

CIL - A Versatile Python Framework for Tomographic Imaging

Evangelos Papoutsellis, Finden

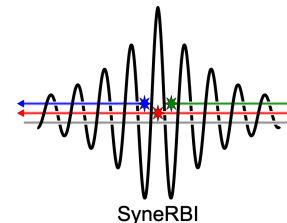
Joint work with

UKRI STFC: *Edoardo Pasca, Gemma Fardell, Laura Murgatroyd, Frank Vidal, Casper da Costa-Luis, Margaret Duff, Hannah Robarts, Danica Sugic*

DTU: *Jakob Sauer Jørgensen*

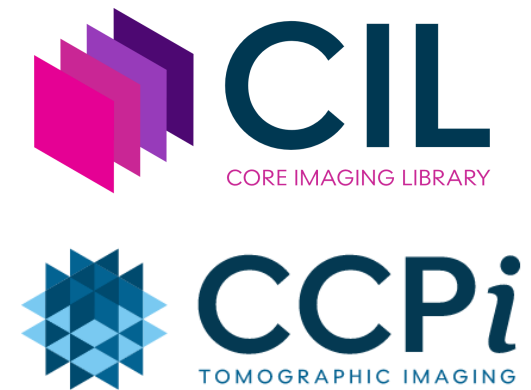
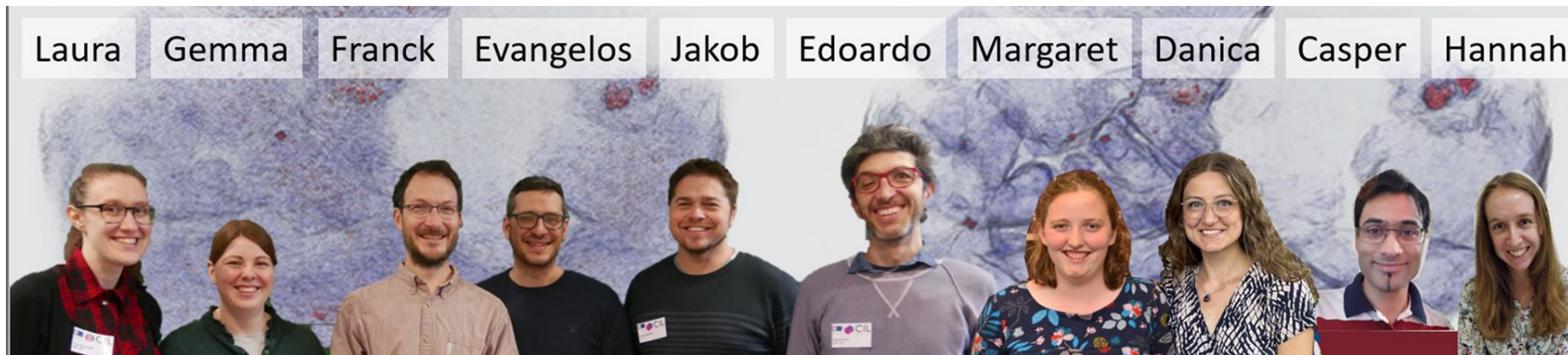


Finden



What is Core Imaging Library (CIL)?

- CIL is an open-source Python library for solving **Imaging Inverse Problems**.
- Emphasis on tomography applications with **challenging data sets**: low-count, non-standard geometries, incomplete, hyper spectral, limited angle.
- "**Near-math**" specification and solution of **optimisation problems**
- **Simple** to get started – **powerful** enough for large, real applications
- Actively developed on GitHub: <https://github.com/TomographicImaging/CIL> (Apache v2)
- Funded by the **Collaborative Computational Project in Tomographic Imaging (CCPi)**



Who is CIL for?

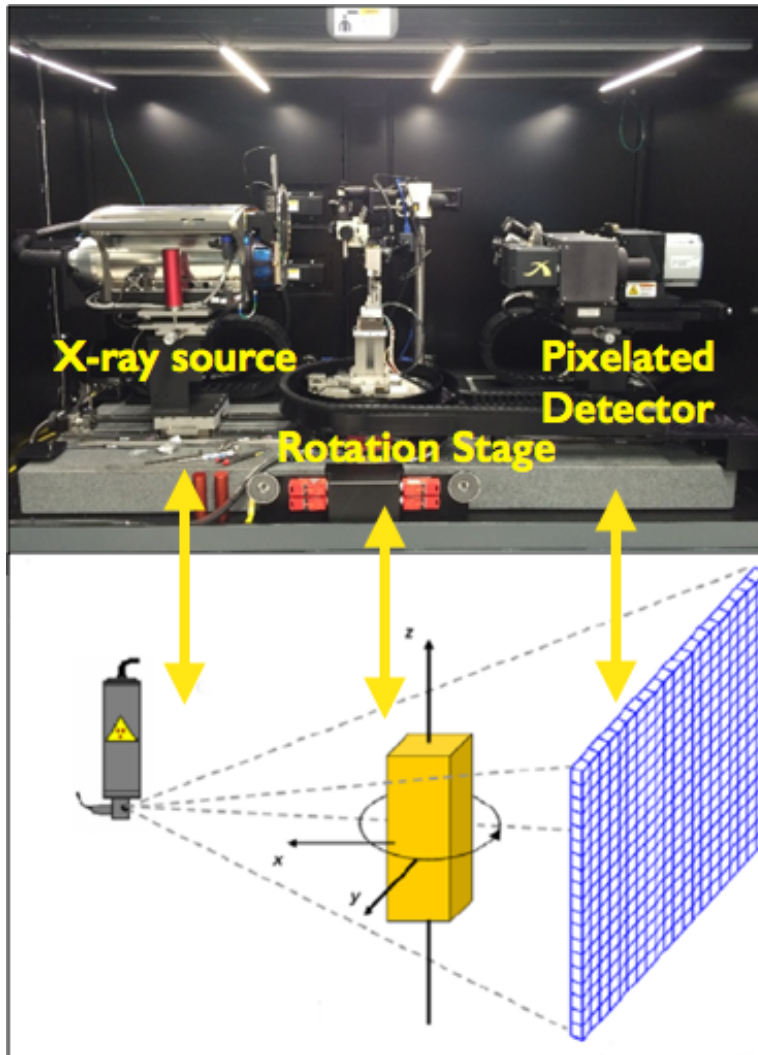
CT Experimentalists and Beamline Scientists

- Apply advanced algorithms for poor data quality or novel imaging modalities
- Optimised standard algorithms for large data
- Batch processing and custom-built open-source data processing

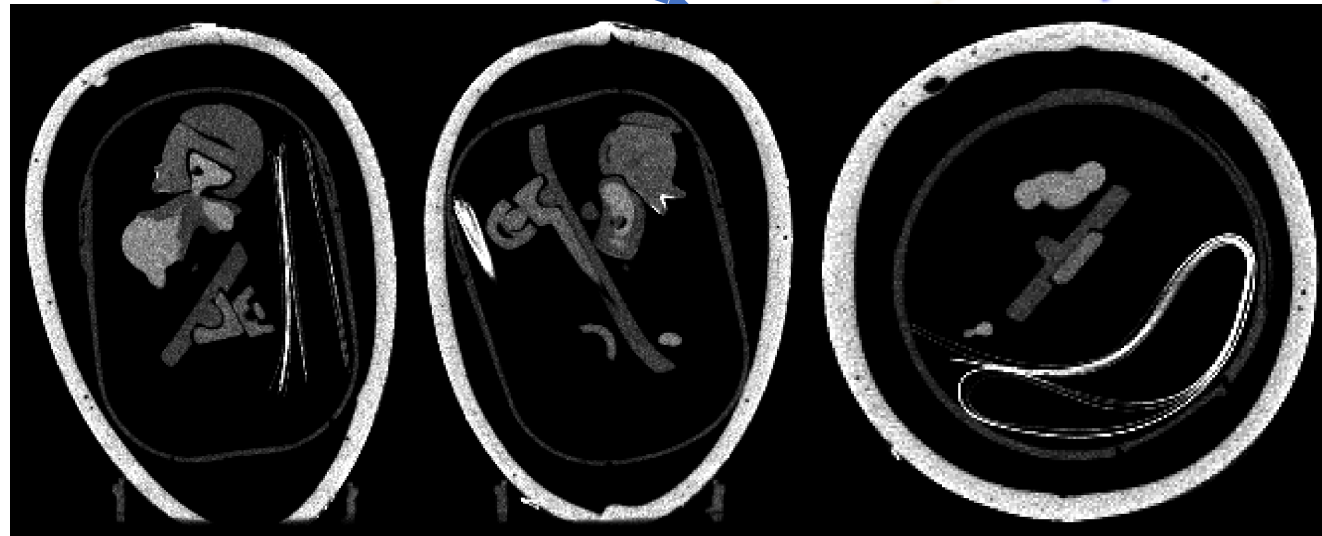
Mathematical imaging researchers

- Prototype new reconstruction algorithms
- Make YOUR METHOD available to community and deploy it to imaging facilities
- Try on real data and assess against existing methods

CIL for X-Ray Computed Tomography (CT)



```
data = TXRMDataReader(filename).read()  
data = TransmissionAbsorptionConverter()(data)  
data = CentreOfRotationCorrector.image_sharpness()(data)  
recon = FBP(data).run()  
show2D(recon)
```



Optimisation in CIL

CIL offers a range of optimisation algorithms, e.g., GD, FISTA, ADMM and (S)PDHG:

$$\operatorname{argmin}_u \quad \alpha_1 \|Au - b\|^2 + \alpha_2 \|Du\|_{2,1} + \alpha_3 \|D_x u\|_1 + \mathbb{I}_{\{0 < u < vm\}}(u)$$

CIL team entry finished 3rd



***beaten by two teams using machine learning
with large amounts of synthetically generated training data***

In PDHG form

$$\operatorname{argmin}_u \quad f(Ku) + g(u), \quad \text{where } f(Ku) = \sum_{i=1}^n f_i(K_i u)$$
$$f = \begin{bmatrix} \alpha_1 \|\cdot - b\|_2^2 \\ \alpha_2 \|\cdot\|_{2,1} \\ \alpha_3 \|\cdot\|_1 \end{bmatrix} \quad K = \begin{bmatrix} A \\ D \\ D_x \end{bmatrix} \quad \mathbb{I}_{\{0 < u < vm\}}(u)$$

Optimisation in CIL

$$f = \begin{bmatrix} \alpha_1 \|\cdot - b\|_2^2 \\ \alpha_2 \|\cdot\|_{2,1} \\ \alpha_3 \|\cdot\|_1 \end{bmatrix}$$

```
F = BlockFunction( a1*L2NormSquared(data),  
                  a2*MixedL21Norm(),  
                  a3*L1Norm() )
```



$$K = \begin{bmatrix} A \\ D \\ D_x \end{bmatrix}$$

```
K = BlockOperator( ProjectionOperator(ig, ag),  
                  GradientOperator(ig),  
                  FiniteDifferenceOperator(ig, 'horizontal_x'))
```

$$\mathbb{I}_{\{0 < u < vm\}}(u)$$

```
G = IndicatorBoxPixelwise( lower=0.0,  
                          upper=v*m )
```

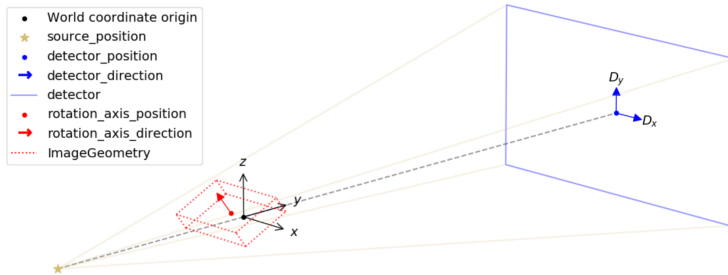
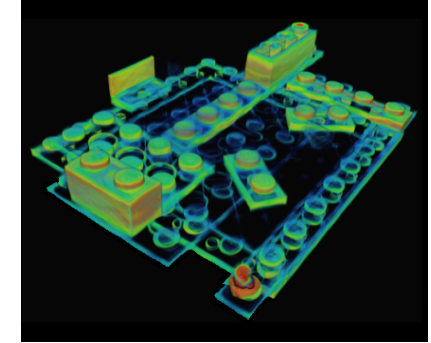
$$\operatorname{argmin}_u f(Ku) + g(u)$$

```
algo = PDHG( initial=ig.allocate(0.0),  
            f=F,  
            g=G,  
            operator=K,  
            max_iteration=2000 )  
algo.run()
```

Case Studies in CLIL : Laminography

Artifacts reduced by TV regularisation

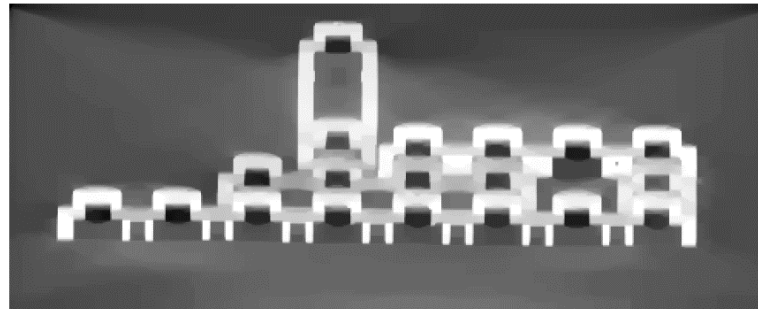
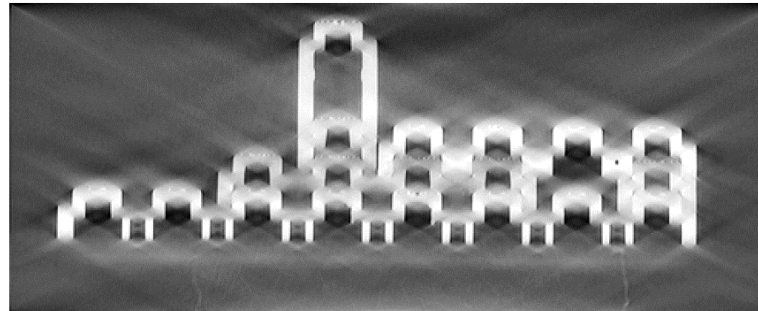
- Planar samples, like composite panels and printed circuit boards are difficult to scan due to different exposure along views.
- Conventional CT scan gives limited-angle artifacts, missing edges. Laminography allows uniform exposure.
- **Dataset** : Planar LEGO-brick



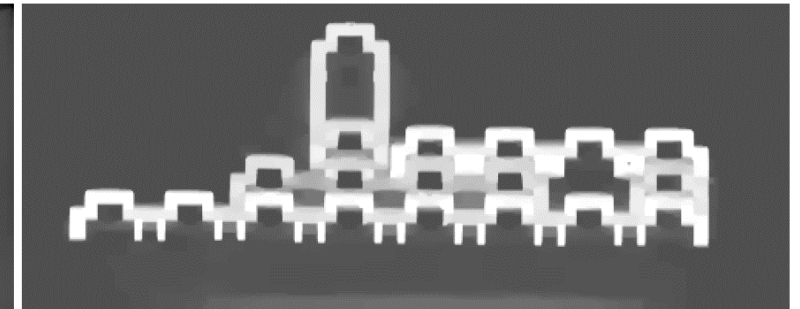
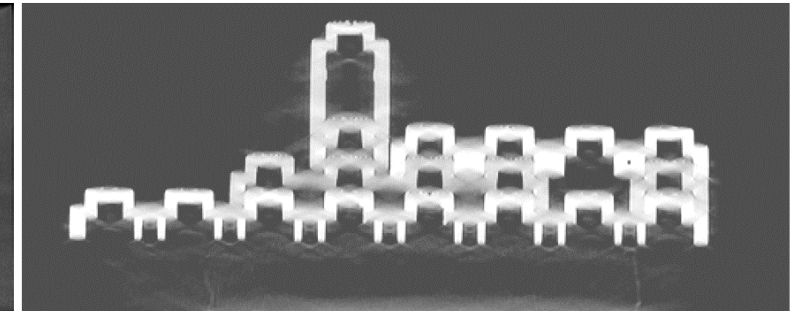
No regu.

TV regu.

Least Squares, unconstrained



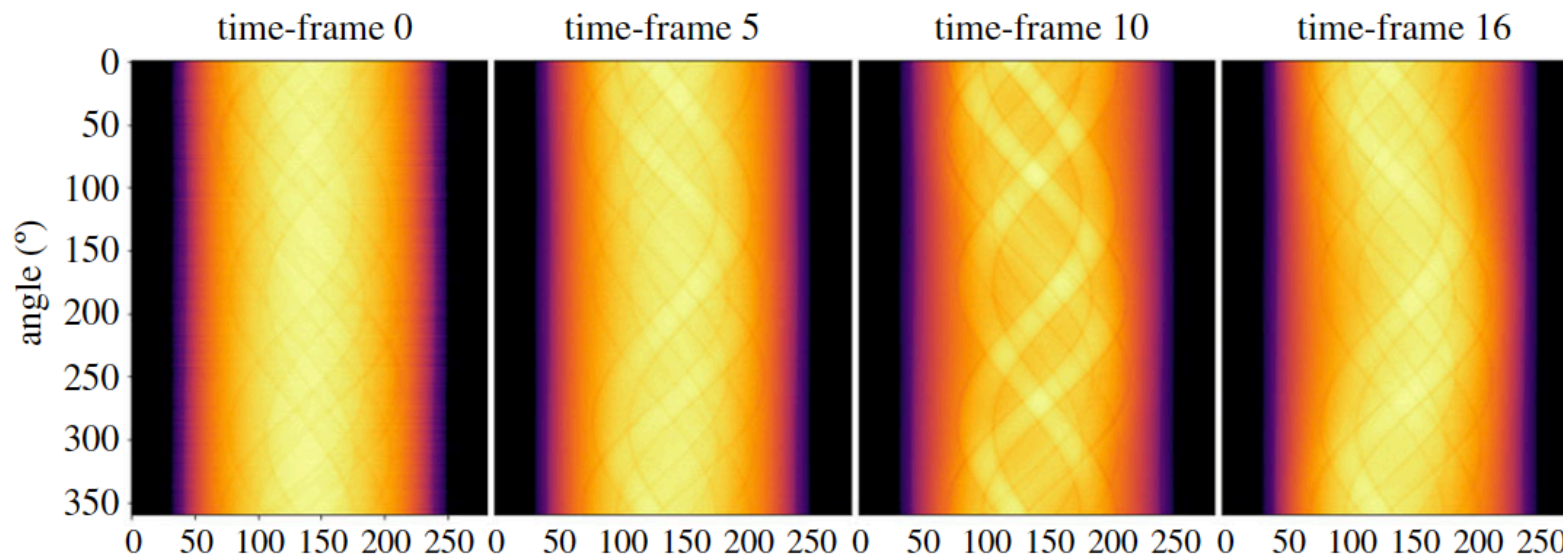
Least squares, nonnegativity



Case Studies in CIL : Dynamic CT

Limited angle Dynamic X-Ray CT reconstruction

- Scan a sample in time that undergoes some change.
- Use iterative reconstruction methods to increase temporal resolution and reduce radiation dose during a CT scan.
- **Dataset:** Simulate diffusion of liquids inside plant stems. Dynamic agarose-gel phantom perfused with a liquid contrast agent (iodine).

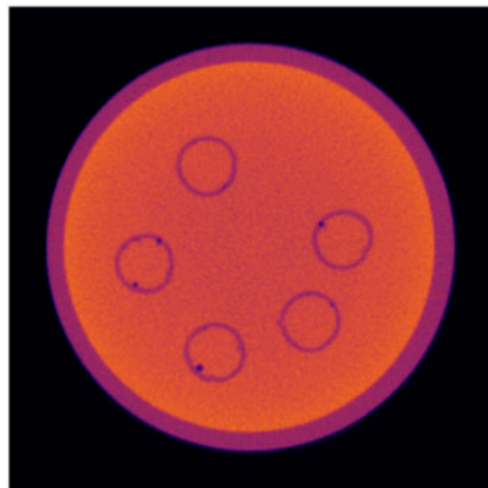


Papoutsellis et al. 2021, Core Imaging Library - Part II: multichannel reconstruction for dynamic and spectral tomography

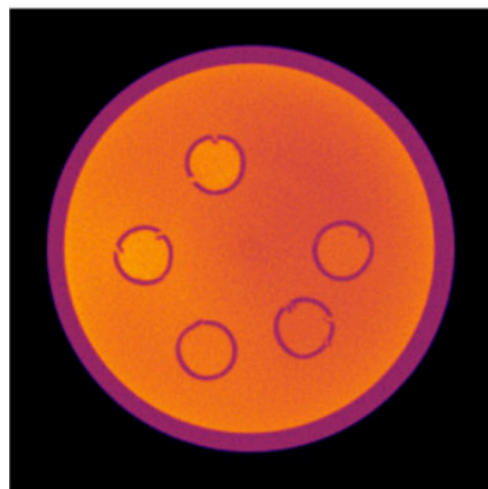
FIPS dataset: Tommi Heikkilä, Hanna Help and Alexander Meaney: "Gel phantom data for dynamic X-ray tomography"

Case Studies in CIL : Dynamic CT

reference (pre-scan)
720 projections



reference (post-scan)
1600 projections



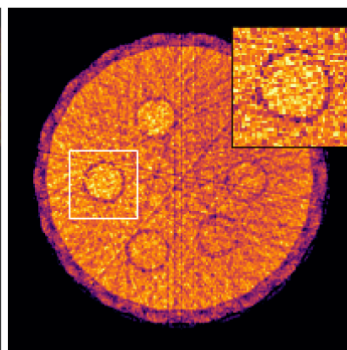
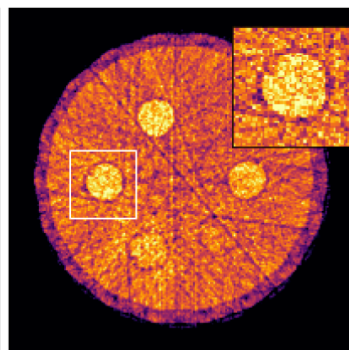
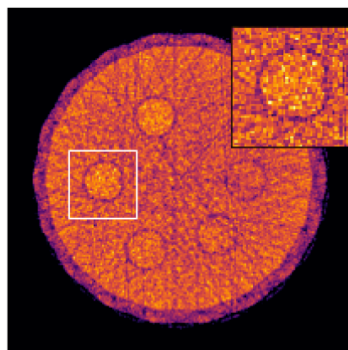
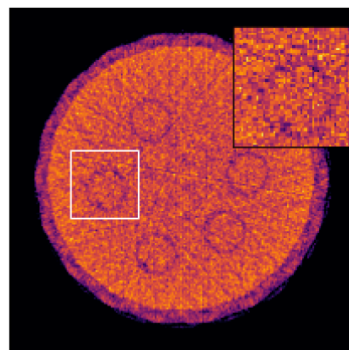
Time-frame 0

Time-frame 5

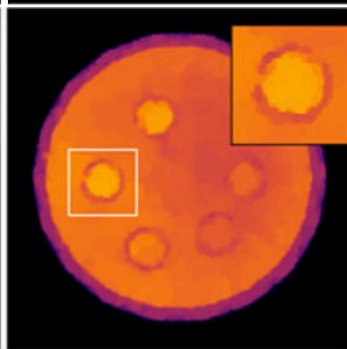
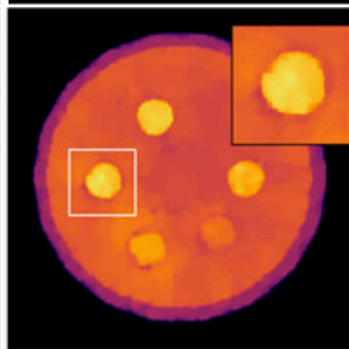
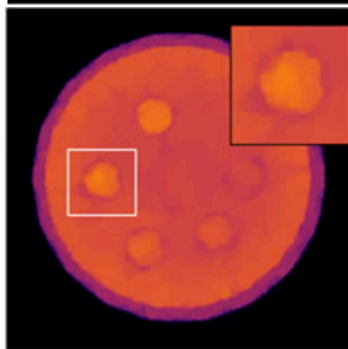
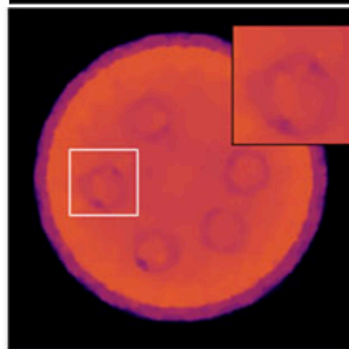
Time-frame 10

Time-frame 16

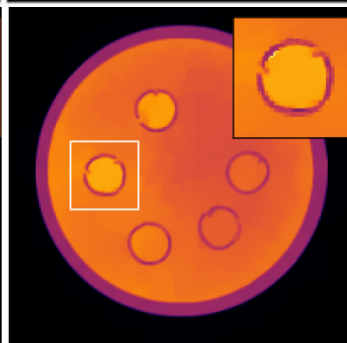
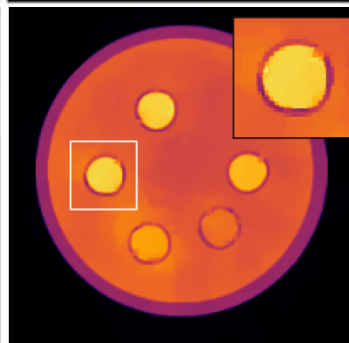
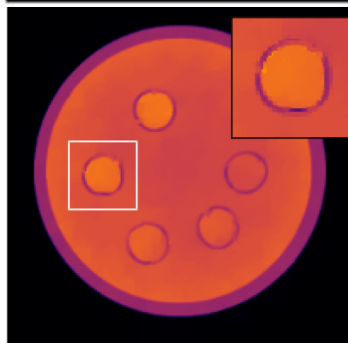
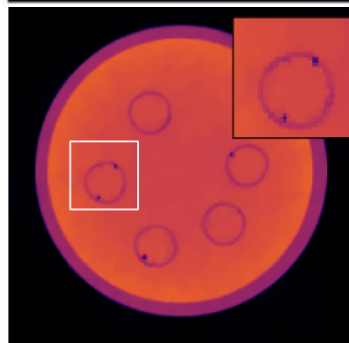
FBP
18 projections



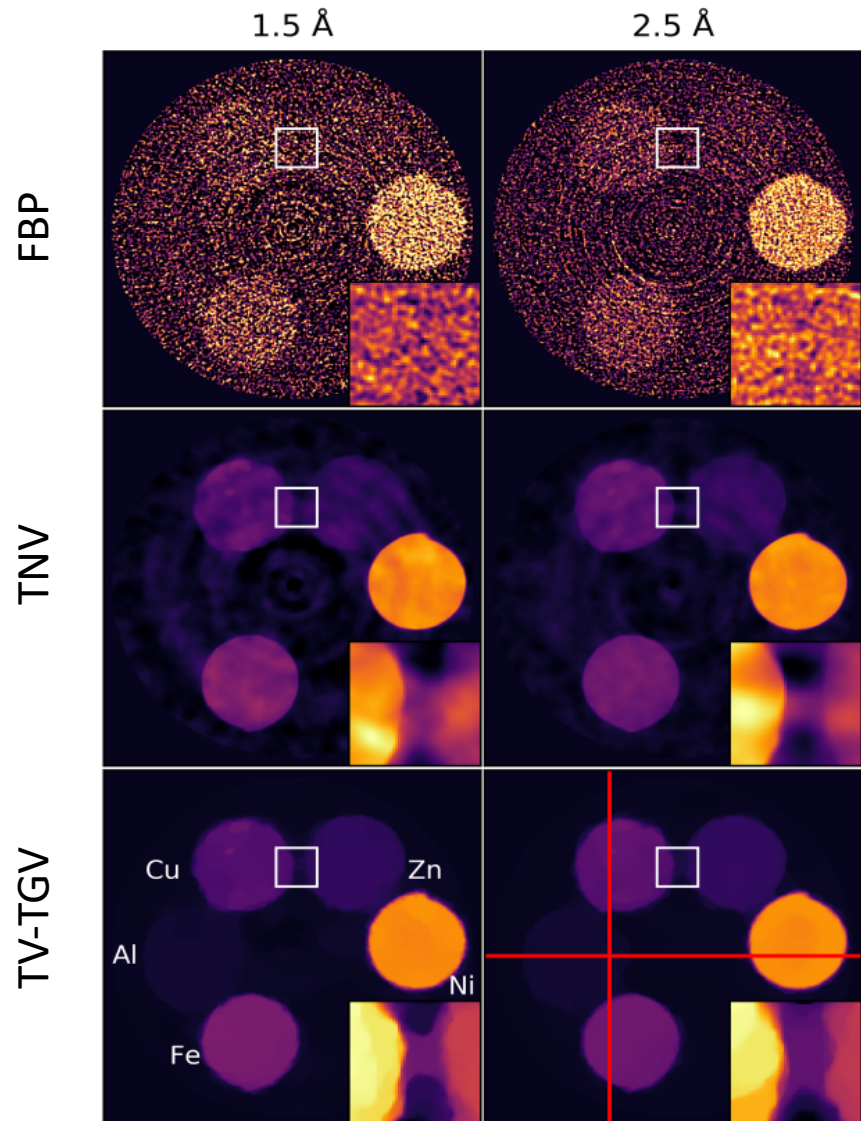
TV
18 projections



dTV
18 projections

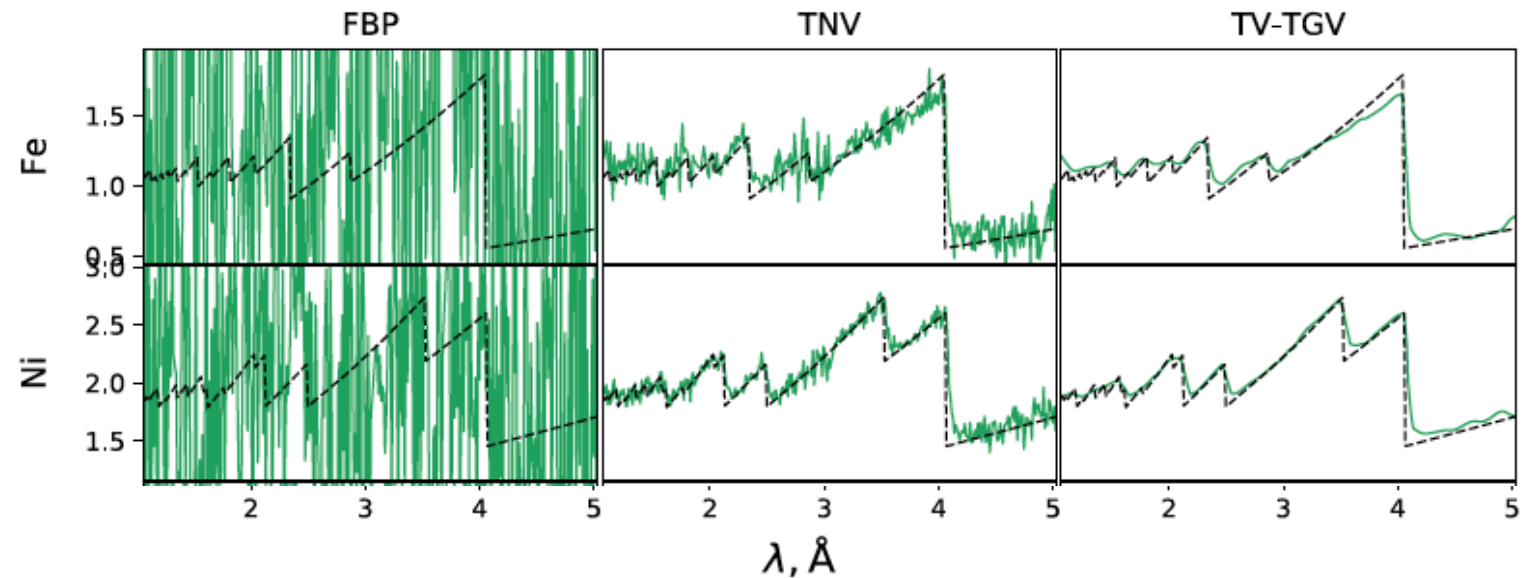


Case Studies in CIL : Hyperspectral Neutron CT

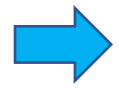


- Proposed spatio-spectral TV-TGV regularization
- Enables clear identification of Bragg edges in 3D

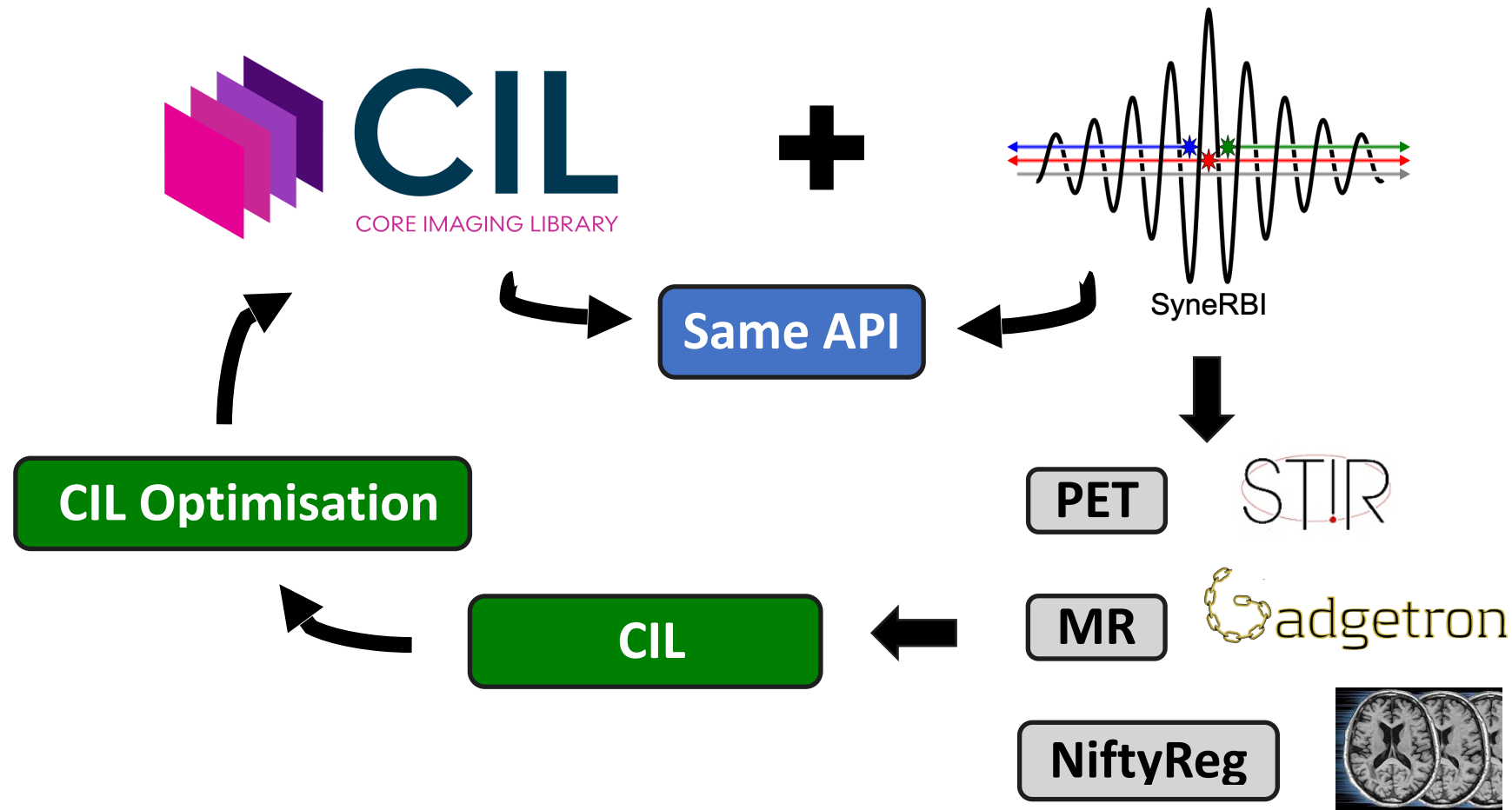
$$\arg \min_u \frac{1}{2} \|Au - b\|^2 + \alpha \text{TV}(u_{space}) + \beta \text{TGV}(u_{spectral})$$



Case Studies in CIL : PET/MR Reconstruction

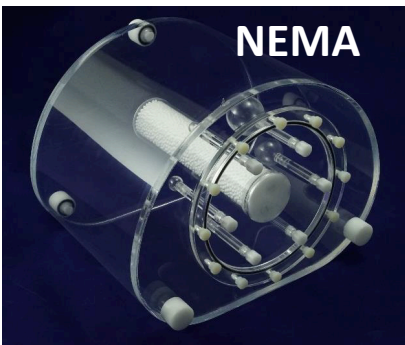


*Synergy between two libraries: **Core Imaging Library (CIL)** and **Synergistic Image Reconstruction Framework (SIRF)** for Medical Imaging Applications*

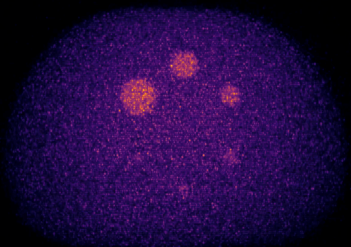


Ovtchinnikov et al. 2017, SIRF: Synergistic Image Reconstruction Framework

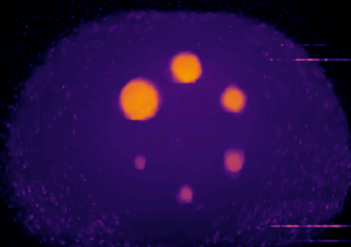
Case Studies in CIL : PET/MR Reconstruction



OSEM



PDHG (CIL) : KL + TV



MRI reconstruction

No motion correction & regularisation

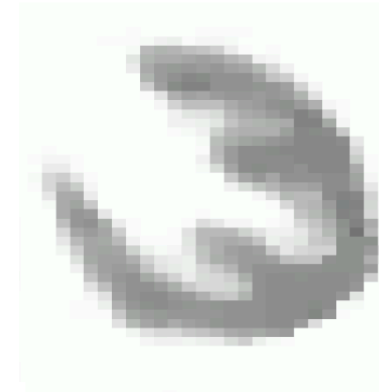


Motion correction with TV



PET reconstruction

Incorporate
information from
MRI



Brown et al. 2021: Motion estimation and correction for simultaneous PET/MR using SIRT and CIL

Case Studies in CLIL : Non Tomography

Colour image inpainting and salt/pepper denoising

$$\arg \min_u \frac{1}{2} \|Mu - g\|_1 + \text{TGV}_{\alpha, \beta}(u)$$

Ground truth



Corrupted image



L1 + TGV



Stochastic Optimisation in CIL

➔ Extend CIL Optimisation (Deterministic) framework to Stochastic Optimisation

Organised: [CCP SyneRBI](#) , [CCPi](#) , [PET++](#)

- 1st Hackathon: November 23-26, 2021
- 2nd Hackathon: April 4-7, 2022
- Finden Ltd - Analysis for Innovators (A4i): Denoising of chemical imaging and tomography data. In collaboration with National Physical Laboratory (05/2023 - 09/2023)

Joint work: Kris Thielemans, Gillman Ashley, Tang Junqi, Zeljko Kereta, Imraj Singh, Gemma Fardell, Evgueni Ovtchinnikov, Matthias Ehrhardt, Laura Murgatroyd, Robert Twyman, Edoardo Pasca, Claire Delplancke, Georg Schramm, Daniel Deidda, Jakob Jørgensen, Sam Porter, Margaret Duff, Antony Vamvakeros, Simon Jacques, Casper da Costa-Luis

Stochastic Optimisation in CIL

➔ Extend CIL Optimisation (Deterministic) framework to Stochastic Optimisation

- Implement splitting for CIL/SIRF DataContainers: CT, PET, SPECT, MRI data
- Implement randomised algorithms in CIL, e.g., SGD, SAG, SAGA, SVRG and more
- Improve CIL optimisation functionality, e.g., step size, preconditioning

$$\begin{aligned} & \min_x f(x) + g(x) \quad , \quad \min_x f(Kx) + g(x) \\ & \min_x \sum_{i=1}^n f_i(x) + g(x) \quad , \quad \min_x \sum_{i=1}^n f_i(K_i x) + g(x) \end{aligned}$$

Defazio et al. 2014, SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives

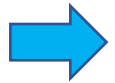
Schmidt et al. 2017, Minimizing finite sums with the stochastic average gradient

Johnson et al., 2013, Accelerating Stochastic Gradient Descent using Predictive Variance Reduction

Stochastic Optimisation in CL

$$\min_x \sum_{i=1}^n f_i(x) + g(x)$$

- *Avoid computing the full gradient per iteration, i.e., gradient for all n .*



- *Select a random index $i_k \in \{1, \dots, n\}$ and compute ∇f_{i_k} per iteration.*

GD	PGA/ISTA	APGA/FISTA
$x_{k+1} = x_k - \gamma_k \nabla f(x_k)$	$x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k))$	$x_{k+1} = \text{prox}_{\gamma_k g}(y_k - \gamma_k \nabla f(y_k))$ $\alpha_{k+1} = \frac{1 + \sqrt{1 + 4\alpha_k^2}}{2}$ $y_k = x_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(x_k - x_{k-1})$

Stochastic Optimisation in CIL

GD	PGA/ISTA	APGA/FISTA
$x_{k+1} = x_k - \gamma_k \nabla f_{i_k}(x_k)$  SGD	$x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f_{i_k}(x_k))$  Prox-SGD	$x_{k+1} = \text{prox}_{\gamma_k g}(y_k - \gamma_k \nabla f_{i_k}(y_k))$ $\alpha_{k+1} = \frac{1 + \sqrt{1 + 4\alpha_k^2}}{2}$ $y_k = x_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(x_k - x_{k-1})$  Acc-Prox-SGD

Stochastic Optimisation Design



- No direct implementation of Stochastic algorithms, e.g., SGD.
- Use a deterministic algorithm, e.g., GD, already available in CIL.
- Implement a Stochastic Gradient “Functions” or Variance-Reduced “Functions”
- Example $\text{SGD} := \text{GD (CIL Algorithm)} + \text{SGFunction (CIL Function)}$

Stochastic Optimisation in CIL

GD	ISTA	FISTA
$x_{k+1} = x_k - \gamma_k \tilde{\nabla} f_{i_k}(x_k)$	$x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \tilde{\nabla} f_{i_k}(x_k))$	$x_{k+1} = \text{prox}_{\gamma_k g}(y_k - \gamma_k \tilde{\nabla} f_{i_k}(y_k))$ $\alpha_{k+1} = \frac{1 + \sqrt{1 + \alpha_k^2}}{2}$ $y_{k+1} = x_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(x_k - x_{k-1})$

`ApproximateGradientSumFunction`

$$\sum_{i=1}^n f_i(x)$$

Stochastic Gradient and Variance-Reduced
CIL “Functions”

`SGFunction`

`SAGFunction`

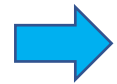
`SVRGFunction`

`SAGAFunction`

`LSVRGFunction`

`and more ...`

Stochastic Optimisation in CL



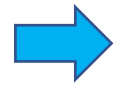
Plug and Play Framework - Different Stochastic Gradient Functions

Algorithms Stochastic Function	GD	ISTA	FISTA
SGFunction	SGD	Prox-SGD	Acc-Prox-SGD
SAGFunction	SAG	Prox-SAG	Acc-Prox-SAG
SAGFunction	SAGA	Prox-SAGA	Acc-Prox-SAGA
SVRGFunction	SVRG	Prox-SVRG	Acc-Prox-SVRG
LSVRGFunction	LSVRG	Prox-LSVRG	Acc-Prox-LSVRG



```
gd = GD(initial=initial, objective_function=f, step_size=step_size,  
        update_objective_interval=1, max_iteration=10)  
gd.run(verbose=1)  
  
pgd = ISTA(initial=initial, f=f, g=g, step_size=step_size,  
           update_objective_interval=1, max_iteration=10)  
pgd.run(verbose=1)
```

Stochastic Optimisation in CL



Plug and Play Framework - Different Stochastic Gradient Functions

Algorithms Stochastic Function	GD	ISTA	FISTA
SGFunction	SGD	Prox-SGD	Acc-Prox-SGD
SAGFunction	SAG	Prox-SAG	Acc-Prox-SAG
SAGAFunction	SAGA	Prox-SAGA	Acc-Prox-SAGA
SVRGFunction	SVRG	Prox-SVRG	Acc-Prox-SVRG
LSVRGFunction	LSVRG	Prox-LSVRG	Acc-Prox-LSVRG



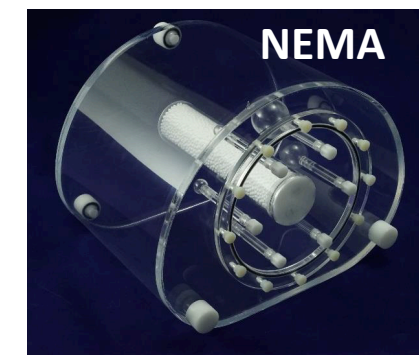
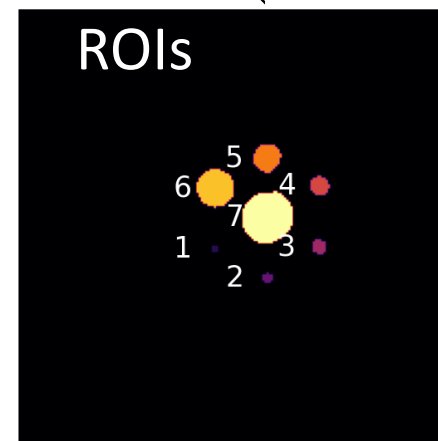
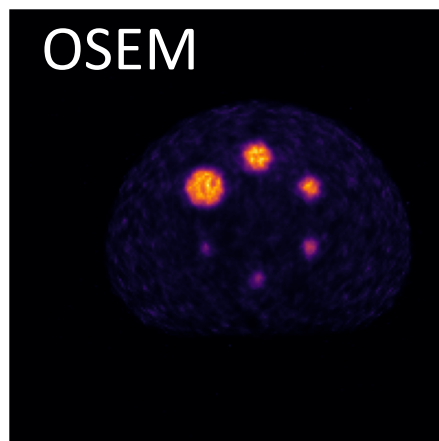
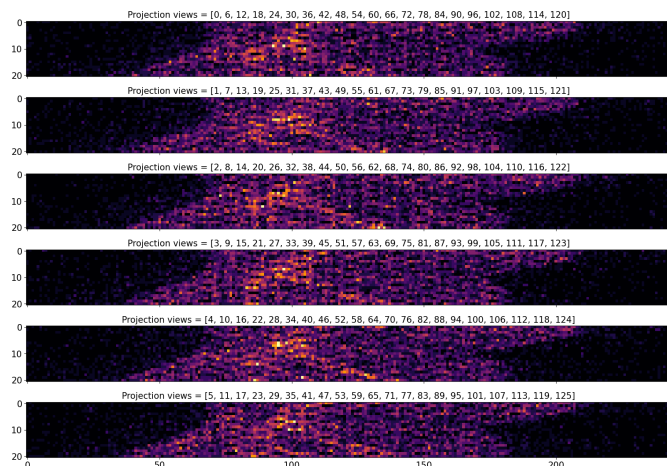
```
sgd = GD(initial=initial, objective_function=SGFunction(fi), step_size=step_size,  
         update_objective_interval=1, max_iteration=10)  
sgd.run(verbose=1)  
  
prox_sgd = ISTA(initial=initial, f=SGFunction(fi), g=g, step_size=step_size,  
               update_objective_interval=1, max_iteration=10)  
prox_sgd.run(verbose=1)
```

Applications (PET - Smooth optimisation)

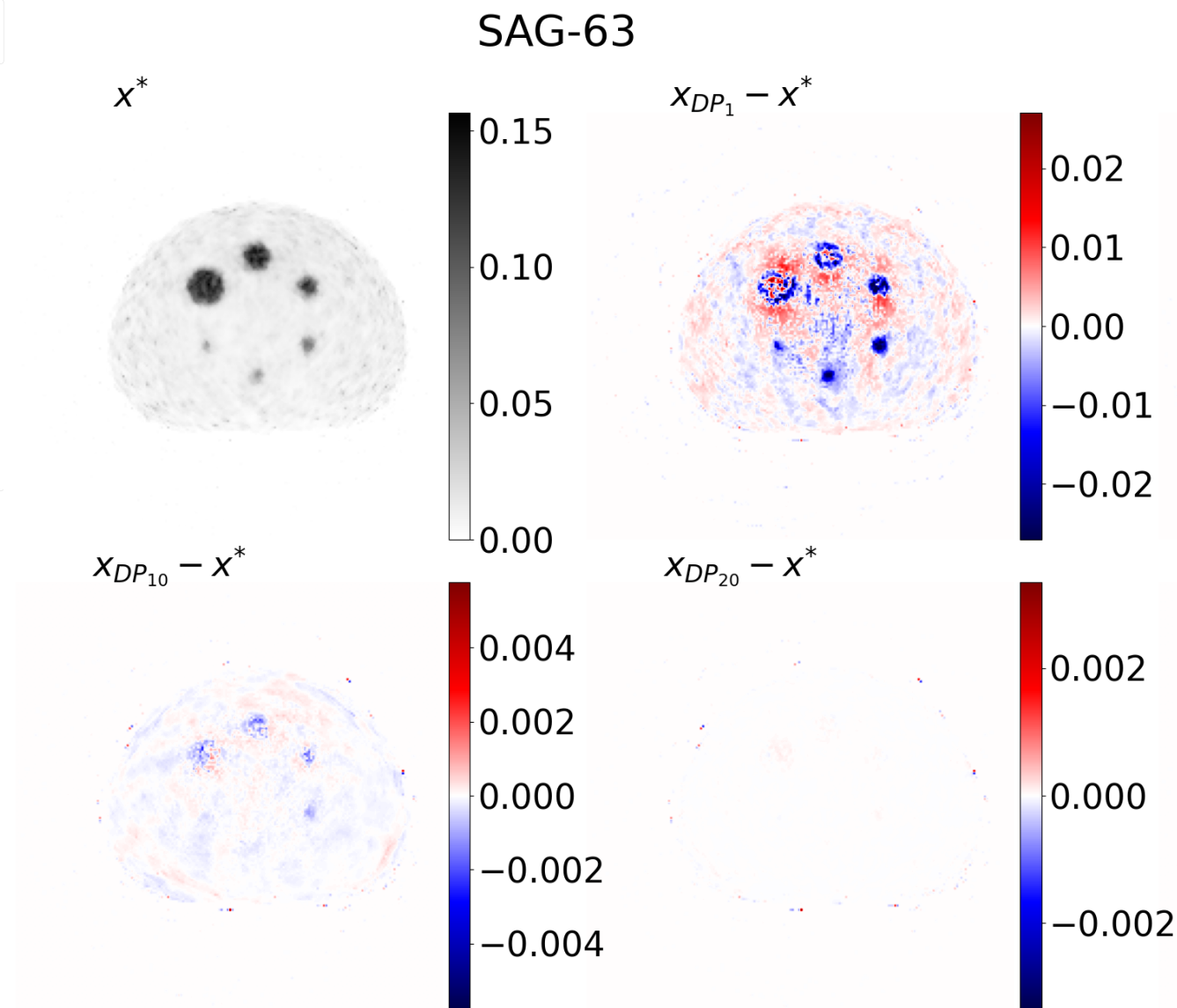
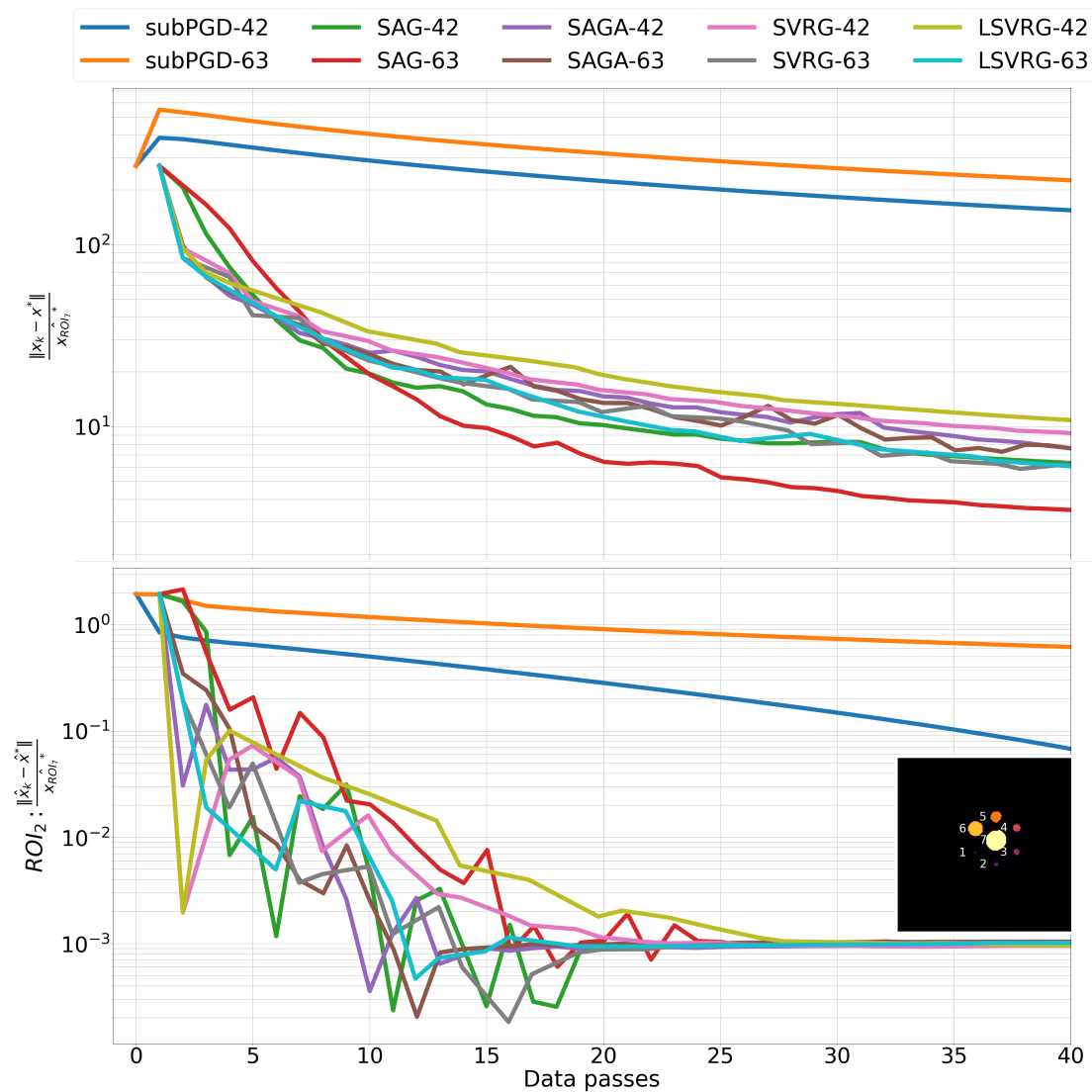
$$\min_u \sum Au - b \log(Au + \eta) + \text{RDP}(u) + \mathbb{I}_{\{u>0\}}(u)$$

- **Splitting Method:** Ordered (126 projections), Subsets = 21, 42, 63
- **Sampling Method (Functions):** Random with replacement
- **Optimal Solution:** (subset) Preconditioned Gradient Descent with relaxed step size
7 subsets (20000 iterations) $\approx$ 10hrs

- **Preconditioning:** $D(x_k) = \frac{x_{OSEM} + \varepsilon}{A^T 1}$
- **Initial:** OSEM
- **Step size:** $\gamma_k = \frac{1}{1 + \eta \left(k // (n/2) \right)}$



Applications (PET - Smooth optimisation)



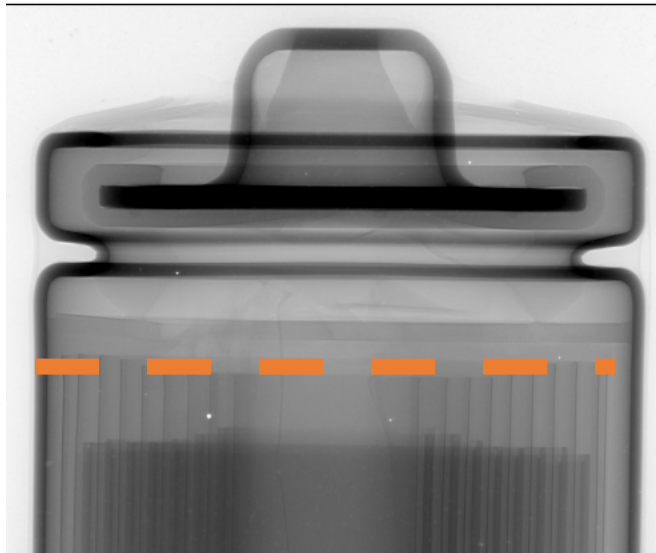
Applications (MicroCT - NonSmooth optimisation)

$$\min_u \frac{1}{2} \|Au - b\|^2 + \alpha \|\nabla u\|_{2,1} + \mathbb{I}_{\{u>0\}}(u)$$

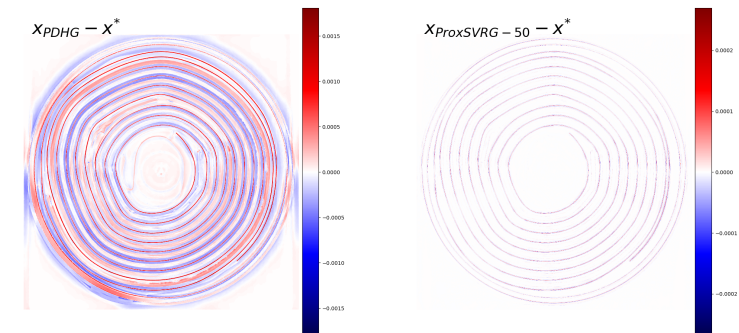
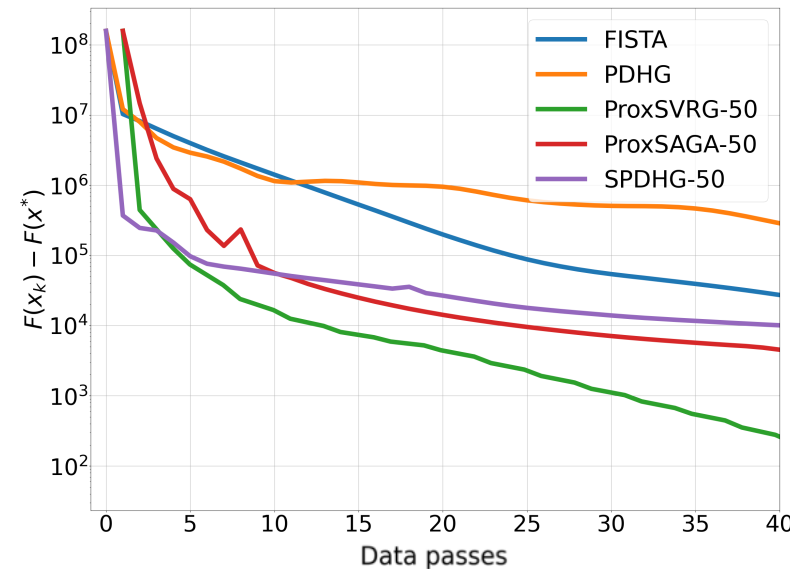
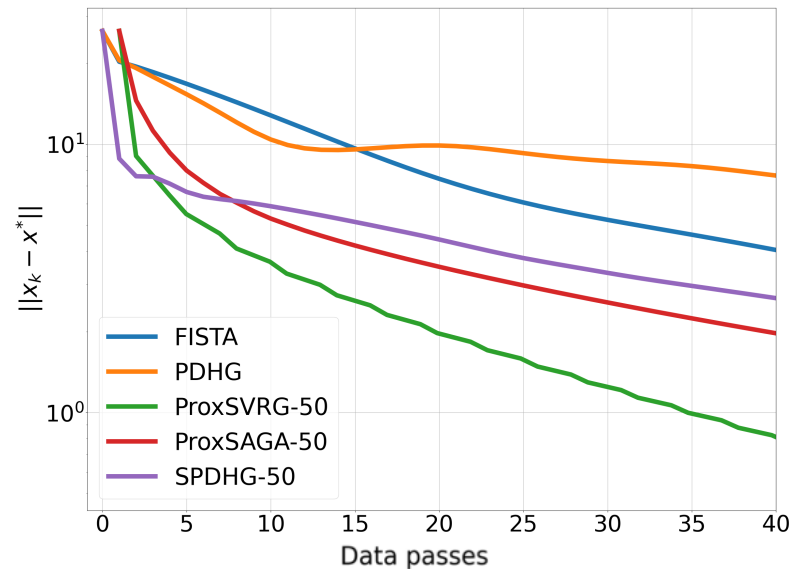
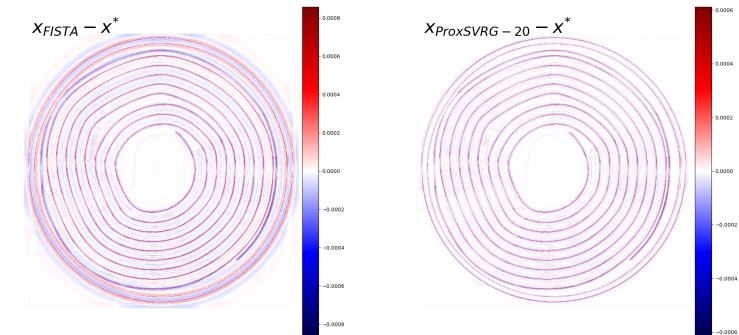
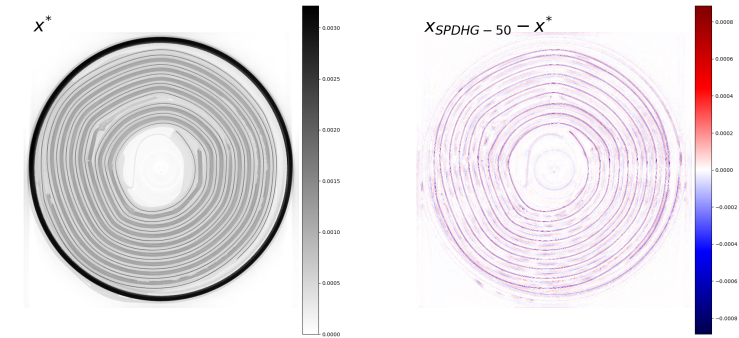
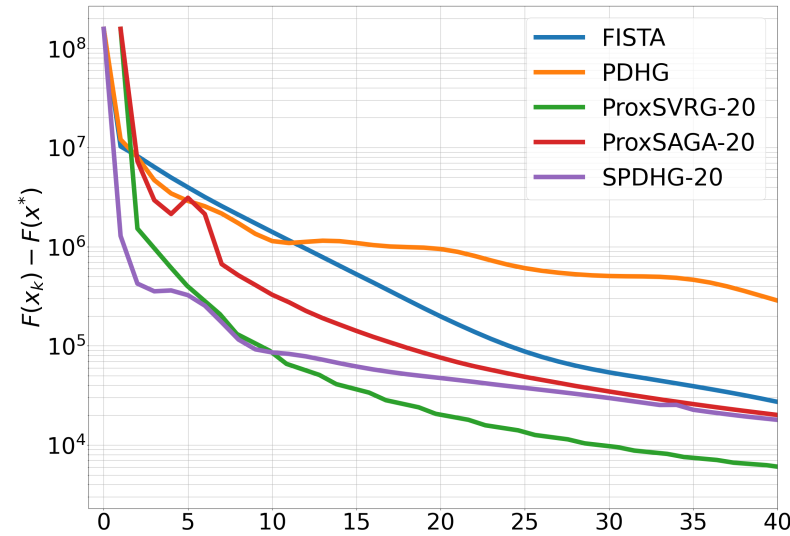
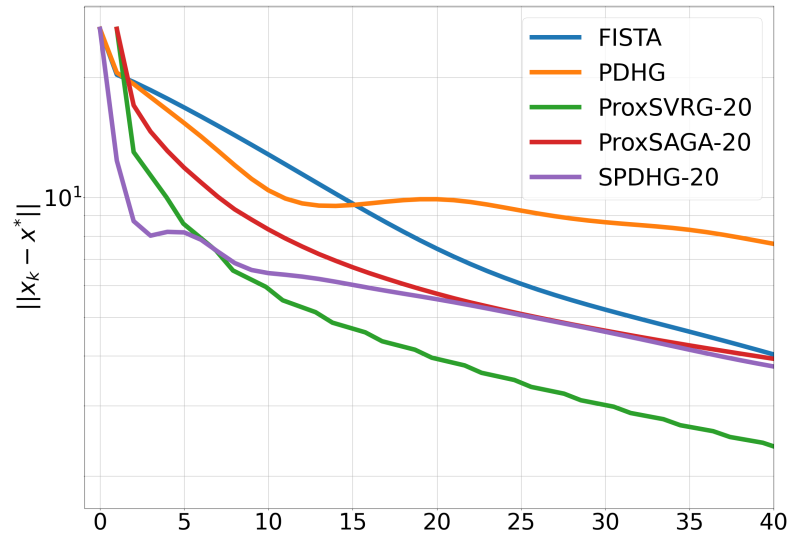
- **Splitting Method:** Ordered (900 projections), Subsets = 20, 50
- **Sampling Method (Functions):** Random with replacement
- **Optimal Solution:** FISTA - 5000 iterations $\approx$ 4days
- **Detector:** 1561x1561, 300 slices



Finden



Applications (MicroCT - NonSmooth optimisation)



Chambolle et al. 2018, Stochastic Primal-Dual Hybrid Gradient Algorithm with Arbitrary Sampling and Imaging Applications Penalised PET Image Reconstruction

Try CIL

```
conda create --name cil -c conda-forge -c intel -c ccpi cil=24.0.0
```

```
docker run --rm --gpus all -p 8888:8888 -it ghcr.io/tomographicimaging/cil:latest
```



CIL on the Cloud

launch binder



- CPU jupyterlab instance
- Easy and free to use
- No installation required

Open in Colab



- GPU (up to 16Gb)
- Easy and free to use
- 4 easy steps to install

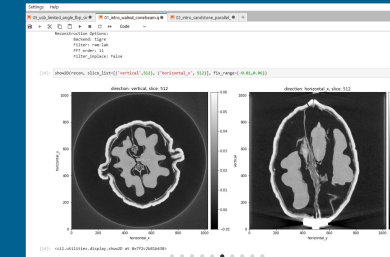


Home News Reports Software Training Events Community

Collaborative Computational Project in Tomographic Imaging

CCPi aims to provide the UK tomography community with a toolbox of algorithms that increases the quality and level of information that can be extracted by computed tomography. Chaired by [Prof Philip Withers \(University of Manchester\)](#), co-ordinated by staff in the [Scientific Computing Department STFC](#), led by a working group of experimental and theoretical academics with links to [Diamond Light Source](#), [ISIS](#) and Industry

[Read more...](#)



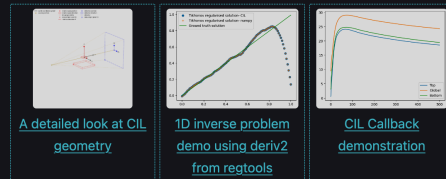
CIL-User-Showcase

001_multibang_regularisation	Multibang (#7)	10 months ago
002_Deblurring	Deblurring Contribution facilitated by Jakob Sauer Jorgen...	9 months ago
003_ID_integral_inverse_problem	1D inverse problems showcase from hackathon (#3)	9 months ago
004_STEMPO-DynamicCT	Create stempo2d_v2_1.ipynb (#4)	10 months ago
005_dynamic_mr_recon	Dyn mr recon hackathon (#5)	8 months ago
006_gVXR2CIL	gVXR2CIL (#2)	6 months ago
007_Hyperspectral_regularisation	Hyperspectral notebook by Ida Puggaard, DTU	7 months ago
008_KL_divergence_and_weighted_least_s...	Notebook from Francesca Bevilacqua comparing LS WLS ...	6 months ago
009_offset_CT_apple	Offset CT scan of an apple, contributed by Estrid Buhl Na...	6 months ago
example_contribution	Create PULL_REQUEST_TEMPLATE.md and example cont...	last year
README.md	Update README.md (#14)	6 months ago

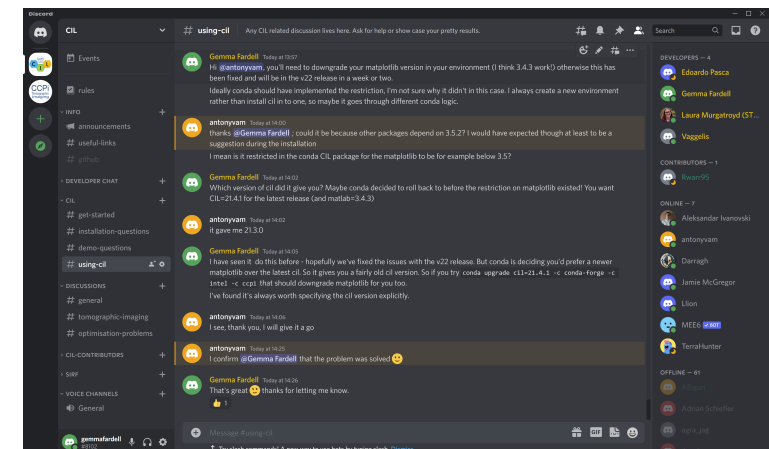
Tutorials

Table of Contents

- Introduction
- Framework
- Read/write AcquisitionData and ImageData
- Optimisation framework
- Block Framework
- Processors
- Recon
- Utilities
- CIL Plugins
- Developers' Guide
- Tutorials



Previous
[Developers' Guide](#)



CIL: Agnostic API for Inverse Imaging Problems

Major Breakthrough in Python Scientific Ecosystem



Data-Array API or NEP 47 or



Python Array API Standard: Toward Array Interoperability in the Scientific Python Ecosystem

Aaron Meurer^{1*}, Athan Reines^{2†}, Ralf Gommers^{3†}, Yao-Lung L. Fang^{4‡}, John Kirkham^{5‡}, Matthew Barber^{6‡}, Stephan Hoyer⁷, Andreas Müller⁸, Sheng Zha^{9*}, Saul Shanabrook, Stephannie Jiménez Gacha[‡], Mario Lezcano-Casado[‡], Thomas J. Fan[‡], Tyler Reddy^{††}, Alexandre Passos, Hyukjin Kwon^{‡‡}, Travis Oliphant[‡], Consortium for Python Data API Standards



CIL User Meeting '24

We are pleased to announce that the 2nd annual CIL User Meeting will take place on the 4th-6th November 2024 at the **Rutherford Appleton Laboratory** (RAL) in Oxfordshire, UK.

Planned events include:

- An exciting technical talk programme from CIL users
- Introductory and advanced CIL user training sessions
- User support sessions and hackathons
- Town Hall user community discussions
- **CCPi Working Group** meeting

ccpi.ac.uk/events/cil-user-meeting-2024



Summary

- CIL is an open-source Python library for solving Imaging Inverse Problems
- Highly flexible and for different imaging modalities
- Modular design for Plug and Play Stochastic optimisation **NEW!**



- **Website** <https://www.ccpi.ac.uk/CIL>
- **Github** <https://tomographicimaging.github.io/CIL>
- **Discord** <https://discord.gg/kmBcU2kebB>
- **Demos** <https://github.com/TomographicImaging/CIL-Demos>
- **Showcase** <https://github.com/TomographicImaging/CIL-User-Showcase>

- Jørgensen et al. 2021: Core Imaging Library - Part I: a versatile Python framework for tomographic imaging, Phil. Trans. R. Soc. A, 379, 20200193: <https://doi.org/10.1098/rsta.2020.0192>
- Papoutsellis et al. 2021: Core Imaging Library - Part II: multichannel reconstruction for dynamic and spectral tomography, Phil. Trans. R. Soc. A, 379, 20200193: <https://doi.org/10.1098/rsta.2020.0193>

Thank you for your attention! Questions?