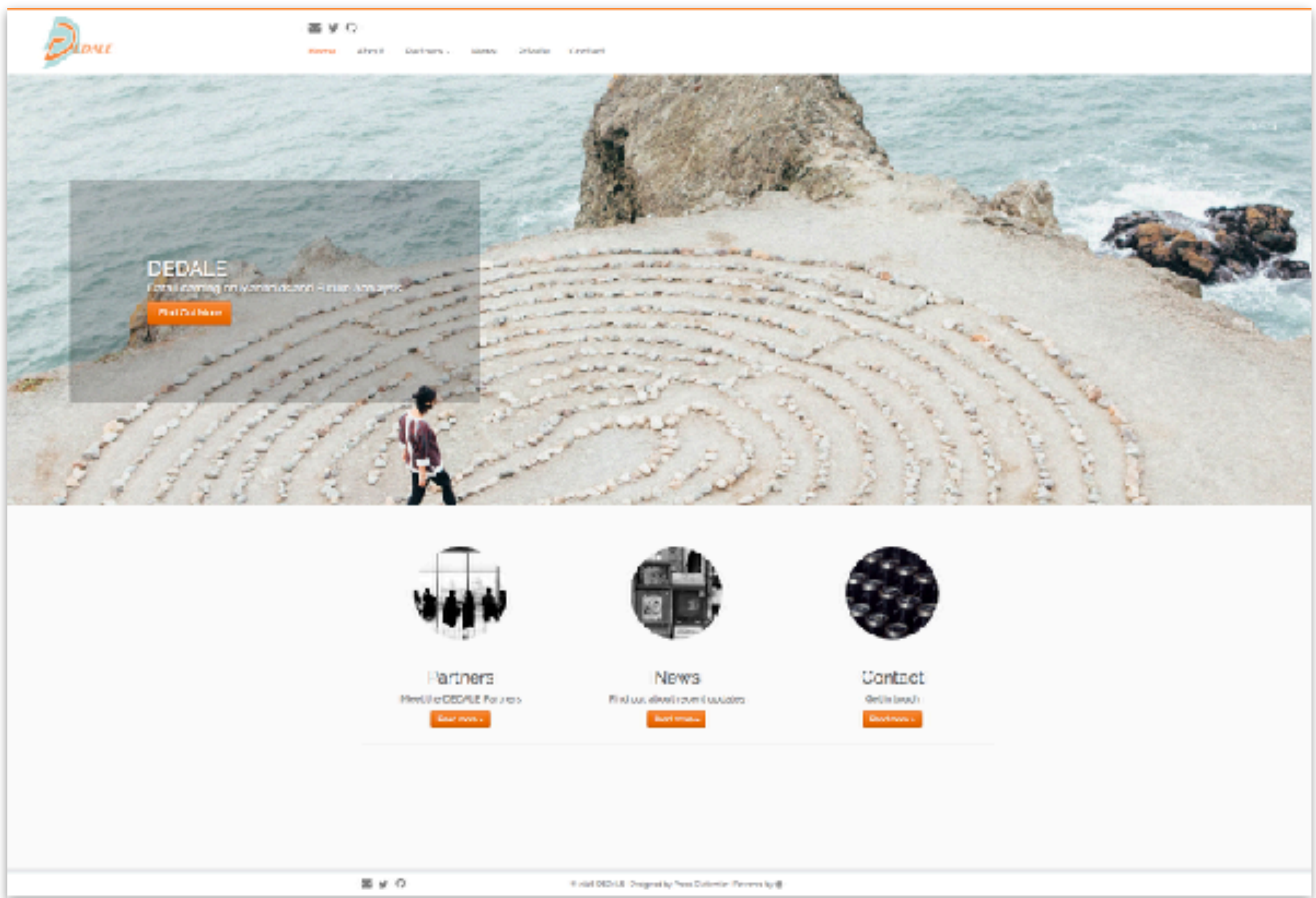
Regularisation techniques for solving inverse problems

Samuel Farrens CEA

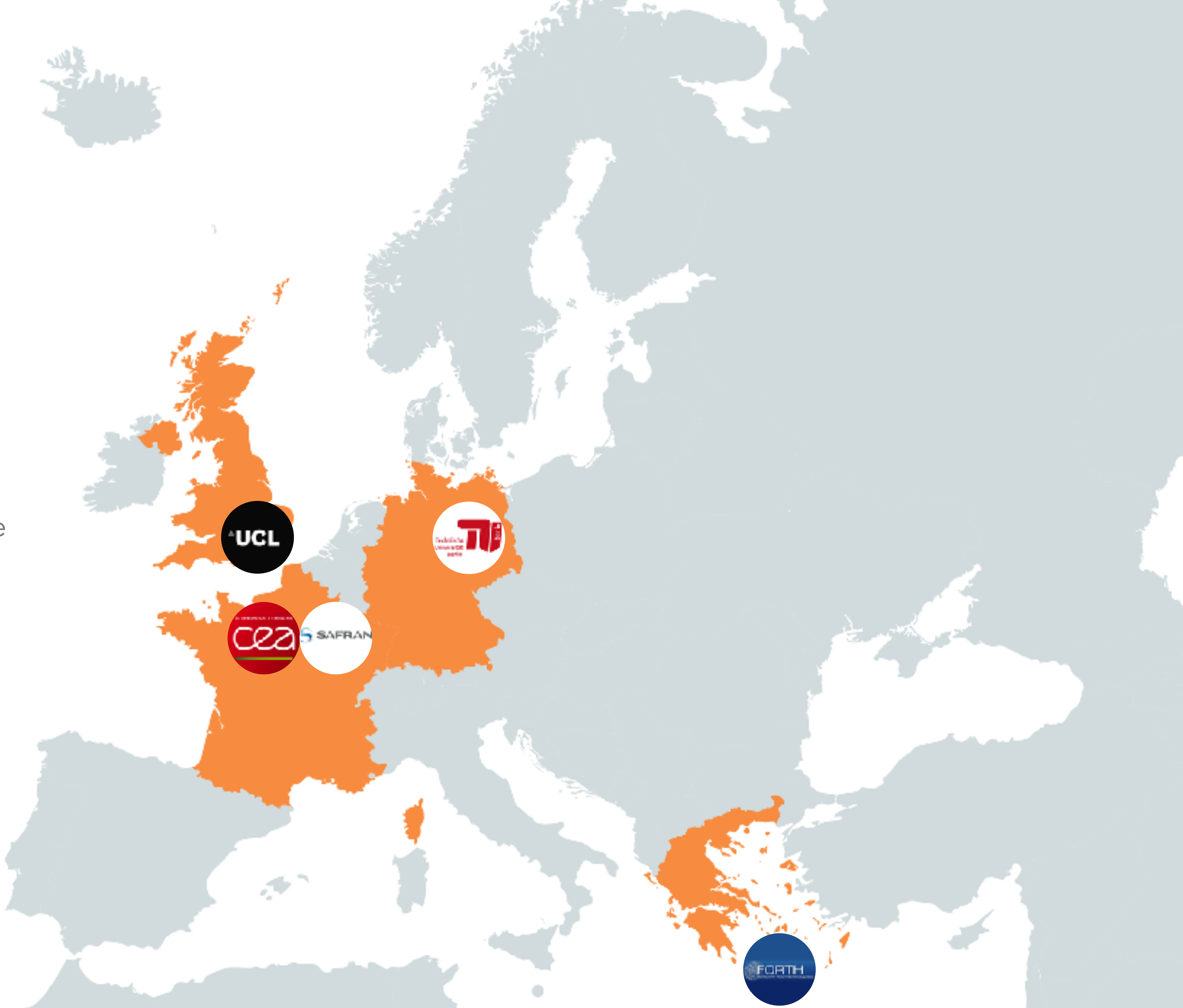
DEDALE

http://dedale.cosmostat.com



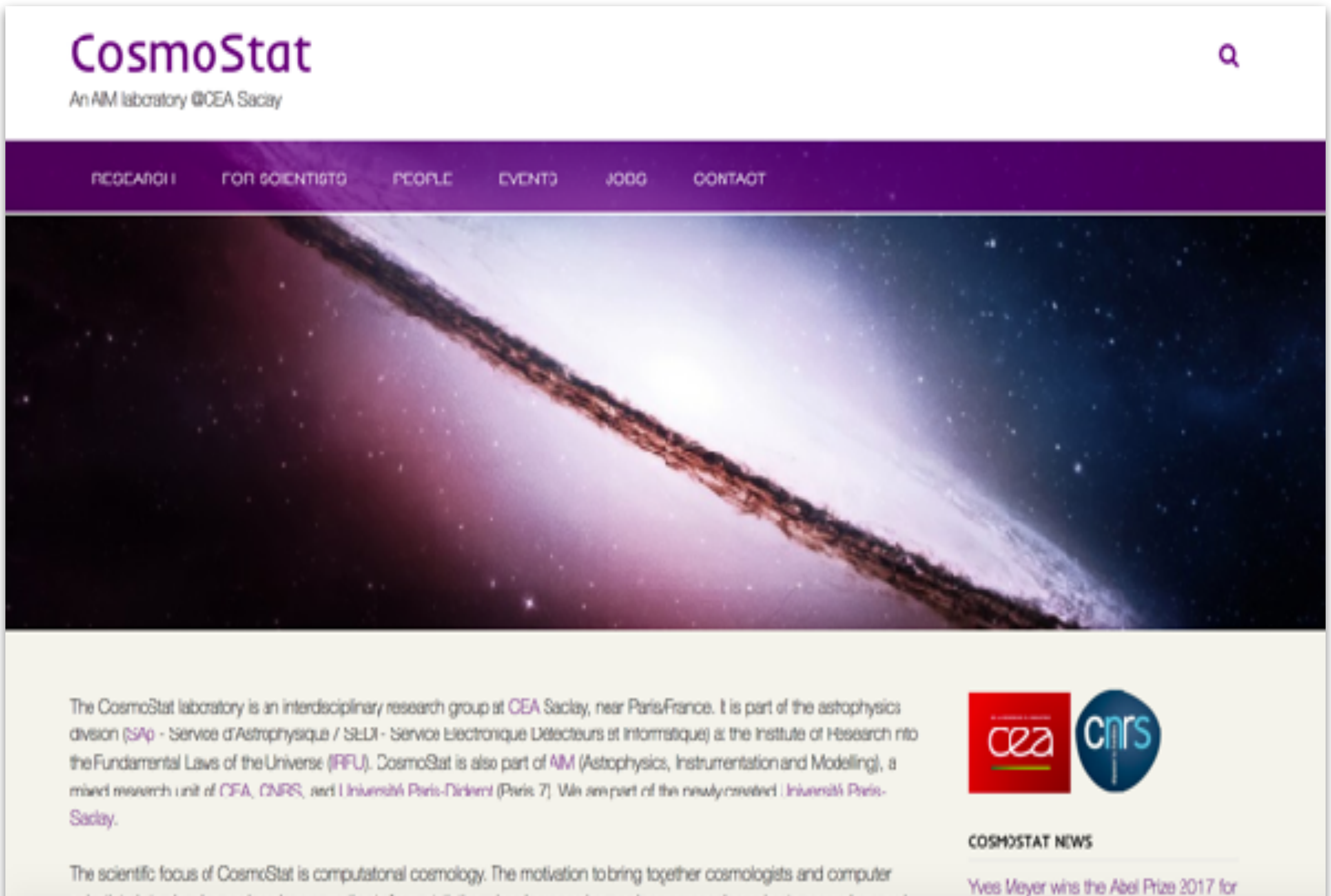
DEDALE

- Mathematics
- Signal Processing
- Computer Science
- Machine Learning
- Astrophysics
- Cosmology



CosmoStat

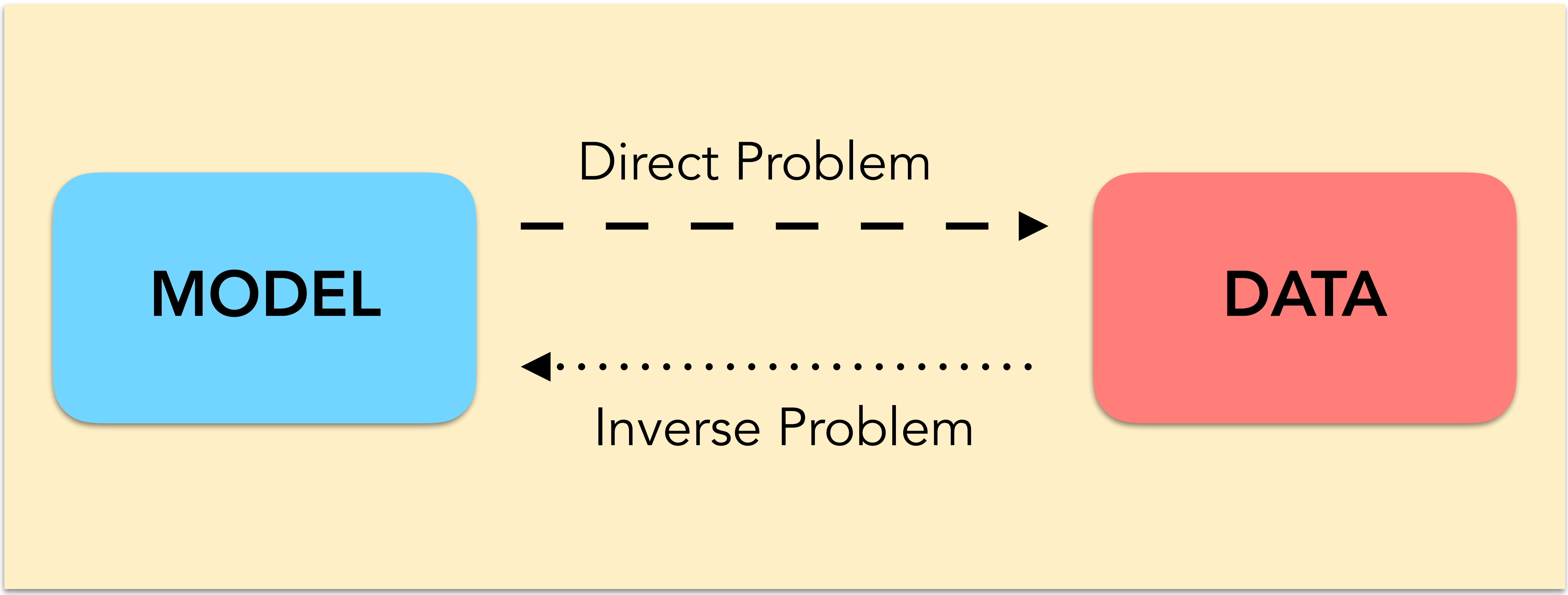
http://www.cosmostat.com



Outline

- **Inverse Problems**
  - Linear Regression
  - Ill-posed Problems
- **Regularisation**
  - Sparsity
  - Low-Rank Approximation
- **Deconvolution of Galaxy Images**
- **Summary**

Inverse Problems



With an inverse problem one attempts to obtain information about a physical system from observed measurements.

Inverse Problems

Linear Inverse Problem

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

$\mathbf{y} \in \mathbb{R}^m$

Data Vector

$\mathbf{A} \in \mathbb{R}^{m \times n}$

Operation Matrix

$\mathbf{x} \in \mathbb{R}^n$

Model Vector

Linear Regression

Straight Line : Direct Problem

$$y = mx + b$$

$$x = \begin{bmatrix} 8 & 2 & 11 & 6 & 5 & 4 & 12 & 9 & 6 & 11 \end{bmatrix}$$

Model

$$m = -1.1$$
$$b = 14$$

Model Vector

$$\mathbf{x} = \begin{bmatrix} 14.0 & 1.1 \end{bmatrix}$$

Operation Matrix

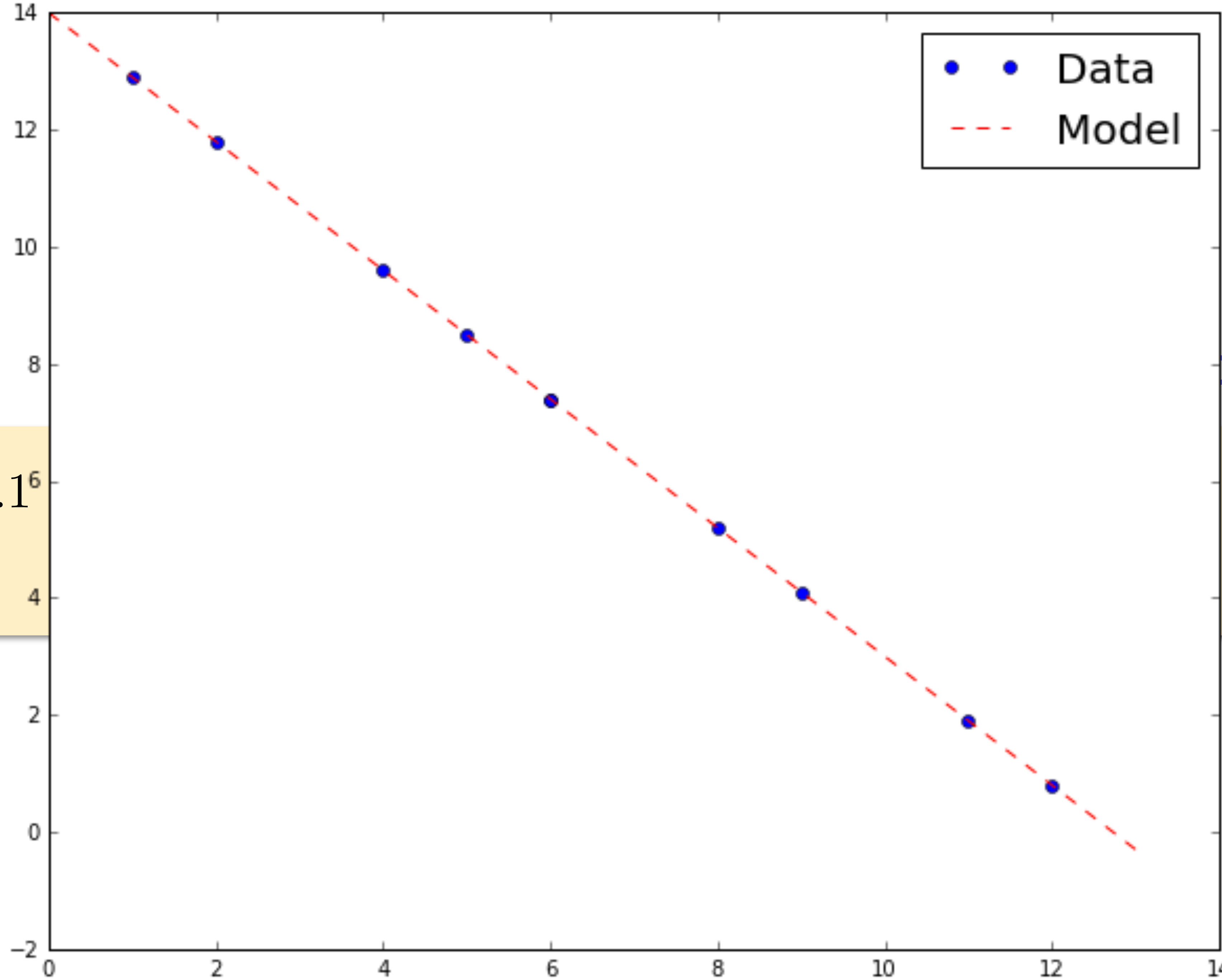
$$\mathbf{A} = \begin{bmatrix} 1 & 8 \\ 1 & 2 \\ 1 & 11 \\ 1 & 6 \\ 1 & 5 \\ 1 & 4 \\ 1 & 12 \\ 1 & 9 \\ 1 & 6 \\ 1 & 11 \end{bmatrix}$$

Data Vector

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

Linear Regression

Straight Line : Direct Problem



Model

$$m = -1.1$$
$$b = 14$$

Data Vector

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

Linear Regression

Straight Line : Inverse Problem

$$y = mx + b$$

$$x = \begin{bmatrix} 8 & 2 & 11 & 6 & 5 & 4 & 12 & 9 & 6 & 11 \end{bmatrix}$$
$$y = \begin{bmatrix} 3 & 10 & 3 & 6 & 8 & 12 & 1 & 4 & 9 & 14 \end{bmatrix}$$

Data Vector

$$\mathbf{y} = \begin{bmatrix} 3 \\ 10 \\ 3 \\ 6 \\ 8 \\ 12 \\ 1 \\ 4 \\ 9 \\ 14 \end{bmatrix}$$

Operation Matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 8 \\ 1 & 2 \\ 1 & 11 \\ 1 & 6 \\ 1 & 5 \\ 1 & 4 \\ 1 & 12 \\ 1 & 9 \\ 1 & 6 \\ 1 & 11 \end{bmatrix}$$

Model Vector

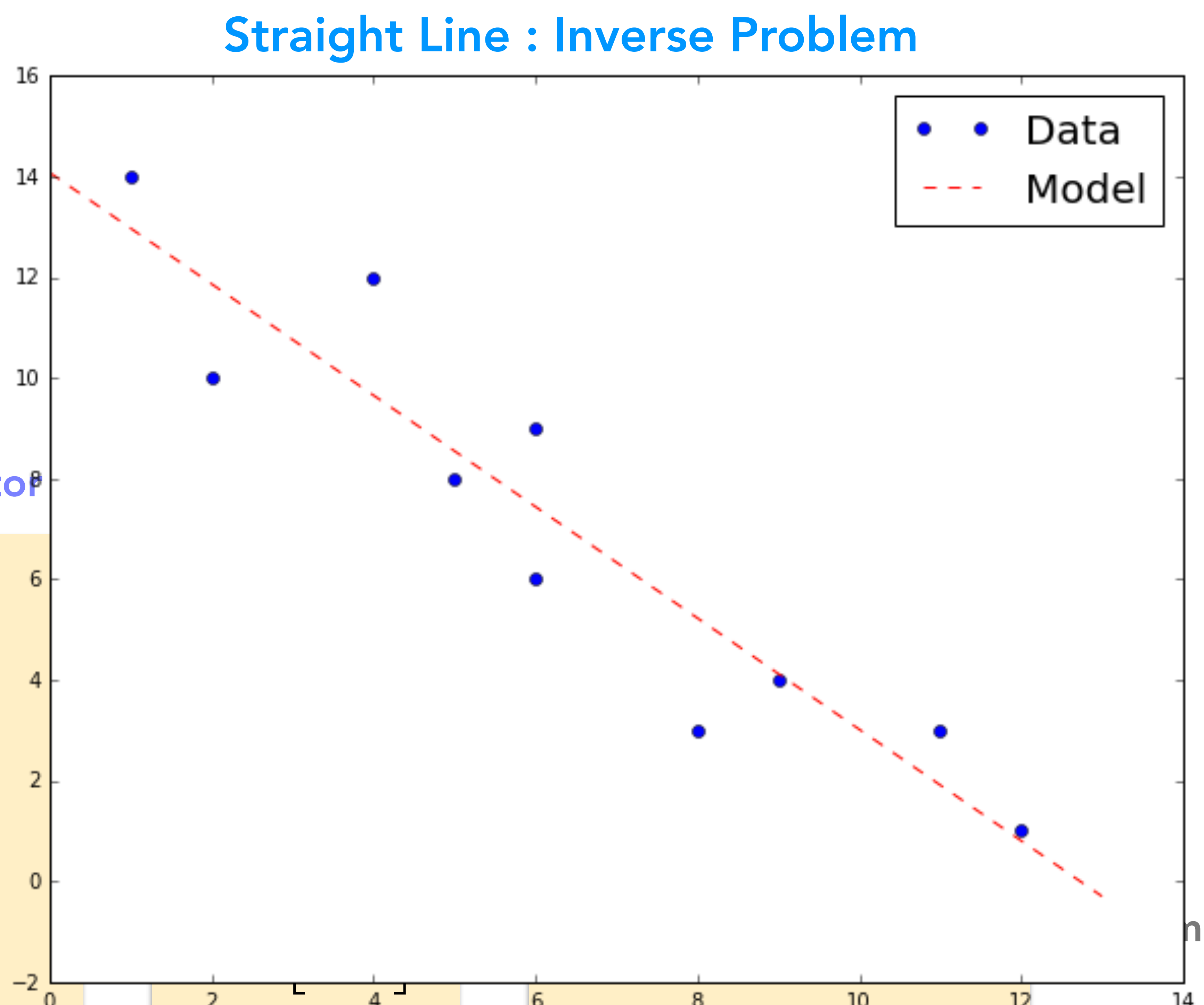
$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$

$$\downarrow$$

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{y}$$

(Normal Equation)

Linear Regression



Data Vector

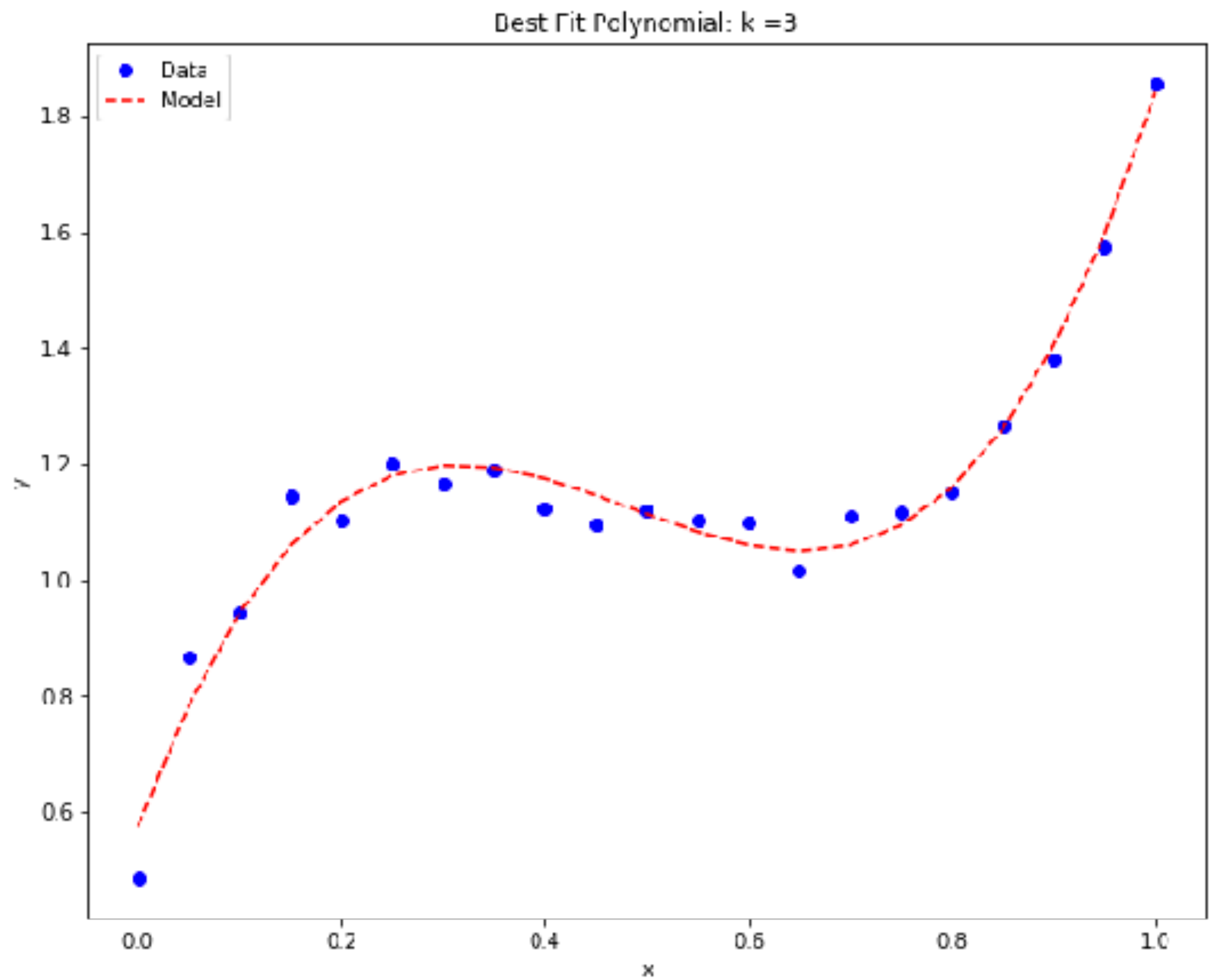
$$\mathbf{y} = \begin{bmatrix} 3 \\ 10 \\ 3 \\ 6 \\ 8 \\ 12 \\ 1 \\ 4 \\ 9 \\ 14 \end{bmatrix}$$

nal Equation)

Linear Regression

Polynomial Line : Inverse Problem

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_k x^k$$



Ill-Posed Problem

Well-Posed Problem

1. A solution exists
2. The solution is unique
3. The solution's behaviour changes continuously with the initial conditions

Ill-Posed Problem

1. No solution exists
2. The solution is not unique
3. The problem is ill-conditioned

Ill-Posed Problem

Well-Conditioned Problem

$$\begin{array}{ccc} \mathbf{y} & \mathbf{A} & \mathbf{x} \\ \begin{bmatrix} 4 \\ 7 \end{bmatrix} = & \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} & \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{array} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 4 \\ 7 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2.01 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.96 \\ 1.02 \end{bmatrix}$$

Ill-Conditioned Problem

$$\begin{bmatrix} 3 \\ 1.47 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0.48 & 0.99 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 3 \\ 1.47 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0.49 & 0.99 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Outline

- **Inverse Problems**
  - Linear Regression
  - Ill-posed Problems
- **Regularisation**
  - Sparsity
  - Low-Rank Approximation
- **Deconvolution of Galaxy Images**
- **Summary**

Regularisation

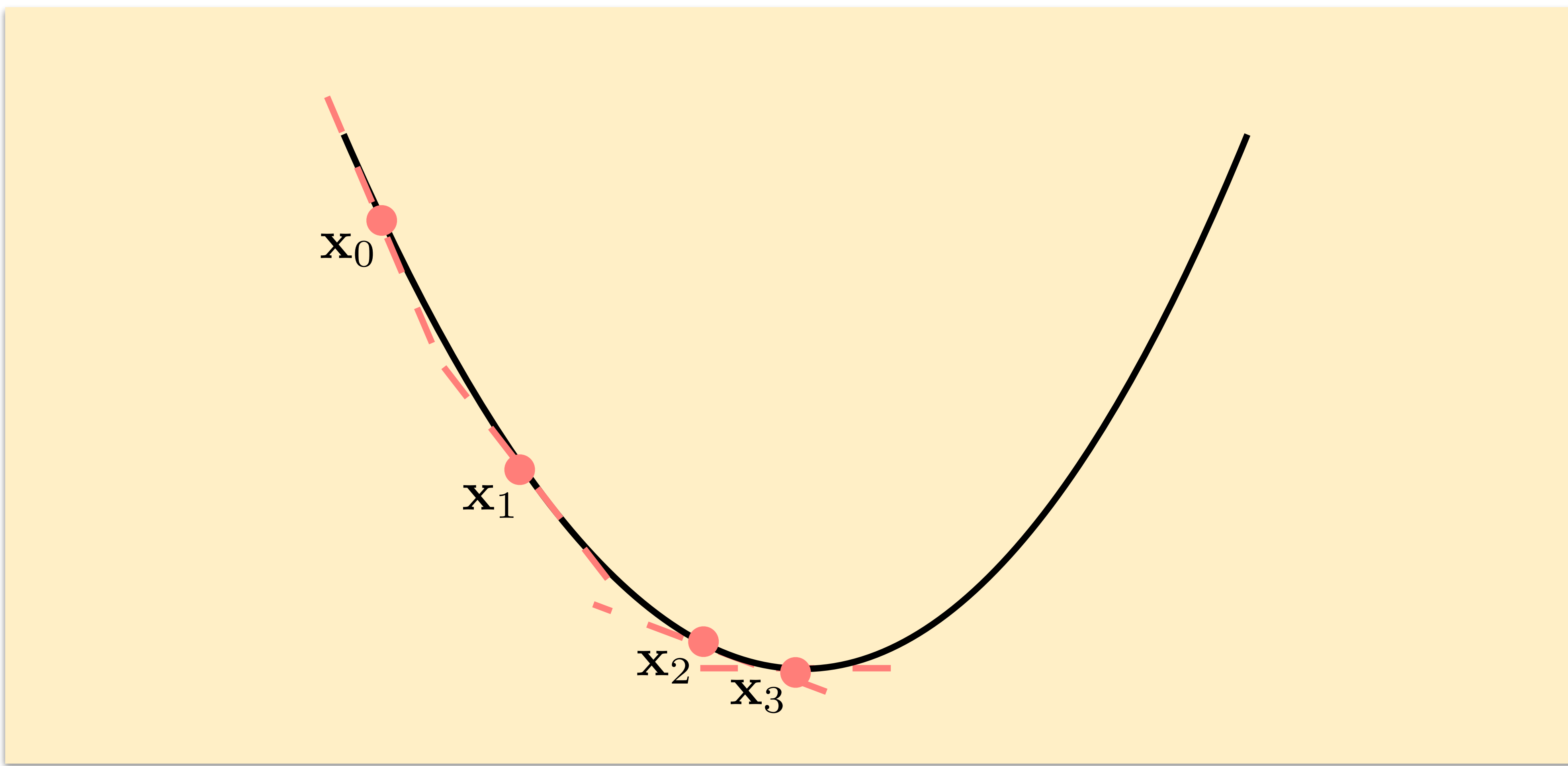
$$\underset{\mathbf{x}}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda R(\mathbf{x})$$

1. Find **x** such that **y-Ax** is small
2. We have some prior knowledge about the properties of **x** given by R(**x**)

Regularisation

$$F(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$$

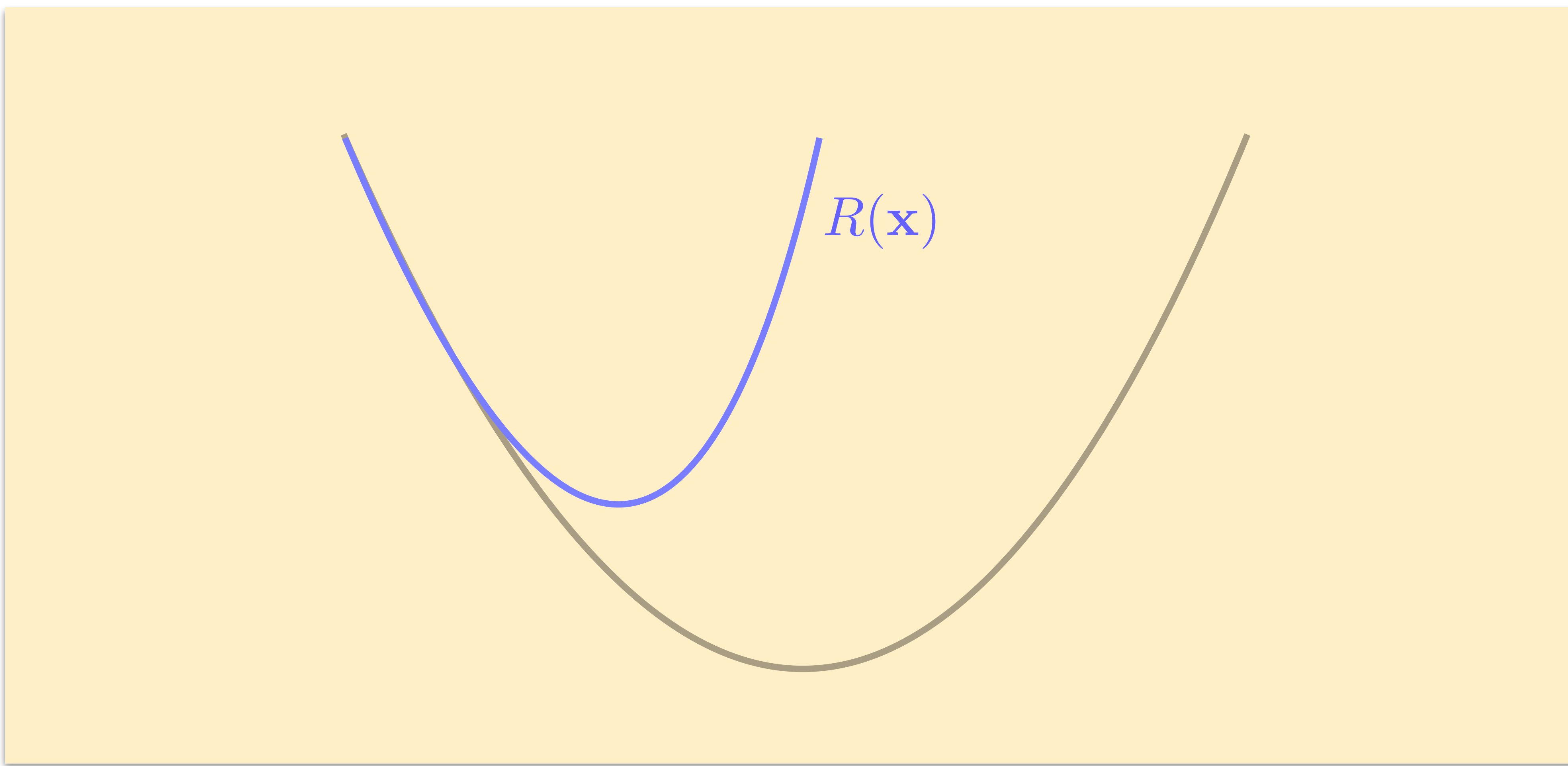
$$\nabla F(\mathbf{x}) = \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x})$$



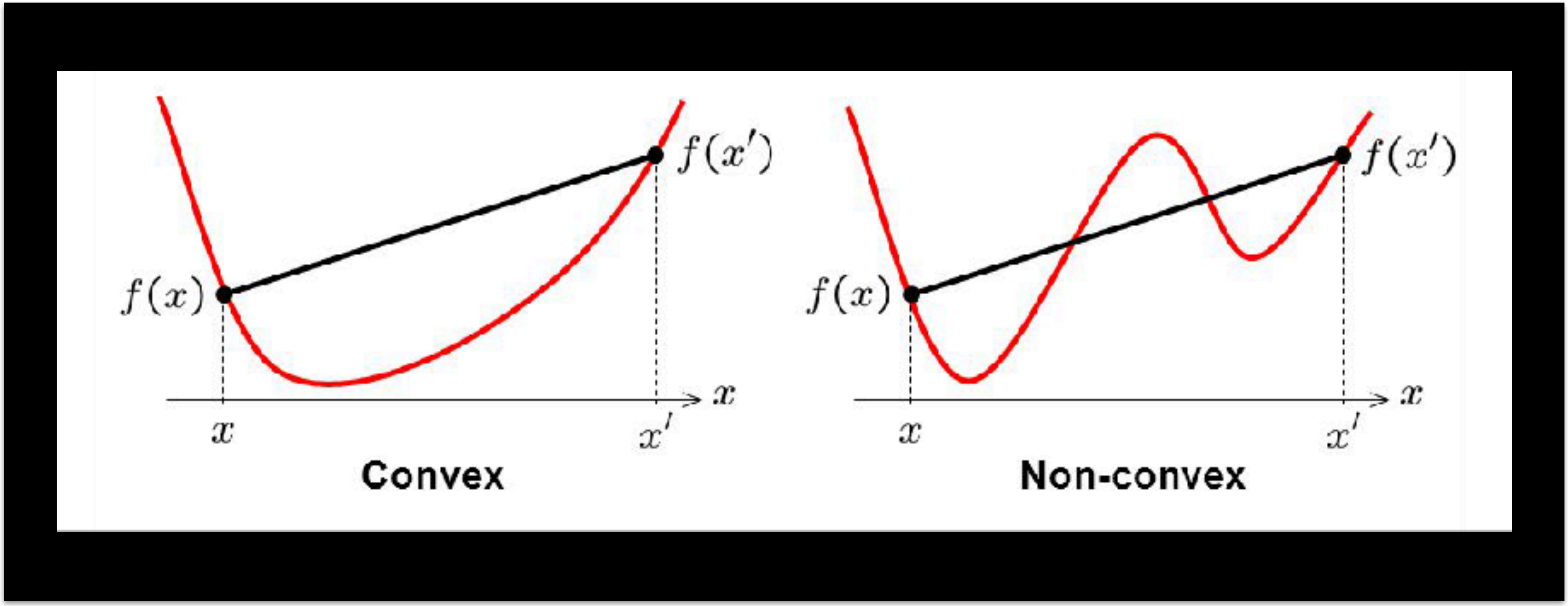
Regularisation

$$F(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$$

$$\nabla F(\mathbf{x}) = \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x})$$



Convexity

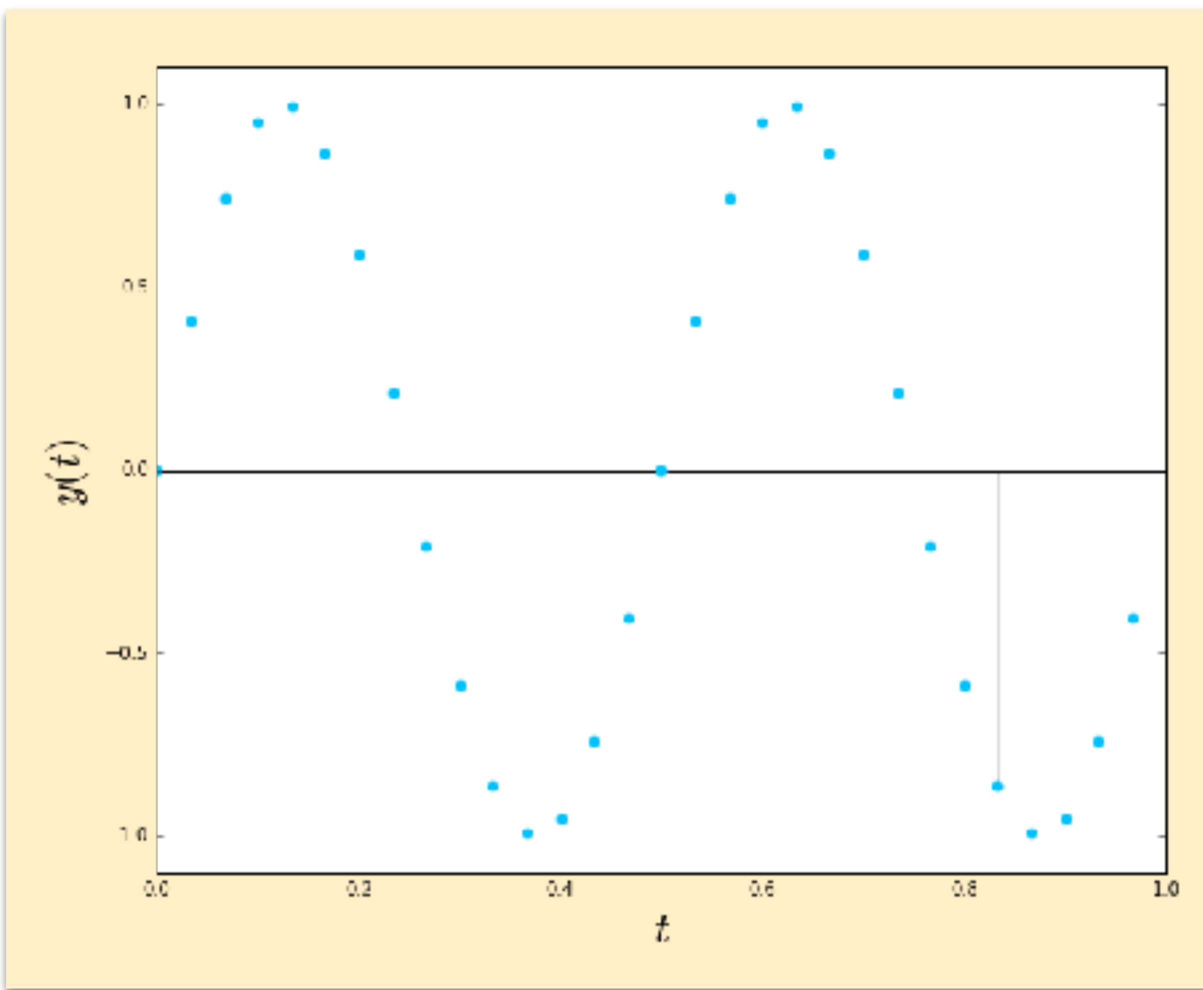


In general we want to preserve convexity

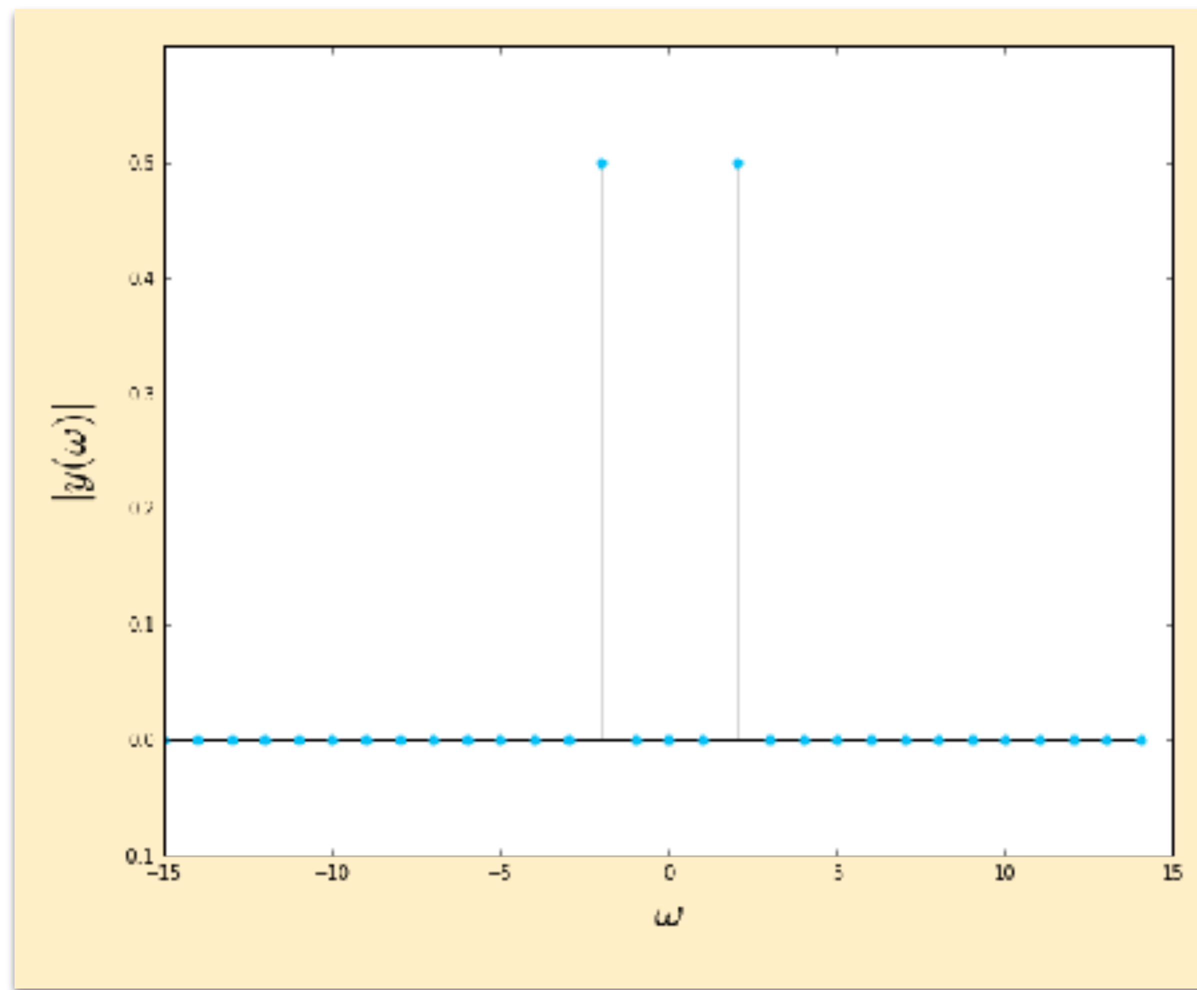
Sparsity

A sparse signal is one that is comprised mostly of zeros when expressed in the appropriate basis.

Direct Space



Sparse Space



Sparsity

$$\mathbf{x} = \phi\alpha = \sum_{i=1}^{n} \phi_i \alpha_i$$

ϕ is the dictionary that converts the signal to a sparse representation. (*e.g.* Fourier transform, wavelet transform, *etc.*)

Measuring Sparsity

$$||\alpha||_0 \rightarrow ||\alpha||_1 = \sum_{i=1}^{n} |\alpha_i|$$

Not convex

Compressive Sensing Theorem

This theorem demonstrates that, under certain conditions regarding the signal and the operation matrix, a perfect reconstruction can be achieved through l_1 minimisation.

No such theorem exists for any other regularisation technique.

Sparsity

Sparse Minimisation

$$\underset{\alpha}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{y} - A\phi\alpha\|_2^2 + \lambda\|\alpha\|_1$$

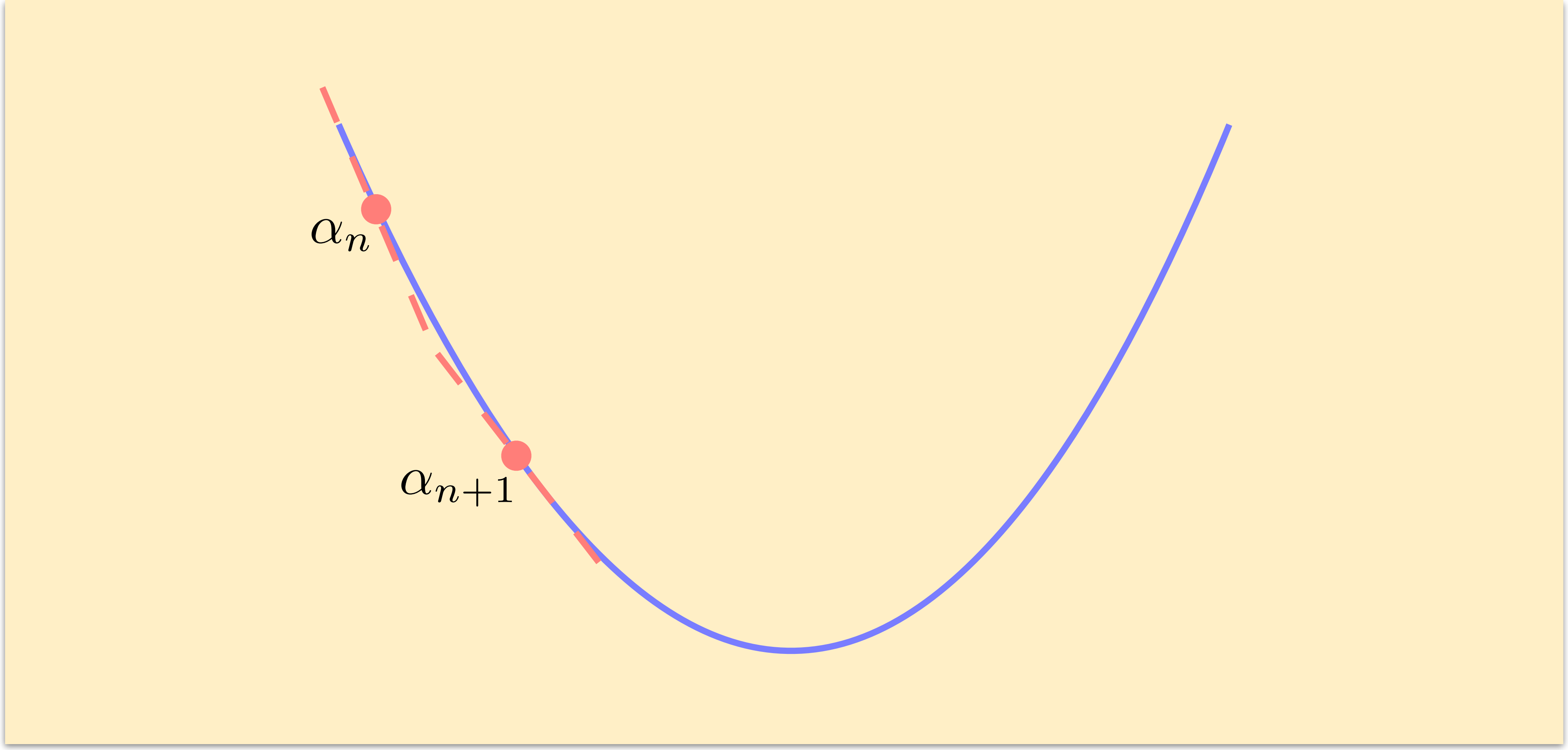
Applications

- Denoising
- **Deconvolution**
- Component Separation
- Inpainting
- Blind Source Separation
- Minimisation algorithms
- Compressed Sensing

Sparsity

Implementation

$$\alpha_{n+1} = \mathrm{ST}_\lambda(\alpha_n - \nabla F(\alpha_n))$$



Sparsity

Soft Threshold

$$ST_\lambda(\mathbf{x}_i) = \begin{cases} \mathbf{x}_i - \lambda \mathrm{sign}(\mathbf{x}_i) & \text{if } |\mathbf{x}_i| \geq \lambda \\ 0 & \text{otherwise} \end{cases}$$

Soft Thresholding of Sparse Coefficients

$$\alpha = \begin{bmatrix} 3 & 0 & 8 \\ 7 & 7 & 1 \\ 2 & 5 & 3 \end{bmatrix} \rightarrow \mathrm{ST}_4(\alpha) = \begin{bmatrix} 0 & 0 & 4 \\ 3 & 3 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Low-Rank Approximation

Rank of a Matrix

The rank of a matrix can be defined in the following ways:

1. the maximum number of linearly independent column vectors in a given matrix
2. the maximum number of linearly independent row vectors in a given matrix

Both of these definitions are equivalent.

$$M = \begin{bmatrix} 1 & 0 & 1 \\ -2 & -3 & 1 \\ 3 & 3 & 0 \end{bmatrix}$$

$$\text{rank}(M) = 2$$

Low-Rank Approximation

Singular Value Decomposition

$$M = U\Sigma V^T$$

$$\Sigma = \begin{bmatrix} \sigma_0 & 0 & \cdots & 0 \\ 0 & \sigma_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix}$$

Measuring Rank

$$\mathrm{rank}(M) = \|\Sigma\|_0 \rightarrow \|M\|_* = \sum_{k=1} \sigma_k(M)$$

Not convex

Low-Rank Approximation

Low-Rank Minimisation

$$\underset{\mathbf{x}}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_*$$

The nuclear norm term can be implemented by preforming a hard thresholding on the singular values of $\mathbf{x}$.

Low-Rank Approximation

Hard Threshold

$$HT_\lambda(\mathbf{x}_i) = \begin{cases} \mathbf{x}_i & \text{if } |\mathbf{x}_i| \geq \lambda \\ 0 & \text{otherwise} \end{cases}$$

Hard Thresholding of Singular Values

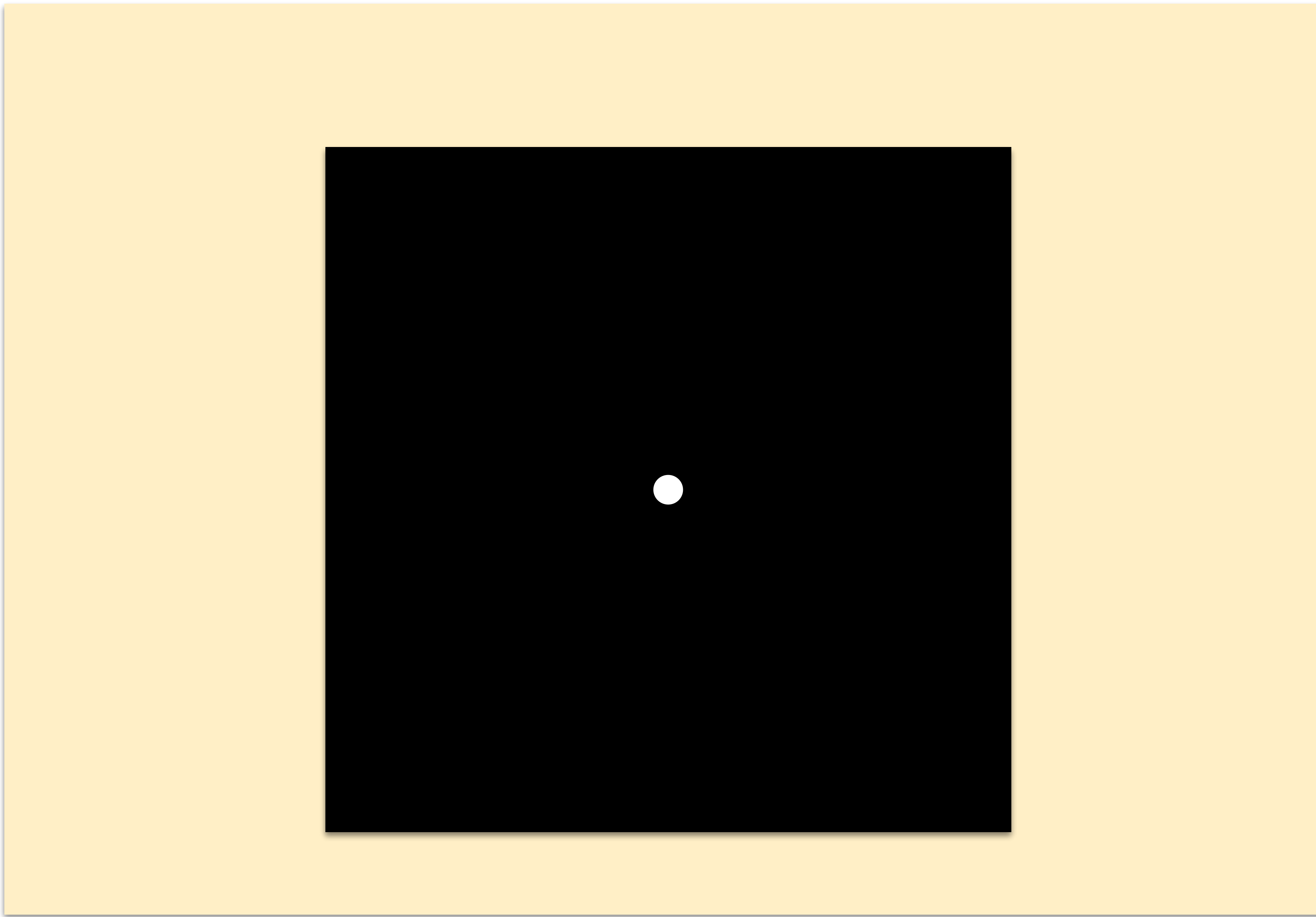
$$\Sigma = \begin{bmatrix} 12 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 3 \end{bmatrix} \rightarrow \mathrm{HT}_4(\Sigma) = \begin{bmatrix} 12 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Outline

- **Inverse Problems**
  - Linear Regression
  - Ill-posed Problems
- **Regularisation**
  - Sparsity
  - Low-Rank Approximation
- **Deconvolution of Galaxy Images**
- **Summary**

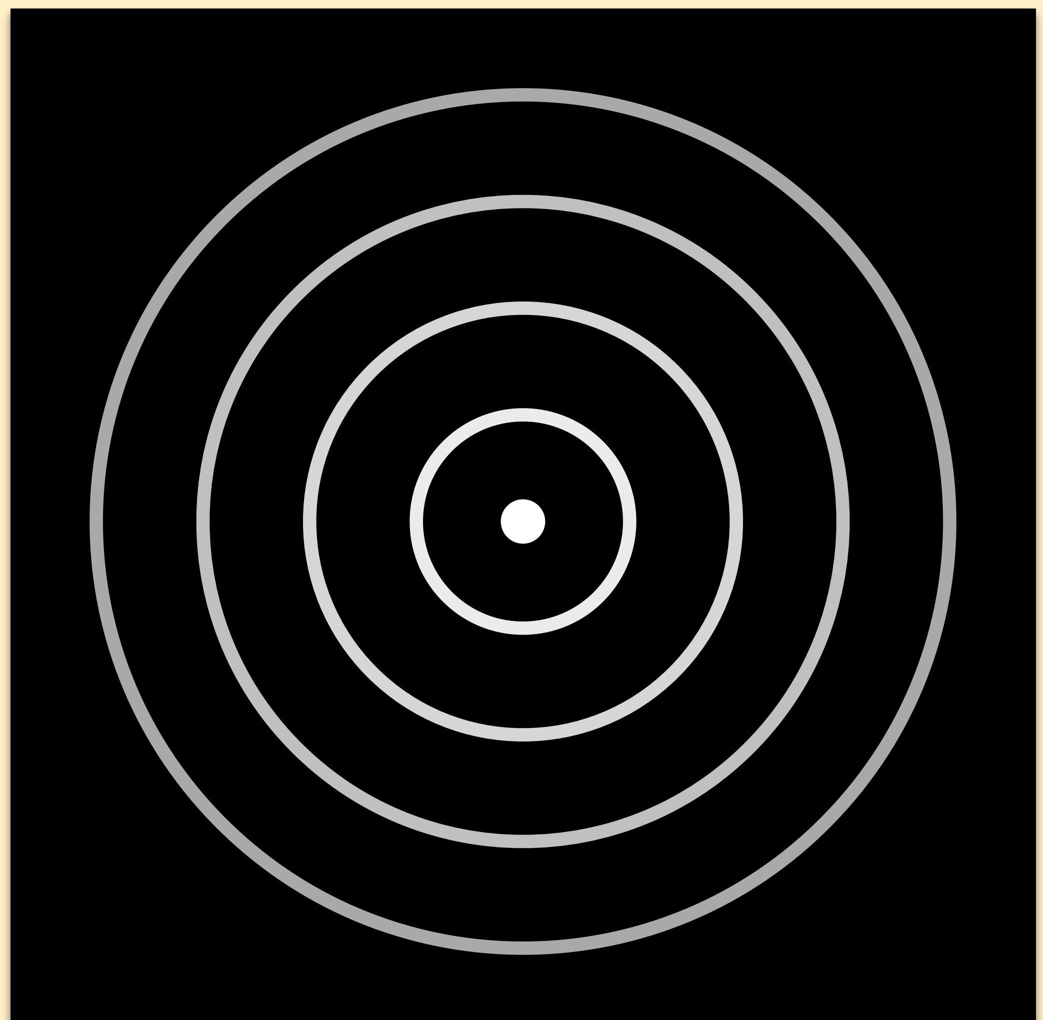
Background

Point Spread Function



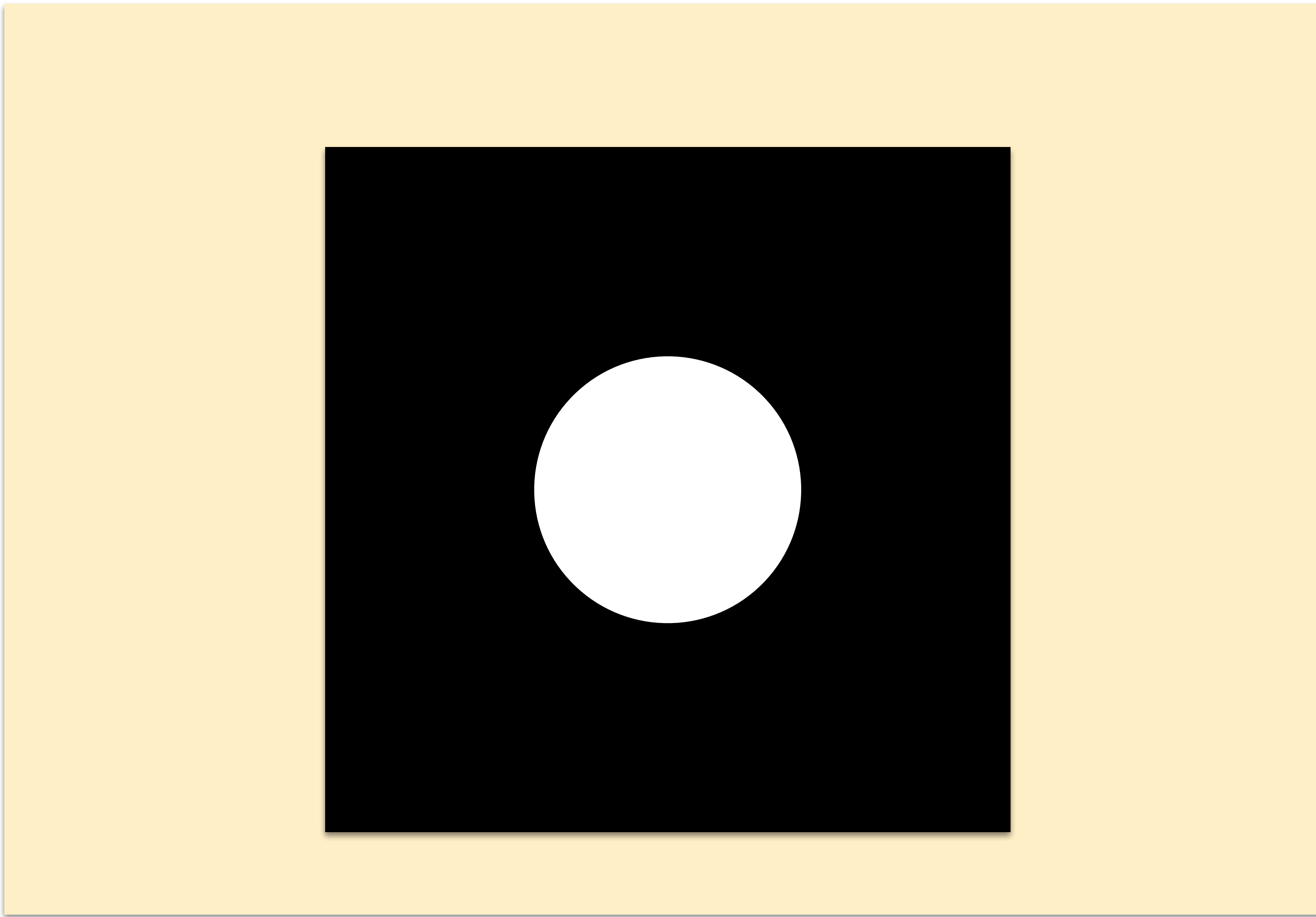
Background

Point Spread Function

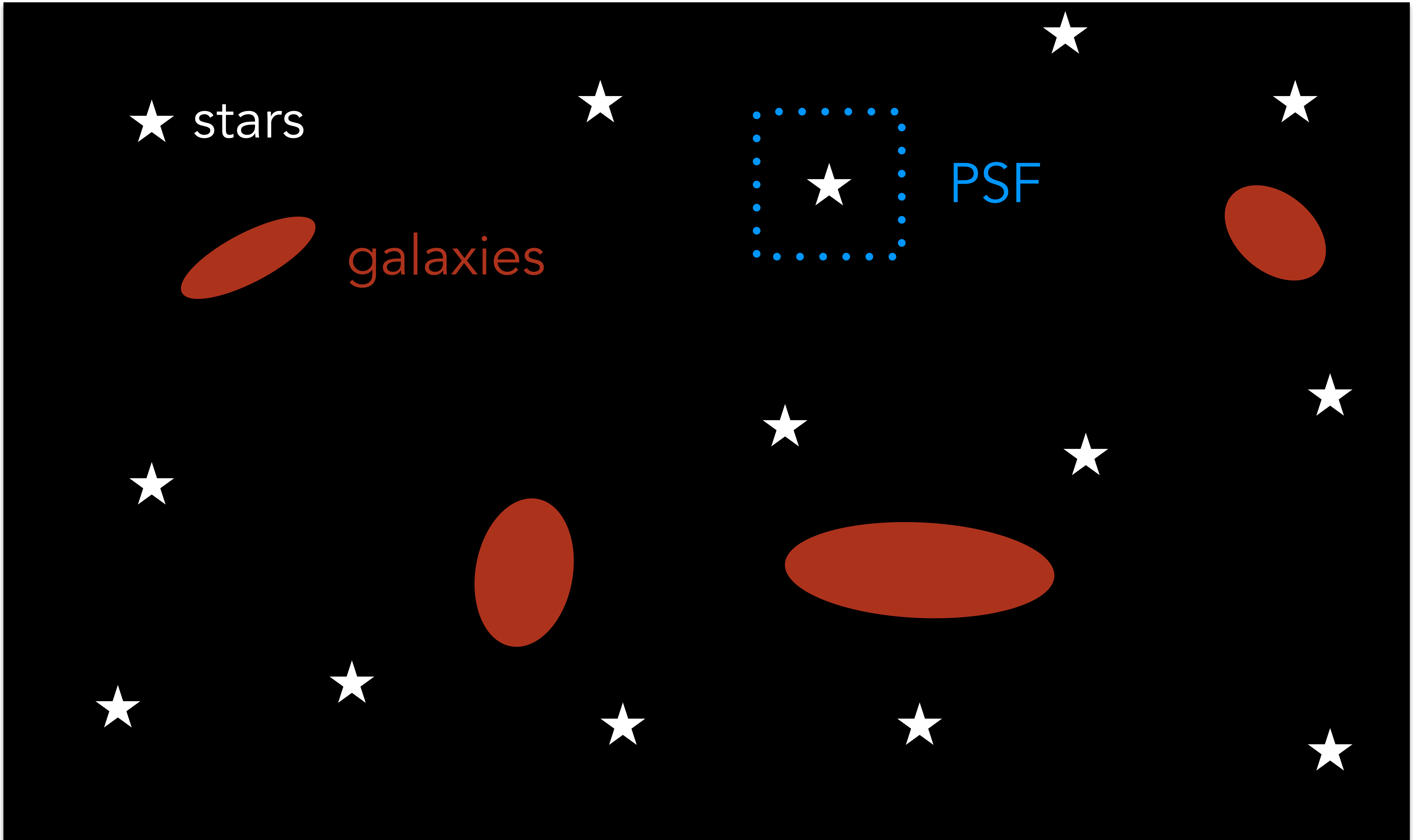


Background

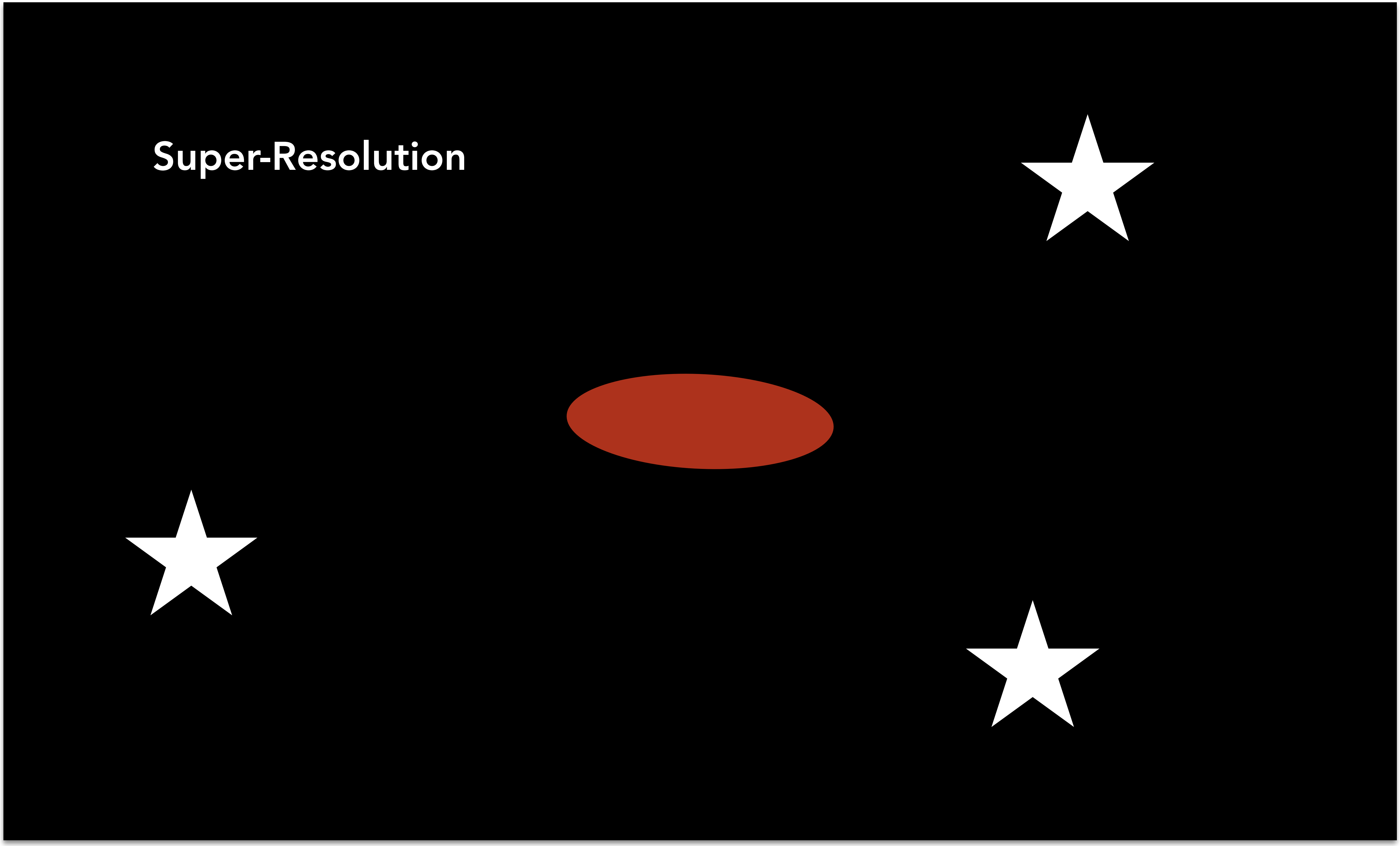
Point Spread Function



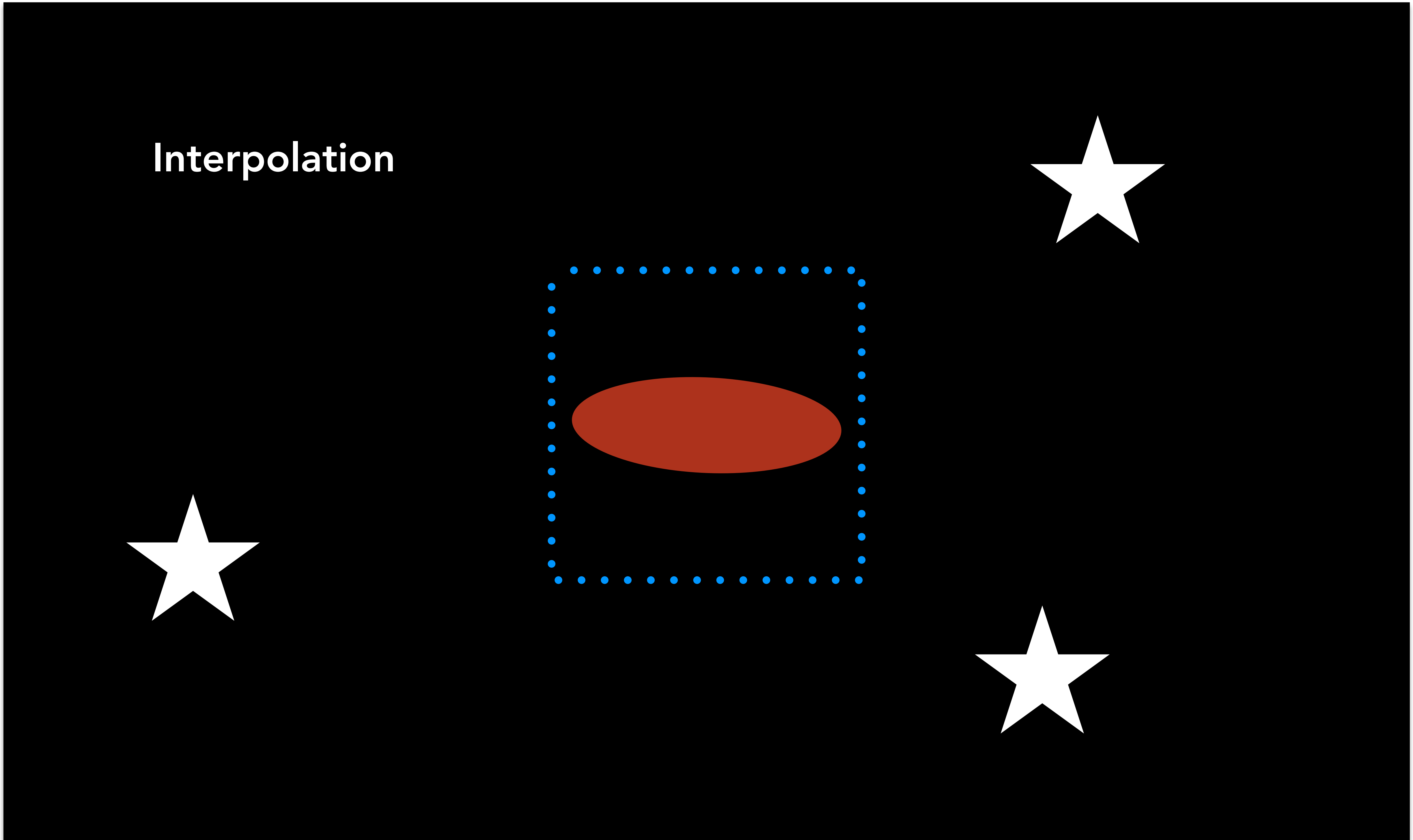
Background



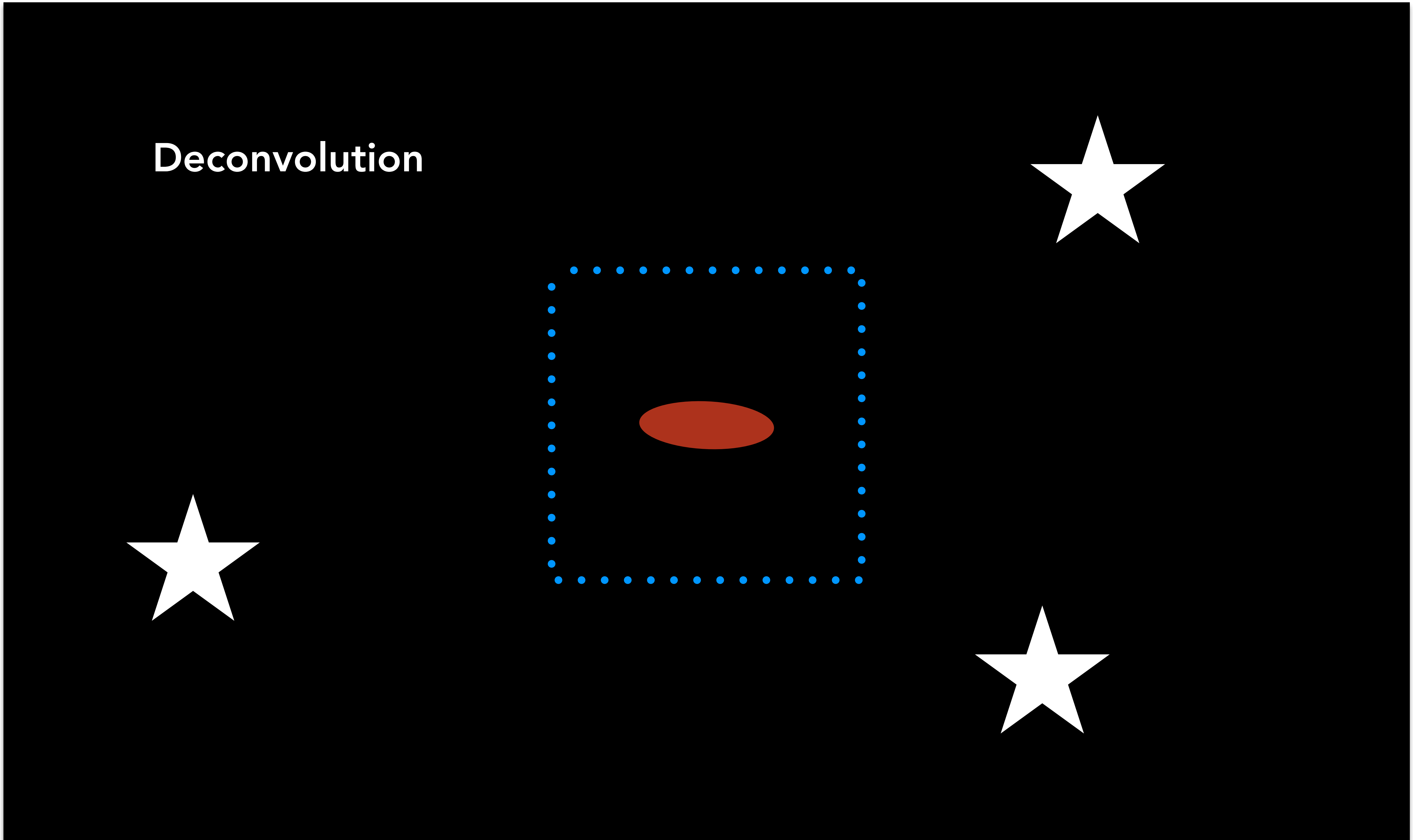
Background



Background

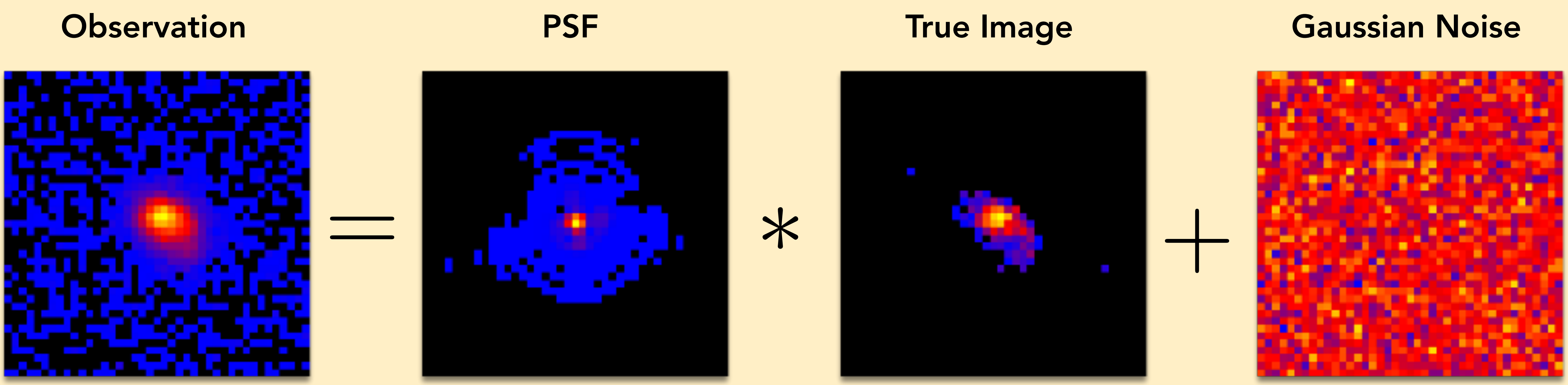


Background



Problem

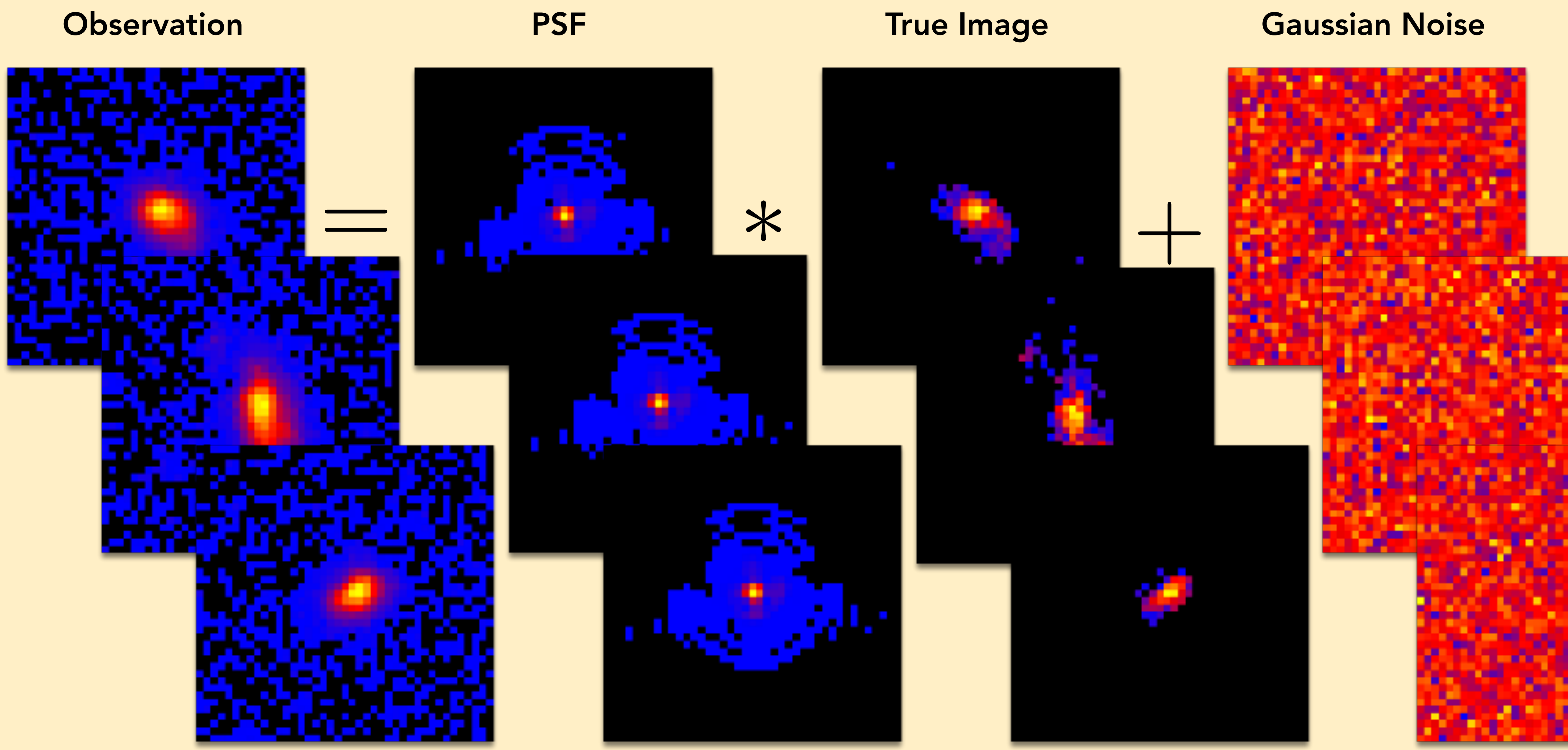
$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$



The problem is ill-posed

Problem

$$\mathbf{Y} = \mathcal{H}(\mathbf{X}) + \mathbf{N}$$



Problem

$$\mathbf{Y} = \mathcal{H}(\mathbf{X}) + \mathbf{N}$$

$$\mathbf{Y} = [\mathbf{y}^0, \mathbf{y}^1, \ldots, \mathbf{y}^n]$$

$$\mathbf{X} = [\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^n]$$

$$\mathbf{N} = [\mathbf{n}^0, \mathbf{n}^1, \ldots, \mathbf{n}^n]$$

$$\mathcal{H}(\mathbf{X}) = [\mathbf{H}^0\mathbf{x}^0, \mathbf{H}^1\mathbf{x}^1, \ldots, \mathbf{H}^n\mathbf{x}^n]$$

Regularisation

$$\underset{\mathbf{X}}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{Y} - \mathcal{H}(\mathbf{X})\|_2^2$$

Positivity Constraint

$$\text{s.t.} \quad \mathbf{X} \geq 0,$$

Sparsity

$$+ \|\mathbf{W}^{(k)} \odot \Phi(\mathbf{X})\|_1$$

Low-Rank

$$+ \lambda\|\mathbf{X}\|_*$$

Sparsity

$$\underset{\mathbf{X}}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{Y} - \mathcal{H}(\mathbf{X})\|_2^2 + \|\mathbf{W}^{(k)} \odot \Phi(\mathbf{X})\|_1 \quad \text{s.t.} \quad \mathbf{X} \geq 0$$

Weights

$$\mathbf{W}_{:,i}^{(0)} = [\mathbf{t}^{i1T}, \cdots, \mathbf{t}^{iJT}]^T,$$

$$\mathbf{t}_m^{ij} = \kappa_j \sigma_i \|\mathbf{\Phi}_{m,:}^j \mathbf{H}^{iT}\|_2.$$

Re-Weighting

$$\mathbf{W}_{i,j}^{(k+1)} = \mathbf{W}_{i,j}^{(k)} \frac{1}{1 + \frac{|\Phi(\hat{\mathbf{X}}^{(k)})_{i,j}|}{\mathbf{W}_{i,j}^{(0)}}},$$

Candès et al. (2008)

Φ - Starlet transform (without the coarse scale)

Low-Rank

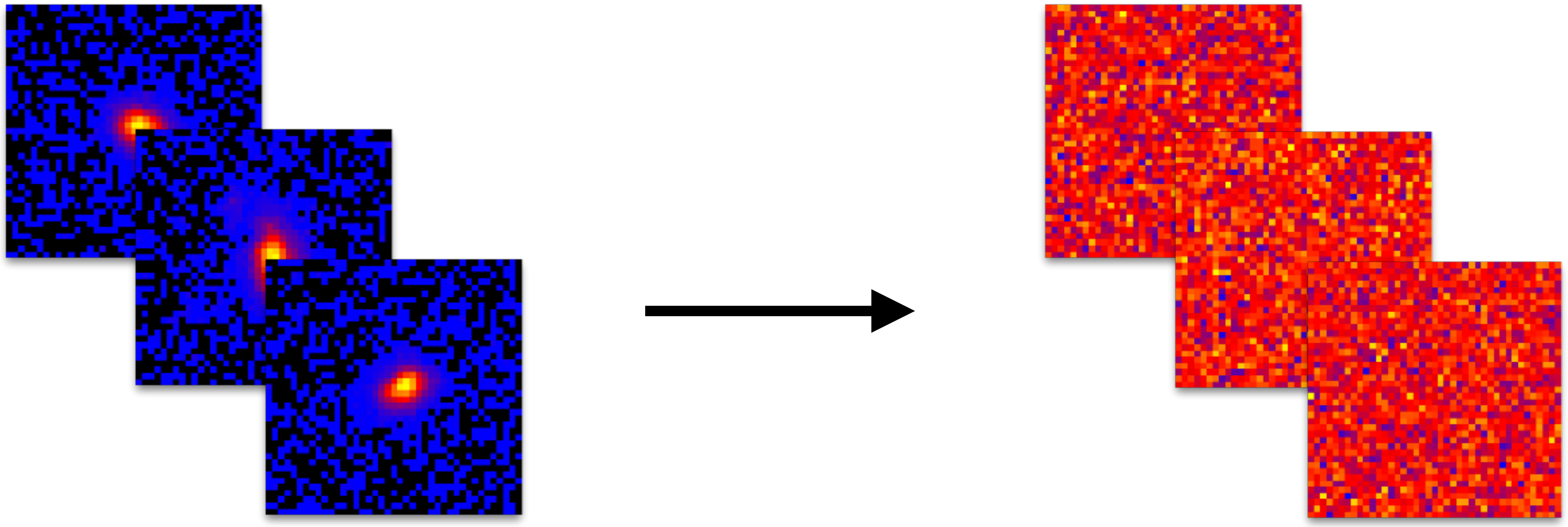
$$\underset{\mathbf{X}}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{Y} - \mathcal{H}(\mathbf{X})\|_2^2 + \lambda\|\mathbf{X}\|_* \quad \text{s.t.} \quad \mathbf{X} \geq 0$$

Threshold

$$\lambda = \alpha\sigma_{est}\sqrt{\max(n+1, p)}\rho(\mathcal{H}),$$

Noise

$$\sigma = 1.4826 \times \mathrm{MAD}(\mathbf{Y})$$



Median Absolute Deviation

$$\mathrm{MAD}((\mathbf{x}_i)_{1\leq i\leq l}) = \mathrm{median}((|\mathbf{x}_i - \mathrm{median}((\mathbf{x}_i)_{1\leq i\leq l})|)_{1\leq i\leq l})$$

Optimisation

Condat (2013) primal-dual splitting

$$1: \tilde{\mathbf{X}}_{k+1} = \mathrm{prox}_{\tau G}(\mathbf{X}_k - \tau\nabla F(\mathbf{X}_k) - \tau\mathcal{L}^*(\mathbf{Y}_k))$$

$$2: \tilde{\mathbf{Y}}_{k+1} = \mathbf{Y}_k + \varsigma\mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k) - \varsigma\mathrm{prox}_{K/\varsigma}\left(\frac{\mathbf{Y}_k}{\varsigma} + \mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k)\right)$$

$$3: (\mathbf{X}_{k+1}, \mathbf{Y}_{k+1}) := \xi(\tilde{\mathbf{X}}_{k+1}, \tilde{\mathbf{Y}}_{k+1}) + (1-\xi)(\mathbf{X}_k, \mathbf{Y}_k)$$

- Primal proximity operator is always the positivity constraint
- Dual proximity operator is either a soft thresholding in Starlet space or a hard thresholding of the singular values

$\mathcal{L}$ - Linear operator (wavelet transform or identity)

Data

Galaxy Images

- 10,000 (space-based) galaxy images obtained from GREAT3
- 0.05 arcsec pixel scale (2× Euclid resolution, hence no aliasing issues)
- Each image is a 41×41 postage stamp
- The data set is well suited for studying Euclid like images as:
  - The ACS PSF can be neglected,
  - the intrinsic noise can easily be removed
  - and it is derived from high-resolution space-based images.

Euclid-like PSFs

- 600 unique PSFs corresponding to different positions across the four CCD chips of the Euclid VIS instrument
- Each PSF has 12× Euclid resolution
- Down-sampled to match galaxy images

Data

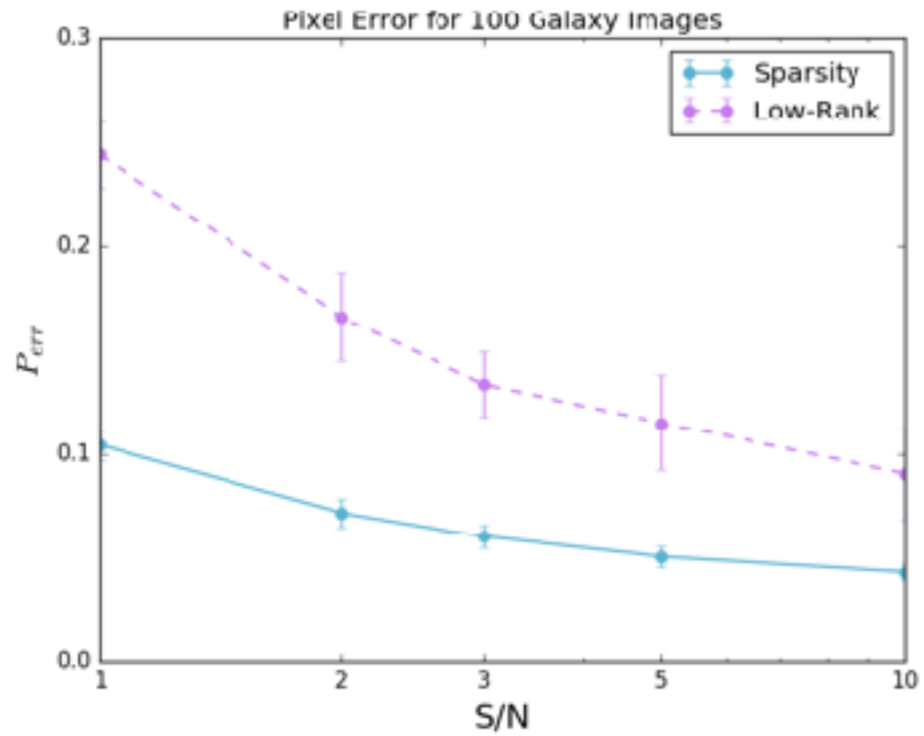
Euclid PSF

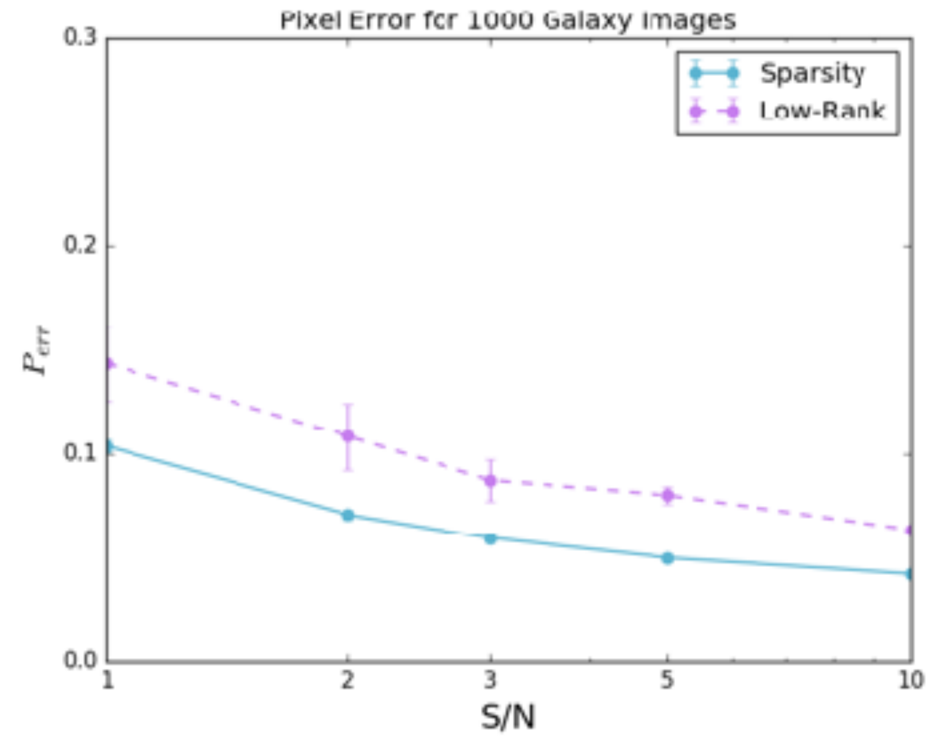
The PSF of the Euclid VIS instrument will be a combination of

- the **instrument optics**, which will introduce additional ellipticity to the galaxy shape measurements owing to aberrations and imperfections in the optical set-up,
- **jitter in the spacecraft pointing**, which will differ from exposure to exposure,
- the **charge spread of the instrument detector**, which will also add ellipticity to the galaxy shape measurements that are aligned with the pixel grid.

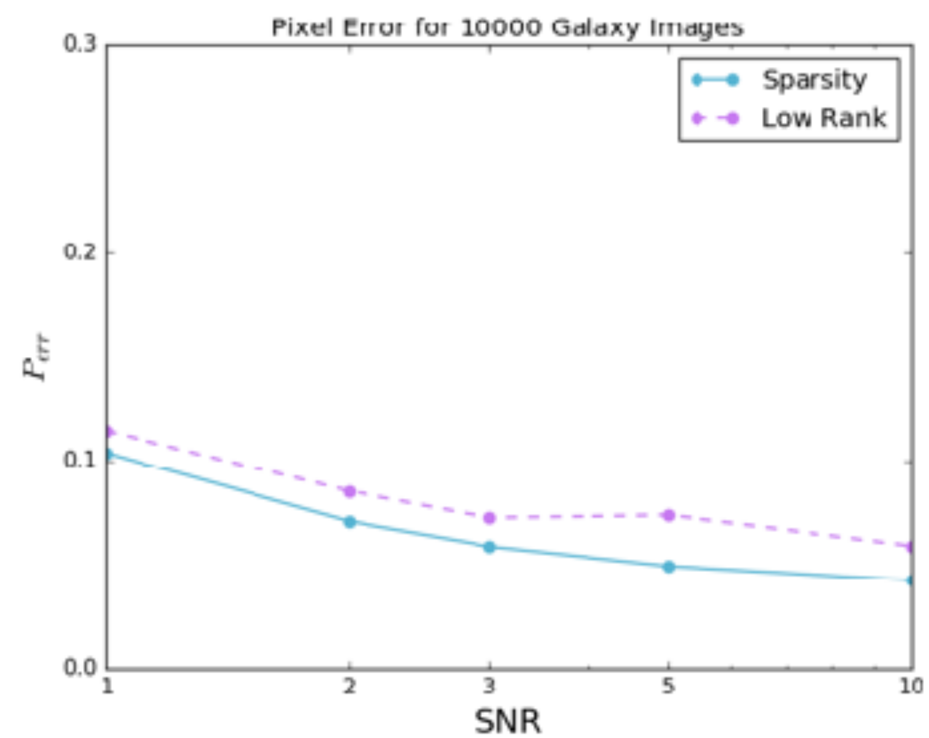
Results

Pixel Error



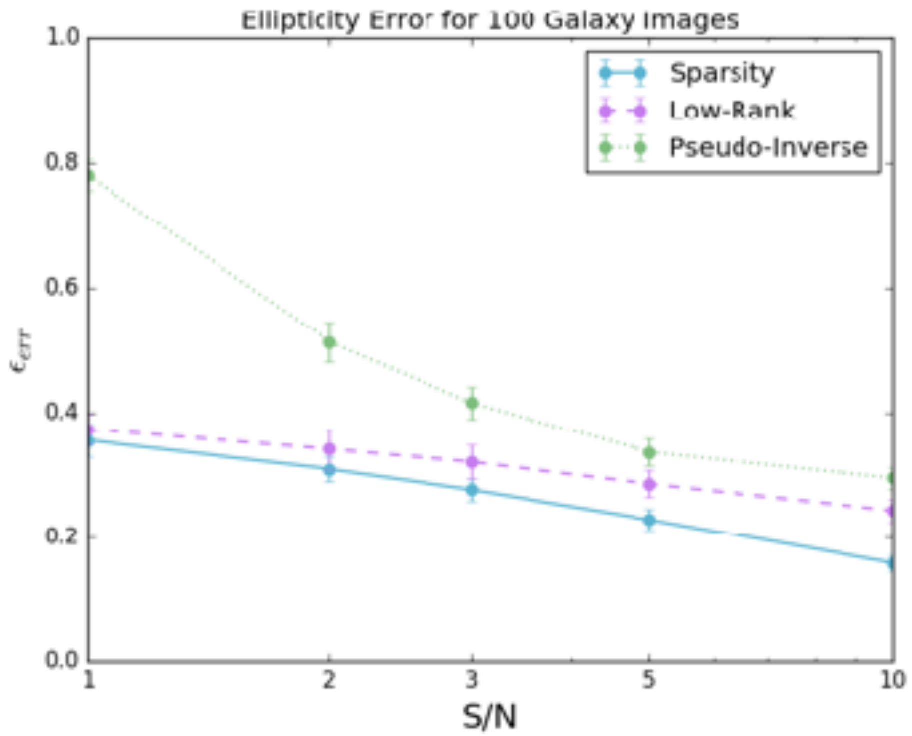


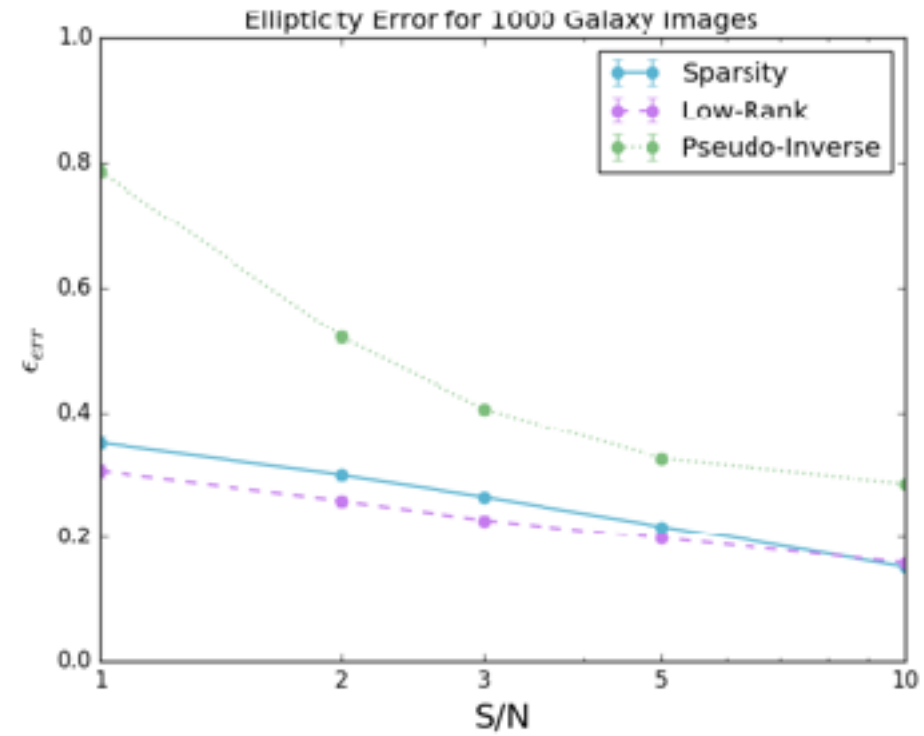
$$P_{err} = \text{median}\left(\frac{\|\mathbf{x}^i - \hat{\mathbf{x}}^i\|_2^2}{\|\mathbf{x}^i\|_2^2}\right)_{1\leq i\leq n}$$



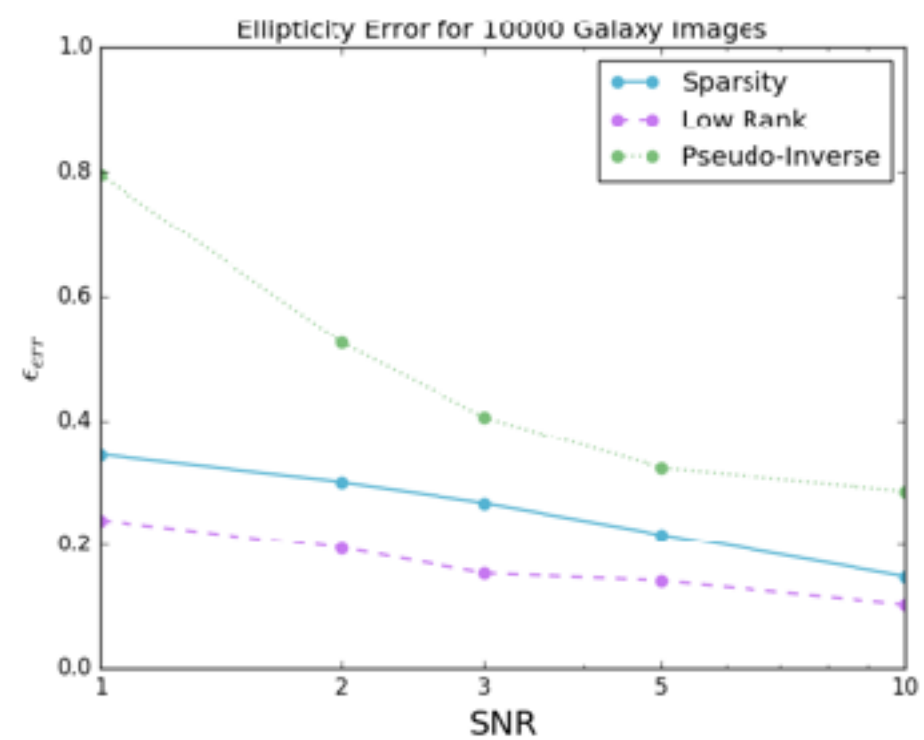
Results

Ellipticity Error

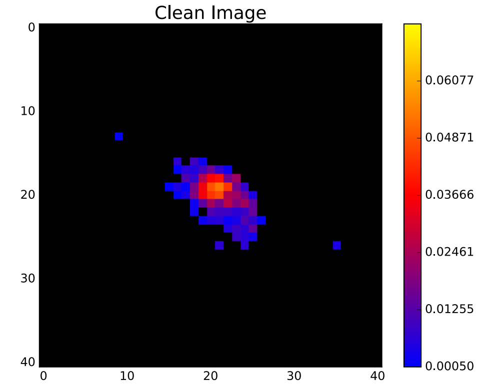




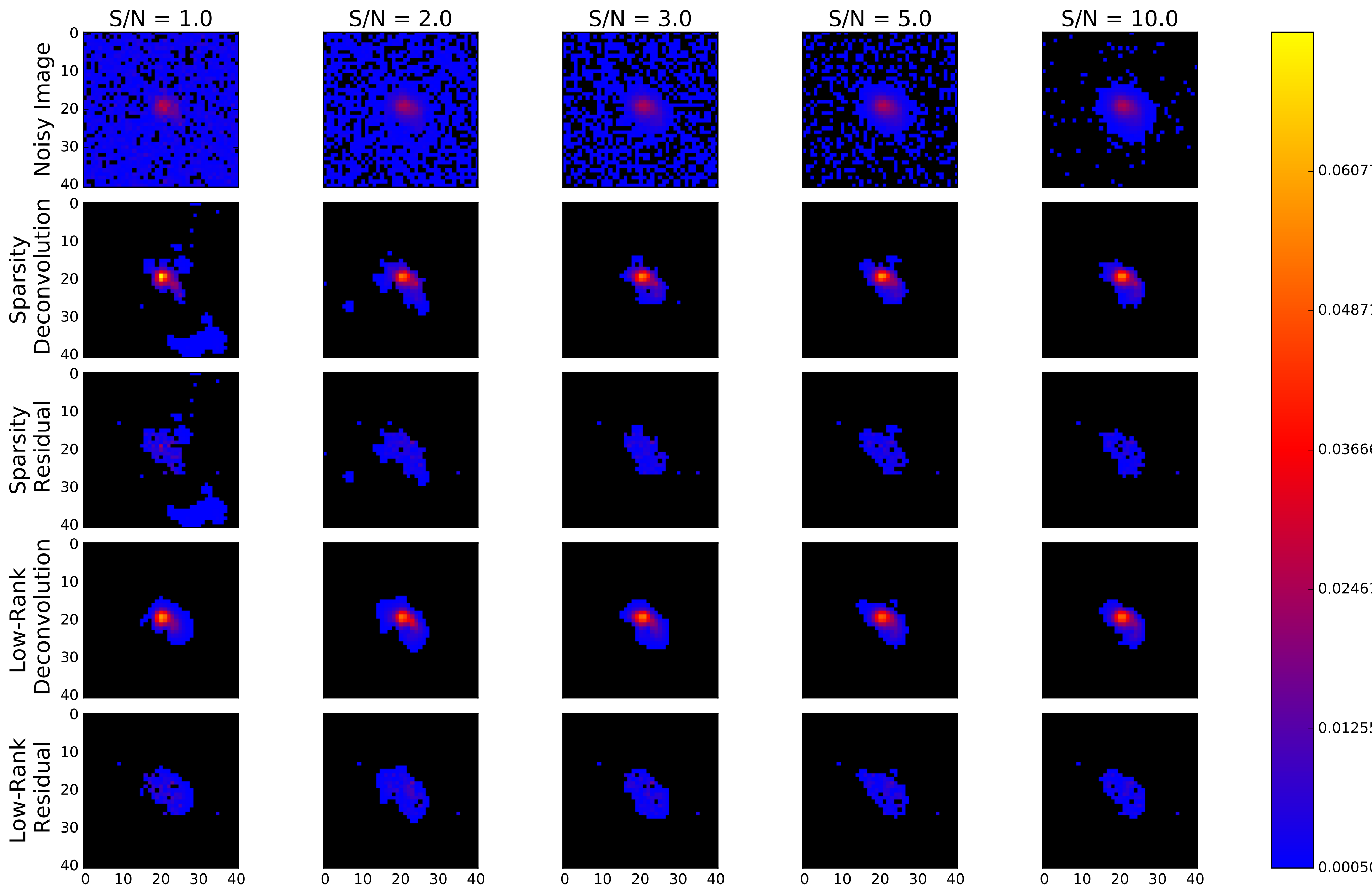
$$\varepsilon_{err} = \text{median}\left(\|\varepsilon(\mathbf{x}^i) - \varepsilon(\hat{\mathbf{x}}^i)\|_2\right)_{1\leq i\leq n}$$



Results

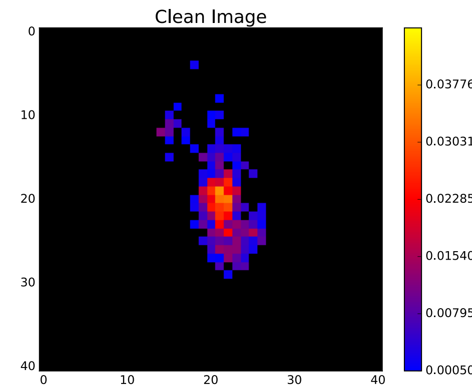


Clean Image
0
10
20
30
40
0
10
20
30
40
0.06077
0.04871
0.03666
0.02461
0.01255
0.00050

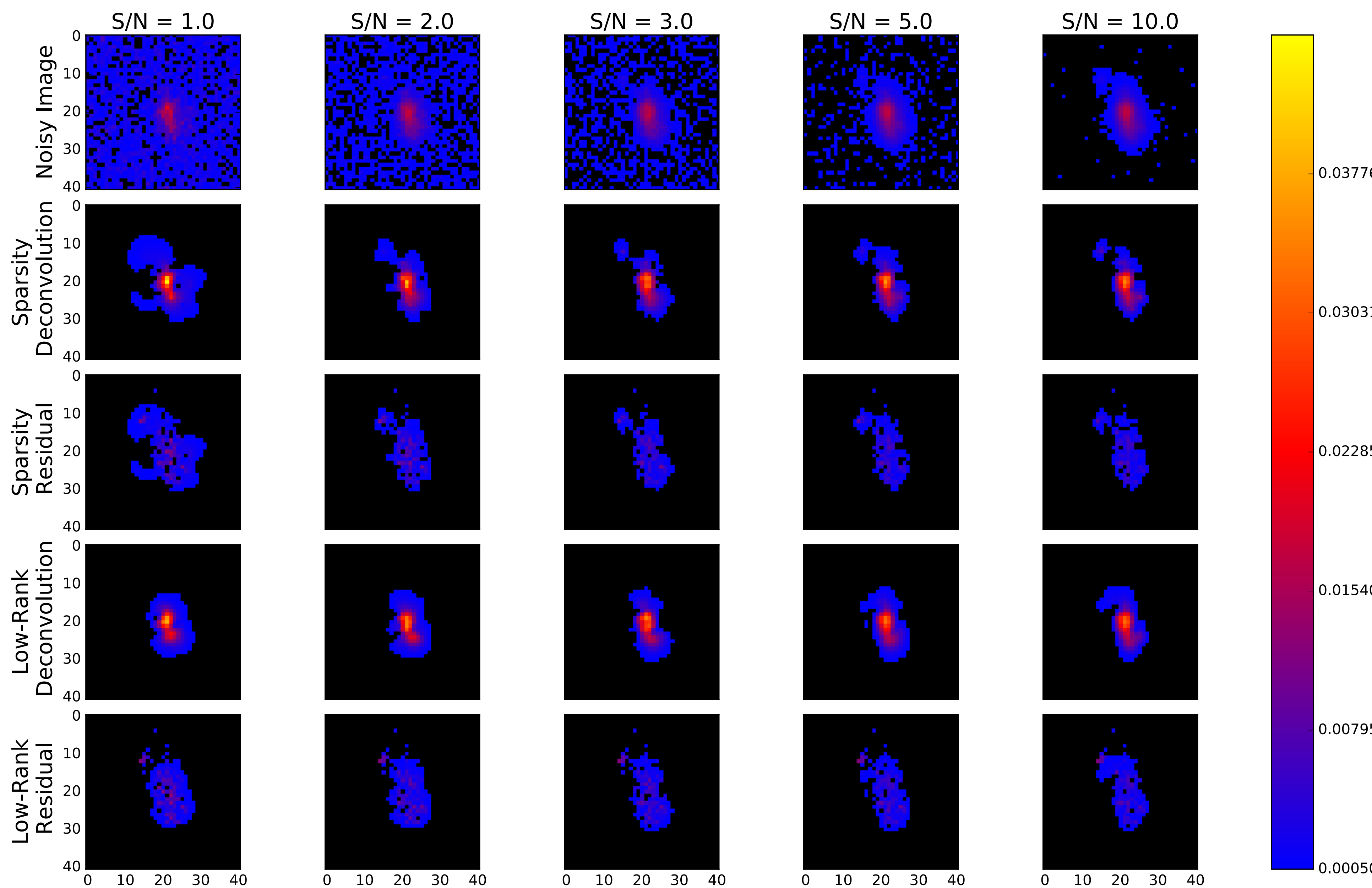


S/N = 1.0
S/N = 2.0
S/N = 3.0
S/N = 5.0
S/N = 10.0
Noisy Image
Sparsity Deconvolution
Sparsity Residual
Low-Rank Deconvolution
Low-Rank Residual
0
10
20
30
40
0.06077
0.04871
0.03666
0.02461
0.01255
0.00050

Results

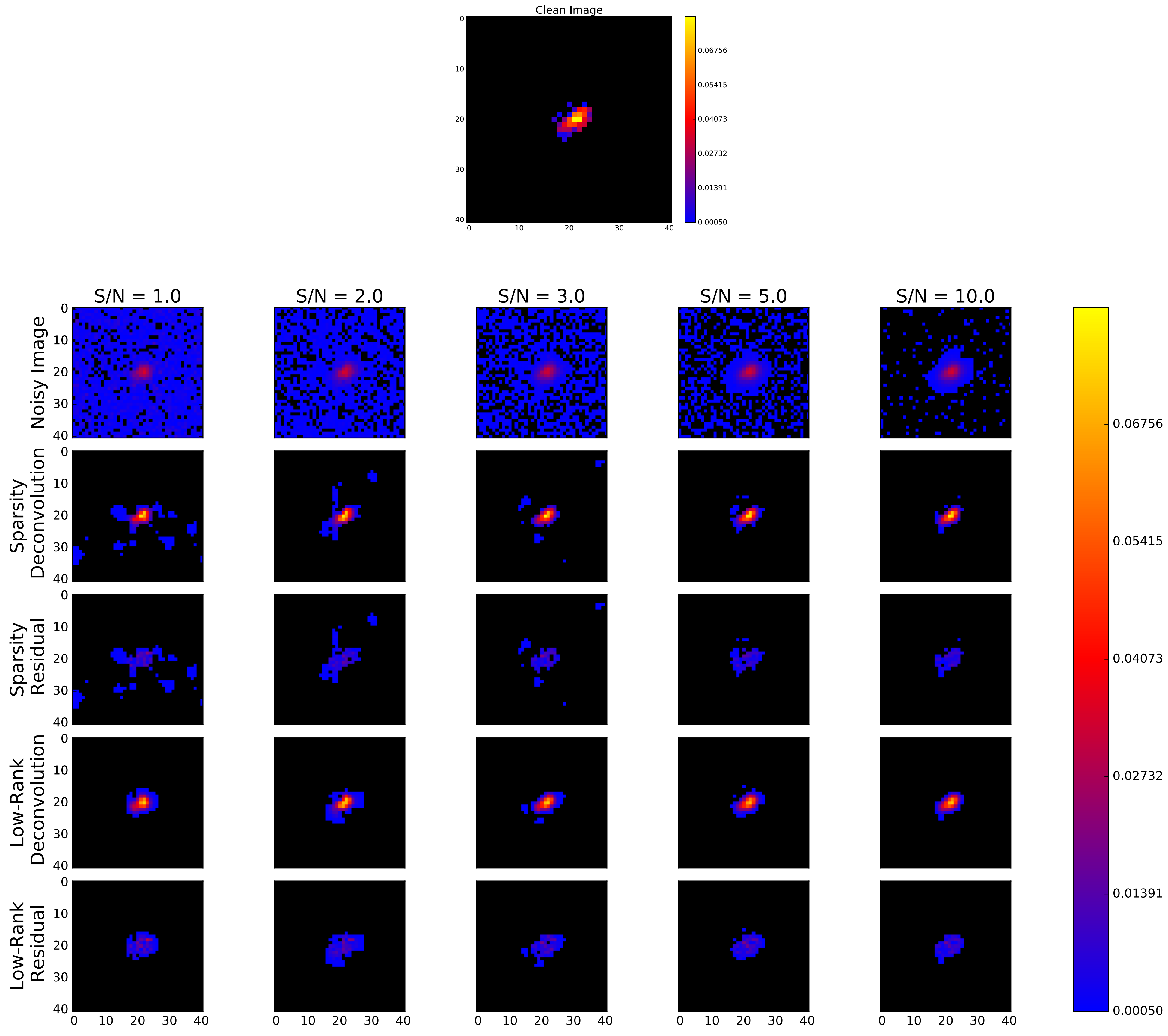


Clean Image
0.03776
0.03031
0.02285
0.01540
0.00795
0.00050



S/N = 1.0
S/N = 2.0
S/N = 3.0
S/N = 5.0
S/N = 10.0
Noisy Image
Sparsity Deconvolution
Sparsity Residual
Low-Rank Deconvolution
Low-Rank Residual
0.03776
0.03031
0.02285
0.01540
0.00795
0.00050

Results

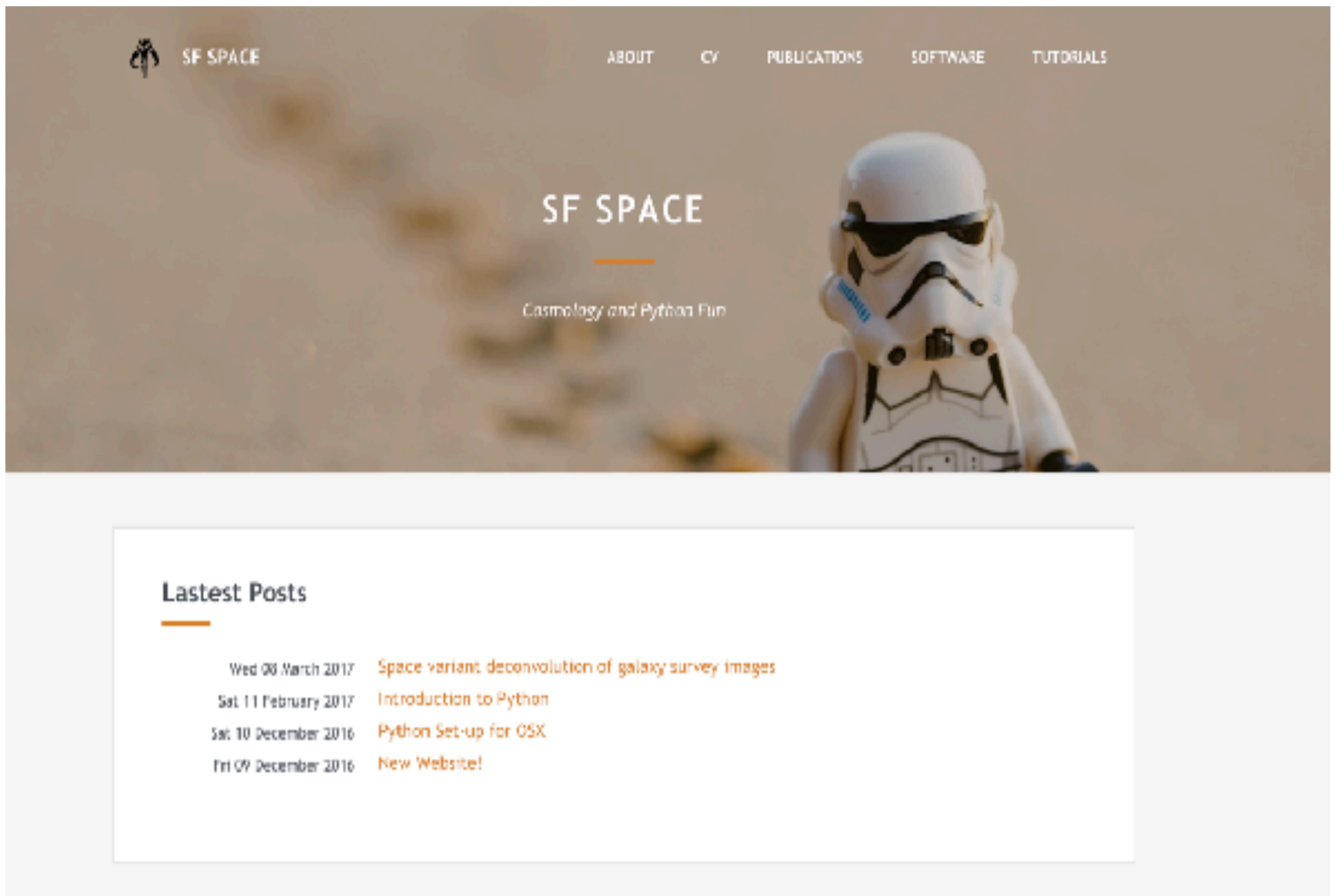

Clean Image
0.06756
0.05415
0.04073
0.02732
0.01391
0.00050
S/N = 1.0
S/N = 2.0
S/N = 3.0
S/N = 5.0
S/N = 10.0
Noisy Image
Sparsity Deconvolution
Sparsity Residual
Low-Rank Deconvolution
Low-Rank Residual

Outline

- **Inverse Problems**
  - Linear Regression
  - Ill-posed Problems
- **Regularisation**
  - Sparsity
  - Low-Rank Approximation
- **Deconvolution of Galaxy Images**
- **Summary**

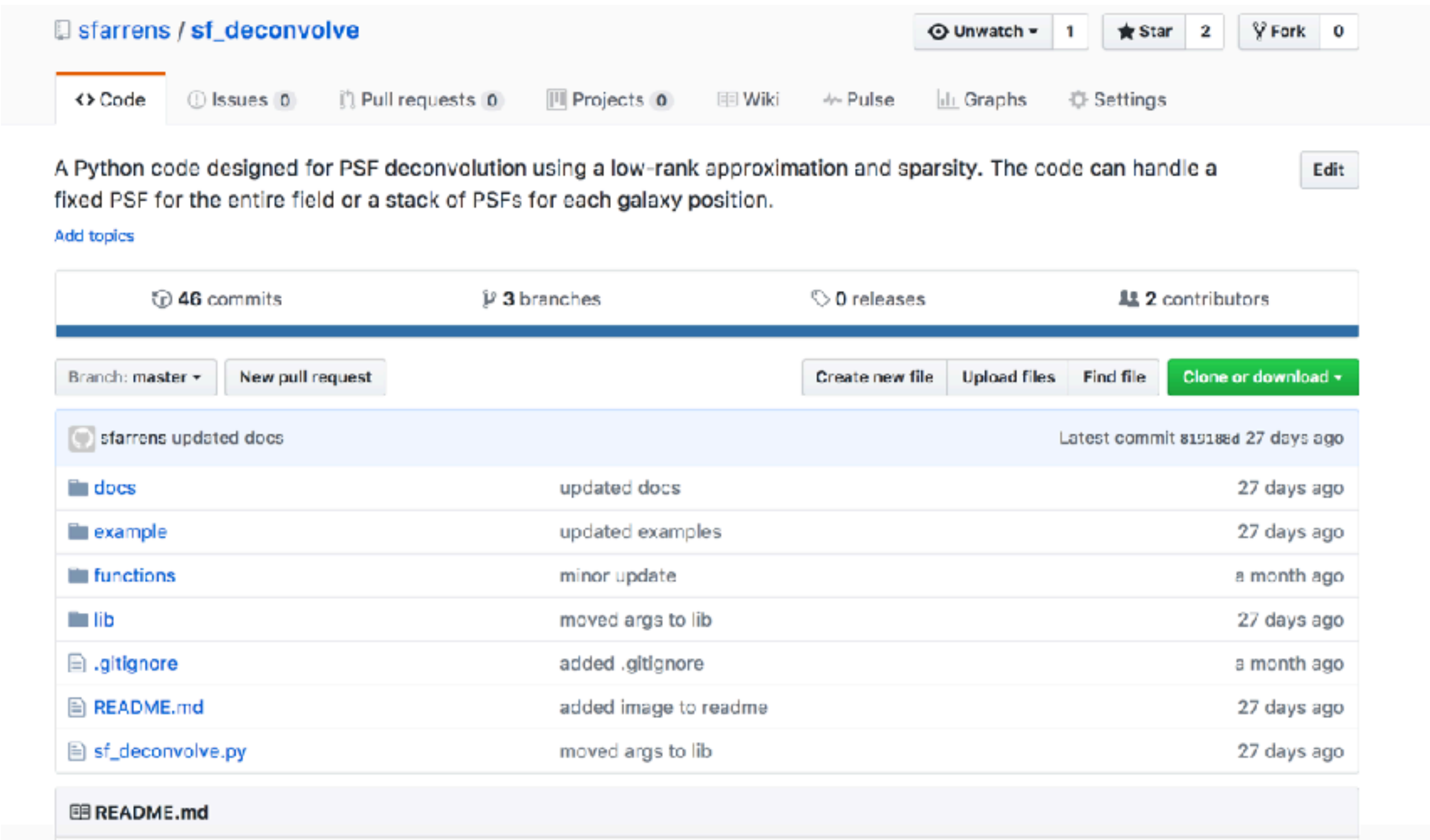
Website

https://sfarrens.github.io/



Code

https://github.com/sfarrens/sf_deconvolve



Code

http://sfarrens.github.io/sf_deconvolve/index.html

sf_deconvolve
3.2

Search docs

sf_deconvolve
- functions package
- lib package
  - Submodules
    - lib.algorithms module
    - lib.convolve module
    - lib.cost module
    - lib.deconvolve module
    - lib.directional module
    - lib.file_io module
    - lib.gradient module
    - lib.linear module
    - lib.noise module
    - **lib.optimisation module**
    - lib.other_methods module
    - lib.plotting module
    - lib.proximity module
    - lib.quality module
    - lib.reweight module
    - lib.sf_deconvolve_args module

Docs » sf_deconvolve » lib package » lib.optimisation module   View page source

lib.optimisation module

OPTIMISATION CLASSES

This module contains classes for optimisation algoritms

Author: Samuel Farrens <samuel.farrens@gmail.com>

Version: 1.2

Date: 05/01/2017

References

1) Condat, A Primal-Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms, 2013, Journal of Optimization Theory and Applications, 158, 2, 460. (C2013) 2) Bauschke et al., Fixed-Point Algorithms for Inverse Problems in Science and Engineering, 2011, Chapter 10. (B2010) 3) Raguet et al., Generalized Forward-Backward Splitting, 2012, , (R2012)

Notes

- x_old is used in place of x_{n}.
- x_new is used in place of x_{n+1}.
- x_prox is used in place of ~{x}_{n+1}.
- x_temp is used for intermediate operations.

Summary

- Farrens et al., A&A, 2017 (arXiv:1703.02305)
- New Python code for deconvolution
  - can handle data with constant or space-variant PSF
  - implements sparse and/or low-rank regularisation
- Results from Euclid-like images show better shape measurements with low-rank regularisation (when sample is sufficiently large)
- For future work we aim to add additional constraints on the deconvolution such as the galaxy shape
- We are also investigating ways to simultaneously estimate the PSF and deconvolve the galaxy images