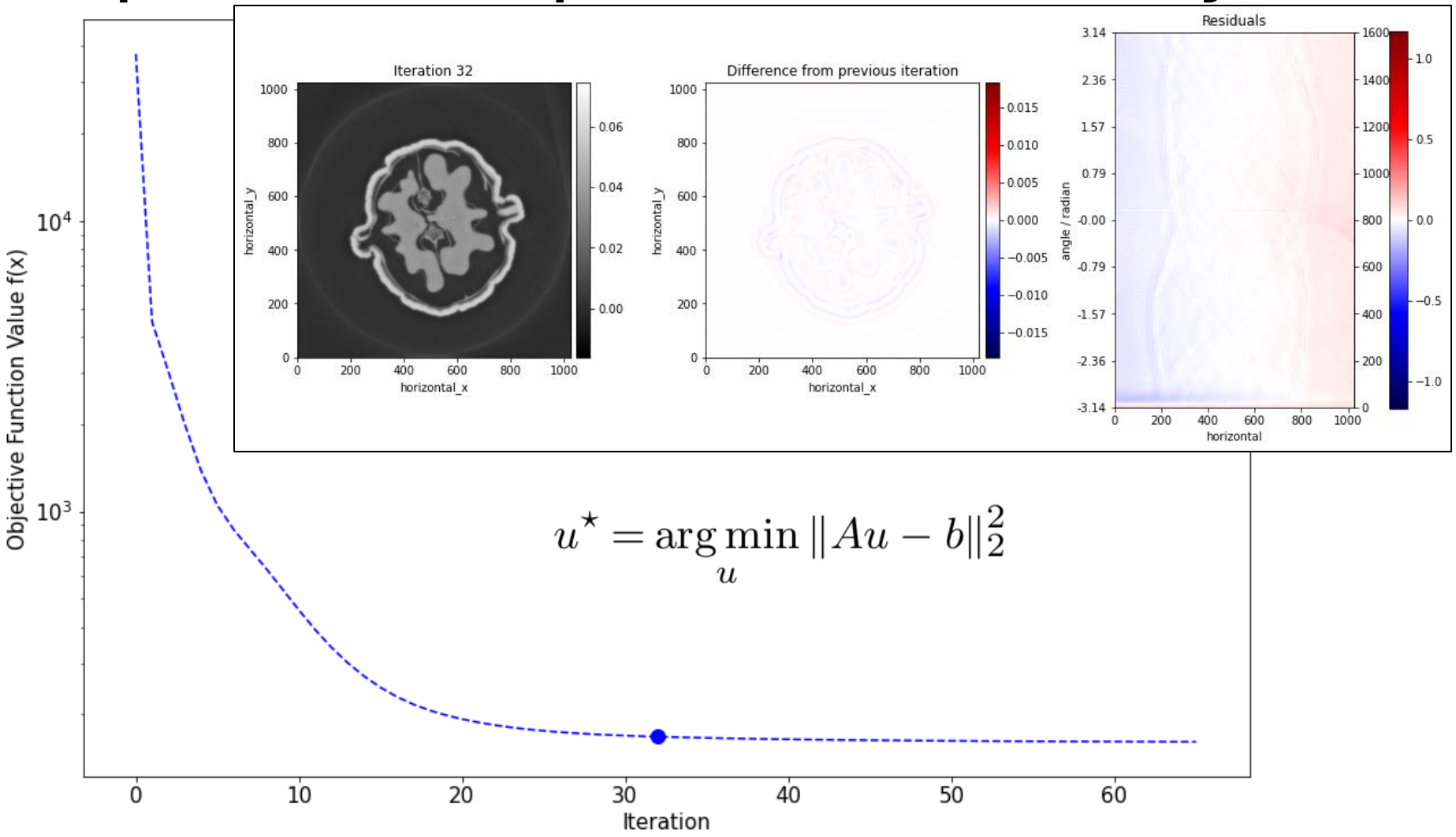
Solve optimisation problem iteratively



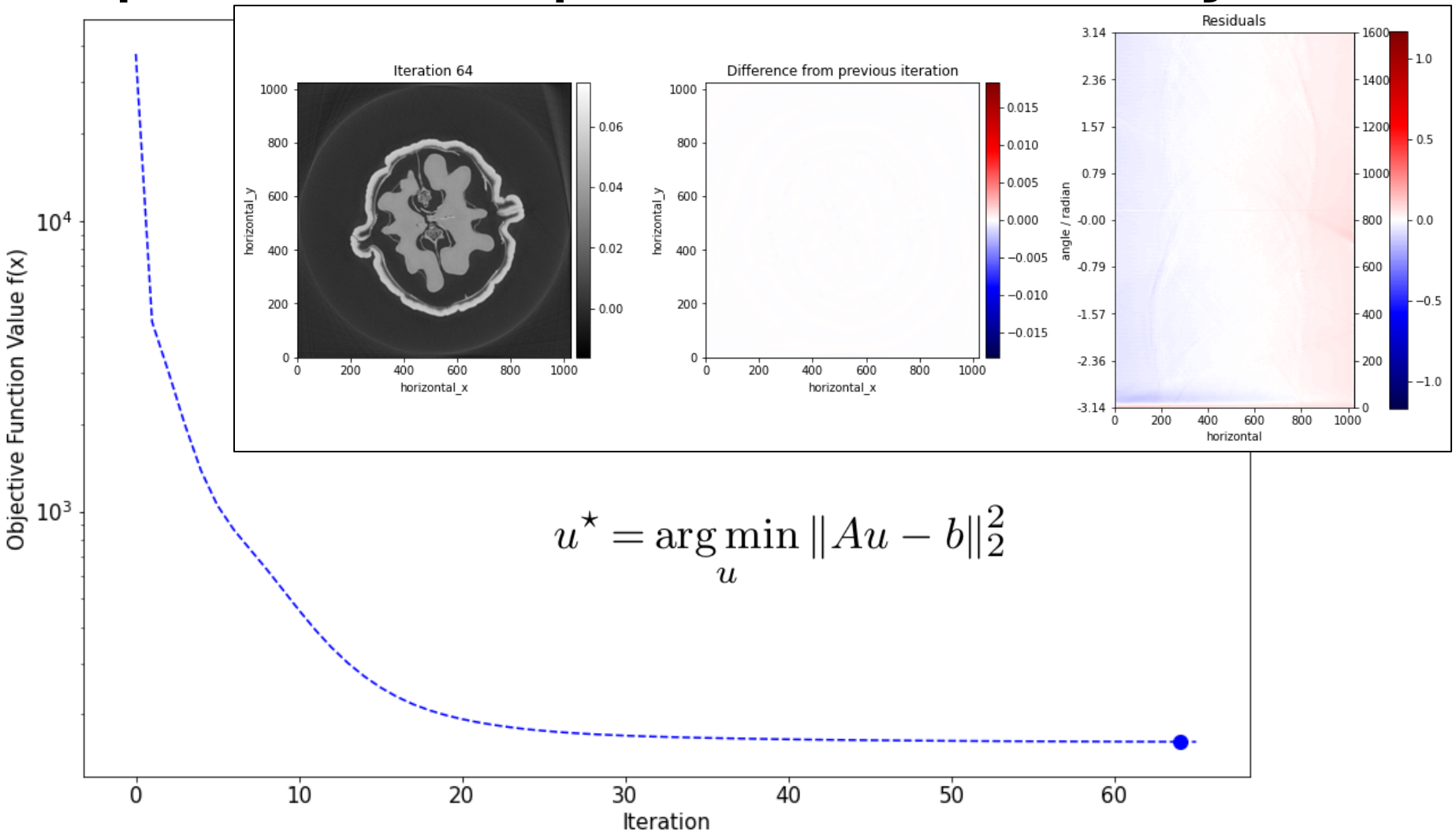
Solve optimisation problem iteratively



Solve optimisation problem iteratively



Solve optimisation problem iteratively



Example with regularisation

$$Au = b$$

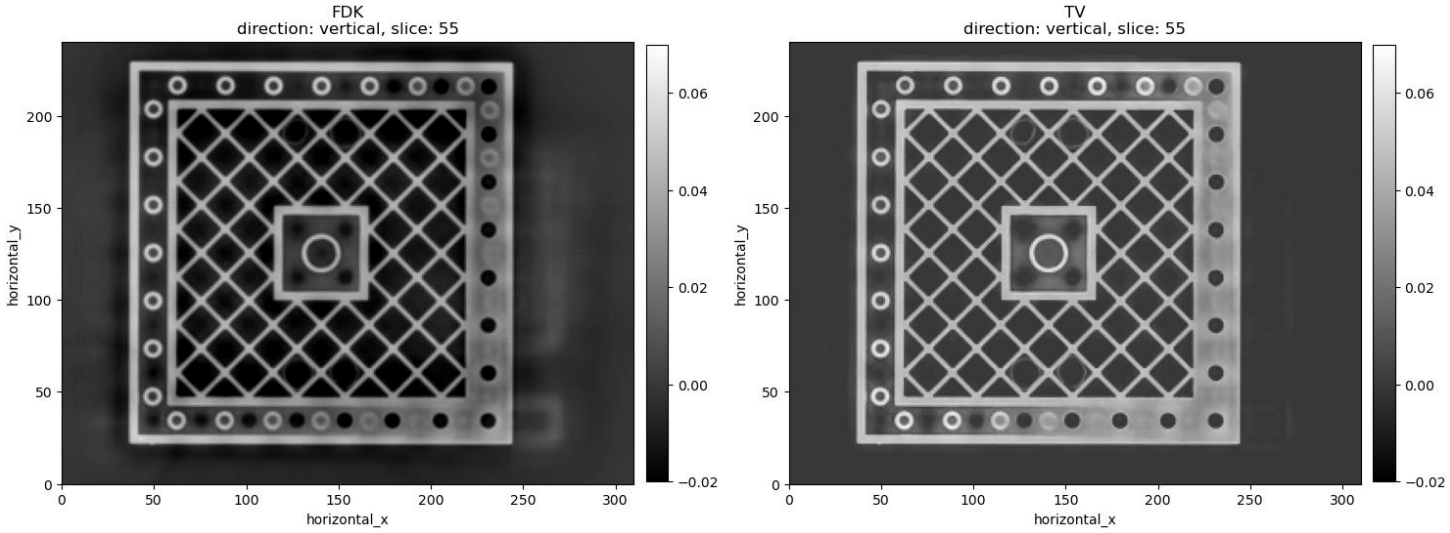
Construct the iterative reconstruction method based on optimisation algorithms and objective functions

$$u^\star = \underset{u}{\operatorname{argmin}} \{\mathcal{D}(Au, b) + \alpha \cdot \mathcal{R}(u)\}$$

```
Projector = ProjectionOperator(ig, ag)
LS = LeastSquares(A=Projector, b=data)
TV = FGP_TV(alpha=0.05, nonnegativity=True, device='gpu’)

fista_TV = FISTA(initial=FDK_reco, f=LS, g=TV,
        max_iteration=1000, update_objective_interval=10)
fista_TV.run(100)
TV_reco = fista_TV.solution

show2D([FDK_recon, TV_recon])
```

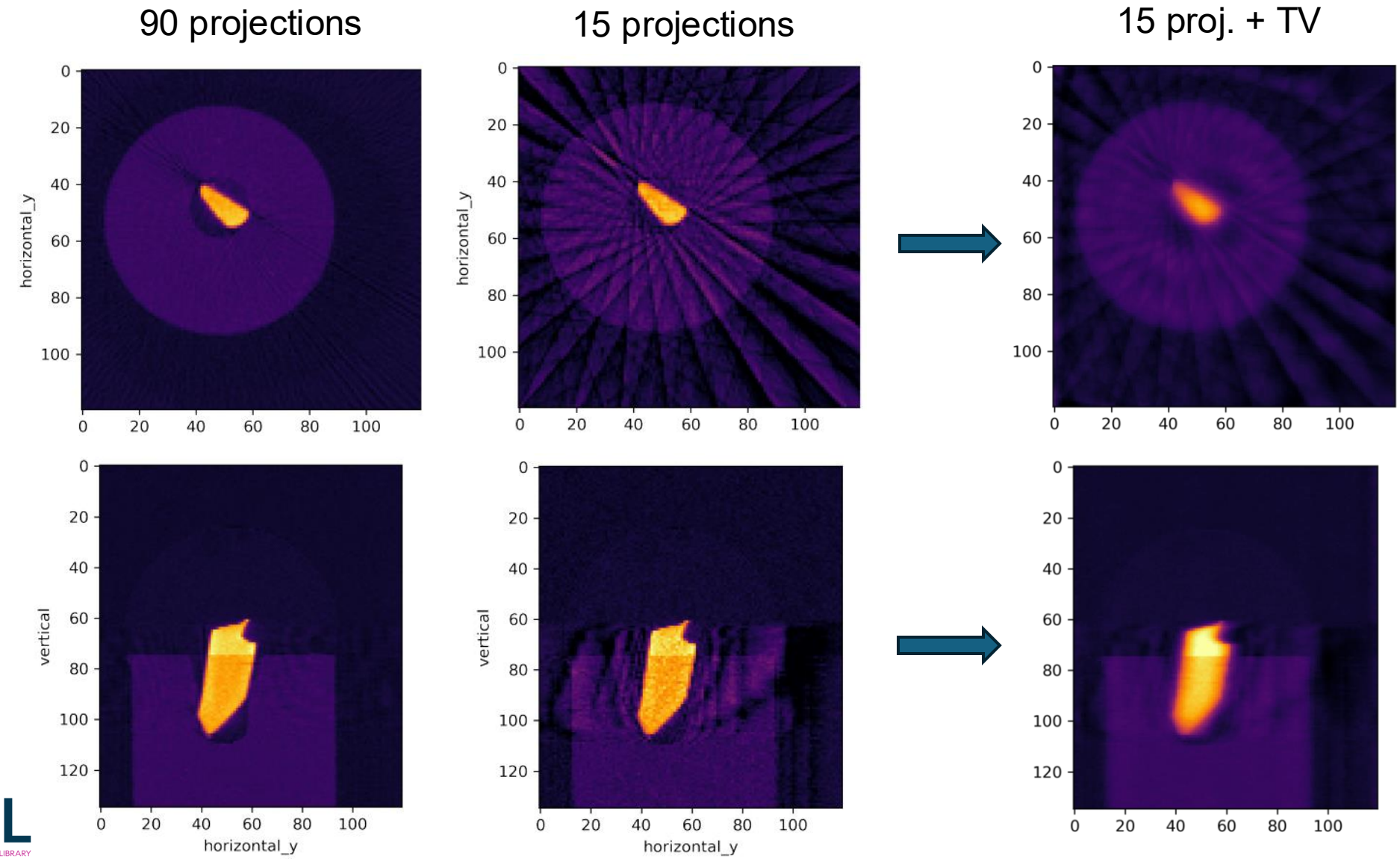


Science and Technology Facilities Council
Scientific Computing

CIL CORE IMAGING LIBRARY

CCPi TOMOGRAPHIC IMAGING

Another example of regularisation



Iterative reconstruction walkthrough

CIL-Demos/blob/main/binder/PyData22_deblurring.ipynb

- See a non-CT inverse problem - deblurring
- See the effect of different regularisation functions

Iterative reconstruction – over to you!

CIL-Demos/blob/main/demos/2_Iterative/01_optimisation_gd_fista.ipynb

- Load a dataset and reconstruct with FBP
- Set-up a least-squares problem to solve using CIL's algorithms, a projection operator and objective function
- Add regularisation to the least-squares problem and compare the results: Tikhonov, Non-negativity, L1-Norm, Total-Variation
- Solve the optimisation problem with the appropriate algorithm: Gradient Descent, FISTA, PDHG

30 minutes to run the notebook, then discussion

- Extension options: https://github.com/TomographicImaging/CIL-Demos/blob/main/demos/3_Multichannel/02_Dynamic_CT.ipynb or https://github.com/TomographicImaging/CIL-Demos/blob/main/demos/4_Deep_Dives/03_htc_2022.ipynb

CIL Optimisation module

name	description	problem type solved
CGLS	conjugate gradient least squares	least squares
SIRT	simultaneous iterative reconstruction technique	weighted least squares
GD	gradient descent	smooth
FISTA	fast iterative shrinkage-thresholding algorithm	smooth + non-smooth
LADMM	linearized alternating direction method of multipliers	non-smooth
PDHG	primal dual hybrid gradient	non-smooth
SPDHG	stochastic primal dual hybrid gradient	non-smooth

MaskOperator	
SymmetrisedGradientOperator	
ZeroOperator	
ProjectionOperator	
ProjectionOperator	

L2NormSquared	squared L^2-norm: $\|x\|_2^2 = \sum_i x_i^2$
LeastSquares	least-squares data fidelity: $\|Ax - b\|_2^2$
MixedL21Norm	mixed $L^{2,1}$-norm: $\|(U_1; U_2)\|_{2,1} = \|(U_1^2 + U_2^2)^{1/2}\|_1$
SmoothMixedL21Norm	smooth $L^{2,1}$-norm: $\|(U_1; U_2)\|_{2,1}^S = \|(U_1^2 + U_2^2 + \beta^2)^{1/2}\|_1$
WeightedL2NormSquared	weighted squared L^2-norm: $\|x\|_w^2 = \sum_i (w_i \cdot x_i^2)$
TotalVariation	$\mathrm{TV}(u) = \|Du\|_{2,1} = \sum_{i,j} \left(\sqrt{(D_y u)^2 + (D_x u)^2} \right)_{i,j}$

Science and Technology Facilities Council
Scientific Computing

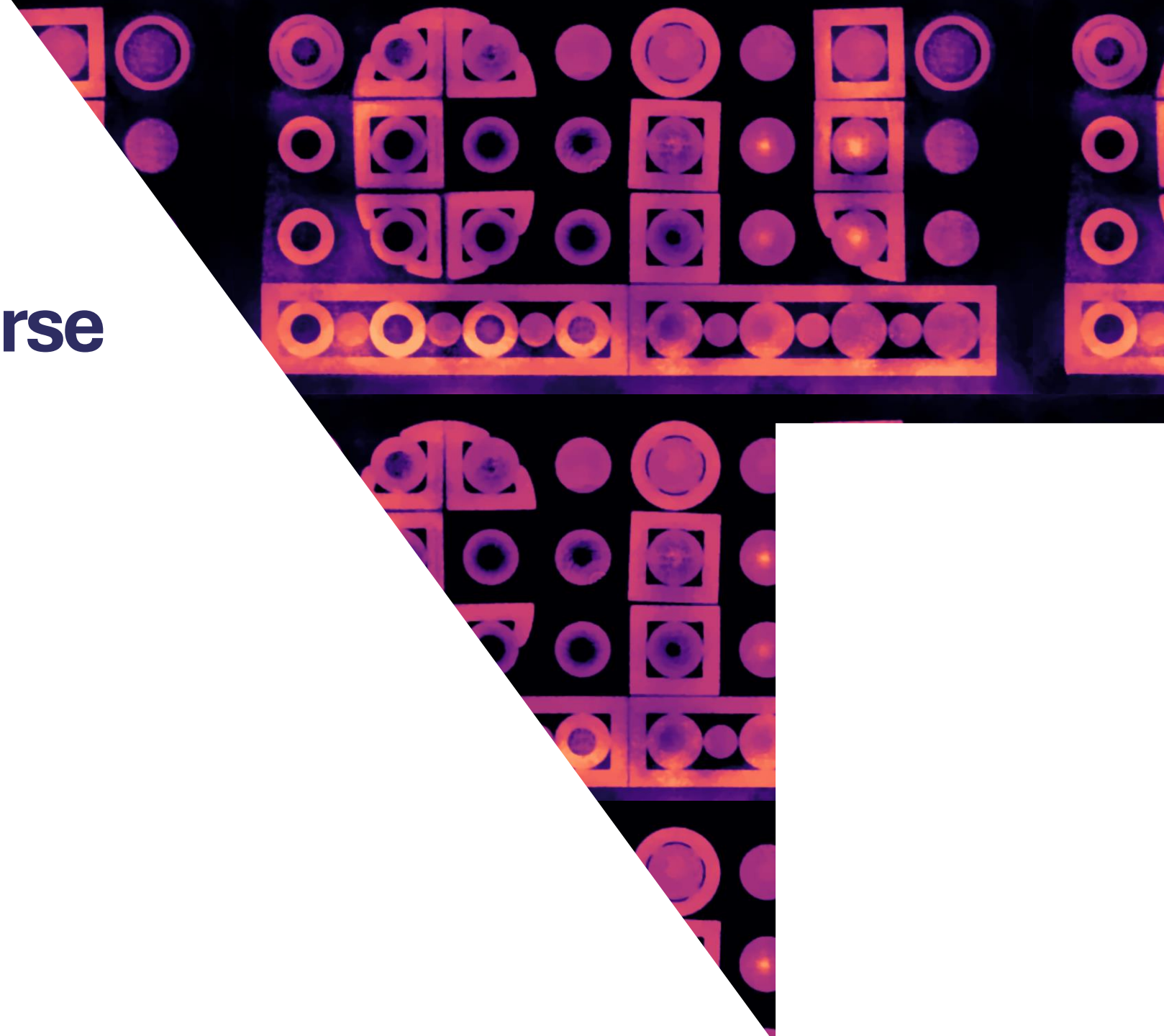
CIL
CORE IMAGING LIBRARY

Optimisation algorithms in CIL

Gradient Descent (GD)	When your objective is convex and differentiable
Conjugate Gradient Least Squares (CGLS)	For minimising a least squares problem e. g. $\min_u \|Au - b\|_2^2$
Simultaneous Iterative Reconstruction Technique (SIRT)	To solve problems of the form $Au = b$ with optional constraints
Iterative Shrinkage-Thresholding Algorithm (ISTA)	To solve problems of the form $\min_u f(u) + g(u)$ where f is convex and differentiable and g is convex with a simple proximal operator
Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)	Like ISTA but accelerated
Primal Dual Hybrid Gradient (PDHG)	To solve problems of the form $\min_u f(Au) + g(u)$ where f is convex and has a "simple" proximal method of its conjugate and g is convex with a "simple" proximal.
Stochastic Primal Dual Hybrid Gradient (SPDHG)	Similar to PDHG but where f can be written as a separable sum
Linearized Alternating Direction Method of Multipliers (LADMM)	To solve problems of the form $\min_u f(u) + g(v)\ f(x)$ subject to $Au + Bv = b$ where both f and g are co
Stochastic algorithms…	Training coming soon…

$$\text{prox}_{\tau g}(u) = \arg\min_v \left\{ \tau g(v) + \frac{1}{2}\|v - u\|_2^2 \right\}$$

Deep learning approaches to inverse problems



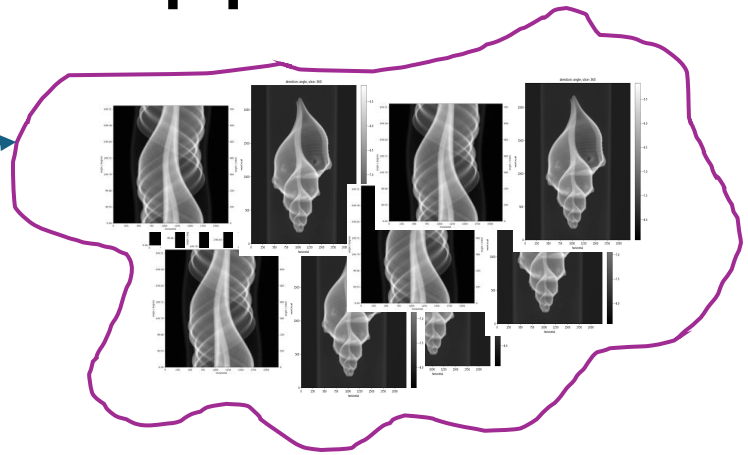
Science and Technology Facilities Council
Scientific Computing

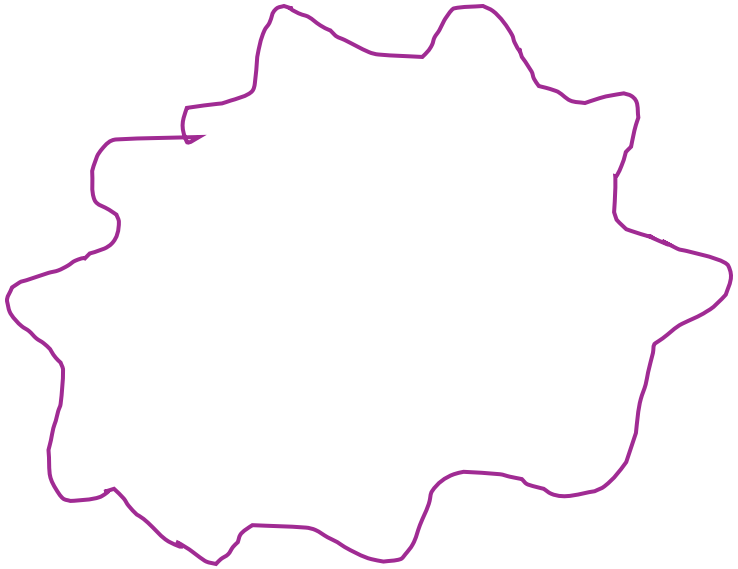
CIL
CORE IMAGING LIBRARY

CCPi
TOMOGRAPHIC IMAGING

Types of approaches

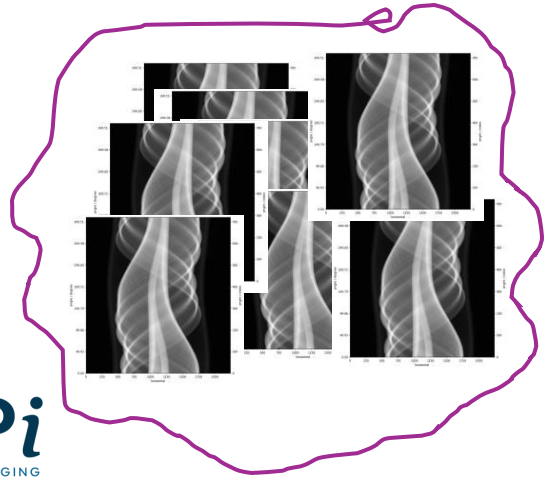
- Supervised



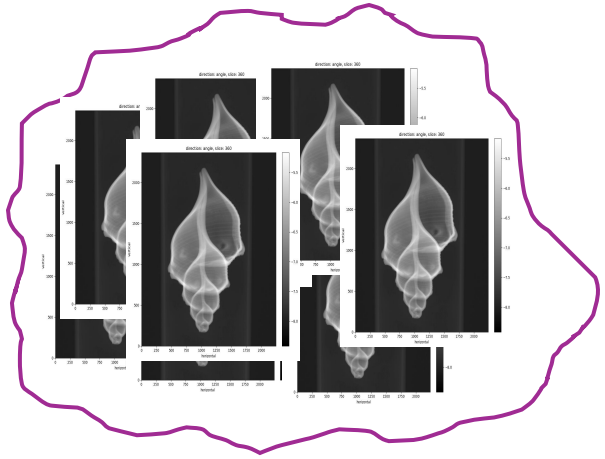


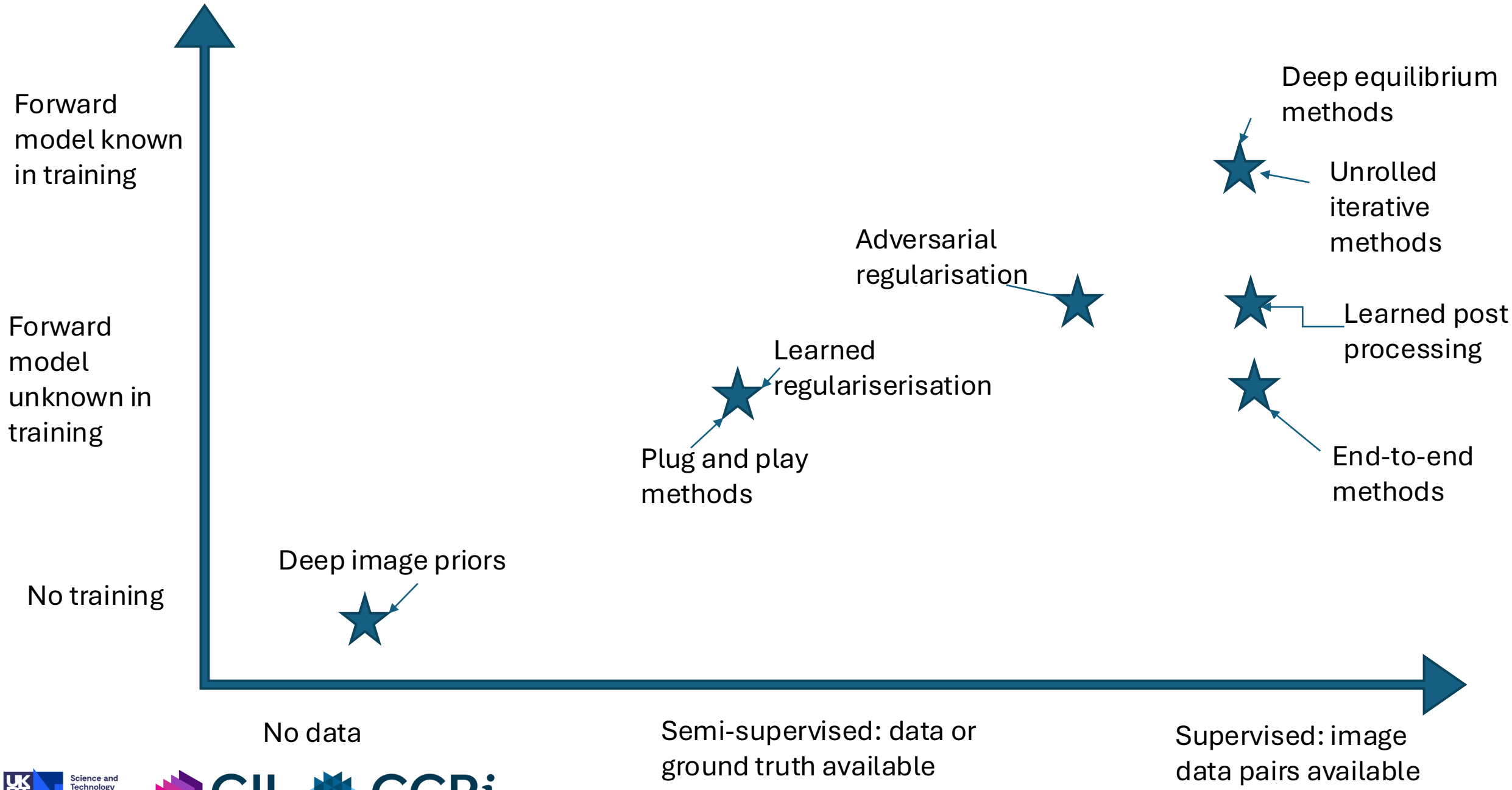
- Unsupervised

- Semi-supervised



or





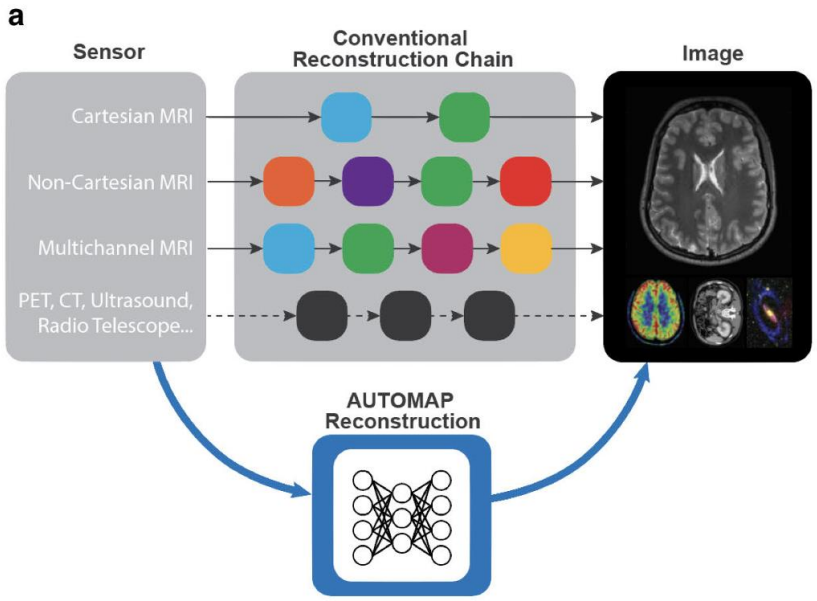
Science and Technology Facilities Council
Scientific Computing

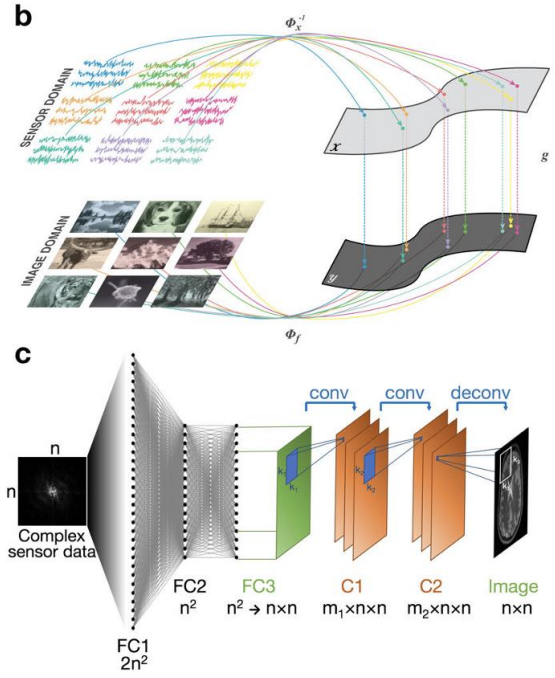
CIL CORE IMAGING LIBRARY

CCPi TOMOGRAPHIC IMAGING

End-to-end methods

Idea: Train a network to go straight from data to reconstruction
e.g. AUTOMAP (2018)





Pros:

- Don't need any understanding on the physics
- Quick to evaluate

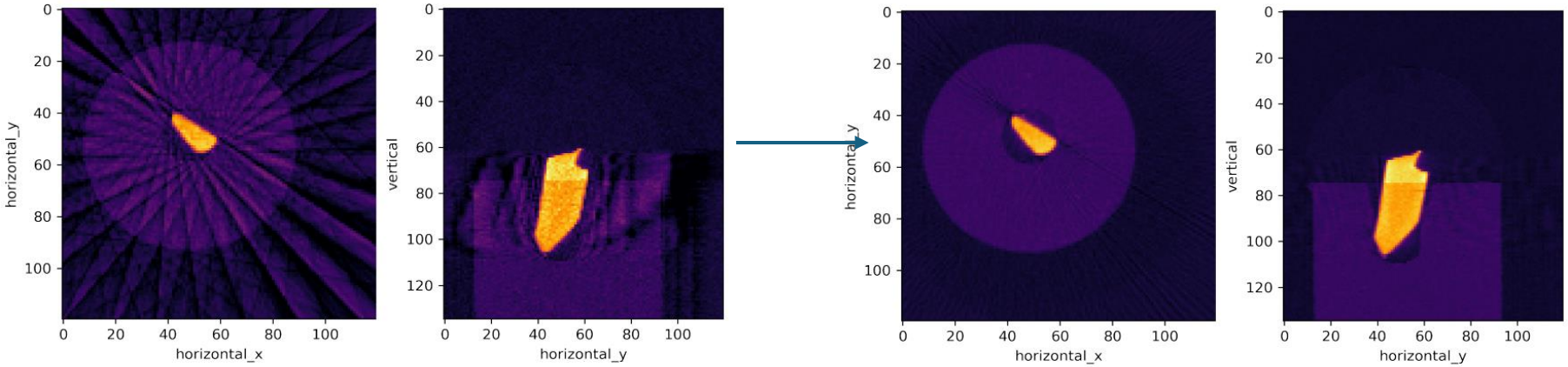
Cons:

- Needs a lot of data to be in any way robust or generalizable
- Networks have to map between very different spaces
- No guarantees the results are even physically reasonable
- Annoys lots of mathematicians and physicists!

Zhu, B., Liu, J., Cauley, S. *et al.* Image reconstruction by domain-transform manifold learning. *Nature* **555**, 487–492 (2018). https://doi.org/10.1038/nature25988

Learned post-processing

Idea: Train a network to go from a rough reconstruction to a good reconstruction



Pros:

- Should do at least as well as the rough reconstruction
- Speed depends on rough reconstruction method but often quick to evaluate

Cons:

- You lose the guarantee that the reconstruction matches the data**
- Depends on the quality of the initial reconstruction
- Requires lots of examples of good data

Ji Zhao, Zhiqiang Chen, Li Zhang, and Xin Jin. “Few-view CT reconstruction method based on deep learning”. In: 2016 IEEE Nuclear Science Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD) (2016), pp. 1–4.

Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. “Deep null space learning for inverse problems: Convergence analysis and rates”. In: Inverse Problems 35 (2 2019), p. 025008.

Deep iterative-unrolled methods

We wish to minimize $D(u) = \|Au - b\|_2^2$ we could use gradient descent

$$u_{k+1} = u_k - \alpha_k \nabla D(u)_k \; for \; k = 0,1,2, \ldots.$$

In iterative unrolled methods we do two things

1. Replace some part of the method with a neural network
2. Fix the final number of iterations

e.g. “Learning to learn by gradient descent”

$$u_{k+1} = u_k - f_{\theta_k}\big(\nabla D(u_k)\big) \; for \; k = 0,1,2, \ldots K$$

Lots more examples for different algorithms!

Pros:

- Quick to evaluate
- Can include physical knowledge e.g. of the forward model
- Could do at least as well as the iterative model

Cons:

- You lose any guarantees
- Needs a lot of data to train

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B. and De Freitas, N., 2016. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, *29*.

Adler, J. and Öktem, O., 2018. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, *37*(6), pp.1322-1332.

Learned regularisation

A couple of examples (in a large field):

- Regularisation by denoising

For a denoiser $W: X \to X$, consider $\underset{u \in X}{\operatorname{argmin}} D(Au, b) + \alpha \|W(u) - u\|_2^2$

- Adversarial regularisation

For a discriminator $W: X \to R_+$ where W is a neural network trained to give large values for "bad" images and small values for "good" images. Consider

$$\underset{u \in X}{\operatorname{argmin}} D(Au, b) + W(u)$$

Pros:

- Can be "explainable"
- Often can be trained using semi-supervised data

Cons:

- Often your networks are not convex so optimization becomes hard
- Iterative optimization can be slow

Romano, Y., Elad, M. and Milanfar, P., 2017. The little engine that could: Regularization by denoising (RED). *SIAM journal on imaging sciences*, *10*(4), pp.1804-1844.
Lunz, S., Öktem, O. and Schönlieb, C.B., 2018. Adversarial regularizers in inverse problems. *Advances in neural information processing systems*, *31*.

Deep Image Prior Methods

Before we wished to minimise

$$u^* \in \underset{u}{\operatorname{argmin}}\, D(Au, b) + \alpha R(u)$$

Consider now

$$\theta^* \in \underset{u}{\operatorname{argmin}}\, D(AG(\theta), b) + \alpha R(u)$$

Where the solution $u^* = G(\theta^*)$ and G is a network with weights θ.

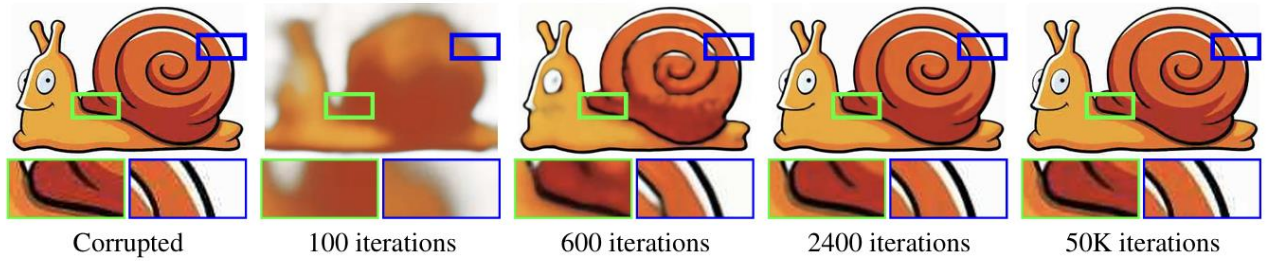


Figure 3: **Blind restoration of a JPEG-compressed image.** (*electronic zoom-in recommended*) Our approach can restore an image with a complex degradation (JPEG compression in this case). As the optimization process progresses, the deep image prior allows to recover most of the signal while getting rid of halos and blockiness (after 2400 iterations) before eventually overfitting to the input (at 50K iterations).

Ulyanov, D., Vedaldi, A. and Lempitsky, V., 2018. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9446-9454).

Pros:

- Requires no training data
- Can benefit from implicit priors in Neural network

Cons:

- Need to train a NN to reconstruct each image
- Often requires early stopping

Plug and play methods

Proximal gradient descent methods optimize objectives such as

$$\underset{u}{\operatorname{argmin}}\, f(u) + g(u)$$

With steps

$$u_{k+1} = \underset{\alpha_k, g}{\operatorname{prox}}(u_k - \alpha_k \nabla f(u_k))$$

where

$$\underset{\alpha_k, g}{\operatorname{prox}}(u) = \underset{x}{\operatorname{argmin}} \frac{1}{2} \|x - u\|_2^2 + \alpha_k g(x)$$

in plug and play methods we replace this proximal with a learned denoiser.

There are many variations with different algorithms and denoisers.

Pros:

- Removes the need to explicitly define a regulariser
- In *some* cases can give convergence guarantees
- Can be trained with semi-supervised data

Cons:

- Iterative optimization methods can be computationally exspensive

Venkatakrishnan, S.V., Bouman, C.A. and Wohlberg, B., 2013, December. Plug-and-play priors for model based reconstruction. In *2013 IEEE global conference on signal and information processing* (pp. 945-948). IEEE.

More areas to explore

We haven't mentioned:

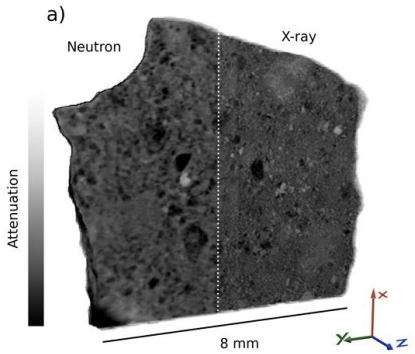
- Lots of extensions and adaptations of these methods
- Deep equilibrium methods
- Noise2noise, Noise2inverse…
- Bilevel learning approaches
- ….

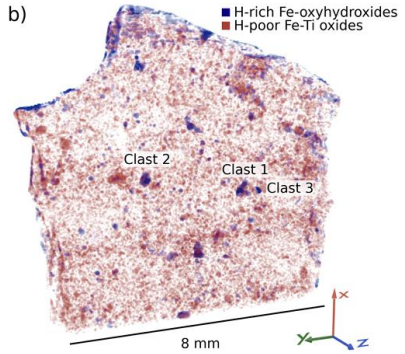
Some places to look further (very incomplete list):

- Deep inverse https://deepinv.github.io/deepinv/
- Learned Iterative Optimisation Networks for CT (LION): https://github.com/CambridgeCIA/LION
- S. Mukherjee, A. Hauptmann, O. Öktem, M. Pereyra and C. -B. Schönlieb, "Learned Reconstruction Methods With Convergence Guarantees: A survey of concepts and applications," in *IEEE Signal Processing Magazine*, vol. 40, no. 1, pp. 164-182, Jan. 2023, doi: 10.1109/MSP.2022.3207451.
- Many more…

Challenges of Deep Learning and Inverse Problem Approaches

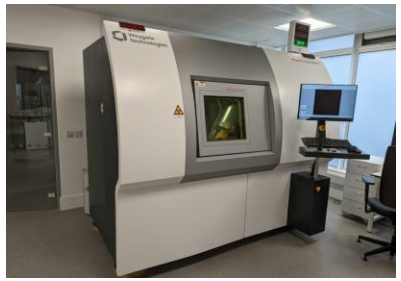
- Not enough data





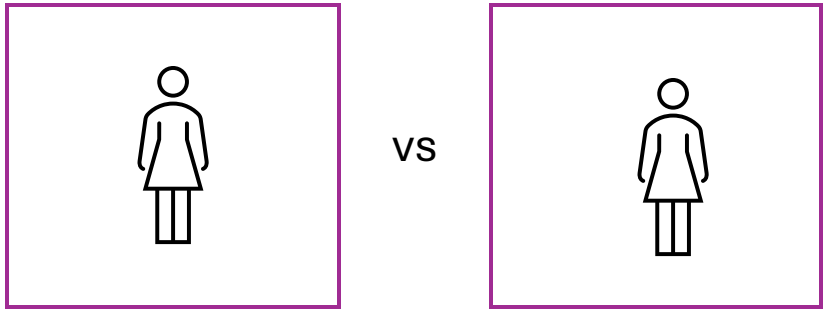
Naver, E.B., et al. 2026. Direct detection of hydrogen reveals a new macroscopic crustal water reservoir on early Mars. *arXiv preprint arXiv:2601.08390*.

- Too much data (data sizes)



Waygate – NXCT Warwick – 4kx4k detector

- The needs of scientists/users
  - How do you define what a good image is?



Science and Technology Facilities Council
Scientific Computing

CIL CORE IMAGING LIBRARY

CCPi TOMOGRAPHIC IMAGING

A CIL example

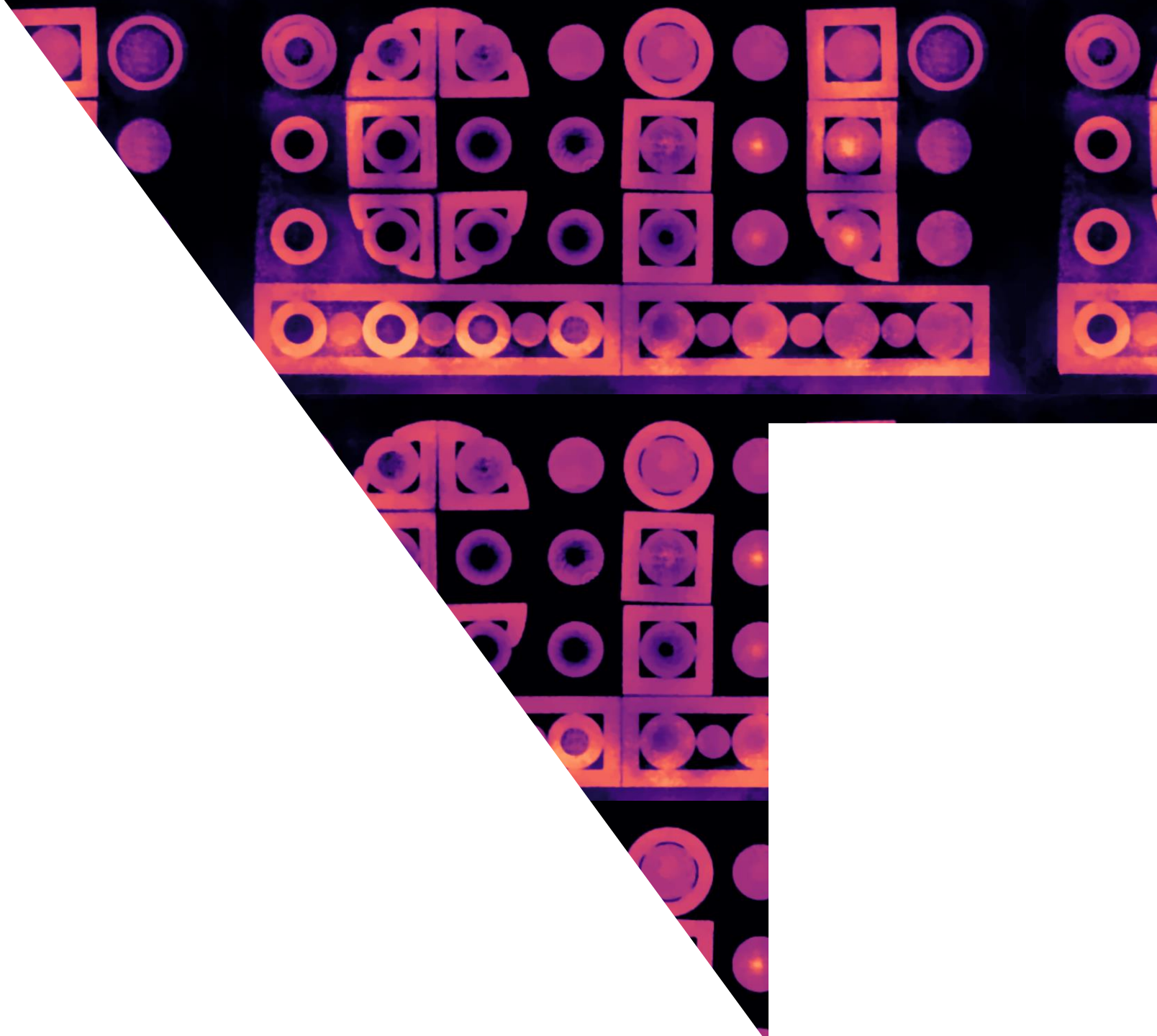
CIL-User-Showcase/016_cil_torch_fista_pnp/fista_with_denoiser.ipynb

- How to create a CIL function to wrap a pytorch function or operator
- Examples of image reconstruction using a pre-trained denoiser in CIL
- Timings of the data copies between pytorch and CIL

Extensions

- Some ideas in the notebook
- Look at https://github.com/TomographicImaging/CIL-Demos/blob/main/demos/4_Deep_Dives/03_htc_2022.ipynb - can you do better with a trained approach?
- Investigate other deep learning and inverse problem approaches
- Look at other notebooks and showcases

Find out more



Science and Technology Facilities Council
Scientific Computing

CIL
CORE IMAGING LIBRARY

CCPi
TOMOGRAPHIC IMAGING

Using CIL on your own

CIL GitHub

README Apache-2.0 license

CIL - Core Imaging Library

build passing | Anaconda.org 25.0.0 | Last updated 13 Aug 2025 | Platforms osx-64,linux-64,win-64 | downloads 32k total

launch binder

The Core Imaging Library (CIL) is an open-source Python framework
imaging with particular emphasis on reconstruction of challenging da
filtered backprojection reconstruction tends to be insufficient for hig
non-standard or multichannel data arising for example in dynamic, sp
tomography. CIL provides an extensive modular optimisation framew
reconstruction methods including sparsity and total variation regular
for loading, preprocessing and visualising tomographic data.

Documentation

The documentation for CIL can be accessed here.

Installation of CIL

Conda

Binary installation of CIL can be achieved with `conda`.

We recommend using either `miniconda` or `miniforge`, which are b
for `conda`. We also recommend a `conda` version of at least `23.10`

CIL

Welcome to CIL's documentation!

The aim of this package is to enable rapid prototyping of optimisation-based reconstruction problems, i.e. defining and solving different optimization problems to enforce different properties on the reconstructed image, while being powerful enough to be employed on real scale problems.

Firstly, it provides a framework to handle acquisition and reconstruction data and metadata; it also provides a basic input/output package to read data from different sources, e.g. Nikon X-Radia, NeXus.

Secondly, it provides an object-oriented framework for defining mathematical operators and functions as well a collection of useful example operators and functions. Both smooth and non-smooth functions can be used.

Further, it provides a number of high-level generic implementations of optimisation algorithms to solve generically formulated optimisation problems constructed from operator and function objects.

Demos and Examples

A number of demos can be found in the CIL-Demos repository.

For detailed information refer to our articles and the repositories with the code to reproduce the article's results.

Science and Technology Facilities Council
Scientific Computing

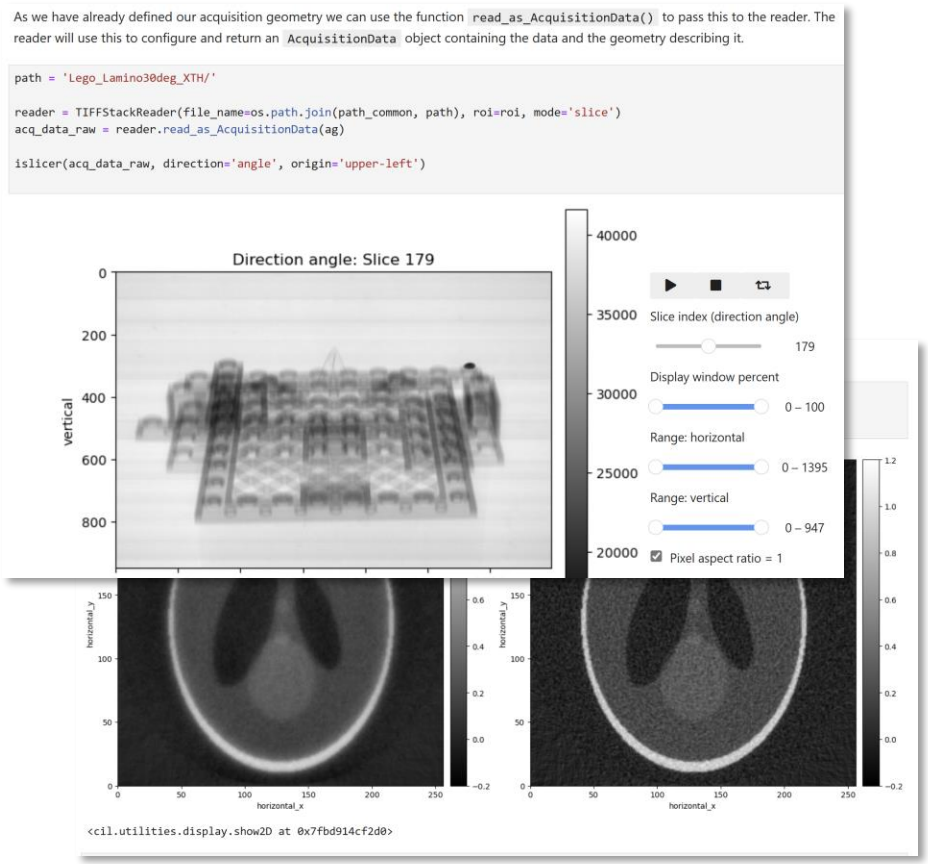
CIL CORE IMAGING LIBRARY

CCPi TOMOGRAPHIC IMAGING

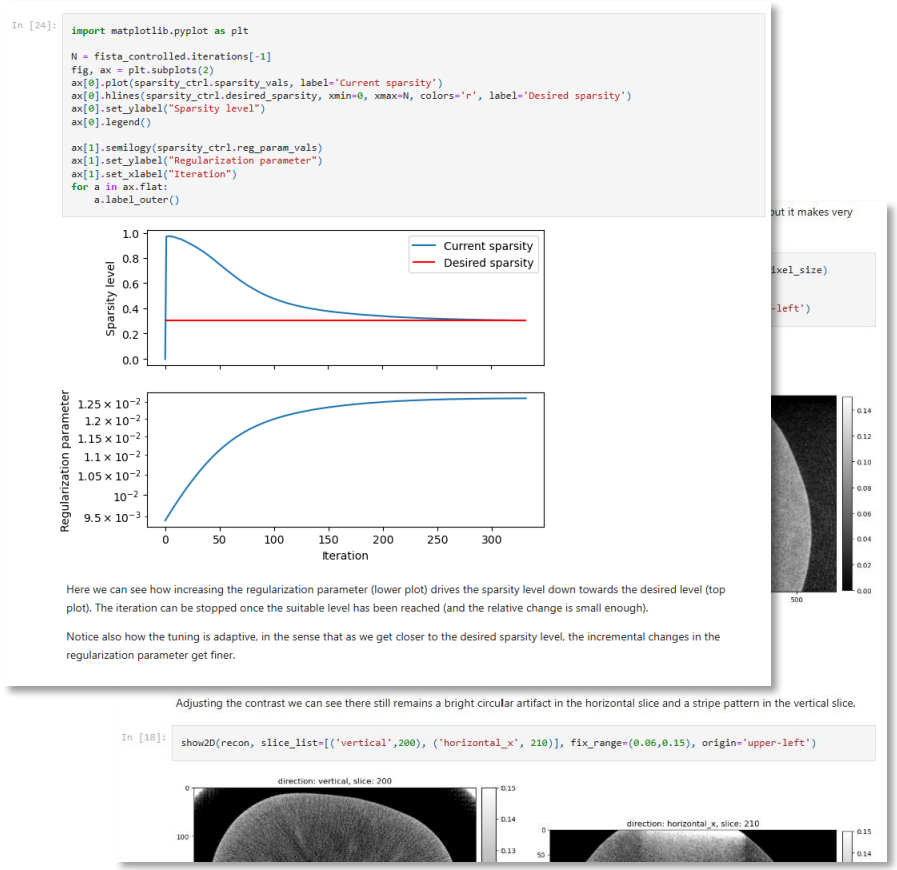
Demos and examples

CIL-Demos

21+ demo notebooks



14 contributed notebooks showcasing interesting uses of CIL



User showcase

Science and Technology Facilities Council
Scientific Computing

CIL CORE IMAGING LIBRARY

CCPi TOMOGRAPHIC IMAGING

CIL Publications

Jørgensen et al.: *Core Imaging Library - Part I: a versatile Python framework for tomographic imaging* Phil. Trans. R. Soc. A. **379** 20200192 (2021) DOI: 10.1098/rsta.2020.0192

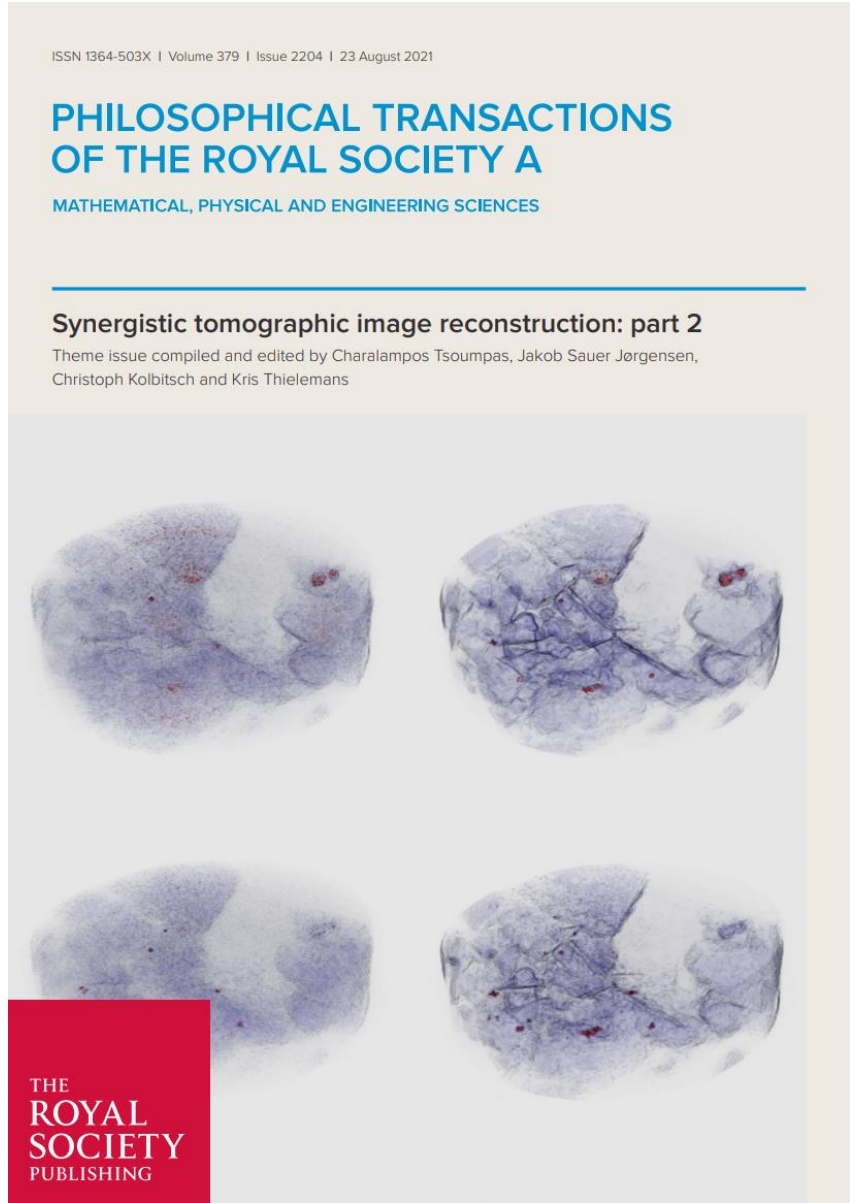
Papoutsellis et al.: *Core Imaging Library - Part II: multichannel reconstruction for dynamic and spectral tomography* Phil. Trans. R. Soc. A.**379**20200193 (2021) DOI: 10.1098/rsta.2020.0193

Jørgensen et al.: *A directional regularization method for the limited-angle Helsinki Tomography Challenge using the Core Imaging Library (CIL),* Applied Mathematics for Modern Challenges, Volume **1**, Issue 2: 143-169. (2023) 10.3934/ammc.2023011

Ametova et al.: *Crystalline phase discriminating neutron tomography using advanced reconstruction methods*, J. Phys. D: Appl. Phys. **54** 325502 (2021) DOI 10.1088/1361-6463/ac02f9

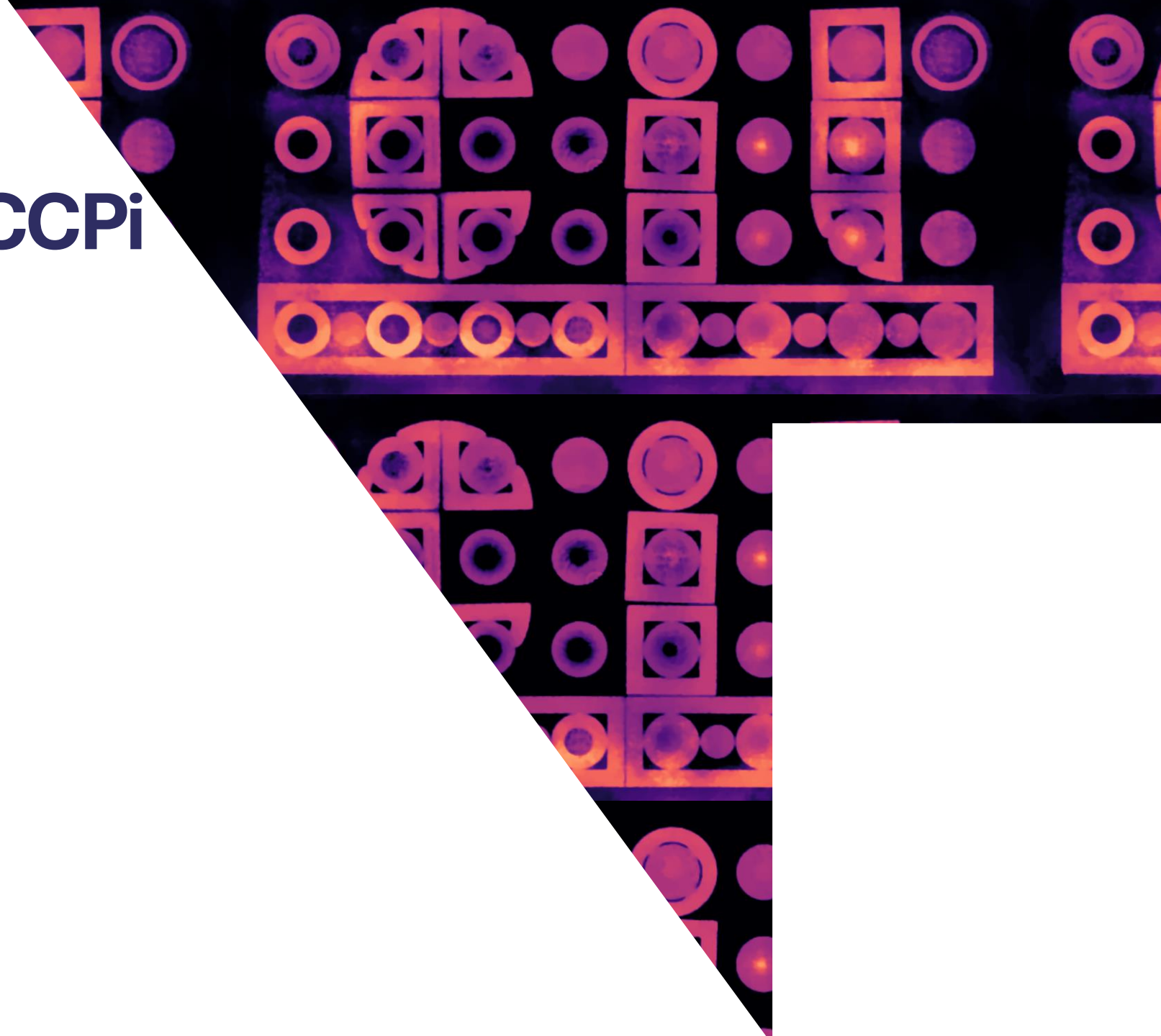
Warr R. et al.: *Enhanced hyperspectral tomography for bioimaging by spatiospectral reconstruction* Sci Rep **11,** 20818 (2021) DOI: 10.1038/s41598-021-00146-4

Brown R. et al.: *Motion estimation and correction for simultaneous PET/MR using SIRF and CIL* Phil. Trans. R. Soc. A **379** 20200208 (2021) DOI: 10.1098/rsta.2020.0208



tomography@stfc.ac.uk

The CIL and Wider CCPi Community



Science and Technology Facilities Council
Scientific Computing

CIL
CORE IMAGING LIBRARY

CCPi
TOMOGRAPHIC IMAGING

CIL User Community

User support through GitHub

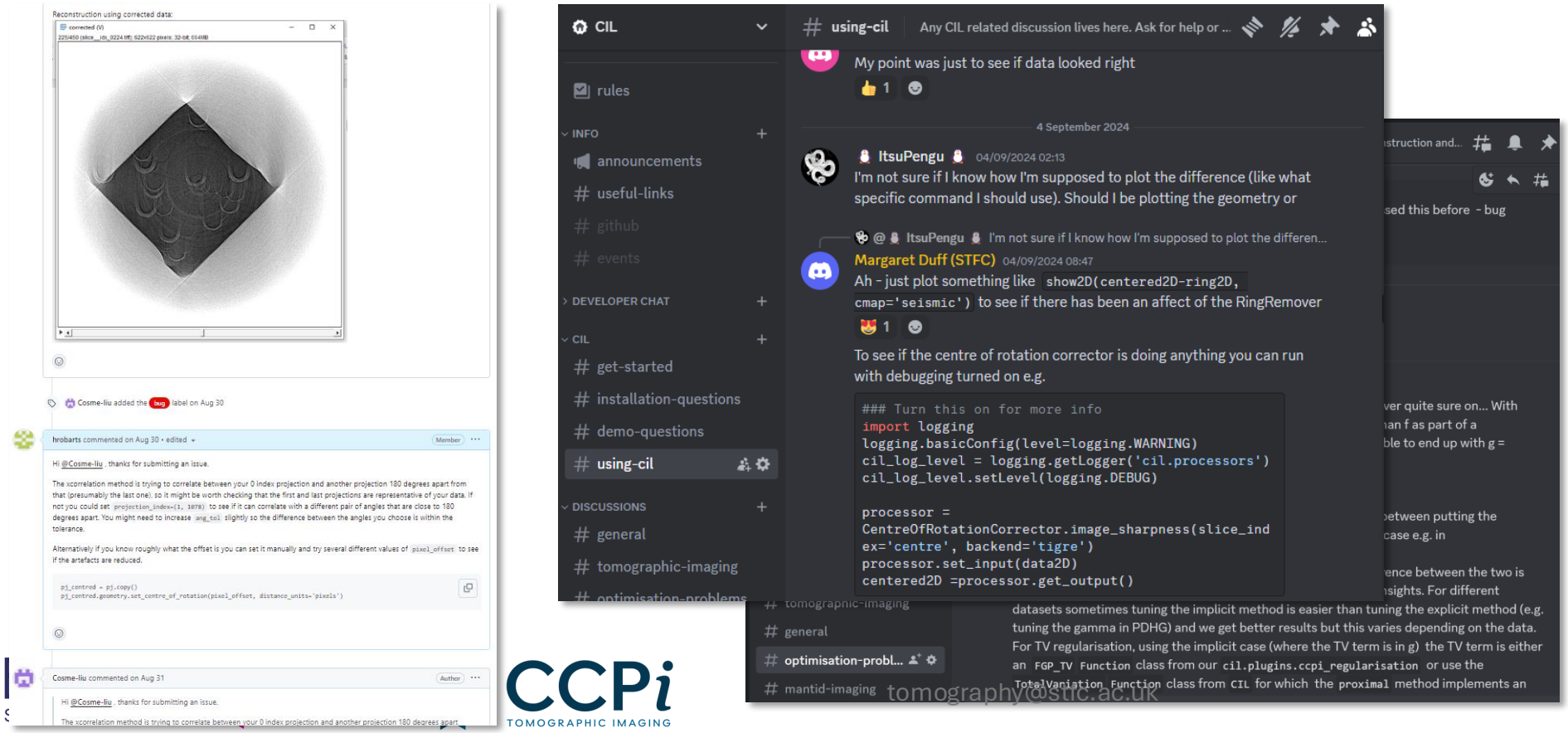
And CIL discord with 200+ users

General and tailored support

Fortnightly User Drop-ins

CIL Discord

CIL GitHub



CCPi TOMOGRAPHIC IMAGING

CIL User Community Events

Travel grants

In person and online training

Annual User Meeting

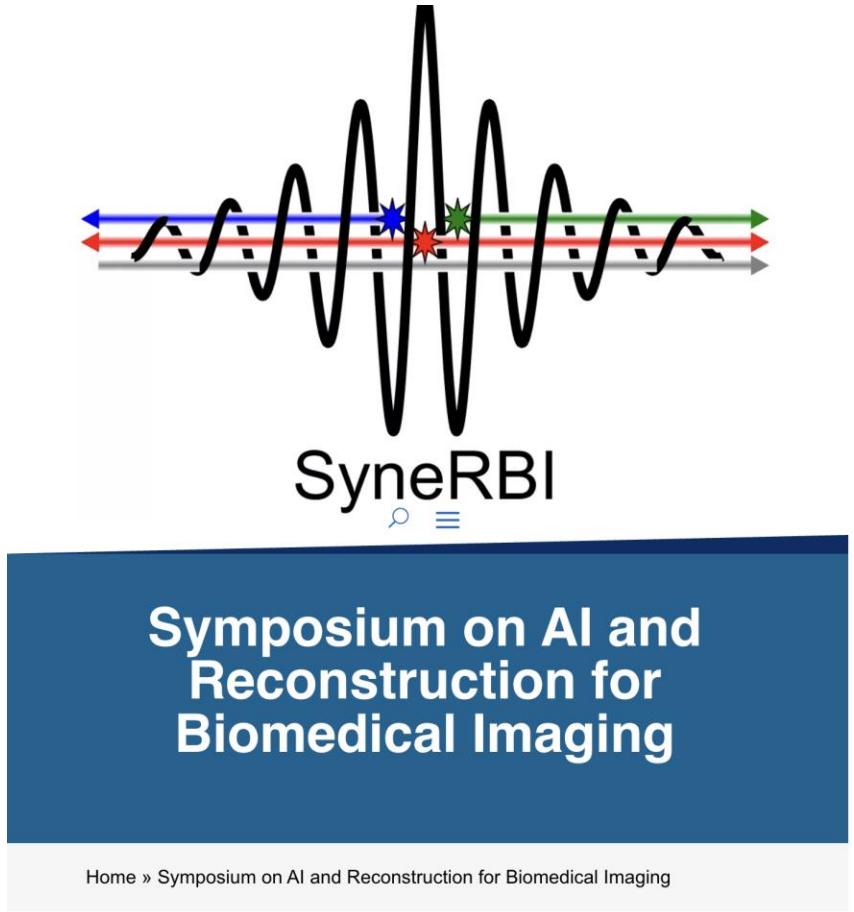
Data and software hackathons

Fortnightly Show and Tell Meetings

For more details…
https://ccpi.ac.uk

Symposium on AI and Reconstruction for Biomedical Imaging

- Online registration still open
- March 9-10th 2026
- https://www.ccpsynerbi.ac.uk/events/airbi/



Science and Technology Facilities Council
Scientific Computing
CIL CORE IMAGING LIBRARY
CCPi TOMOGRAPHIC IMAGING

Wrap up

- CIL is an Open Source mostly Python library for all your tomographic needs:
  - I/O
  - pre-processing
  - Reconstruction
  - Visualisation
- Developer Support, user driven, long term funding
- Join the community Discord
- https://www.ccpi.ac.uk/CIL