Krylov Subspace Regularization for Inverse Problems

James Nagy, Emory University

Atlanta, Georgia, USA

Work supported in part by the United States National Science Foundation

Aims of this talk

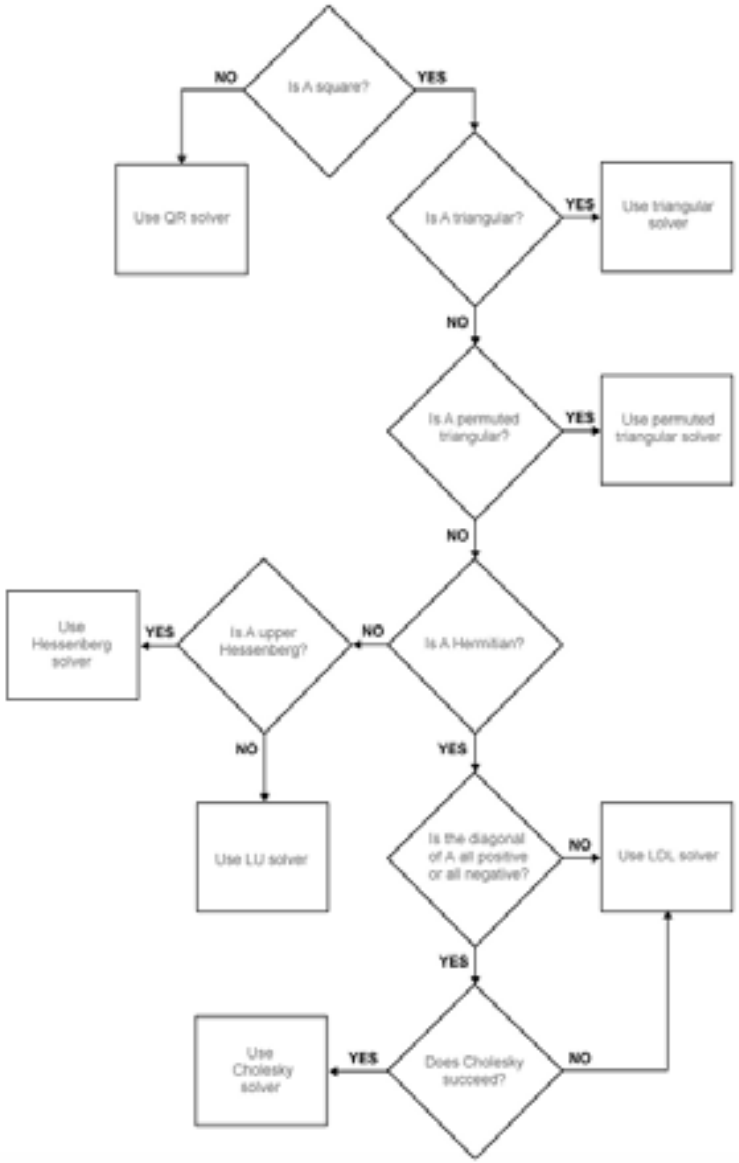
Discuss iterative methods to solve large-scale inverse problems:

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}$$

1. Introduction and Basics
2. Standard Krylov Solvers
3. Hybrid Krylov Solvers
4. Krylov Methods for Sparse Solutions
5. Krylov Methods for Low-Rank Solutions
6. Concluding Remarks

MATLAB backslash operator is amazing!

The simple command $\mathbf{x} = \mathbf{A} \setminus \mathbf{b}$ can solve many different problems.



Backslash for $\mathbf{b} = \mathbf{A}\mathbf{x} + \eta$?

- $\mathbf{b}$ is known vector (measured data)
- $\mathbf{x}$ is unknown vector (want to find this)
- η is unknown vector (noise)
- $\mathbf{A}$ is large, ill-conditioned matrix, and generally
  - large singular values $\leftrightarrow$ low frequency singular vectors
  - small singular values $\leftrightarrow$ high frequency singular vectors

ignore noise, and "solve" $\mathbf{A}\hat{\mathbf{x}} = \mathbf{b} \quad \Rightarrow \quad \hat{\mathbf{x}} = \mathbf{A}\backslash\mathbf{b} = \mathbf{A}^{-1}\mathbf{b} = \mathbf{x} + \mathbf{A}^{-1}\eta \not\approx \mathbf{x}$

Backslash for $\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}$?

Computing approximate requires regularization.

If $\mathbf{A}$ is not too large, can use $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ (SVD), e.g.

- Truncated SVD (TSVD): $\hat{\mathbf{x}} = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i, \quad n = \text{rank}(A),\ k < n$
- Tikhonov: $\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \alpha^2 \|\mathbf{x}\|_2^2 \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$
- General SVD Filtering: $\hat{\mathbf{x}} = \sum_{i=1}^{n} \phi_i(\alpha) \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i \qquad \phi_i(\alpha) \approx \begin{cases} 1, & \text{“for large” } \sigma_i \\ 0, & \text{“for small” } \sigma_i \end{cases}$

Good choice for regularization parameter α depends on problem.

With SVD, many methods to help guide choosing α, such as:

- discrepancy principle
- generalized cross validation

Backslash for $\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}$?

Summary: When **A** is not too large, SVD based methods can be effective.

Good MATLAB software:

Regularization Tools, `http://www.imm.dtu.dk/~pcha/Regutools/`

[Hansen. *Rank-Deficient and Discrete Ill-Posed Problems.* SIAM,1997.]

[Hansen. *Regularization Tools Version 4.0 for Matlab 7.3.* Numerical Algorithms, 2007]

[Hansen. *Discrete Inverse Problems: Insight and Algorithms.* SIAM, 2010]

Backslash for $\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}$?

If $\mathbf{A}$ is large, then we need iterative methods, e.g.,

- Truncated iterations using, e.g., CG, LSQR, GMRES, ...
  - mimics truncated SVD
- Iterative methods to solve variational problems: $\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \alpha\mathcal{R}(\mathbf{x})$
  - Choose your favorite $\mathcal{R}$, e.g., $\|\mathbf{L}\mathbf{x}\|_2^2$, $\mathrm{TV}(\mathbf{x})$, $\|\mathbf{x}\|_1$, ...
  - How to choose α?

One method does not fit all problems, all data.

No simple flowchart for choices like backslash!

But Krylov combined with SVD can help.

Standard Krylov Solvers for Inverse Problems

Framework for Krylov Subspace (e.g., conjugate gradient) methods.
At iteration k:

- Expand the Krylov subspace:
$$\mathbf{A}\mathbf{V}_k = \mathbf{U}_{k+1}\mathbf{T}_k\,,$$
with $\mathbf{T}_k \in \mathbb{R}^{(k+1)\times k}$ (tiny matrix), $\mathbf{V}_k$ and $\mathbf{U}_{k+1}$ orthonormal columns.
- Solve a projected LS problem:
$$\mathbf{y}_k = \arg\min_{\mathbf{y}\in\mathbb{R}^k} \|\mathbf{d}_k - \mathbf{T}_k\mathbf{y}\|_2$$
- Approximation at iteration k: $\mathbf{x}_k = \mathbf{V}_k\mathbf{y}_k$.

Important remarks:

- $\mathbf{x}_k = \text{argmin}_{\mathbf{x}\in\mathcal{R}(\mathbf{V}_k)} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$
- $\mathbf{x}_k$ mimics a TSVD solution
- Stop iteration when solutions is "good enough"

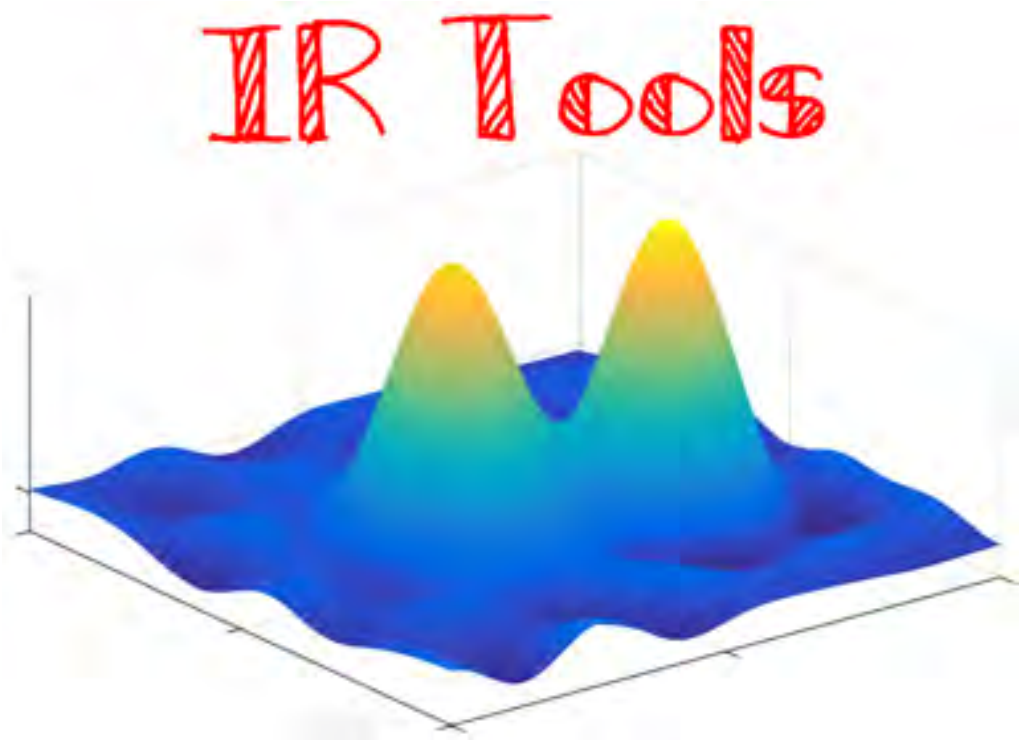
[Hanke. *Conjugate Gradient Type Methods for Ill-Posed Problems.* CRC, 1995]

[Hansen. *SIAM* 1997, 2010]

[Engl, Hanke, Neubauer. *Regularization of Inverse Problems.* Kluwer, 2000]

[Gazzola, Novati, Russo. *On Krylov projection methods and Tikhonov regularization.* ETNA, 2015]

Try it out in MATLAB ...



Silvia Gazzola	Per Christian Hansen	James Nagy
University of Bath, UK	Tech. Univ. of Denmark	Emory University, USA
S.Gazzola@bath.ac.uk	pcha@dtu.dk	jnagy@emory.edu

https://github.com/jnagy1/IRtools

Let's try: `[x, IterInfo] = IRcgls(A, b);`

Standard Krylov Solvers in IR Tools

How to determine good stopping iteration?

- If noise level, $\|\boldsymbol{\eta}\|_2$ is known, stop when $\|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|_2 \approx \|\eta\|_2$ (called discrepancy principle)

```
options = IRset('NoiseLevel', 0.01);
[x, IterInfo] = IRcgls(A, b, options);
```

```
[x, IterInfo] = IRcgls(A, b, options);
```

Standard Krylov Solvers in IR Tools

How to determine good stopping iteration?

- If noise level, $\|\boldsymbol{\eta}\|_2$ is known, stop when $\|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|_2 \approx \|\eta\|_2$ (called discrepancy principle)

```
options = IRset('NoiseLevel', 0.01);
[x, IterInfo] = IRcgls(A, b, options);
```

- Or, use simple Tikhonov regularization:

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \alpha^2 \|\mathbf{x}\|_2^2$$

Need to specify α, e.g.,

```
options = IRset('RegParam', 20);
[x, IterInfo] = IRcgls(A, b, options);
```

```
[x, IterInfo] = IRcgls(A, b, options);
```

Standard Krylov Solvers in IR Tools

Next question: I don't know the noise level, or a good initial α. What can I do?

Hybrid Krylov Solvers for Inverse Problems

Krylov-based methods $\Rightarrow$ at iteration k,

$$\begin{aligned} \text{solve tiny projected problem: } \mathbf{y}_k &= \arg\min_{\mathbf{y}\in\mathbb{R}^k} \|\mathbf{d}_k - \mathbf{T}_k\mathbf{y}\|_2^2 \\ \text{project back: } \mathbf{x}_k &= \mathbf{V}_k\mathbf{y}_k \end{aligned}$$

Hybrid Krylov Solvers for Inverse Problems

Hybrid Krylov-based regularization methods $\Rightarrow$ at iteration k,

$$\begin{aligned} \text{solve tiny projected problem: } \mathbf{y}_{\alpha_k,k} &= \arg\min_{\mathbf{y}\in\mathbb{R}^k} \|\mathbf{d}_k - \mathbf{T}_k\mathbf{y}\|_2^2 + \alpha_k^2\|\mathbf{y}\|_2^2 \\ \text{project back: } \mathbf{x}_{\alpha_k,k} &= \mathbf{V}_k\mathbf{y}_{\alpha_k,k} \end{aligned}$$

Hybrid Krylov Solvers for Inverse Problems

Hybrid Krylov-based regularization methods $\Rightarrow$ at iteration k,

$$\begin{aligned} \text{solve tiny projected problem: } \mathbf{y}_{\alpha_k,k} &= \arg\min_{\mathbf{y}\in\mathbb{R}^k} \|\mathbf{d}_k - \mathbf{T}_k\mathbf{y}\|_2^2 + \alpha_k^2\|\mathbf{y}\|_2^2 \\ \text{project back: } \mathbf{x}_{\alpha_k,k} &= \mathbf{V}_k\mathbf{y}_{\alpha_k,k} \end{aligned}$$

For tiny problem:

- Use GCV (or discrepancy) to choose regularization parameter, α_k.
- Compute information to determine stopping iteration, k.

Golub-Kahan Bidiagonalization (GKB) Approaches

[O'Leary, Simmons. *A bidiag.-reg. procedure for large scale discretizations of ill-posed problems.* SISSC, 1981]

[Björck. *A bidiag. alg. for solving sarge and sparse ill-posed systems of lin. eqs.* BIT, 1988]

[Björck, Grimme, van Dooren. *An implicit shift bidiag. alg. for ill-posed systems of lin. eqs.* BIT, 1994)]

[Hanke. *On Lanczos based methods for the reg. of discrete ill-posed problems.* BIT, 2001]

[Chung, Nagy, O'Leary. *A weighted GCV method for Lanczos hybrid regularization.* ETNA, 2008]

[Chung. *Numerical approaches for large-scale ill-posed inverse problems.* PhD Thesis, Emory Univ., 2009]

- IR Tools method `IRhybrid_lsqr` uses GKB to solve:

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \alpha^2 \|\mathbf{x}\|_2^2$$

- Underlying Krylov subspace is used to:
  - estimate regularization parameter α via
    - generalized cross validation (default)
    - discrepancy principle if noise level is provided
  - determine stopping iteration
- Can use as:

```
[x, IterInfo] = IRhybrid_lsqr(A, b);
[x, IterInfo] = IRhybrid_lsqr(A, b, K);
[x, IterInfo] = IRhybrid_lsqr(A, b, options);
[x, IterInfo] = IRhybrid_lsqr(A, b, K, options);
```

```
[x, IterInfo] = IRhybrid_lsqr(A, b);
```

Arnoldi-based Krylov Methods

Arnoldi-Tikhonov based methods are similar.

[Calvetti, Morigi, Reichel, Sgallari. *Tik. reg. and the L-curve for large discrete ill-posed probs.* JCAM, 2000]

[Lewis, Reichel. *Arnoldi-Tikhonov regularization methods.* JCAM, 2009]

[Reichel, Sgallari, Ye. *Tik. reg. based on generalized Krylov subspace methods.* Appl. Numer. Math., 2012]

[Gazzola, Novati. *Automatic parameter setting for Arnoldi-Tikhonov methods.* JCAM, 2014]

[Gazzola. *Reg. Tech. Based on Krylov Subspace Methods for Ill-Posed Lin. Syst.* PhD Thesis, Padova, 2014]

[Gazzola, Novati, Russo. *On Krylov projection methods and Tikhonov regularization.* ETNA, 2015]

IR Tools basic implementations:

`IRhybrid_gmres`
`IRhybrid_fgmres`

Hybrid methods for general form regularization [Kilmer, Hansen, Español. *SISC*, 2007]:

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \alpha^2 \|\mathbf{L}\mathbf{x}\|_2^2$$

Beyond the classical Krylov Methods and the 2-norm

Consider

$$\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \alpha\|\mathbf{x}\|_1 \,,$$

and adopt an iteratively reweighted norm approach:
Let $\mathbf{x}_0$ be an initial guess, then replace $\|\mathbf{x}\|_1$ with

$$\|\mathbf{x}\|_1 \approx \|\mathbf{W}_k\mathbf{x}\|_2^2, \quad \text{where} \quad \mathbf{W}_k = \text{diag}\left(1/\sqrt{|\mathbf{x}_{k-1}|}\right) \,.$$

Iterate: $\mathbf{x}_k = \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \alpha\|\mathbf{W}_k\mathbf{x}\|_2^2$
We can efficiently handle it by:

- transforming into std form:

$$\widetilde{\mathbf{x}}_\alpha = \arg\min_{\widetilde{\mathbf{x}}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{W}_k^{-1}\widetilde{\mathbf{x}}-\mathbf{b}\|_2^2 + \alpha\|\widetilde{\mathbf{x}}\|_2^2, \quad \widetilde{\mathbf{x}}_\alpha = \mathbf{W}_k\mathbf{x}_\alpha \,.$$

- iteration dependent "preconditioner" $\Rightarrow$ need flexible Krylov method

[Gazzola, Nagy. *GAT for sparse reconstruction.* SISC, 2014]

[Chung, Gazzola. *Flexible Krylov methods for ℓ_p regularization.* SISC, 2018]

[Gazzola, Nagy, Sabaté Landman. *Iter. Reweighted FGMRES and FLSQR for Sparse Recon.*, SISC, to appear.]

Beyond the classical Krylov Methods and the 2-norm

$$\mathbf{x}_\alpha = \arg\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \alpha\|\mathbf{x}\|_1 \,,$$

In IR Tools, use:

```
[x, IterInfo] = IRell1(A, b);
[x, IterInfo] = IRell1(A, b, K);
[x, IterInfo] = IRell1(A, b, options);
[x, IterInfo] = IRell1(A, b, K, options);
```

Makes use of `IRhybrid_fgmres`.

```
[x, IterInfo] = IRell1(A, b);
```

Computing low-rank (LR) solutions

Here we want to consider the problem:

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}$$

with the constraint that $\mathbf{x}$ has "low rank", i.e.,

$$\mathbf{x} = \mathbf{x}_1 \otimes \mathbf{y}_1 + \mathbf{x}_2 \otimes \mathbf{y}_2 + \cdots + \mathbf{x}_r \otimes \mathbf{y}_r$$

or equivalently

$$\mathbf{X} = \mathbf{y}_1\mathbf{x}_1^T + \mathbf{y}_2\mathbf{x}_2^T + \cdots + \mathbf{y}_r\mathbf{x}_r^T, \qquad \text{where } \mathbf{x} = \mathrm{vec}(\mathbf{X})$$

Computing low-rank (LR) solutions

Our first approach was motivated by:

[Kressner, Tobler. *Low-rank tensor Krylov methods for parametrized systems.* SIMAX, 2011]
[Stoll, Breiten. *A low-rank in time approach for PDE-constrained optimization.* SISC, 2015]
[Lee, Elman. *A Preconditioned Low-Rank Projection Method for SDEs.* SISC, 2017]

Basic idea:

- Enforce low (Kronecker) rank of solution $\mathbf{x}$
- Use flexible Golub-Kahan [Chung, Gazzola. *SISC*, 2018] to generate:

$$\mathbf{A}\mathbf{Z}_k = \mathbf{U}_{k+1}\mathbf{M}_k \quad \text{and} \quad \mathbf{A}^T\mathbf{U}_{k+1} = \mathbf{V}_{k+1}\mathbf{T}_{k+1}$$

  where
  - $\mathbf{U}_{k+1}$ and $\mathbf{V}_{k+1}$ contain orthonormal vectors
  - $\mathbf{Z}_k$ contain iteration dependent vectors
  - $\mathbf{M}_k$ is upper Hessenberg, $\mathbf{T}_k$ is upper triangular
  - $\mathbf{x}_k = \mathbf{x}_0 + \mathbf{Z}_k\mathbf{y}_k$, where $\mathbf{y}_k = \arg\min_{\mathbf{y}} \|\mathbf{M}_k\mathbf{y} - \beta_1\mathbf{e}_1\|_2^2 + \lambda\|\mathbf{y}\|_2^2$
- Use truncation approach to restrict Kronecker ranks of:
  - columns of $\mathbf{Z}_k$, and
  - approximate solution $\mathbf{x}_k$

Alternative Approach: Nuclear Norm Regularization

Let $\text{vec}^{-1}(\mathbf{x}) = \mathbf{X} = \mathbf{U}_\mathbf{X}\mathbf{\Sigma}_\mathbf{X}\mathbf{V}_\mathbf{X}^\mathsf{T}$. Solve the NNR problem:

$$\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda\|\text{vec}^{-1}(\mathbf{x})\|_*, \quad \text{where } \|\text{vec}^{-1}(\mathbf{x})\|_* = \sum_{i=1}^n \sigma_i(\mathbf{X})$$

Insight: "singular value sparsity"

Solve the NNRp problem:

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda\|\text{vec}^{-1}(\mathbf{x})\|_{*,p}, \text{ where } \|\mathbf{X}\|_{*,p} = \sum_{i=1}^n (\sigma_i(\mathbf{X}))^p,\ 0 < p \le 1$$

Insight: even more "singular value sparsity"

Solution techniques:

- Projected Gradient Descent (singular value thresholding (SVT))

  [Cai, Candès, Shen. *Singular value thresholding for matrix completion*. SIOPT, 2010]

- Iteratively Reweighted Norm (IRN) algorithms

  [Fornasier, Rauhut, Ward. *LR matrix recovery via IRLS minimization*. SIOPT, 2011]

  [Mohan, Fazel. *Iterative reweighted alg. for rank min.*. J. Mach. Learn. Res., 2012]

Low Rank and Krylov Solvers

at the kth iteration solve

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \widetilde{\lambda}\| \underbrace{\left(\mathbf{I} \otimes (\Sigma^2_{\mathbf{X}_k} + \gamma \mathbf{I})^{p/4-1/2}\right)}_{=:(\mathbf{W}_p^\gamma)_k} \overbrace{\left(\mathbf{V}^T_{\mathbf{X}_k} \otimes \mathbf{U}^T_{\mathbf{X}_k}\right)}^{=:\mathbf{S}_k} \mathbf{x}\|_2^2$$

Algorithm 1 IRN-NNRp

1: Inputs: $\mathbf{A}$, $\mathbf{b}$, $(\mathbf{W}_p^\gamma)_0 = \mathbf{I}$, $\mathbf{S}_0 = \mathbf{I}$
2: **for** $k = 0, 1, \ldots$ until a stopping criterion holds **do**
3: Solve problem the $(k+1)$th linear problem
4: "Decrease" γ
5: Update $(\mathbf{W}_p^\gamma)_{k+1}$ and $\mathbf{S}_{k+1}$
6: **end for**

Low Rank and Krylov Solvers

Perform *inner-outer iteration cycles*; at the kth (*outer*) iteration solve

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \widetilde{\lambda}\| \underbrace{\left(\mathbf{I} \otimes (\Sigma_{\mathbf{X}_k}^2 + \gamma \mathbf{I})^{p/4-1/2}\right)}_{=:(\mathbf{W}_p^\gamma)_k} \overbrace{\left(\mathbf{V}_{\mathbf{X}_k}^T \otimes \mathbf{U}_{\mathbf{X}_k}^T\right)}^{=:\mathbf{S}_k} \mathbf{x}\|_2^2$$

Algorithm 1 IRN-NNRp

1: Inputs: $\mathbf{A}$, $\mathbf{b}$, $(\mathbf{W}_p^\gamma)_0 = \mathbf{I}$, $\mathbf{S}_0 = \mathbf{I}$
2: **for** $k = 0, 1, \ldots$ until a stopping criterion holds (*outer* iterations) **do**
3: Solve problem the $(k+1)$th linear problem (*inner* iterations)
4: "Decrease" γ
5: Update $(\mathbf{W}_p^\gamma)_{k+1}$ and $\mathbf{S}_{k+1}$
6: **end for**

A summary of IRN-LSQR-NNRp and IRN-GMRES-NNRp

[Gazzola, Meng, Nagy. *Krylov Methods for Low-Rank Regularization*. SIMAX, 2020]

Final step: Transform to standard form, and use hybrid method:

$$\min_{\widehat{\mathbf{x}}} \|\mathbf{A}\mathbf{S}_k^T(\mathbf{W}_p^\gamma)_k^{-1}\widehat{\mathbf{x}} - \mathbf{b}\|_2^2 + \widetilde{\lambda}\|\widehat{\mathbf{x}}\|_2^2$$

Algorithm 2 IRN-LSQR-NNRp, IRN-GMRES-NNRp

1: Inputs: $\mathbf{A}$, $\mathbf{b}$, $(\mathbf{W}_p^\gamma)_0 = \mathbf{I}$, $\mathbf{S}_0 = \mathbf{I}$
2: **for** $k = 0, 1, \ldots$ until a stopping criterion is satisfied **do**
3: **for** $m = 1, 2, \ldots$ until a stopping criterion is satisfied **do**
4: update the relevant partial factorizations
5: solve the relevant projected problems, tuning $\widetilde{\lambda}_m$
6: **end for**
7: "Decrease" γ
8: Update the new $(\mathbf{W}_p^\gamma)_{k+1}$ and $\mathbf{S}_{k+1}$.
9: **end for**

A summary of FLSQR-NNRp and FGMRES-NNRp

[Gazzola, Meng, Nagy. *Krylov Methods for Low-Rank Regularization*. SIMAX, 2020]

Algorithm 3 FLSQR-NNRp and FGMRES-NNRp

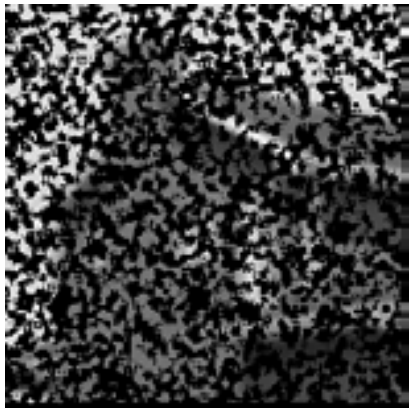
1: Inputs: $\mathbf{A}$, $\mathbf{b}$, $(\mathbf{W}_p^\gamma)_0 = \mathbf{I}$, $\mathbf{S}_0 = \mathbf{I}$
2: **for** $i = 1, 2, \ldots$ until a stopping criterion is satisfied **do**
3: update the relevant partial flexible factorizations
4: solve the relevant projected problems, tuning $\widetilde{\lambda}_i$ if necessary
5: "Decrease" γ
6: Update the new $(\mathbf{W}_p^\gamma)_i$ and $\mathbf{S}_i$, using $\mathbf{X}_i = \text{vec}^{-1}(\mathbf{x}_i) = \mathbf{U}_{\mathbf{X}_i} \mathbf{\Sigma}_{\mathbf{X}_i} \mathbf{V}_{\mathbf{X}_i}^{\mathbf{T}}$
7: **end for**

Remark: With flexible Krylov subspace framework, can use alternative $(\mathbf{W}_p^\gamma)_i$ and $\mathbf{S}_i$ that are effective in producing low-rank solutions.

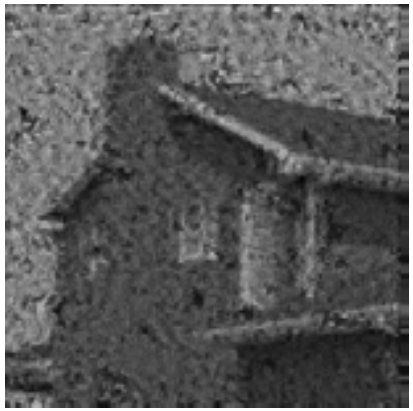
Example: Image deblurring and inpainting

exact

corrupted



LSQR



LR-FLSQR

FLSQR-NNR(v)

FLSQR-NNR

IRN-LSQR-NNR

Concluding Remarks

Krylov methods can be powerful approaches for large-scale inverse problems

- Allows to use adaptive guides to choosing regularization parameters.
- Can encode regularization through iteration-dependent "preconditioning".

IR Tools MATLAB Package

- Implements many Krylov methods.
- Includes many other well-known methods, some with box constraints.
- Combines with AIR Tools II for tomography examples and iterative methods

  [Hansen, Jørgensen. *AIR Tools II: algeb. iter. recon. methods, improved impl.* Numer. Algor., 2018.]
- Package can be used in many ways:
  - Use our implementations to solve your problems.
  - Experiment with different regularization approaches, constraints, etc.
  - Use our test problems to evaluate your new algorithms.
  - Compare your best/new algorithms with our implementations.
- Get the paper, Numer Algor. (2018): https://doi.org/10.1007/s11075-018-0570-7
- Get the software from GitHub: https://github.com/jnagy1/IRtools