



Generalized Sparsity-Promoting Solvers and Samplers for Hierarchical Inverse Problems

SIAM UQ24

27th February 2024



Jonathan Lindblom[†], Jan Glaubitz, and Anne Gelb

Outline

1. Overview
2. Generalizations
3. Priorconditioning
4. Application to sampling
5. Other directions



OVERVIEW

Problem setup

We would like to reconstruct an unknown signal $\mathbf{x} \in \mathbb{R}^n$ given an indirect noisy observation

$$\mathbf{y} = \mathbf{F}\mathbf{x} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n).$$

Here $\mathbf{y}, \boldsymbol{\epsilon} \in \mathbb{R}^m$ with linear forward operator $\mathbf{F} \in \mathbb{R}^{m \times n}$.

Ground truth



Blur



Under-sampling



Up-sampling



Maximum likelihood

Observation (no noise)



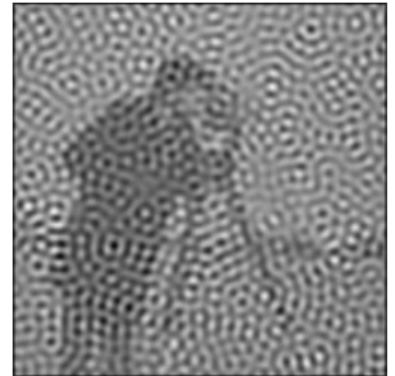
Least-squares reconstruction



Observation (with noise)



Least-squares reconstruction



$$\mathbf{x}^{\text{MLE}} = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{F}\mathbf{x} - \mathbf{y}\|_2$$

ℓ_2 regularization

Ground truth



Observation

 $\lambda = 1$  $\lambda = 100$  $\lambda = 1000$ 

$$\arg \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2\sigma^2} \|\mathbf{F}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \|\mathbf{R}\mathbf{x}\|_2^2$$

Statistical interpretation

$$\mathbf{y} | \mathbf{x} \sim \mathcal{N} \left(\mathbf{F}\mathbf{x}, \sigma^2 \mathbf{I}_n \right)$$

$$\mathbf{x} \sim \mathcal{N} \left(\mathbf{0}, \left(\lambda \mathbf{R}^T \mathbf{R} \right)^{-1} \right)$$

$$\implies \mathbf{x} | \mathbf{y} \sim \mathcal{N} \left(\boldsymbol{\mu}_{\text{post}}, \mathbf{Q}_{\text{post}}^{-1} \right), \quad \mathbf{Q}_{\text{post}} = \sigma^{-2} \mathbf{F}^T \mathbf{F} + \lambda \mathbf{R}^T \mathbf{R}, \quad \boldsymbol{\mu}_{\text{post}} = \sigma^{-2} \mathbf{Q}_{\text{post}}^{-1} \mathbf{F}^T \mathbf{y}$$

$$\begin{aligned} -\log \pi(\mathbf{x} | \mathbf{y}) + C &= \frac{1}{2\sigma^2} \|\mathbf{F}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \|\mathbf{R}\mathbf{x}\|_2^2 \\ &:= \mathcal{G}(\mathbf{x}) \end{aligned}$$

Statistical interpretation

$$\mathbf{y} | \mathbf{x} \sim \mathcal{N} \left(\mathbf{F}\mathbf{x}, \sigma^2 \mathbf{I}_n \right)$$

$$\mathbf{x} \sim \mathcal{N} \left(\mathbf{0}, \left(\lambda \mathbf{R}^T \mathbf{R} \right)^{-1} \right)$$

$$\implies \mathbf{x} | \mathbf{y} \sim \mathcal{N} \left(\boldsymbol{\mu}_{\text{post}}, \mathbf{Q}_{\text{post}}^{-1} \right), \quad \mathbf{Q}_{\text{post}} = \sigma^{-2} \mathbf{F}^T \mathbf{F} + \lambda \mathbf{R}^T \mathbf{R}, \quad \boldsymbol{\mu}_{\text{post}} = \sigma^{-2} \mathbf{Q}_{\text{post}}^{-1} \mathbf{F}^T \mathbf{y}$$

$$\begin{aligned} -\log \pi(\mathbf{x} | \mathbf{y}) + C &= \frac{1}{2\sigma^2} \|\mathbf{F}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \|\mathbf{R}\mathbf{x}\|_2^2 \\ &:= \mathcal{G}(\mathbf{x}) \end{aligned}$$

Statistical interpretation

$$\mathbf{y} | \mathbf{x} \sim \mathcal{N} \left(\mathbf{F}\mathbf{x}, \sigma^2 \mathbf{I}_n \right)$$

$$\mathbf{x} \sim \mathcal{N} \left(\mathbf{0}, \left(\lambda \mathbf{R}^T \mathbf{R} \right)^{-1} \right)$$

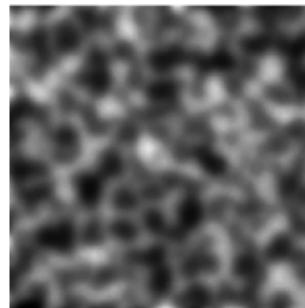
$$\implies \mathbf{x} | \mathbf{y} \sim \mathcal{N} \left(\boldsymbol{\mu}_{\text{post}}, \mathbf{Q}_{\text{post}}^{-1} \right), \quad \mathbf{Q}_{\text{post}} = \sigma^{-2} \mathbf{F}^T \mathbf{F} + \lambda \mathbf{R}^T \mathbf{R}, \quad \boldsymbol{\mu}_{\text{post}} = \sigma^{-2} \mathbf{Q}_{\text{post}}^{-1} \mathbf{F}^T \mathbf{y}$$

$$\begin{aligned} -\log \pi(\mathbf{x} | \mathbf{y}) + C &= \frac{1}{2\sigma^2} \|\mathbf{F}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \|\mathbf{R}\mathbf{x}\|_2^2 \\ &:= \mathcal{G}(\mathbf{x}) \end{aligned}$$

Gaussian vs. non-Gaussian priors

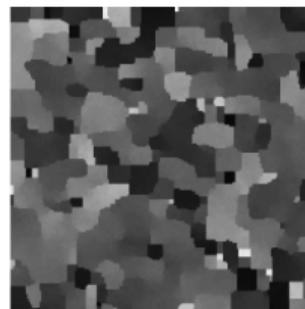
Gaussian:

- MAP estimation under a Gaussian prior amounts to solving a linear system / least squares problem 
- Sampling high-dimensional Gaussians has been studied extensively 



Non-Gaussian:

- MAP estimation usually requires minimizing non-quadratic/nonsmooth objective 
- To sample must usually resort to MCMC, but even this is challenging 



M. Markkanen, L. Roininen, J. M. Huttunen, and S. Lasanen,
Cauchy difference priors for edge-preserving bayesian
inversion, Journal of Inverse and Ill-posed Problems, 27 (2019),
pp. 225–24.

Scale mixture representations

$$\begin{aligned} \mathbf{y} | \mathbf{x} &\sim \mathcal{N}\left(\mathbf{F}\mathbf{x}, \sigma^2 \mathbf{I}_n\right) \\ \mathbf{x} &\sim \pi(\mathbf{x}) \end{aligned}$$



$$\begin{aligned} \mathbf{y} | \mathbf{x} &\sim \mathcal{N}\left(\mathbf{F}\mathbf{x}, \sigma^2 \mathbf{I}_n\right) \\ \mathbf{x} | \boldsymbol{\theta} &\sim \mathcal{N}\left(\mathbf{0}, \left(\mathbf{R}^T \mathbf{D}_{\boldsymbol{\theta}}^{-1} \mathbf{R}\right)^{-1}\right) \\ \theta_i &\stackrel{\text{ind}}{\sim} \text{Hyper-prior} \end{aligned}$$

$$\pi(\mathbf{x}) = \int \pi(\mathbf{x} | \boldsymbol{\theta}) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

Scale mixture representations

$$\begin{array}{ccc} \boldsymbol{y} | \boldsymbol{x} \sim \mathcal{N} \left(\boldsymbol{F}\boldsymbol{x}, \sigma^2 \boldsymbol{I}_n \right) & \iff & \boldsymbol{y} | \boldsymbol{x} \sim \mathcal{N} \left(\boldsymbol{F}\boldsymbol{x}, \sigma^2 \boldsymbol{I}_n \right) \\ \boldsymbol{x} \sim \pi(\boldsymbol{x}) & & \boldsymbol{x} | \boldsymbol{\theta} \sim \mathcal{N} \left(\mathbf{0}, \left(\boldsymbol{R}^T \boldsymbol{D}_{\boldsymbol{\theta}}^{-1} \boldsymbol{R} \right)^{-1} \right) \\ & & \theta_i \stackrel{\text{ind}}{\sim} \text{Hyper-prior} \end{array}$$

$$\pi(\boldsymbol{x}) = \int \pi(\boldsymbol{x} | \boldsymbol{\theta}) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

Generalized Gamma hyper-prior

$$\pi_{\text{GG}}(\theta | r, \beta, \vartheta) \propto \theta^{r\beta-1} \exp \left\{ - \left(\frac{\theta}{\vartheta} \right)^r \right\} \mathbb{1}_{\mathbb{R}_+}(\theta)$$

- Defined for $r \in \mathbb{R} \setminus \{0\}, \beta > 0, \vartheta > 0$

Generalized Gamma hyper-prior

$$\pi_{\text{GG}}(\theta | r, \beta, \vartheta) \propto \theta^{r\beta-1} \exp \left\{ - \left(\frac{\theta}{\vartheta} \right)^r \right\} \mathbb{1}_{\mathbb{R}_+}(\theta)$$

- Defined for $r \in \mathbb{R} \setminus \{0\}, \beta > 0, \vartheta > 0$
- Encompasses exponential, Gamma, inverse Gamma distributions

Generalized Gamma hyper-prior

$$\pi_{\text{GG}}(\theta | r, \beta, \vartheta) \propto \theta^{r\beta-1} \exp \left\{ - \left(\frac{\theta}{\vartheta} \right)^r \right\} \mathbb{1}_{\mathbb{R}_+}(\theta)$$

- Defined for $r \in \mathbb{R} \setminus \{0\}, \beta > 0, \vartheta > 0$
- Encompasses exponential, Gamma, inverse Gamma distributions
- Parameters (r, s) set the parametric family of marginal prior, ϑ controls scale

Generalized Gamma hyper-prior

$$\pi_{\text{GG}}(\theta | r, \beta, \vartheta) \propto \theta^{r\beta-1} \exp \left\{ - \left(\frac{\theta}{\vartheta} \right)^r \right\} \mathbb{1}_{\mathbb{R}_+}(\theta)$$

- Defined for $r \in \mathbb{R} \setminus \{0\}, \beta > 0, \vartheta > 0$
- Encompasses exponential, Gamma, inverse Gamma distributions
- Parameters (r, s) set the parametric family of marginal prior, ϑ controls scale
- In certain limits, captures ℓ_p -norm priors for $0 < p < 2$, Student- t , Cauchy

Hierarchical representation of ℓ_1 prior

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2\sigma^2} \|\mathbf{F}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{R}\mathbf{x}\|_1$$



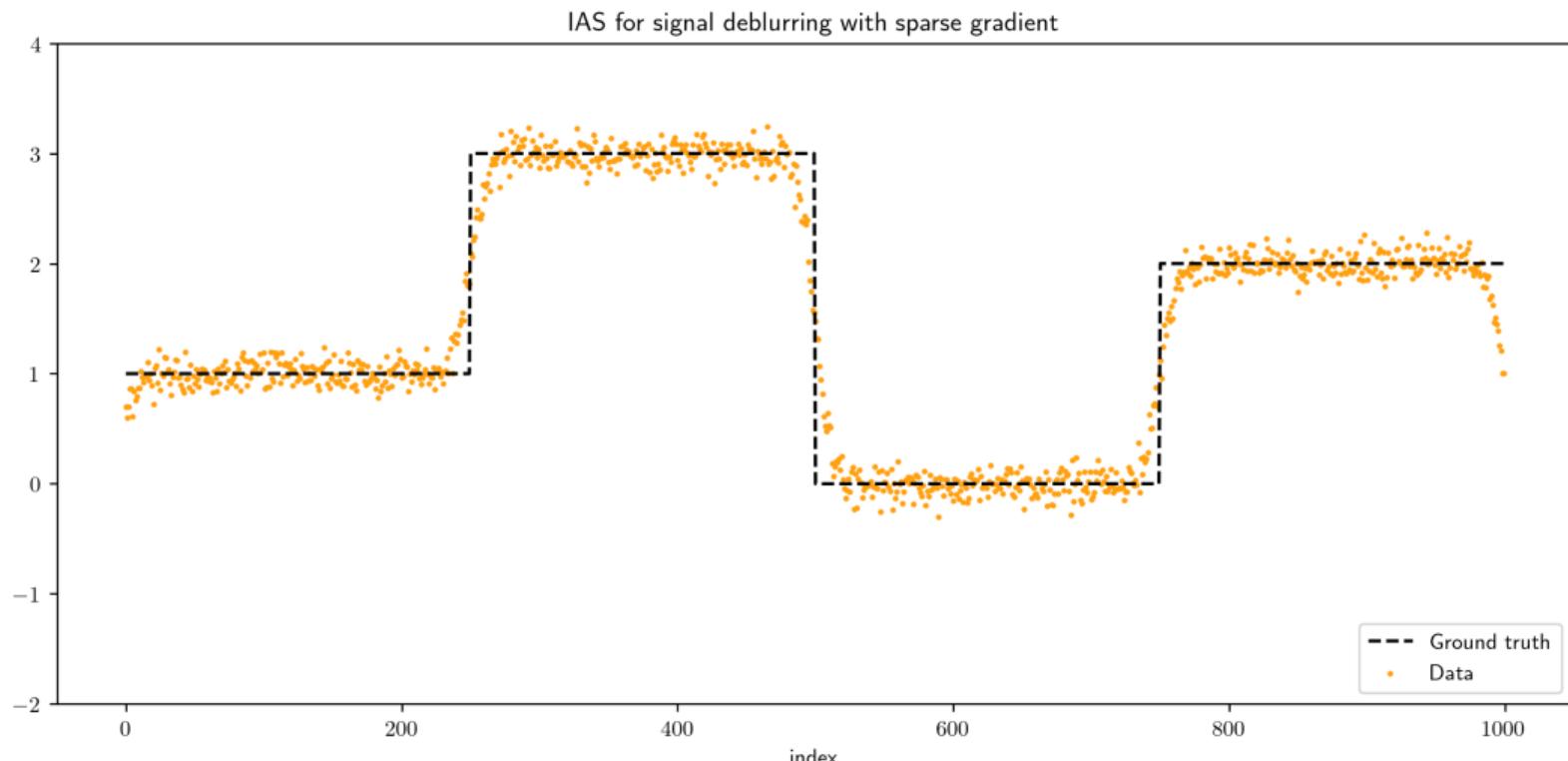
$$(\mathbf{x}^*, \boldsymbol{\theta}^*) \stackrel{\eta \rightarrow 0}{=} \arg \min_{\mathbf{x}, \boldsymbol{\theta}} \frac{1}{2\sigma^2} \|\mathbf{F}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{1}{2} \|\mathbf{D}_{\boldsymbol{\theta}}^{-1/2} \mathbf{R}\mathbf{x}\|_2^2 + \frac{1}{2} \sum_{i=1}^k \log \theta_i - \log \pi(\boldsymbol{\theta})$$

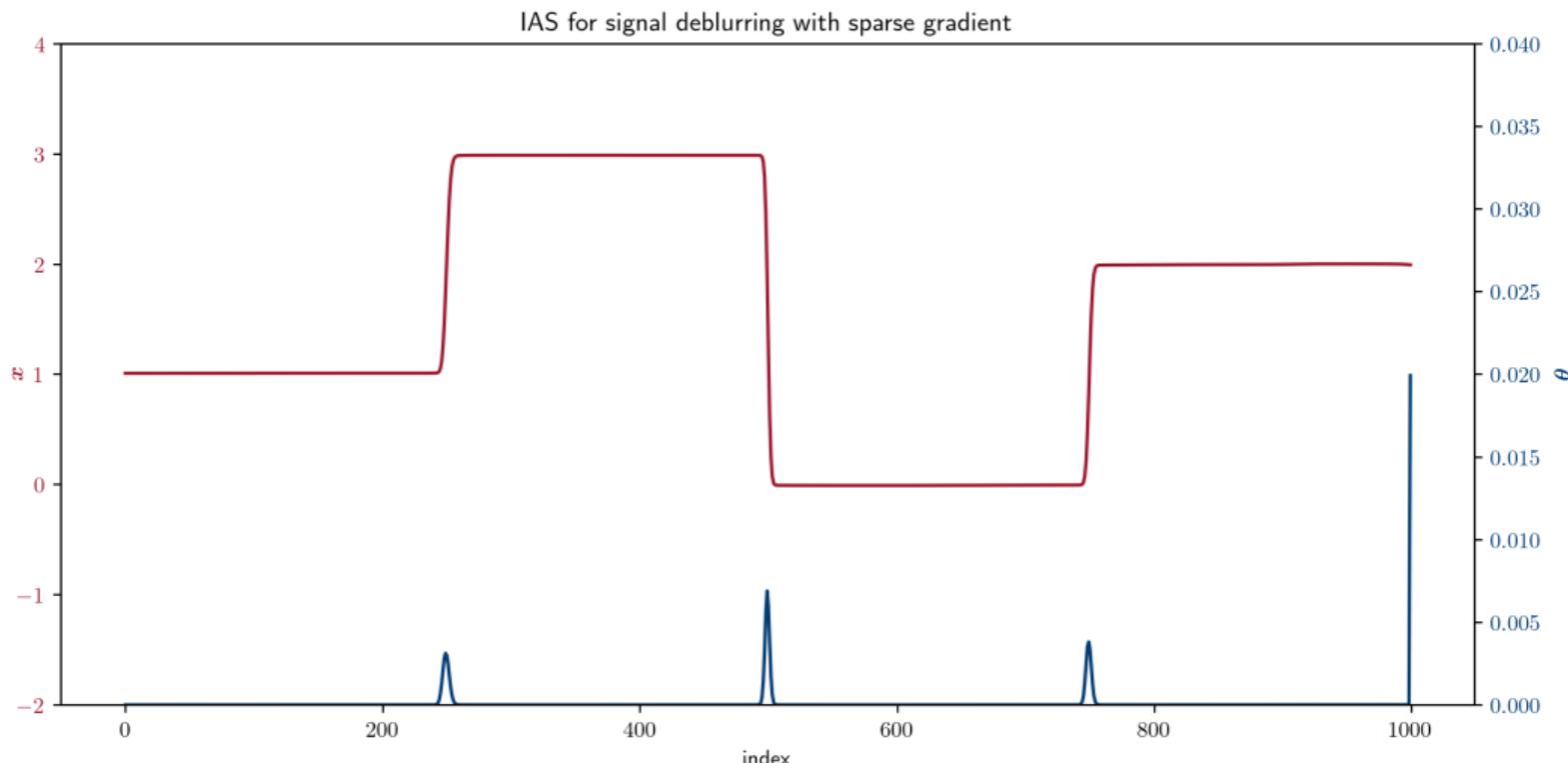
Block coordinate descent (IAS) for MAP estimation

IAS Algorithm

- 1: **Input:** Data \mathbf{y} , noise variance σ^2 , forward operator \mathbf{F} , sparsifying transformation \mathbf{R} , hyper-parameters $(r, \beta, \boldsymbol{\vartheta})$, and initialization $\mathbf{x}^{(0)}$
 - 2: **Output:** Approximate MAP estimate $(\mathbf{x}^{\text{MAP}}, \boldsymbol{\theta}^{\text{MAP}})$ for the joint posterior $\pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y})$
 - 3: **repeat**
 - 4: Update the hyper-parameters $\boldsymbol{\theta}$ by $\boldsymbol{\theta}^{(k+1)} = \arg \max_{\boldsymbol{\theta}} \pi(\mathbf{x}^{(k)}, \boldsymbol{\theta} | \mathbf{y}) = \arg \min_{\boldsymbol{\theta}} \mathcal{G}(\mathbf{x}^{(k)}, \boldsymbol{\theta})$
 - 5: Update the parameter vector \mathbf{x} by $\mathbf{x}^{(k+1)} = \arg \max_{\mathbf{x}} \pi(\mathbf{x}, \boldsymbol{\theta}^{(k+1)} | \mathbf{y}) = \arg \min_{\mathbf{x}} \mathcal{G}(\mathbf{x}, \boldsymbol{\theta}^{(k+1)})$
 - 6: **until** convergence or the maximum number of iterations is reached
-

$$\mathcal{G}(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\mathbf{F}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{1}{2} \|\mathbf{D}_{\boldsymbol{\theta}}^{-1/2} \mathbf{R}\mathbf{x}\|_2^2 + \sum_{i=1}^k \left(\frac{\theta_i}{\vartheta_i} \right)^r - \sum_{i=1}^k (r\beta - 3/2) \log(\theta_i) + \delta_{\mathbb{R}_+^N}(\boldsymbol{\theta})$$





GENERALIZATIONS

Convexity

Previous convexity result

Let $\eta := r\beta - \frac{3}{2}$ and $\mathbf{R} = \mathbf{I}_n$. Then $(\mathbf{x}, \boldsymbol{\theta})$ is globally strict convex if $r \geq 1$ and $\eta > 0$, and is locally convex at $(\mathbf{x}, \boldsymbol{\theta})$ if either (i) $0 < r < 1$ and $\eta > 0$ or (ii) $r < 0$ and

$$\theta_i < \vartheta_i \left(\eta / (r|r-1|) \right)^{1/r}, \quad i = 1, \dots, n.$$

Convexity with general \mathbf{R}

The previous result holds for any \mathbf{R} so long as $\ker(\mathbf{F}) \cap \ker(\mathbf{R}) = \{\mathbf{0}_n\}$.

Convexity

Previous convexity result

Let $\eta := r\beta - \frac{3}{2}$ and $\mathbf{R} = \mathbf{I}_n$. Then $(\mathbf{x}, \boldsymbol{\theta})$ is globally strict convex if $r \geq 1$ and $\eta > 0$, and is locally convex at $(\mathbf{x}, \boldsymbol{\theta})$ if either (i) $0 < r < 1$ and $\eta > 0$ or (ii) $r < 0$ and

$$\theta_i < \vartheta_i \left(\eta / (r|r-1|) \right)^{1/r}, \quad i = 1, \dots, n.$$

Convexity with general \mathbf{R}

The previous result holds for any \mathbf{R} so long as $\ker(\mathbf{F}) \cap \ker(\mathbf{R}) = \{\mathbf{0}_n\}$.

Convergence

Previous convergence result

When $r = 1$ and $\eta > 0$, IAS converges to the unique global minimizer of $\mathcal{G}(\mathbf{x}, \boldsymbol{\theta})$.

General convex case

When $r \geq 1$ and $\eta > 0$, IAS converges to the unique global minimizer of $\mathcal{G}(\mathbf{x}, \boldsymbol{\theta})$.

Nonconvex case

For any valid hyper-prior parameters $(r, \beta, \boldsymbol{\vartheta})$, the sequence $\{\mathbf{x}^{(k)}, \boldsymbol{\theta}^{(k)}\}$ is bounded and any limit point must be a stationary point of $\mathcal{G}(\mathbf{x}, \boldsymbol{\theta})$.

Convergence

Previous convergence result

When $r = 1$ and $\eta > 0$, IAS converges to the unique global minimizer of $\mathcal{G}(\boldsymbol{x}, \boldsymbol{\theta})$.

General convex case

When $r \geq 1$ and $\eta > 0$, IAS converges to the unique global minimizer of $\mathcal{G}(\boldsymbol{x}, \boldsymbol{\theta})$.

Nonconvex case

For any valid hyper-prior parameters $(r, \beta, \boldsymbol{\vartheta})$, the sequence $\{\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)}\}$ is bounded and any limit point must be a stationary point of $\mathcal{G}(\boldsymbol{x}, \boldsymbol{\theta})$.

Learning noise variance

- In some settings, we may not have a good estimate of the noise variance σ^2
- Could try initial reconstruction \mathbf{x} to estimate σ^2 , but not always feasible
- Can adopt an “uninformative” prior for the noise variance

$$\mathbf{y} | \mathbf{x}, \nu \sim \mathcal{N}(\mathbf{F}\mathbf{x}, \nu \mathbf{I}_n)$$

$$\mathbf{x} | \boldsymbol{\theta} \sim \mathcal{N}\left(\mathbf{0}, \left(\mathbf{R}^T \mathbf{D}_{\boldsymbol{\theta}}^{-1} \mathbf{R}\right)^{-1}\right)$$

$$\theta_i \stackrel{\text{ind}}{\sim} \mathcal{G}\mathcal{G}(r, \beta, \vartheta_i)$$

$$\nu \sim \mathcal{G}\mathcal{G}(\tilde{r}, \tilde{\beta}, \tilde{\vartheta})$$

Generalized IAS Algorithm

- 1: **Input:** Data \mathbf{y} , forward operator \mathbf{F} , sparsifying transformation \mathbf{R} , hyper-parameters $(r, \beta, \boldsymbol{\vartheta})$ and $(\tilde{r}, \tilde{\beta}, \tilde{\vartheta})$, initialization $\mathbf{x}^{(0)}$
 - 2: **Output:** Approximate MAP estimate $(\mathbf{x}^{\text{MAP}}, \boldsymbol{\theta}^{\text{MAP}}, \nu^{\text{MAP}})$ for the joint posterior $\pi(\mathbf{x}, \boldsymbol{\theta}, \nu | \mathbf{y})$
 - 3: **repeat**
 - 4: Update $\boldsymbol{\theta}$ by $\boldsymbol{\theta}^{(k+1)} = \arg \max_{\boldsymbol{\theta}} \pi(\mathbf{x}^{(k)}, \boldsymbol{\theta}, \nu^{(k)} | \mathbf{y}) = \arg \min_{\boldsymbol{\theta}} \mathcal{G}(\mathbf{x}^{(k)}, \boldsymbol{\theta}, \nu^{(k)})$
 - 5: Update ν by $\nu = \arg \max_{\nu} \pi(\mathbf{x}^{(k)}, \boldsymbol{\theta}^{(k+1)}, \nu | \mathbf{y}) = \arg \min_{\nu} \mathcal{G}(\mathbf{x}^{(k)}, \boldsymbol{\theta}^{(k+1)}, \nu)$
 - 6: Update \mathbf{x} by $\mathbf{x}^{(k+1)} = \arg \max_{\mathbf{x}} \pi(\mathbf{x}, \boldsymbol{\theta}^{(k+1)}, \nu^{(k+1)} | \mathbf{y}) = \arg \min_{\mathbf{x}} \mathcal{G}(\mathbf{x}, \boldsymbol{\theta}^{(k+1)}, \nu^{(k+1)})$
 - 7: **until** convergence or the maximum number of iterations is reached
-

$$\begin{aligned} \mathcal{G}(\mathbf{x}, \boldsymbol{\theta}, \nu) &= \frac{1}{2\nu} \|\mathbf{F}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{1}{2} \|\mathbf{D}_{\boldsymbol{\theta}}^{-1/2} \mathbf{R}\mathbf{x}\|_2^2 + \left(\frac{\nu}{\tilde{\vartheta}} \right)^{\tilde{r}} + \sum_{i=1}^k \left(\frac{\theta_i}{\vartheta_i} \right)^r \\ &\quad - \left(\tilde{r}\tilde{\beta} - [m+2]/2 \right) \log(\nu) - (r\beta - 3/2) \sum_{i=1}^k \log(\theta_i) + \delta_{\mathbb{R}_+^{k+1}}(\boldsymbol{\theta}, \nu), \end{aligned}$$

Convexity

Convexity result

Let $\eta = r\beta - 3/2$, $\tilde{\eta} = \tilde{r}\beta - (m+2)/2$. Then $\mathcal{G}(\mathbf{x}, \boldsymbol{\theta}, \nu)$ is globally strictly convex if $r, \tilde{r} \geq 1$ and $\eta, \tilde{\eta} > 0$.

- The use of an uninformative noise variance prior such as $r = -1$, $\beta = 1$, $\vartheta \approx 0$ generally leads to a nonconvex objective

Convexity

Convexity result

Let $\eta = r\beta - 3/2$, $\tilde{\eta} = \tilde{r}\beta - (m+2)/2$. Then $\mathcal{G}(\mathbf{x}, \boldsymbol{\theta}, \nu)$ is globally strictly convex if $r, \tilde{r} \geq 1$ and $\eta, \tilde{\eta} > 0$.

- The use of an uninformative noise variance prior such as $r = -1$, $\beta = 1$, $\vartheta \approx 0$ generally leads to a nonconvex objective

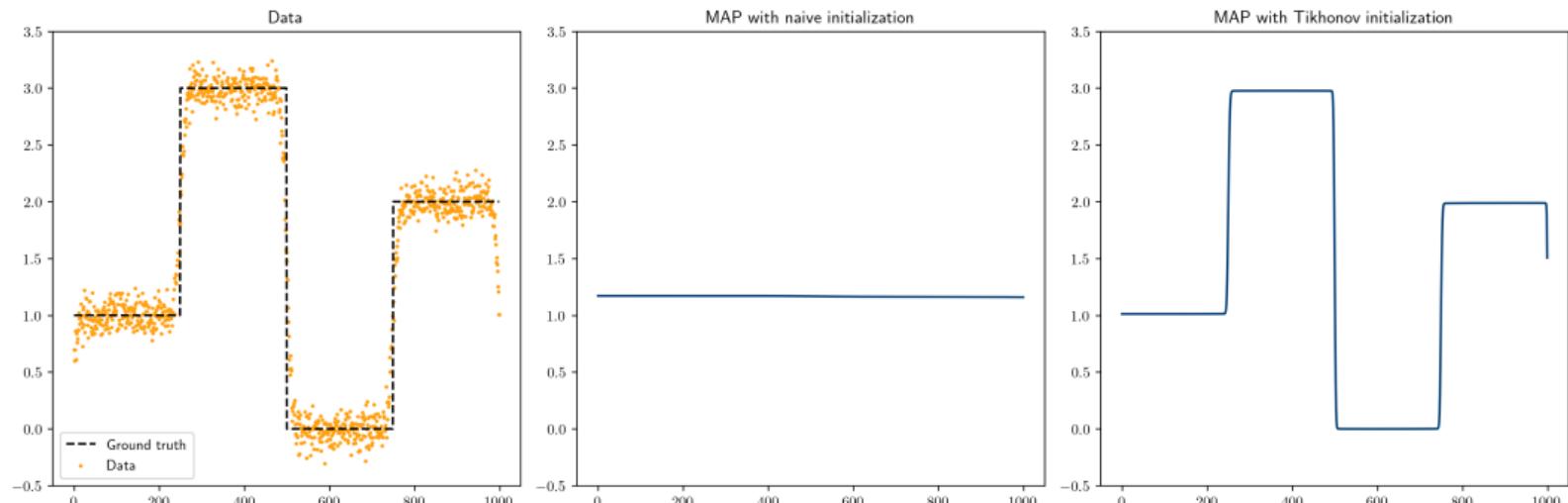
Convergence

General convex case

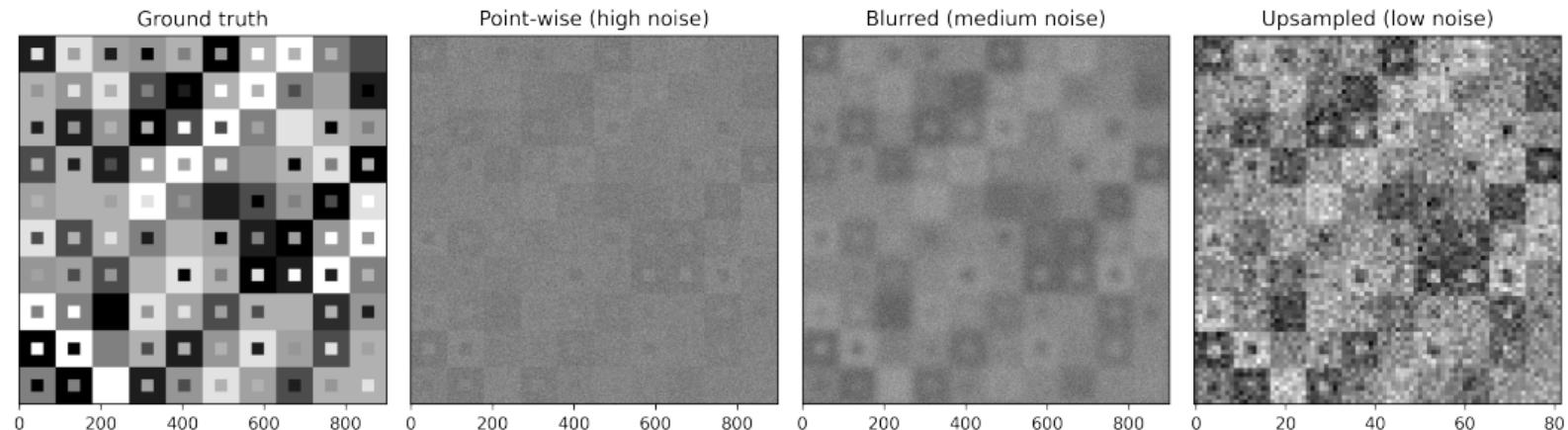
When $r, \tilde{r} \geq 1$ and $\eta, \tilde{\eta} > 0$, generalized IAS converges to the unique global minimizer of $\mathcal{G}(\boldsymbol{x}, \boldsymbol{\theta}, \nu)$.

Nonconvex case

For any valid hyper-prior parameters $(r, \beta, \boldsymbol{\vartheta})$, the sequence $\{\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)}, \nu^{(k)}\}$ is bounded and any limit point must be a stationary point of $\mathcal{G}(\boldsymbol{x}, \boldsymbol{\theta}, \nu)$.

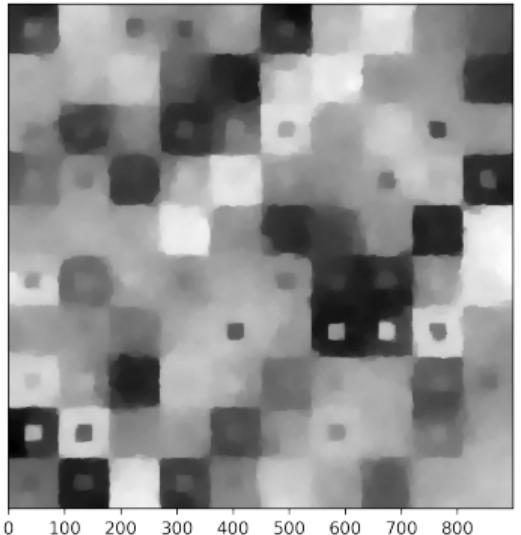


Data fusion

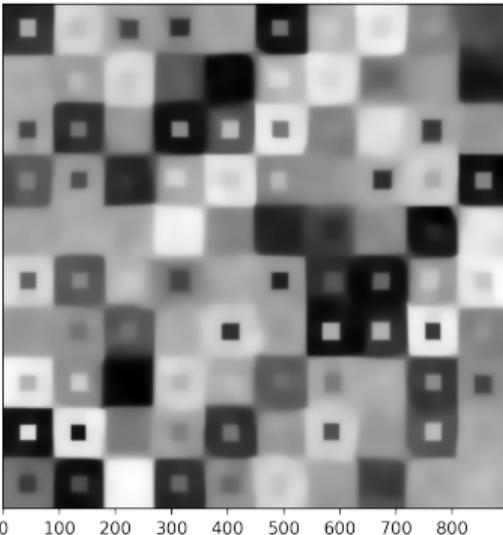


Data fusion

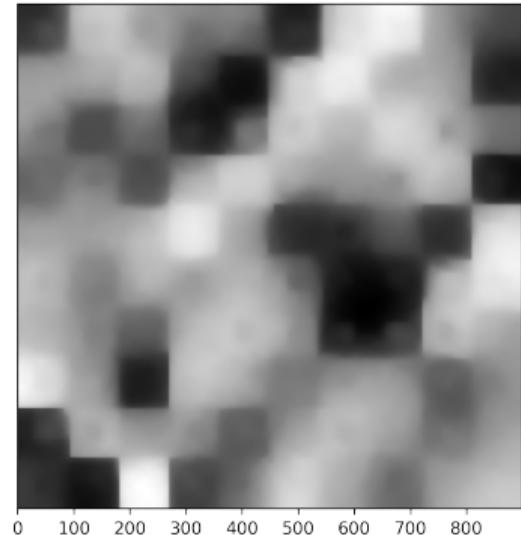
From point-wise observation



From blurred observation

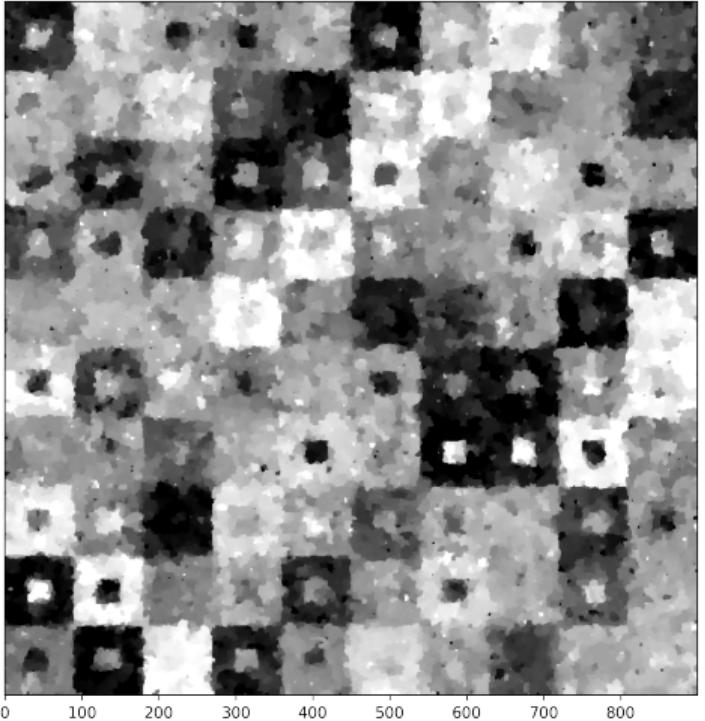


From up-sampled observation

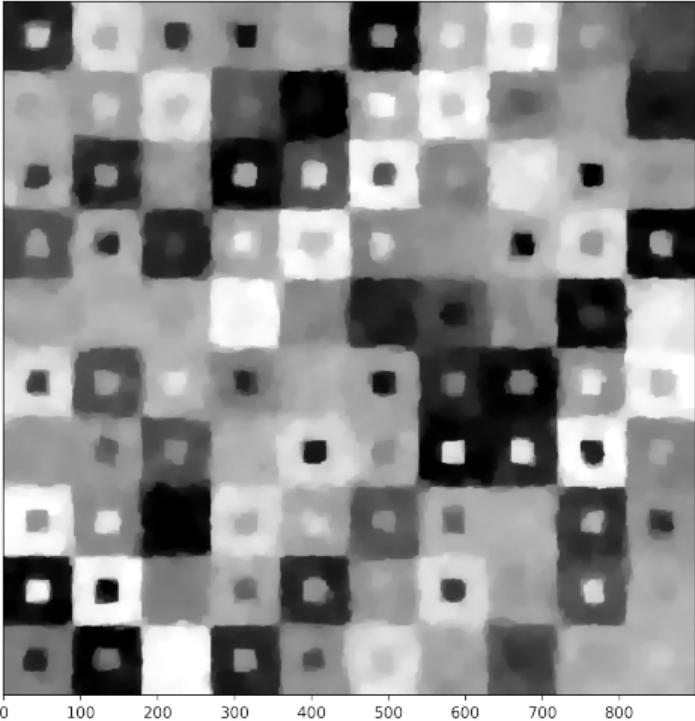


Data fusion

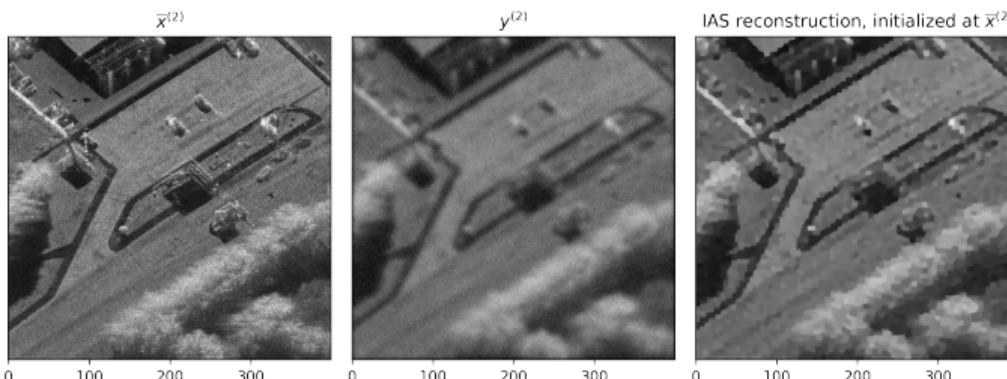
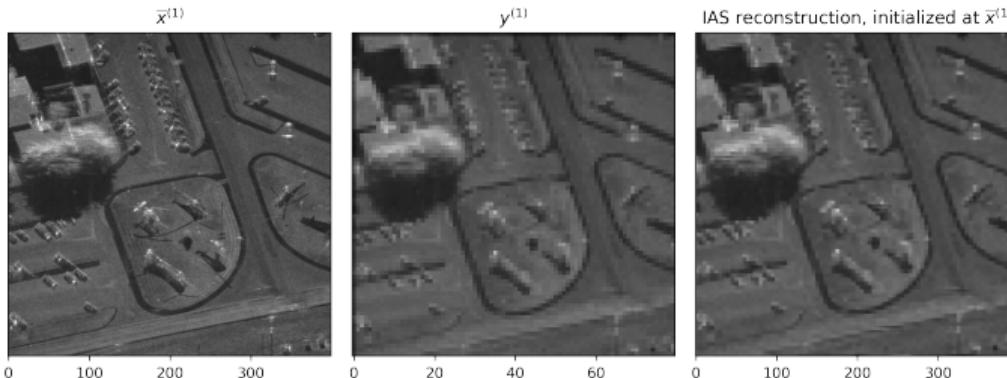
Stacked reconstruction



Reconstruction with three variances



Data fusion



PRIORCONDITIONING

The most computationally intensive step of the IAS method is the solution of quadratic optimization problems of the form

$$\boldsymbol{x}^* = \arg \min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{F}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \|\boldsymbol{D}_{\theta}^{-1/2} \boldsymbol{R}\boldsymbol{x}\|_2^2,$$

which is given explicitly by

$$\boldsymbol{x}^* = \left(\boldsymbol{F}^T \boldsymbol{F} + \boldsymbol{R}^T \boldsymbol{D}_{\theta}^{-1} \boldsymbol{R} \right)^{-1} \boldsymbol{F}^T \boldsymbol{y}.$$

or equivalently as the solution of the least squares problems

$$\begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{D}_{\theta}^{-1/2} \boldsymbol{R} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{0}_k \end{bmatrix}.$$

Usually, we must use a Krylov iterative method such as CGLS.

The most computationally intensive step of the IAS method is the solution of quadratic optimization problems of the form

$$\boldsymbol{x}^* = \arg \min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{F}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \|\boldsymbol{D}_{\theta}^{-1/2} \boldsymbol{R}\boldsymbol{x}\|_2^2,$$

which is given explicitly by

$$\boldsymbol{x}^* = \left(\boldsymbol{F}^T \boldsymbol{F} + \boldsymbol{R}^T \boldsymbol{D}_{\theta}^{-1} \boldsymbol{R} \right)^{-1} \boldsymbol{F}^T \boldsymbol{y}.$$

or equivalently as the solution of the least squares problems

$$\begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{D}_{\theta}^{-1/2} \boldsymbol{R} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{y} \\ \mathbf{0}_k \end{bmatrix}.$$

Usually, we must use a Krylov iterative method such as CGLS.

The most computationally intensive step of the IAS method is the solution of quadratic optimization problems of the form

$$\boldsymbol{x}^* = \arg \min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{F}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \|\boldsymbol{D}_{\theta}^{-1/2} \boldsymbol{R}\boldsymbol{x}\|_2^2,$$

which is given explicitly by

$$\boldsymbol{x}^* = \left(\boldsymbol{F}^T \boldsymbol{F} + \boldsymbol{R}^T \boldsymbol{D}_{\theta}^{-1} \boldsymbol{R} \right)^{-1} \boldsymbol{F}^T \boldsymbol{y}.$$

or equivalently as the solution of the least squares problems

$$\begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{D}_{\theta}^{-1/2} \boldsymbol{R} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{y} \\ \mathbf{0}_k \end{bmatrix}.$$

Usually, we must use a Krylov iterative method such as CGLS.

Preconditioning can improve efficiency. One such technique is called priorconditioning.

Let $\mathbf{R}_\theta = \mathbf{D}_\theta^{-1/2} \mathbf{R}$.

If \mathbf{R}^{-1} exists:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_\theta \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix} \quad \mathbf{x} = \mathbf{R}_\theta^{-1} \mathbf{w} \rightarrow \begin{bmatrix} \mathbf{F} \mathbf{R}_\theta^{-1} \\ \mathbf{I}_n \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix}$$

If $\ker(\mathbf{R}) = \{\mathbf{0}\}$:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_\theta \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix} \quad \mathbf{x} = \mathbf{R}_\theta^\dagger \mathbf{w} \rightarrow \begin{bmatrix} \mathbf{F} \mathbf{R}_\theta^\dagger \\ \mathbf{I}_k \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_k \end{bmatrix}$$

★ What about if $\ker(\mathbf{R}) \neq \{\mathbf{0}\}$?

Preconditioning can improve efficiency. One such technique is called priorconditioning.

Let $\mathbf{R}_\theta = \mathbf{D}_\theta^{-1/2} \mathbf{R}$.

If \mathbf{R}^{-1} exists:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_\theta \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix} \quad \mathbf{x} = \mathbf{R}_\theta^{-1} \mathbf{w} \rightarrow \begin{bmatrix} \mathbf{F} \mathbf{R}_\theta^{-1} \\ \mathbf{I}_n \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix}$$

If $\ker(\mathbf{R}) = \{\mathbf{0}\}$:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_\theta \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix} \quad \mathbf{x} = \mathbf{R}_\theta^\dagger \mathbf{w} \rightarrow \begin{bmatrix} \mathbf{F} \mathbf{R}_\theta^\dagger \\ \mathbf{I}_k \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_k \end{bmatrix}$$

★ What about if $\ker(\mathbf{R}) \neq \{\mathbf{0}\}$?

Preconditioning can improve efficiency. One such technique is called priorconditioning.

Let $\mathbf{R}_\theta = \mathbf{D}_\theta^{-1/2} \mathbf{R}$.

If \mathbf{R}^{-1} exists:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_\theta \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix} \quad \mathbf{x} = \mathbf{R}_\theta^{-1} \mathbf{w} \rightarrow \begin{bmatrix} \mathbf{F} \mathbf{R}_\theta^{-1} \\ \mathbf{I}_n \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix}$$

If $\ker(\mathbf{R}) = \{\mathbf{0}\}$:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_\theta \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix} \quad \mathbf{x} = \mathbf{R}_\theta^\dagger \mathbf{w} \rightarrow \begin{bmatrix} \mathbf{F} \mathbf{R}_\theta^\dagger \\ \mathbf{I}_k \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_k \end{bmatrix}$$

★ What about if $\ker(\mathbf{R}) \neq \{\mathbf{0}\}$?

Preconditioning can improve efficiency. One such technique is called priorconditioning.

Let $\mathbf{R}_\theta = \mathbf{D}_\theta^{-1/2} \mathbf{R}$.

If \mathbf{R}^{-1} exists:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_\theta \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix} \quad \mathbf{x} = \mathbf{R}_\theta^{-1} \mathbf{w} \rightarrow \begin{bmatrix} \mathbf{F} \mathbf{R}_\theta^{-1} \\ \mathbf{I}_n \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix}$$

If $\ker(\mathbf{R}) = \{\mathbf{0}\}$:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_\theta \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_n \end{bmatrix} \quad \mathbf{x} = \mathbf{R}_\theta^\dagger \mathbf{w} \rightarrow \begin{bmatrix} \mathbf{F} \mathbf{R}_\theta^\dagger \\ \mathbf{I}_k \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_k \end{bmatrix}$$

★ What about if $\ker(\mathbf{R}) \neq \{\mathbf{0}\}$?

For sparsifying transformations with non-trivial kernels, we can use the oblique pseudoinverse, which for \mathbf{W} s.t. $\ker(\mathbf{R}) = \text{col}(\mathbf{W})$ is given by

$$\mathbf{R}_{\theta}^{\#} = (\mathbf{I}_n - \mathbf{W}(\mathbf{F}\mathbf{W})^{\dagger}\mathbf{F}) \mathbf{R}_{\theta}^{\dagger}$$

We make the change of variables

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_{\theta} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_k \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} \mathbf{F}\mathbf{R}_{\theta}^{\#} \\ \mathbf{I}_k \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_k \end{bmatrix}$$

Then, the solution is given by

$$\mathbf{x}^* = \underbrace{\mathbf{R}_{\theta}^{\#} \mathbf{w}^*}_{\in \ker(\mathbf{R})^{\perp_F}} + \underbrace{\mathbf{W}(\mathbf{F}\mathbf{W})^{\dagger} \mathbf{y}}_{\in \ker(\mathbf{R})}$$

For sparsifying transformations with non-trivial kernels, we can use the oblique pseudoinverse, which for \mathbf{W} s.t. $\ker(\mathbf{R}) = \text{col}(\mathbf{W})$ is given by

$$\mathbf{R}_{\theta}^{\#} = (\mathbf{I}_n - \mathbf{W}(\mathbf{F}\mathbf{W})^{\dagger}\mathbf{F}) \mathbf{R}_{\theta}^{\dagger}$$

We make the change of variables

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_{\theta} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_k \end{bmatrix} \quad \mathbf{x} = \mathbf{R}_{\theta}^{\#} \mathbf{w} \rightarrow \begin{bmatrix} \mathbf{F} \mathbf{R}_{\theta}^{\#} \\ \mathbf{I}_k \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_k \end{bmatrix}$$

Then, the solution is given by

$$\mathbf{x}^* = \underbrace{\mathbf{R}_{\theta}^{\#} \mathbf{w}^*}_{\in \ker(\mathbf{R})^{\perp_F}} + \underbrace{\mathbf{W}(\mathbf{F}\mathbf{W})^{\dagger} \mathbf{y}}_{\in \ker(\mathbf{R})}$$

For sparsifying transformations with non-trivial kernels, we can use the oblique pseudoinverse, which for \mathbf{W} s.t. $\ker(\mathbf{R}) = \text{col}(\mathbf{W})$ is given by

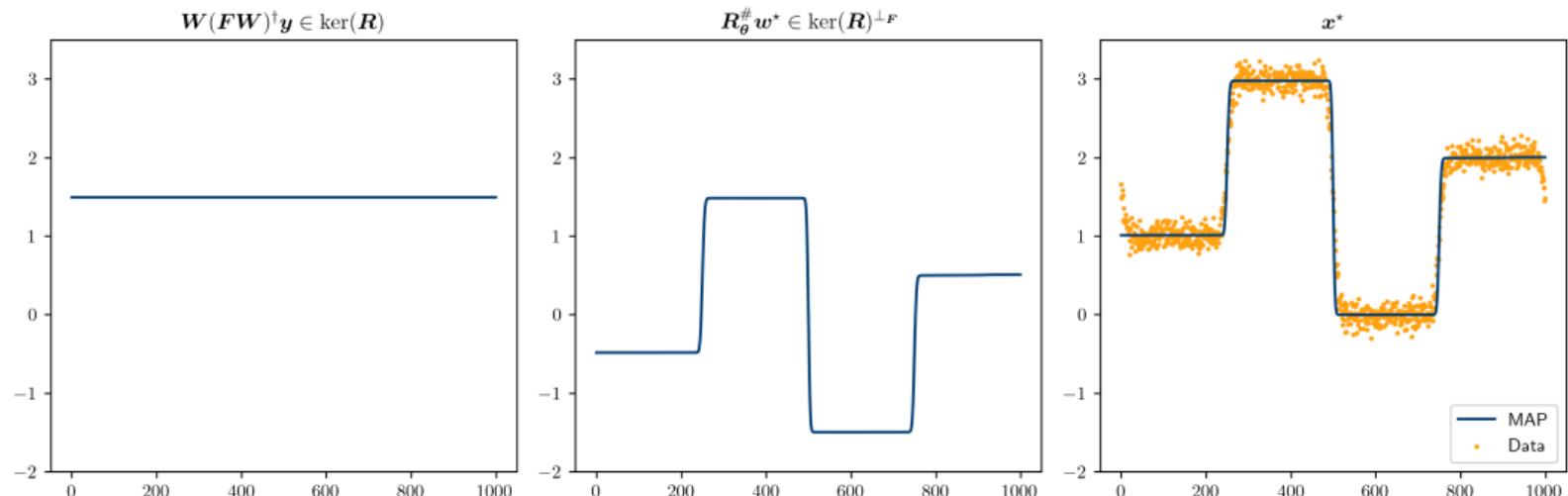
$$\mathbf{R}_{\theta}^{\#} = (\mathbf{I}_n - \mathbf{W}(\mathbf{F}\mathbf{W})^{\dagger}\mathbf{F}) \mathbf{R}_{\theta}^{\dagger}$$

We make the change of variables

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_{\theta} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_k \end{bmatrix} \quad \mathbf{x} = \mathbf{R}_{\theta}^{\#} \mathbf{w} \rightarrow \begin{bmatrix} \mathbf{F} \mathbf{R}_{\theta}^{\#} \\ \mathbf{I}_k \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_k \end{bmatrix}$$

Then, the solution is given by

$$\mathbf{x}^* = \underbrace{\mathbf{R}_{\theta}^{\#} \mathbf{w}^*}_{\in \ker(\mathbf{R})^{\perp_F}} + \underbrace{\mathbf{W}(\mathbf{F}\mathbf{W})^{\dagger} \mathbf{y}}_{\in \ker(\mathbf{R})}$$



In terms of linear systems?

If \mathbf{R}^{-1} exists:

$$Q_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_\theta^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_\theta^{-1} w} \quad Q_t = \mathbf{R}_\theta^{-T} \mathbf{F}^T \mathbf{F} \mathbf{R}_\theta^{-1} + \mathbf{I}_n$$

If $\ker(\mathbf{R}) = \{\mathbf{0}_n\}$:

$$Q_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_\theta^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_\theta^\dagger w} \quad Q_t = (\mathbf{R}_\theta^\dagger)^T \mathbf{F}^T \mathbf{F} \mathbf{R}_\theta^\dagger + \mathbf{I}_k$$

If $\ker(\mathbf{R}) \neq \{\mathbf{0}_n\}$:

$$Q_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_\theta^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_\theta^\# w} \quad Q_t = (\mathbf{R}_\theta^\#)^T \mathbf{F}^T \mathbf{F} \mathbf{R}_\theta^\# + \mathbf{I}_k$$

In terms of linear systems?

If \mathbf{R}^{-1} exists:

$$\mathbf{Q}_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_{\theta}^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_{\theta}^{-1} w} \quad \mathbf{Q}_t = \mathbf{R}_{\theta}^{-T} \mathbf{F}^T \mathbf{F} \mathbf{R}_{\theta}^{-1} + \mathbf{I}_n$$

If $\ker(\mathbf{R}) = \{\mathbf{0}_n\}$:

$$\mathbf{Q}_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_{\theta}^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_{\theta}^{\dagger} w} \quad \mathbf{Q}_t = (\mathbf{R}_{\theta}^{\dagger})^T \mathbf{F}^T \mathbf{F} \mathbf{R}_{\theta}^{\dagger} + \mathbf{I}_k$$

If $\ker(\mathbf{R}) \neq \{\mathbf{0}_n\}$:

$$\mathbf{Q}_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_{\theta}^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_{\theta}^{\#} w} \quad \mathbf{Q}_t = (\mathbf{R}_{\theta}^{\#})^T \mathbf{F}^T \mathbf{F} \mathbf{R}_{\theta}^{\#} + \mathbf{I}_k$$

In terms of linear systems?

If \mathbf{R}^{-1} exists:

$$\mathbf{Q}_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_{\theta}^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_{\theta}^{-1} w} \quad \mathbf{Q}_t = \mathbf{R}_{\theta}^{-T} \mathbf{F}^T \mathbf{F} \mathbf{R}_{\theta}^{-1} + \mathbf{I}_n$$

If $\ker(\mathbf{R}) = \{\mathbf{0}_n\}$:

$$\mathbf{Q}_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_{\theta}^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_{\theta}^{\dagger} w} \quad \mathbf{Q}_t = (\mathbf{R}_{\theta}^{\dagger})^T \mathbf{F}^T \mathbf{F} \mathbf{R}_{\theta}^{\dagger} + \mathbf{I}_k$$

If $\ker(\mathbf{R}) \neq \{\mathbf{0}_n\}$:

$$\mathbf{Q}_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_{\theta}^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_{\theta}^{\#} w} \quad \mathbf{Q}_t = (\mathbf{R}_{\theta}^{\#})^T \mathbf{F}^T \mathbf{F} \mathbf{R}_{\theta}^{\#} + \mathbf{I}_k$$

In terms of linear systems?

If \mathbf{R}^{-1} exists:

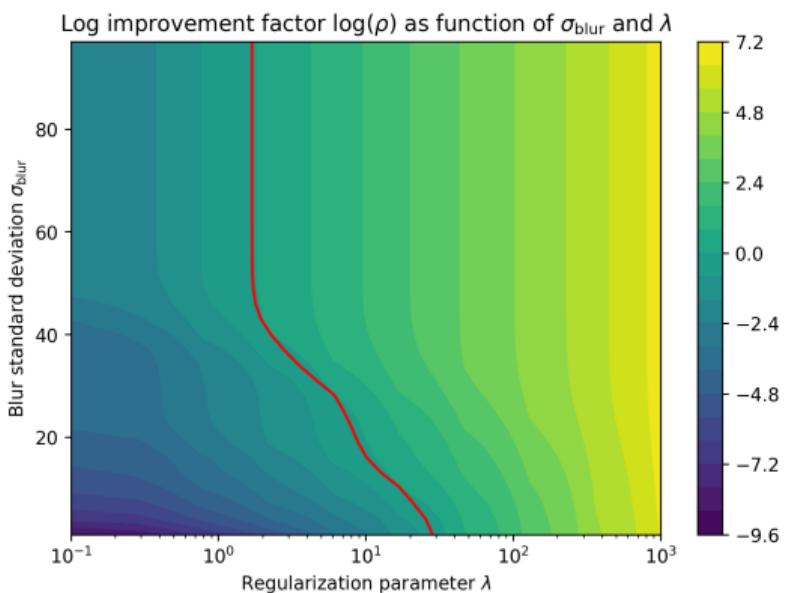
$$\mathbf{Q}_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_{\theta}^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_{\theta}^{-1} w} \quad \mathbf{Q}_t = \mathbf{R}_{\theta}^{-T} \mathbf{F}^T \mathbf{F} \mathbf{R}_{\theta}^{-1} + \mathbf{I}_n$$

If $\ker(\mathbf{R}) = \{\mathbf{0}_n\}$:

$$\mathbf{Q}_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_{\theta}^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_{\theta}^{\dagger} w} \quad \mathbf{Q}_t = (\mathbf{R}_{\theta}^{\dagger})^T \mathbf{F}^T \mathbf{F} \mathbf{R}_{\theta}^{\dagger} + \mathbf{I}_k$$

If $\ker(\mathbf{R}) \neq \{\mathbf{0}_n\}$:

$$\mathbf{Q}_0 = \mathbf{F}^T \mathbf{F} + \mathbf{R}^T \mathbf{D}_{\theta}^{-1} \mathbf{R} \quad \xrightarrow{x=\mathbf{R}_{\theta}^{\#} w} \quad \mathbf{Q}_t = (\mathbf{R}_{\theta}^{\#})^T \mathbf{F}^T \mathbf{F} \mathbf{R}_{\theta}^{\#} + \mathbf{I}_k$$



Consider the condition number improvement factor

$$\rho = \frac{\kappa(Q_0)}{\kappa(Q_t)}$$

- $\rho > 1 \Rightarrow$ priorconditioning gives a good preconditioner
- $\rho = 1 \Rightarrow$ indifferent
- $\rho < 1 \Rightarrow$ priorconditioning worsens the conditioning

Implementing pseudoinverses

To implement prior conditioning efficiently, we need a good way to compute matvecs and rmatvecs with

$$\mathbf{R}_{\theta}^{\#} = \left(\mathbf{I}_n - \mathbf{W}(\mathbf{F}\mathbf{W})^{\dagger}\mathbf{F} \right) \mathbf{R}_{\theta}^{\dagger}$$

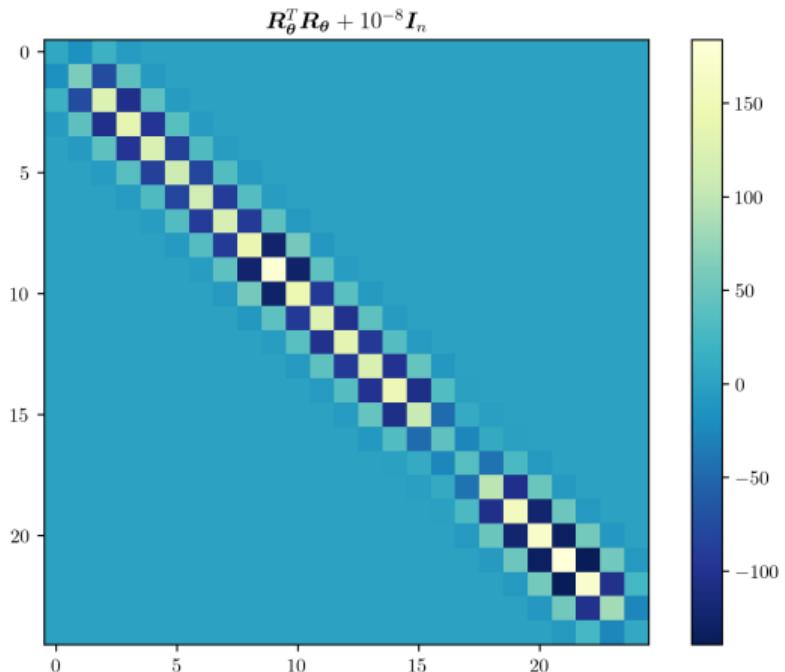
- In typical cases $\dim(\ker(\mathbf{R}))$ is small, so \mathbf{W} is "skinny" and $(\mathbf{F}\mathbf{W})^{\dagger}$ can be computed quickly using a QR factorization (and only once)
- However, the QR approach is very costly for $\mathbf{R}_{\theta}^{\dagger}$ since it must be re-computed for each θ , and $\mathbf{R}_{\theta} \in \mathbb{R}^{k \times n}$ with $k \approx n$ (or larger)

One strategy is to use the approximations

$$\mathbf{R}_{\theta}^{\dagger} \approx (\mathbf{R}_{\theta}^T \mathbf{R}_{\theta} + \delta \mathbf{I}_n)^{-1} \mathbf{R}_{\theta}^T,$$

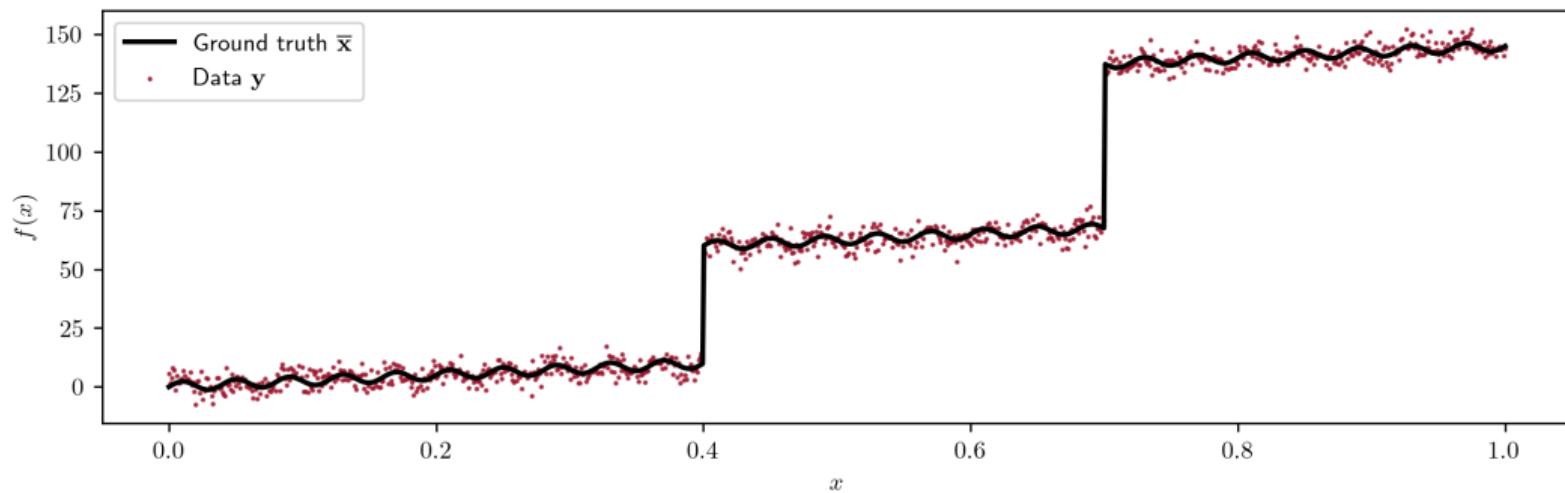
$$(\mathbf{R}_{\theta}^{\dagger})^T \approx \mathbf{R}_{\theta} (\mathbf{R}_{\theta}^T \mathbf{R}_{\theta} + \delta \mathbf{I}_n)^{-1}$$

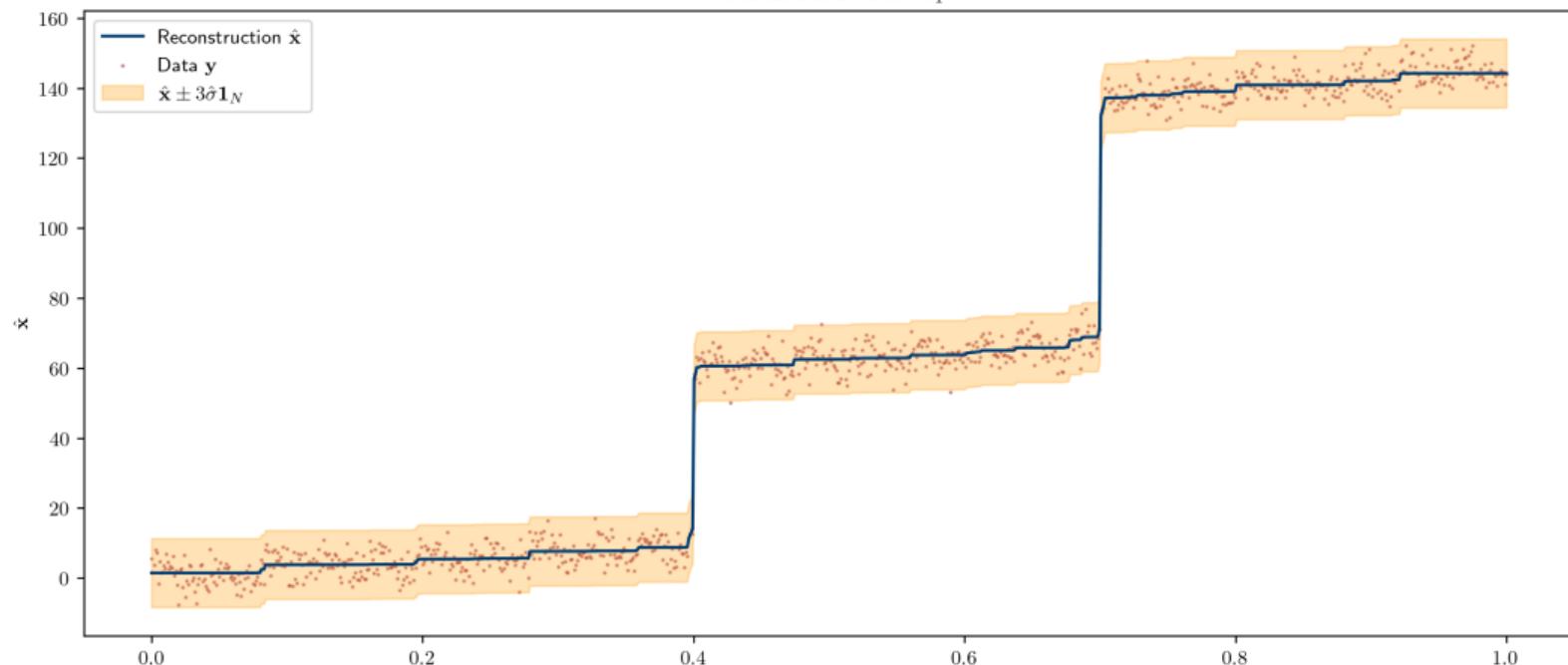
- If $\mathbf{R}_{\theta}^T \mathbf{R}_{\theta}$ has bandwidth p , costs $\mathcal{O}(p^2 n)$ flops using a sparse Cholesky routine 
- Re-computed for each new θ 

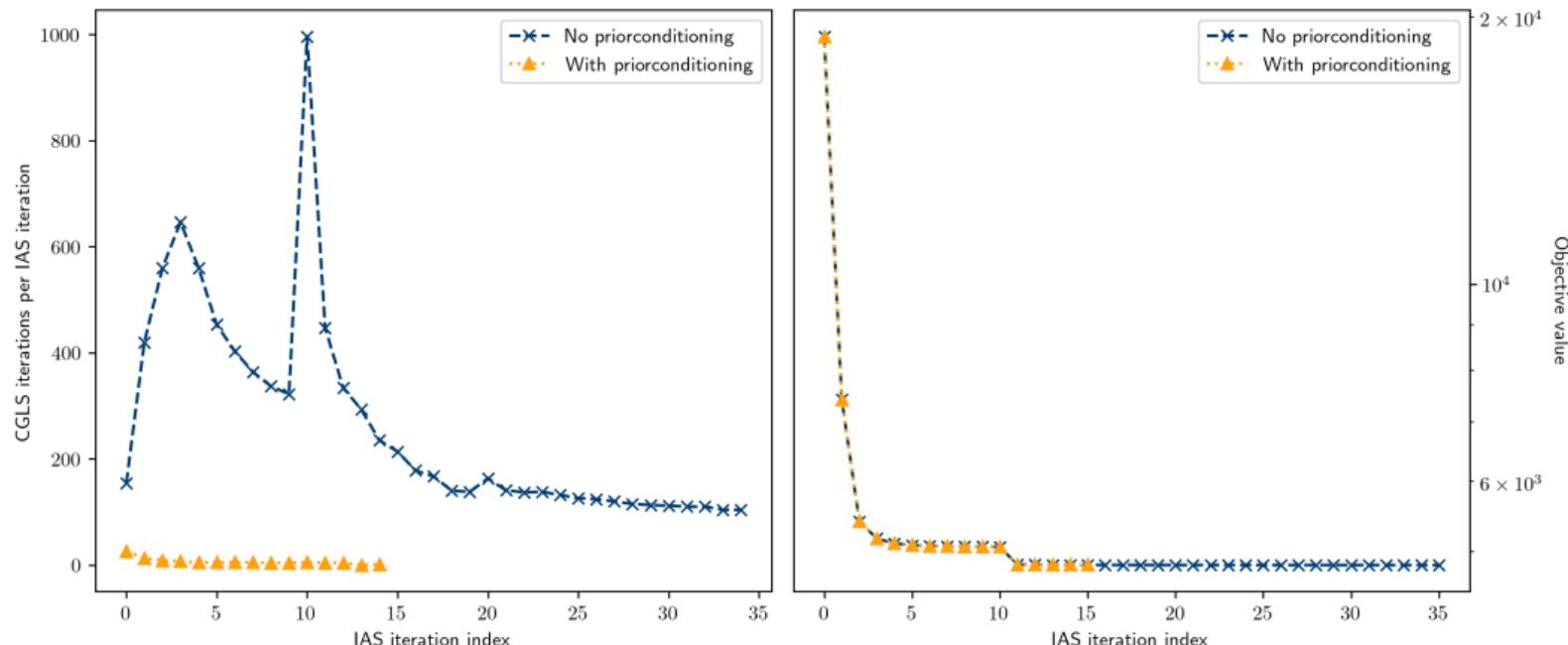


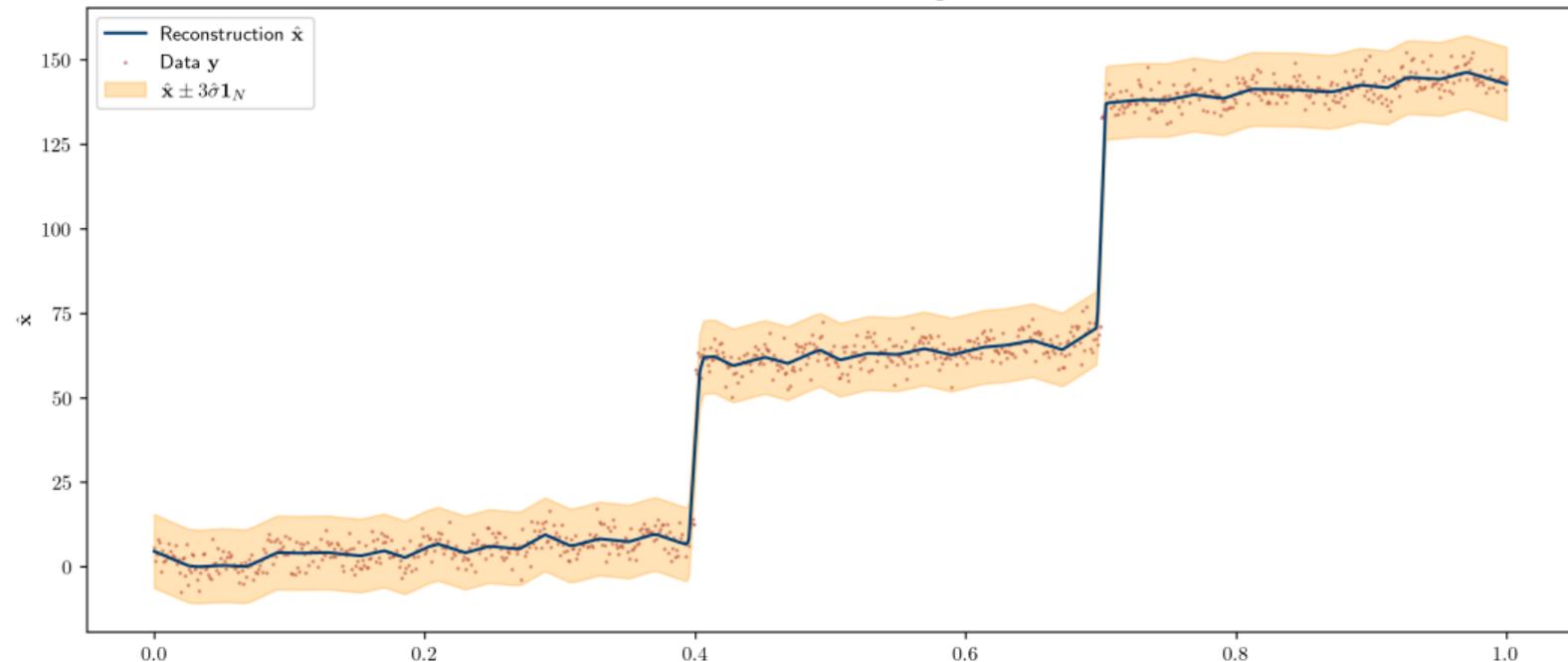
$$\mathbf{R}_1 = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \quad \mathbf{R}_2 = \begin{bmatrix} -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & 1 \end{bmatrix} \quad \mathbf{R}_3 = \begin{bmatrix} -1 & 3 & -3 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & 3 & -3 & 1 \end{bmatrix}$$

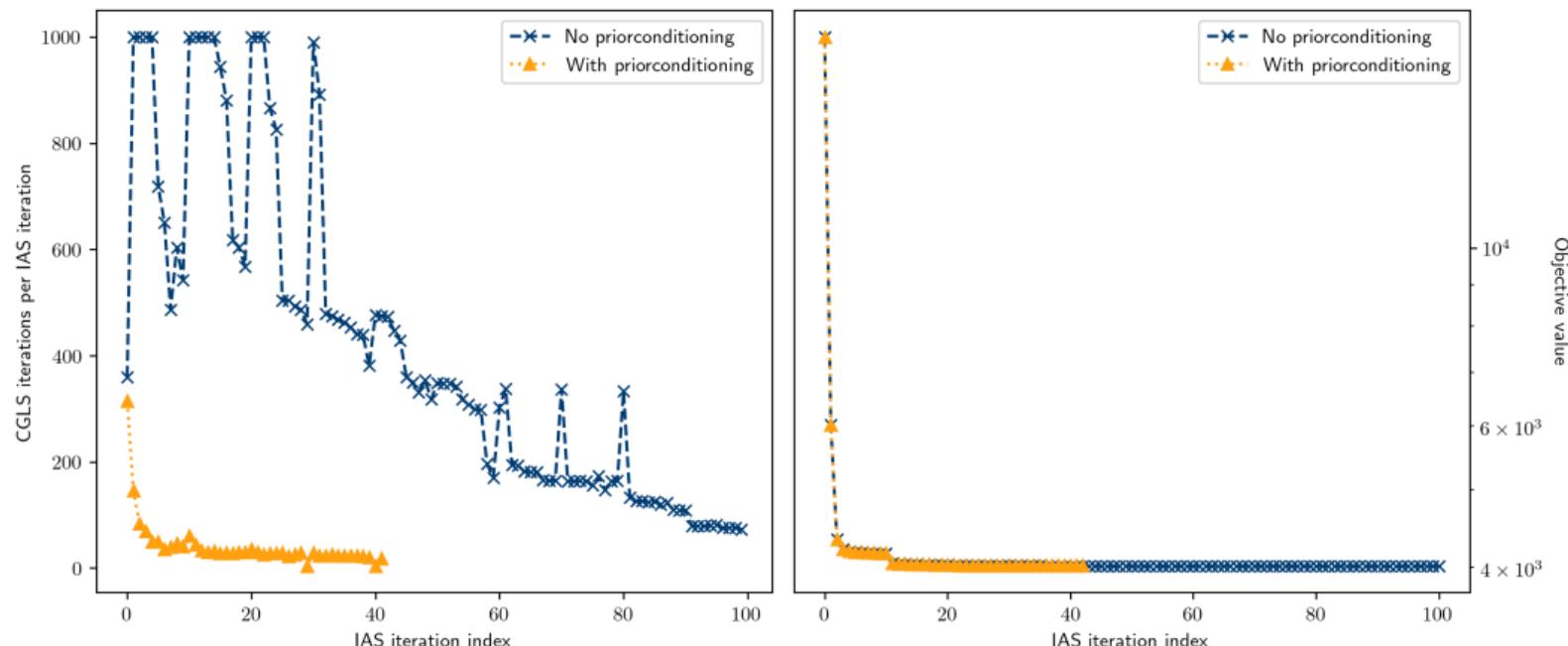
$$\mathbf{W}_1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \quad \mathbf{W}_2 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 1 & n-1 \\ 1 & n \end{bmatrix} \quad \mathbf{W}_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \\ \vdots & \vdots & \vdots \\ 1 & n-2 & \\ 1 & n-1 & n(n-1)/2 \\ 1 & n & n(n+1)/2 \end{bmatrix}$$

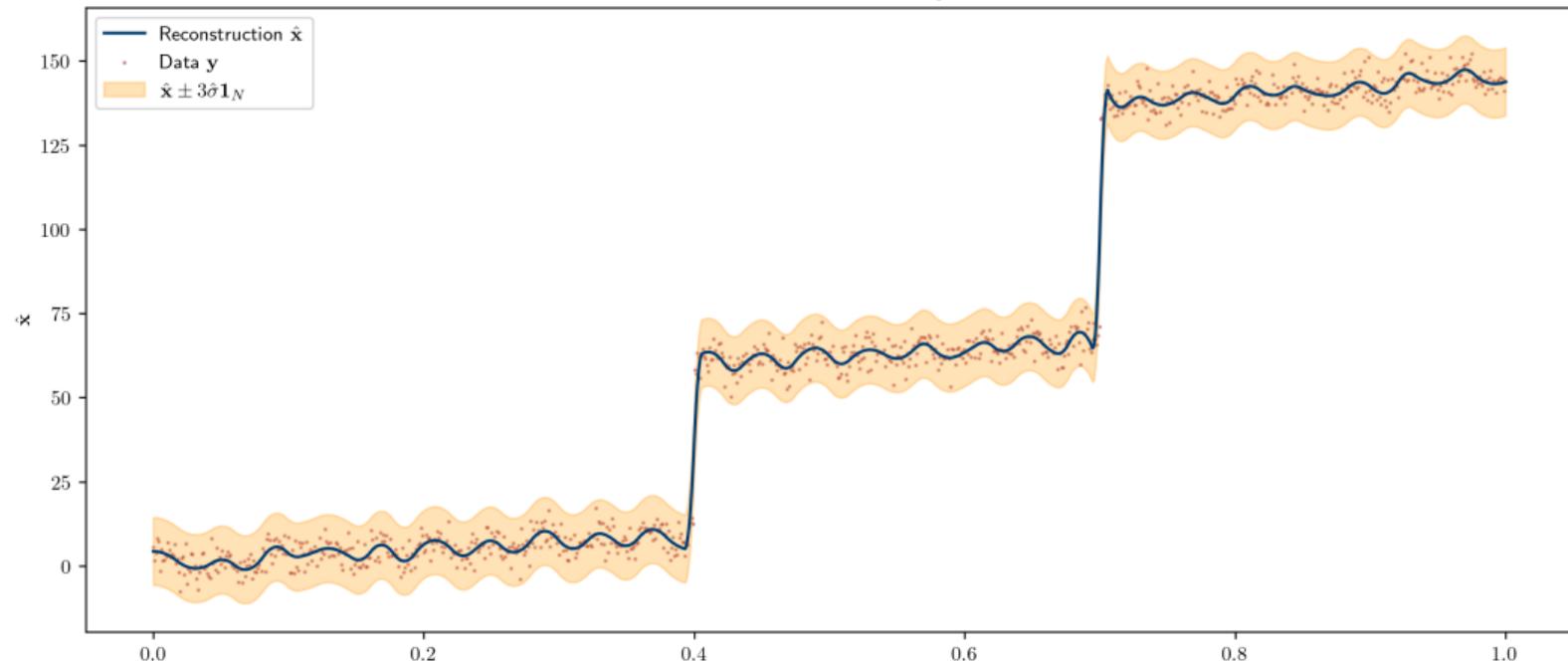


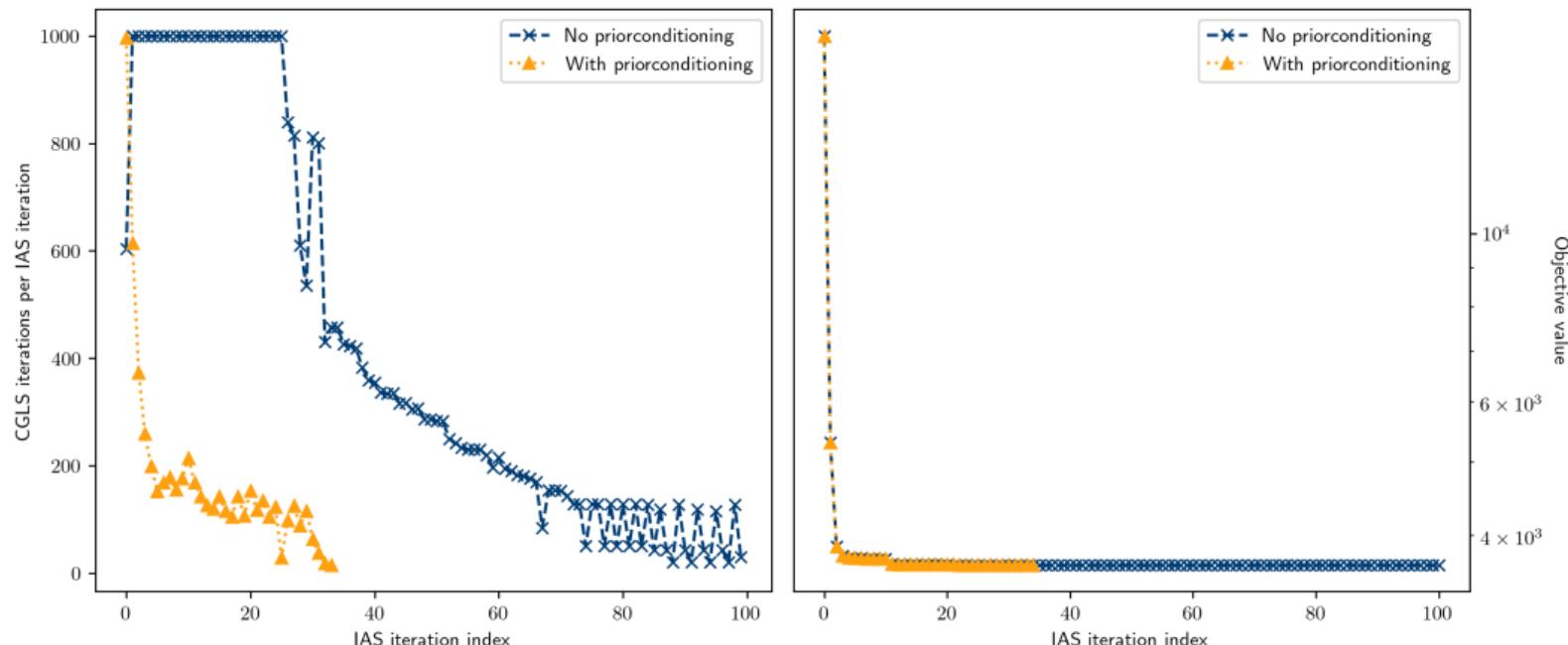
Reconstruction with R_1 

Reconstruction with R_1 

Reconstruction with R_2 

Reconstruction with R_2 

Reconstruction with R_3 

Reconstruction with R_3 

However, the factorization approach is infeasible for large problems (even if sparse).

An alternative approach is to use an iterative method for computing pseudoinverses.

Idea: the matrix-vector product $\mathbf{z} = \mathbf{R}^\dagger \mathbf{v}$ can be computed approximately by applying the CG method to the solution of

$$\mathbf{R}^T \mathbf{R} \mathbf{z} = \mathbf{R}^T \mathbf{v}$$

with any initialization $\mathbf{z}^{(0)} \in \text{col}(\mathbf{R}^T)$, e.g., $\mathbf{z}^{(0)} = \mathbf{0}_n$.

However, the factorization approach is infeasible for large problems (even if sparse).

An alternative approach is to use an iterative method for computing pseudoinverses.

Idea: the matrix-vector product $\mathbf{z} = \mathbf{R}^\dagger \mathbf{v}$ can be computed approximately by applying the CG method to the solution of

$$\mathbf{R}^T \mathbf{R} \mathbf{z} = \mathbf{R}^T \mathbf{v}$$

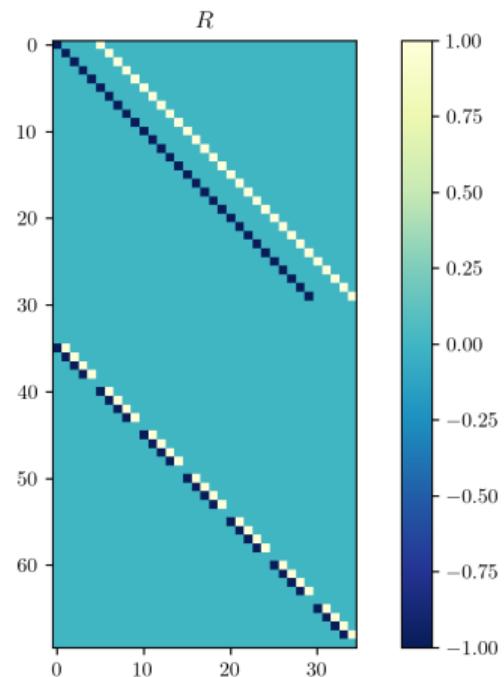
with any initialization $\mathbf{z}^{(0)} \in \text{col}(\mathbf{R}^T)$, e.g., $\mathbf{z}^{(0)} = \mathbf{0}_n$.

Case study: \mathbf{R} = discrete gradient with Neumann boundary

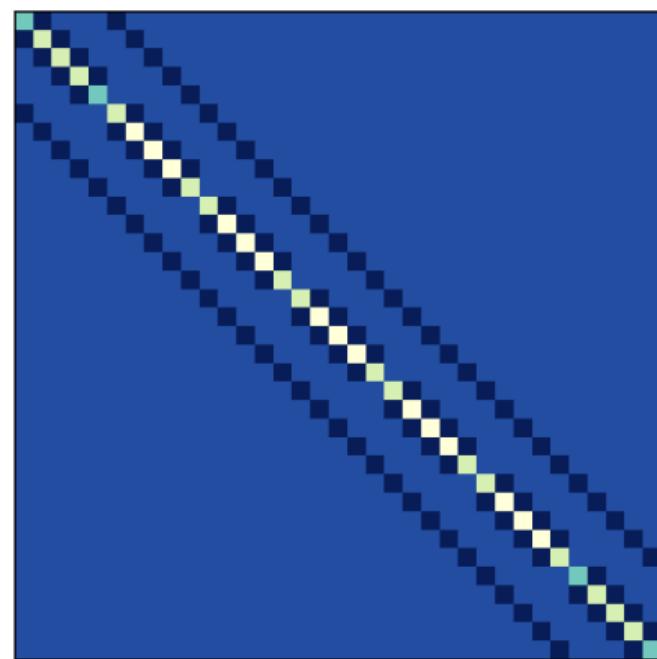
$$\mathbf{R}_l = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & & 0 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{n_1} \otimes \mathbf{I}_{n_2} \\ \mathbf{I}_{n_1} \otimes \mathbf{R}_{n_2} \end{bmatrix}$$

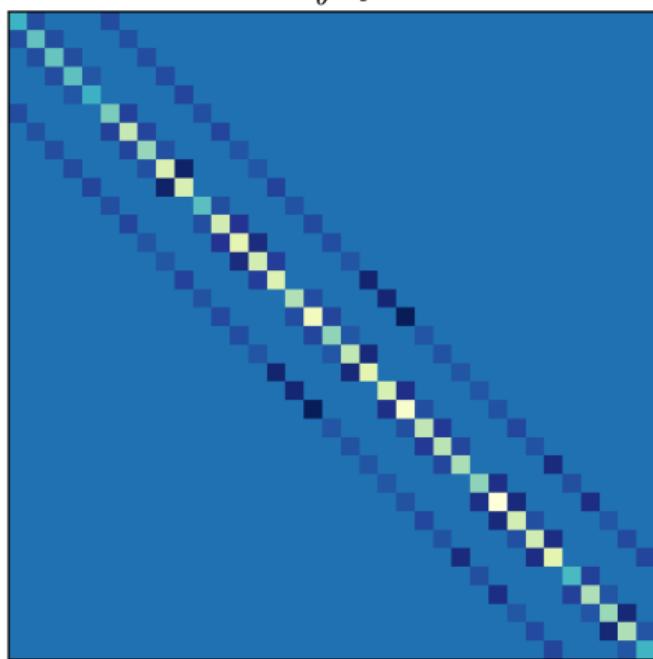
$$\mathbf{1}_{n_1 n_2} \in \ker(\mathbf{R})$$



$$\mathbf{R}^T \mathbf{R}$$



$$\mathbf{R}_\theta^T \mathbf{R}_\theta$$



$$\mathbf{R}^T \mathbf{R} = \mathbf{B}_{\text{DCT}}^T \boldsymbol{\Lambda} \mathbf{B}_{\text{DCT}}$$

$$\mathbf{R}_\theta^T \mathbf{R}_\theta = ???$$

Preconditioning

Can we use the unweighted Laplacian as a preconditioner for the CG method computing $z = R_\theta^\dagger v$?

Lemma B.1. Let $A, M \in \mathbb{R}^{N \times N}$ be symmetric matrices such that $A, M \succeq 0$ (the matrices are positive semi-definite) and $\text{col}(M) = \text{col}(A)$, and let $\mathbf{b} \in \text{col}(A)$. Let L be any matrix such that $M = LL^T$, such as (but not limited to) the spectral square root. Then the problem

$$(B.3) \quad \text{find } \mathbf{x} \in \text{col}(A) \text{ s.t. } A\mathbf{x} = \mathbf{b}$$

has a unique solution \mathbf{x}^* , which is also given by $\mathbf{x}^* = (L^T)^\dagger \mathbf{z}^*$ where \mathbf{z}^* is the unique solution to the problem

$$(B.4) \quad \text{find } \mathbf{z} \in \text{col}(A) \text{ s.t. } L^\dagger A(L^T)^\dagger \mathbf{z} = L^\dagger \mathbf{b}.$$

Preconditioning

Algorithm B.2 Preconditioned conjugate gradient method for SPD system.

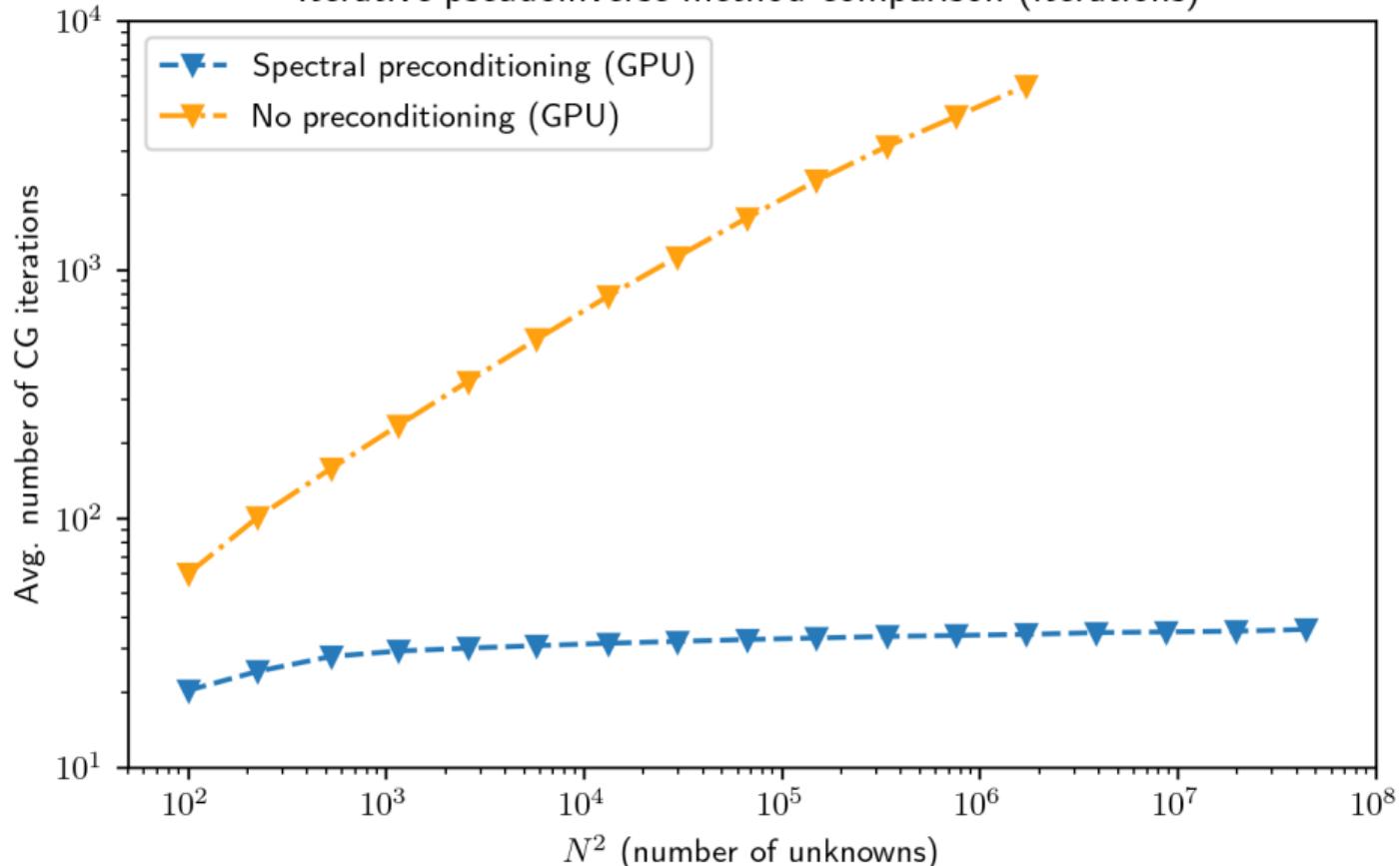
- 1: **Input:** SPD $A \in \mathbb{R}^{N \times N}$, $\mathbf{b} \in \mathbb{R}^N$, $\mathbf{x}^{(0)} \in \mathbb{R}^N$, **SPD** $M \in \mathbb{R}^{N \times N}$
- 2: **Output:** Solution $\mathbf{x}^* \approx A^{-1}\mathbf{b}$
- 3: $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$
- 4: $\mathbf{z}^{(0)} = M^{-1}\mathbf{r}^{(0)}$
- 5: $\mathbf{d}^{(0)} = \mathbf{z}^{(0)}$
- 6: $k = 0$
- 7: **repeat**
- 8: $\alpha^{(k)} = (\mathbf{r}^{(k)})^T \mathbf{z}^{(k)} / (\mathbf{d}^{(k)})^T A \mathbf{d}^{(k)}$
- 9: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$
- 10: $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha^{(k)} A \mathbf{d}^{(k)}$
- 11: $\mathbf{z}^{(k+1)} = M^{-1}\mathbf{r}^{(k+1)}$
- 12: $\beta^{(k+1)} = (\mathbf{r}^{(k+1)})^T \mathbf{z}^{(k+1)} / (\mathbf{r}^{(k)})^T \mathbf{z}^{(k)}$
- 13: $\mathbf{d}^{(k+1)} = \mathbf{z}^{(k+1)} + \beta^{(k)} \mathbf{d}^{(k)}$
- 14: $k = k + 1$
- 15: **until** convergence criterion satisfied
- 16: **return** \mathbf{x}^*

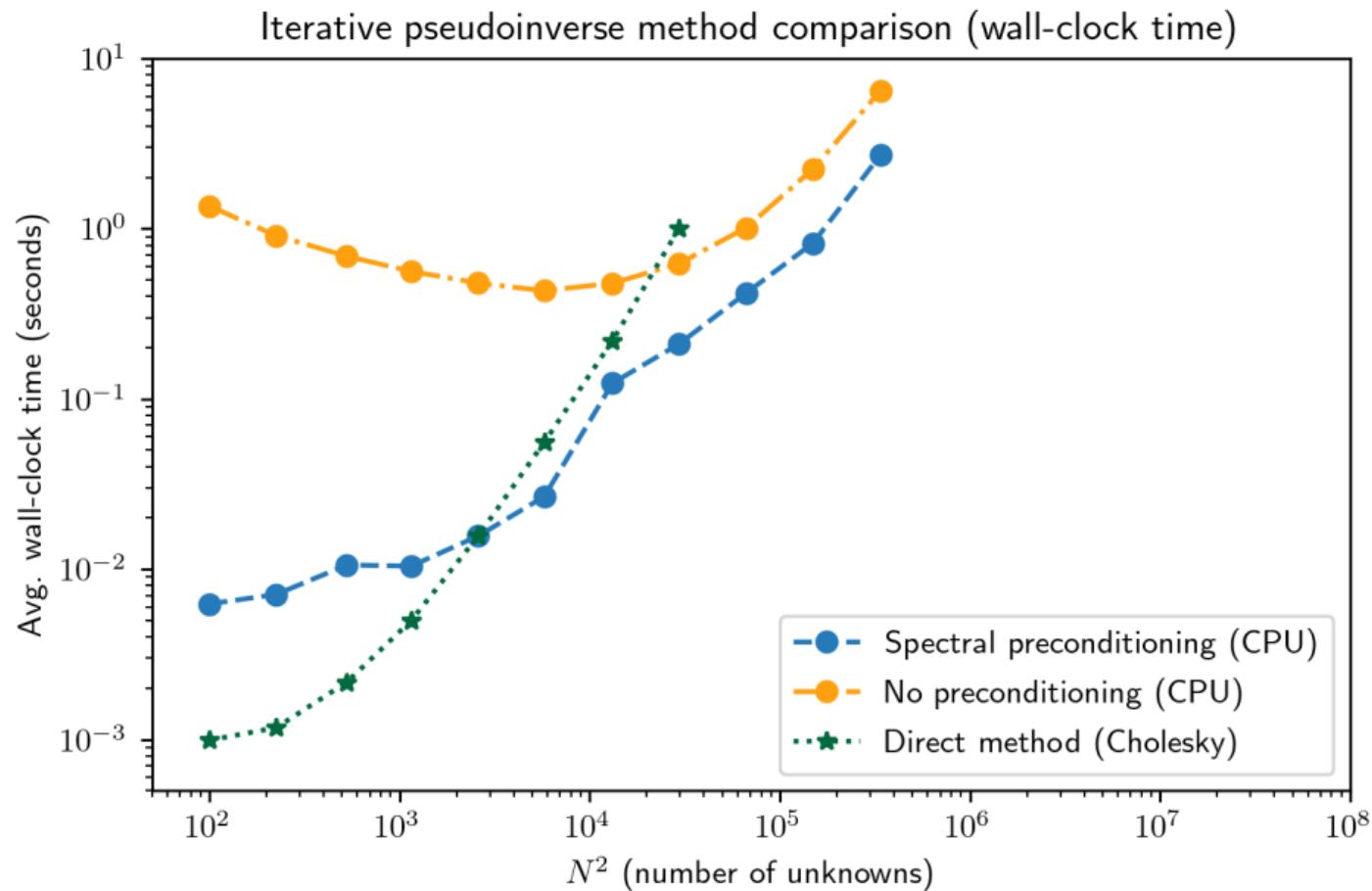
Preconditioning

Algorithm B.1 Preconditioned conjugate gradient method for singular system.

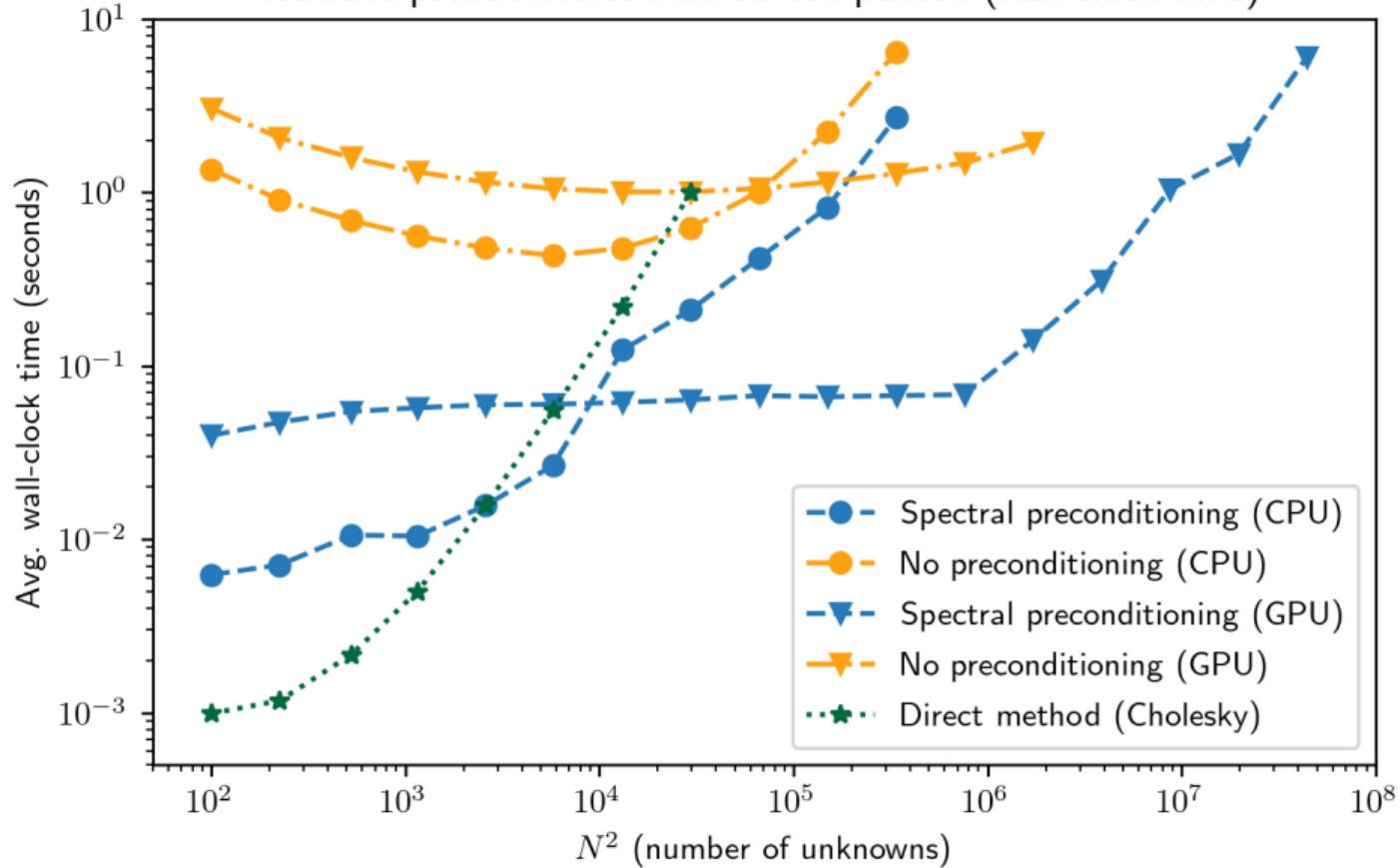
```
1: Input: SSPD  $A \in \mathbb{R}^{N \times N}$ ,  $\mathbf{b} \in \text{col}(A)$ ,  $\mathbf{x}^{(0)} \in \text{col}(A)$ , SSPD  $M \in \mathbb{R}^{N \times N}$  s.t.  $\text{col}(M) = \text{col}(A)$ 
2: Output: Solution  $\mathbf{x}^* \approx A^\dagger \mathbf{b}$ 
3:  $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$ 
4:  $\mathbf{z}^{(0)} = M^\dagger \mathbf{r}^{(0)}$ 
5:  $\mathbf{d}^{(0)} = \mathbf{z}^{(0)}$ 
6:  $k = 0$ 
7: repeat
8:    $\alpha^{(k)} = (\mathbf{r}^{(k)})^T \mathbf{z}^{(k)} / (\mathbf{d}^{(k)})^T A \mathbf{d}^{(k)}$ 
9:    $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$ 
10:   $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha^{(k)} A \mathbf{d}^{(k)}$ 
11:   $\mathbf{z}^{(k+1)} = M^\dagger \mathbf{r}^{(k+1)}$ 
12:   $\beta^{(k+1)} = (\mathbf{r}^{(k+1)})^T \mathbf{z}^{(k+1)} / (\mathbf{r}^{(k)})^T \mathbf{z}^{(k)}$ 
13:   $\mathbf{d}^{(k+1)} = \mathbf{z}^{(k+1)} + \beta^{(k)} \mathbf{d}^{(k)}$ 
14:   $k = k + 1$ 
15: until convergence criterion satisfied
16: return  $\mathbf{x}^*$ 
```

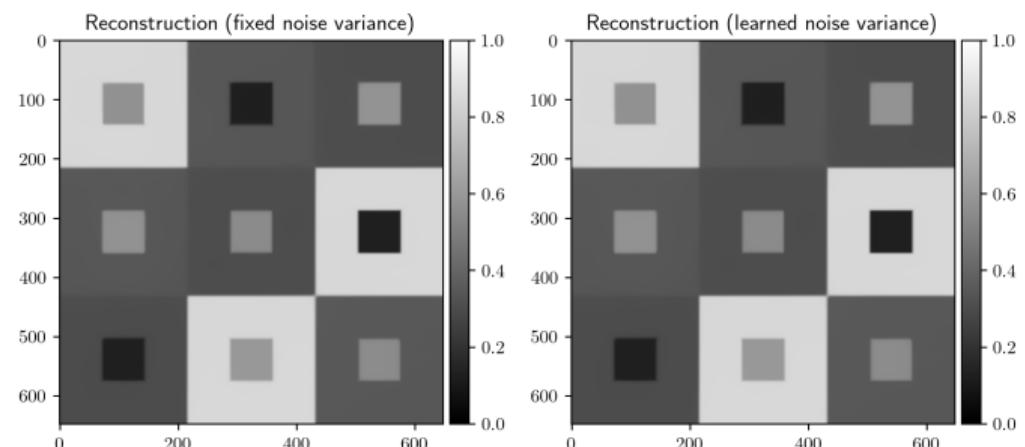
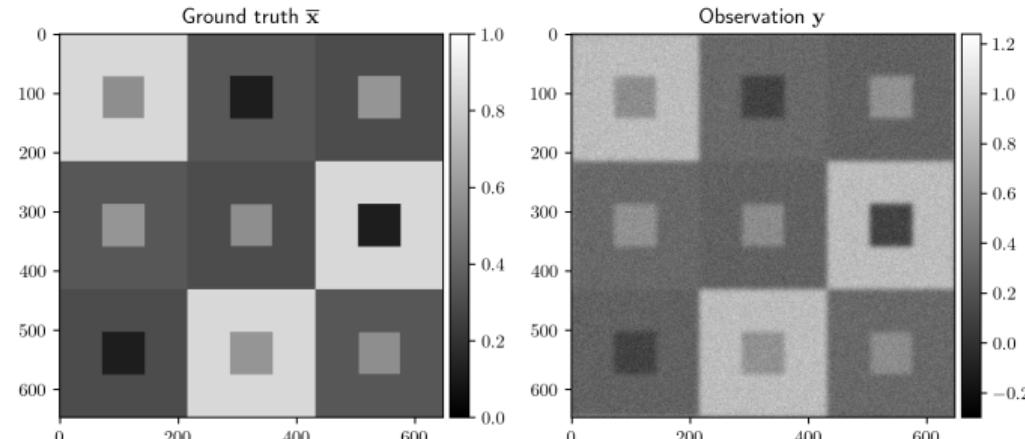
Iterative pseudoinverse method comparison (iterations)





Iterative pseudoinverse method comparison (wall-clock time)





APPLICATION TO SAMPLING

Posterior sampling

- Challenging since high-dimensional, strong correlations, and multimodal
- Gibbs approaches (with or without de-correlating transformation such as non-centering) rely on efficient sampling of the conditional $\boldsymbol{x} | \boldsymbol{\theta}$

Perturbation optimization

Idea: drawing a sample from the conditional $\boldsymbol{x} | \boldsymbol{\theta}$ can be cast as solving a randomized linear system / least squares problem / quadratic optimization problem.

$$\left(\mathbf{F}^T \mathbf{F} + \mathbf{R}_{\boldsymbol{\theta}}^T \mathbf{R}_{\boldsymbol{\theta}} \right) \hat{\boldsymbol{x}} = \mathbf{F}^T (\mathbf{y} + \mathcal{N}(\mathbf{0}_m, \mathbf{I}_m)) + \mathbf{R}_{\boldsymbol{\theta}}^T \mathcal{N}(\mathbf{0}_k, \mathbf{I}_k)$$

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R}_{\boldsymbol{\theta}} \end{bmatrix} \hat{\boldsymbol{x}} = \begin{bmatrix} \mathbf{y} + \mathcal{N}(\mathbf{0}_m, \mathbf{I}_m) \\ \mathcal{N}(\mathbf{0}_k, \mathbf{I}_k) \end{bmatrix}$$

$$\hat{\boldsymbol{x}} = \arg \min_{\boldsymbol{x} \in \mathbb{R}^n} \left\| \mathbf{F}\boldsymbol{x} - \mathbf{y} - \mathcal{N}(\mathbf{0}_m, \mathbf{I}_m) \right\|_2^2 + \left\| \mathbf{R}_{\boldsymbol{\theta}}\boldsymbol{x} - \mathcal{N}(\mathbf{0}_k, \mathbf{I}_k) \right\|_2^2$$

Lemma (standard form for sampling)

The least squares solution \mathbf{x}^* of the system

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{R} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

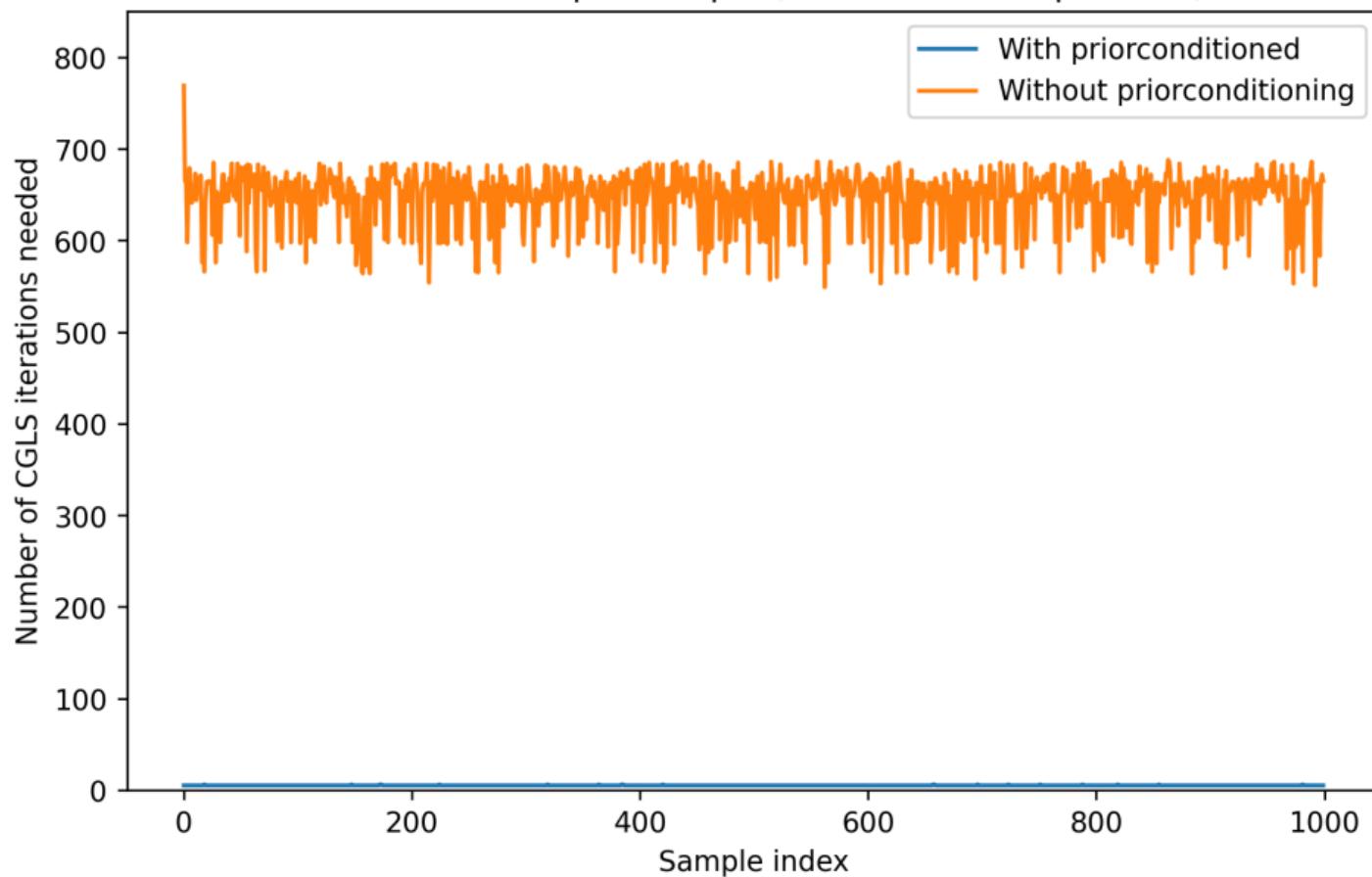
can be expressed as $\mathbf{x} = \mathbf{x}_{\text{ker}} + \mathbf{x}_{\perp}$, where

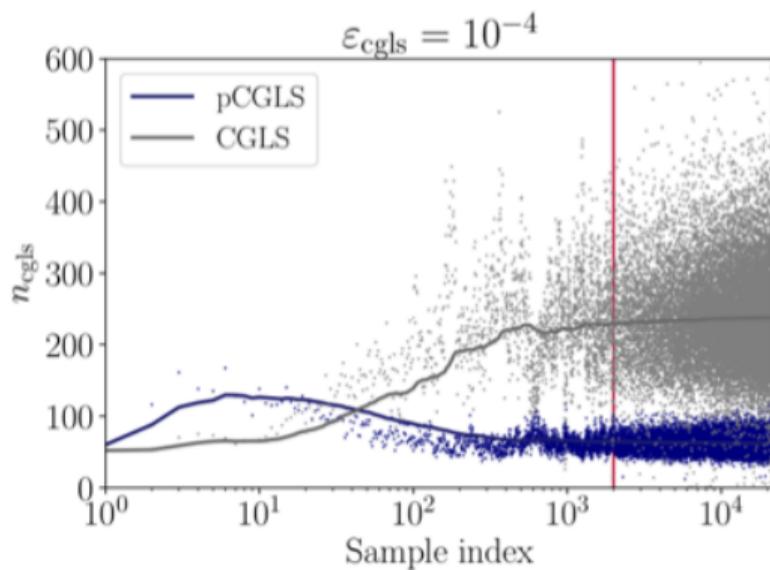
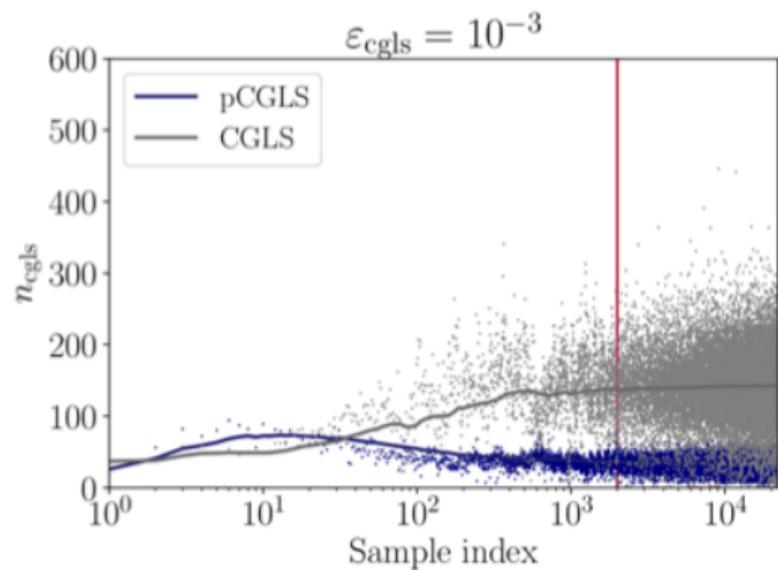
$$\mathbf{x}_{\text{ker}} = \mathbf{W}(\mathbf{F}\mathbf{W})^\dagger \mathbf{b}_1, \quad \mathbf{x}_{\perp} = \mathbf{R}^\# \mathbf{z},$$

and \mathbf{z}^* is the least squares solution of

$$\begin{bmatrix} \mathbf{F}\mathbf{R}^\# \\ \mathbf{I}_k \end{bmatrix} \mathbf{z} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{R}\mathbf{R}^\dagger \mathbf{b}_2 \end{bmatrix}.$$

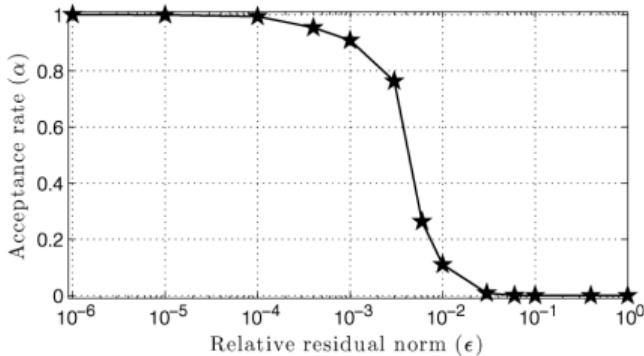
CGLS iterations per sample (1K dimensional problem)





Reversible jump perturbation optimization (RJPO)

- Solution via direct method can be very costly (Cholesky factorizations)
- Approximate solution by an iterative method may be cheaper, but will not preserve the stationary distribution
- It is possible to equip these methods with a simple accept-reject step based on the residual norm (CG sampler) to correct this bias

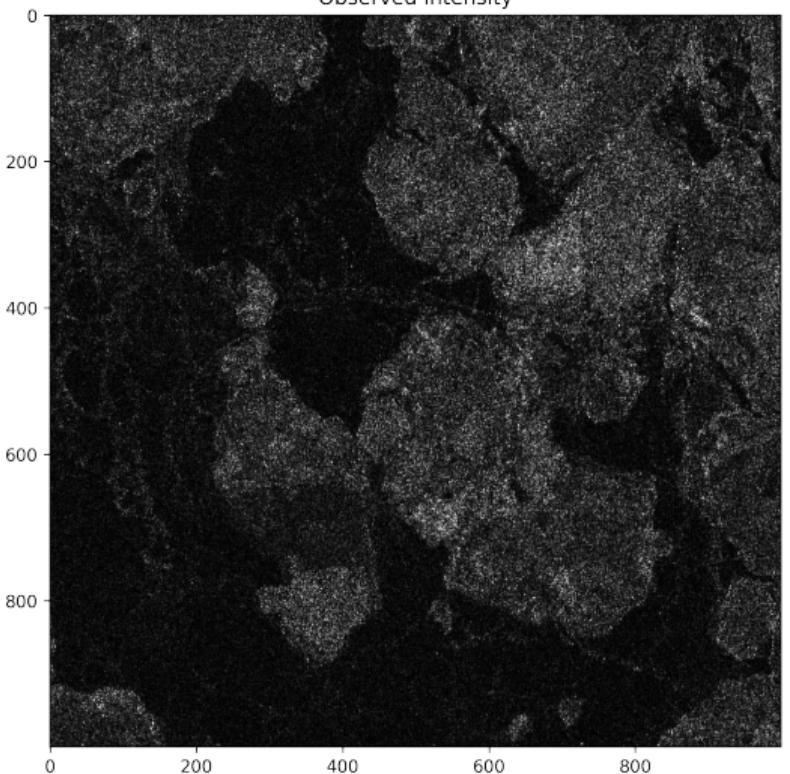


OTHER DIRECTIONS

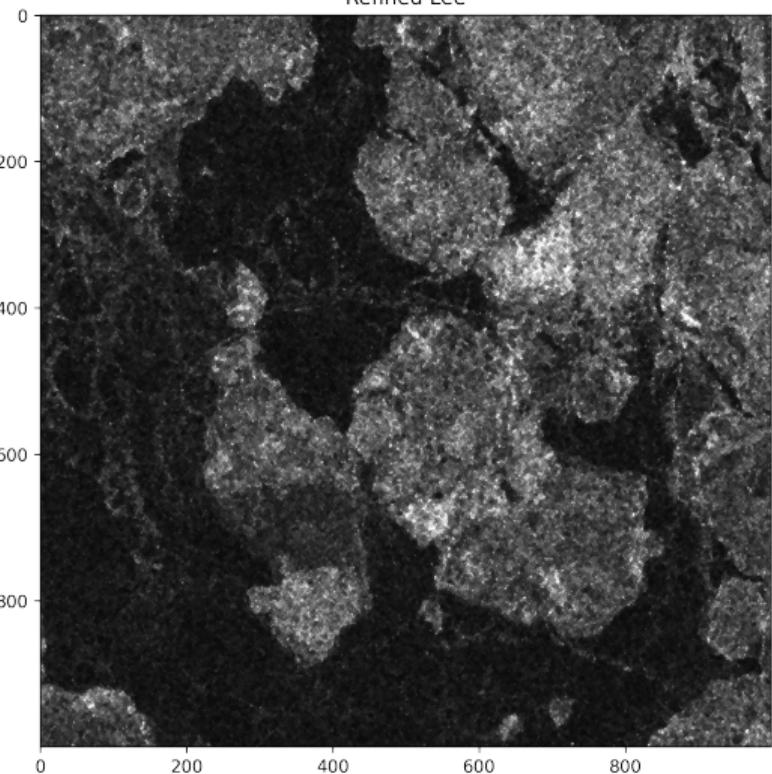


Sentinel-1, courtesy of ESA

Observed intensity



Refined Lee



Hierarchical model for multiplicative denoising

$$\begin{aligned} [\mathbf{f}]_i | \mathbf{u} &\stackrel{\text{ind}}{\sim} \text{Gamma}\left(L, \frac{L}{[\mathbf{A}\mathbf{u}]_i}\right), \quad i = 1, \dots, m, \\ \mathbf{u} | \boldsymbol{\theta} &\sim \mathcal{N}\left(\mathbf{0}, \left(\mathbf{R}^T \mathbf{D}_{\boldsymbol{\theta}}^{-1} \mathbf{R}\right)^{-1}\right), \\ [\boldsymbol{\theta}]_i &\stackrel{\text{iid}}{\sim} \mathcal{GG}(r, \beta, \vartheta), \quad i = 1, \dots, k. \end{aligned}$$

$$\mathcal{G}(\mathbf{u}, \boldsymbol{\theta}) = \sum_{i=1}^m L \log ([\mathbf{A}\mathbf{u}]_i) + L \frac{[\mathbf{f}]_i}{[\mathbf{A}\mathbf{u}]_i} + \frac{1}{2} \|\mathbf{D}_{\boldsymbol{\theta}}^{-1/2} \mathbf{R}\mathbf{u}\|_2^2 - \log \pi(\boldsymbol{\theta}),$$

Changes to coordinate descent from Gaussian case?

- The method for performing the θ -update is essentially unchanged.
- The u -update corresponds to finding

$$\arg \min_{\mathbf{u} \in \mathbb{R}_{++}^n} \sum_{i=1}^m L \log ([\mathbf{A}\mathbf{u}]_i) + L \frac{[\mathbf{f}]_i}{[\mathbf{A}\mathbf{u}]_i} + \frac{1}{2} \|\mathbf{D}_{\theta}^{-1/2} \mathbf{R}\mathbf{u}\|_2^2$$

- We can apply Newton's method to this subproblem
- Newton's method can be expressed as solving a sequence of least squares problems, to which priorconditioning can be applied!

