Regularization by inexact Krylov methods

Silvia Gazzola
S.Gazzola@bath.ac.uk

Joint work **M. Sabaté Landman**

Department of Mathematical Sciences
UNIVERSITY OF BATH

Workshop on Recent Advances in Iterative Reconstruction
UCL – may 22-23, 2023

Setting the stage: linear inverse problem

Solution of

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|_2, \quad \text{where} \quad Ax_{\text{true}}+e=b$$

and

$b\in\mathbb{R}^m$	available observations or measurements
$x_{\text{true}}\in\mathbb{R}^n$	unknown quantity of interest
$A\in\mathbb{R}^{m\times n}$	available ill-conditioned matrix models forward process
$\mathrm{e}\in\mathbb{R}^m$	additive Gaussian white noise

Setting the stage: linear inverse problem

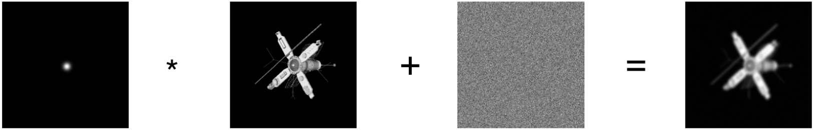
Solution of

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2, \quad \text{where} \quad Ax_{\text{true}} + e = b$$

and

$b \in \mathbb{R}^m$	available observations or measurements
$x_{\text{true}} \in \mathbb{R}^n$	unknown quantity of interest
$A \in \mathbb{R}^{m \times n}$	available ill-conditioned matrix models forward process
$e \in \mathbb{R}^m$	additive Gaussian white noise

Example: image deblurring



PSF * exact + noise = data

Here $m = n = 65536$.

Setting the stage: separable nonlinear inverse problem

Solution of

$$\min_{x\in\mathbb{R}^n, y\in\mathbb{R}^p} \|A(y)x - b\|_2, \quad \text{where} \quad A(y_{\text{true}})x_{\text{true}} + e = b$$

and

$b \in \mathbb{R}^m$	available observations or measurements
$x_{\text{true}} \in \mathbb{R}^n$	unknown quantity of interest
$y_{\text{true}} \in \mathbb{R}^p$	unknown parameters defining A, $p \ll n$
$A(y) \in \mathbb{R}^{m\times n}$	ill-conditioned matrix models forward process
$e \in \mathbb{R}^m$	additive Gaussian white noise

Example: image (semi-)blind deblurring, with Gaussian PSF $P(y)$

$$[P(y)]_{i,j} = c(\sigma_1,\sigma_2,\rho)\exp\left(-\frac{1}{2}\begin{bmatrix} i-\chi_1 \\ j-\chi_2\end{bmatrix}^T \begin{bmatrix} \sigma_1^2 & \rho^2 \\ \rho^2 & \sigma_2^2\end{bmatrix}^{-1}\begin{bmatrix} i-\chi_1 \\ j-\chi_2\end{bmatrix}\right)$$

Note: $\sigma_1^2\sigma_2^2 - \rho^4 > 0$; $\sum_{i,j=1}^N [P(y)]_{i,j} = 1$.

Here $y = [\sigma_1, \sigma_2, \rho]^T \in \mathbb{R}^3$. For illustrations: $y_{\text{true}} = [2.5, 2.5, 0]^T$.

Dealing with ill-posedness: introducing regularization

- For (large-scale) linear inverse problems
  - early termination of Krylov methods (LSQR,CGLS...), applied to

$$\min_x \|Ax - b\| \qquad \text{(from now on, } \|\cdot\| = \|\cdot\|_2\text{)}$$

  - combining variational (e.g., Tikhonov) regularization methods

$$z_\lambda = \arg\min_{z\in\mathbb{R}^n} \|Az - r_0\|^2 + \lambda^2\|z\|^2, \quad \text{where} \quad \begin{array}{rcl} r_0 & = & b - Ax_0 \\ x_\lambda & = & x_0 + z_\lambda \end{array}$$

    and Krylov methods... equivalently
    - first project then regularize
    - first regularize then project

Main ingredient (for hybrid solvers): shift-invariance of Krylov subspaces

$$\mathcal{K}_k(A^TA, A^Tr_0) = \mathcal{K}_k(A^TA + \lambda^2 I, A^Tr_0)$$

See papers by: Bjorck, Buccini, Calvetti, Chung, Donatelli, Espanol, Fenu, G., Hansen, Hanke, Hnetynkova, Hochstenbach, Kilmer, Morigi, Nagy, Novati, O'Leary, Renaut, Reichel, Sgallari

Dealing with ill-posedness: introducing regularization

- For (large-scale) separable nonlinear inverse problems

$$(z_\lambda, y^*) = \arg \min_{z\in\mathbb{R}^n, y\in\mathbb{R}^p} \|A(y)z - r_0\|^2 + \lambda^2\|z\|^2, \quad \text{where} \quad \begin{array}{rcl} r_0 & = & b - A(y)x_0 \\ x_\lambda & = & x_0 + z_\lambda \end{array}$$

Trick: exploit separability!

In particular: apply the variable projection method (inner-outer iterations)

- implicitly 'eliminates' z (hybrid solver)
- y is updated using a NLLS solver (e.g., Gauss–Newton)

[Golub and Pereyra, *Inverse Problems*, 2003] [Chung and Nagy, *SISC*, 2010]

Dealing with ill-posedness: introducing regularization

- For (large-scale) separable nonlinear inverse problems

$$(z_\lambda, y^*) = \arg \min_{z\in\mathbb{R}^n, y\in\mathbb{R}^p} \|A(y)z - r_0\|^2 + \lambda^2\|z\|^2, \quad \text{where} \quad \begin{array}{rcl} r_0 & = & b - A(y)x_0 \\ x_\lambda & = & x_0 + z_\lambda \end{array}$$

Trick: exploit separability!

In particular: apply the variable projection method (inner-outer iterations)

- implicitly 'eliminates' z (hybrid solver)
- y is updated using a NLLS solver (e.g., Gauss–Newton)

[Golub and Pereyra, *Inverse Problems*, 2003] [Chung and Nagy, *SISC*, 2010]

- *In this talk*:
  - introduce inexact Krylov methods (iLSQR, iCGLS) for regularization
  - introducing hybrid iLSQR and hybrid iCGLS for regularization
  - adopting inexact solvers within the variable projection method (application to blind deblurring)

Transitioning from exact to inexact Golub–Kahan

Inspired by: [Simoncini and Szyld, *SIMAX*, 2003], [Gaaf and Simoncini, *Appl.Num.Math.*, 2017]

exact (GKB) | **inexact (iGK)**

'iteration-wise'

exact (GKB):

$$u_1 = r_0/\beta,\ v_1 = A^T u_1/\alpha_1$$
$$u_{i+1} = (Av_i - \alpha_i u_i)/\beta_{i+1}$$
$$v_{i+1} = (A^T u_{i+1} - \beta_{i+1} v_i)/\alpha_{i+1}$$

inexact (iGK):

$$u_1 = r_0/\beta,\ v_1 = (A + F_1)^T u_1/[L_{k+1}]_{1,1}$$
$$u_{i+1} = (I - U_i U_i^T)(A + E_i)v_i/[M_k]_{i+1,i+1}$$
$$v_{i+1} = (I - V_i V_i^T)(A + F_{i+1})^T u_{i+1}/[L_{k+1}]_{i+1,i+1}$$

'factorization-wise'

exact (GKB):

$$AV_k = U_{k+1}\bar{B}_k$$
$$A^T U_{k+1} = V_{k+1}B_{k+1}^T$$

inexact (iGK):

$$[(A+E_1)v_1, \dots, (A+E_k)v_k] = U_{k+1}M_k$$
$$\left[(A+F_1)^T u_1, \dots, (A+F_{k+1})^T u_{k+1}\right] = V_{k+1}L_{k+1}^T$$

where $V_{k+1} = [v_1, \dots, v_{k+1}]$, $U_{k+1} = [u_1, \dots, u_{k+1}]$

'compactly factorization-wise'

$$(A+\mathcal{E}_k)V_k = U_{k+1}M_k$$
$$(A+\mathcal{F}_{k+1})^T U_{k+1} = V_{k+1}L_{k+1}^T$$

where
$$\mathcal{E}_k = \sum_{i=1}^{k} E_i v_i v_i^T$$
$$\mathcal{F}_{k+1} = \sum_{i=1}^{k+1} (u_i u_i^T)F_i$$

links with symmetric Lanczos

exact (GKB):

$$A^T A V_k = V_{k+1}B_{k+1}^T \bar{B}_k$$

inexact (iGK):

$$(A^T A + \mathcal{F}_{k+1}^T A + A^T\mathcal{E}_k + \mathcal{F}_{k+1}^T\mathcal{E}_k)V_k = V_{k+1}L_{k+1}^T M_k$$

Transitioning from exact to inexact linear system solvers

Inspired by: [Simoncini and Szyld, *SIMAX*, 2003]

$x_k = x_0 + z_k = x_0 + V_k s_k$

GKB: $AV_k = U_{k+1}\bar{B}_k, \dots$

LSQR

$q_k = \arg\min_{q\in\mathcal{R}(U_{k+1}\bar{B}_k)=\mathcal{R}(AV_k)} \|q - r_0\|$

equivalently

$s_k = \arg\min_{s\in\mathbb{R}^k} \|\bar{B}_k s - \beta e_1\|$

equivalently

$(\bar{B}_k^T \bar{B}_k)s_k = \bar{B}_k^T(\beta e_1)$

iGK: $(A + \mathcal{E}_k)V_k = U_{k+1}M_k, \dots$

inexact LSQR (iLSQR)

$q_k = \arg\min_{q\in\mathcal{R}(U_{k+1}M_k)} \|q - r_0\|$

$s_k = \arg\min_{s\in\mathbb{R}^k} \|M_k s - \beta e_1\|$

does not minimize the true residual!

$(M_k^T M_k)s_k = M_k^T(\beta e_1)$

Transitioning from exact to inexact linear system solvers

Inspired by: [Simoncini and Szyld, *SIMAX*, 2003]

$x_k = x_0 + z_k = x_0 + V_k s_k$

GKB: $AV_k = U_{k+1}\bar{B}_k, \dots$

LSQR

$q_k = \arg\min_{q \in \mathcal{R}(U_{k+1}\bar{B}_k) = \mathcal{R}(AV_k)} \|q - r_0\|$

equivalently

$s_k = \arg\min_{s \in \mathbb{R}^k} \|\bar{B}_k s - \beta e_1\|$

equivalently

$(\bar{B}_k^T \bar{B}_k) s_k = \bar{B}_k^T (\beta e_1)$

equivalently, **CGLS**

$V_k^T (A^T A) V_k s_k = V_k^T A^T r_0 = \bar{B}_k^T \beta e_1$

equivalently

$q_k \in \mathcal{R}(V_{k+1}\bar{T}_k),\ A^T r_0 - q_k \perp \mathcal{R}(V_k)$

iGK: $(A + \mathcal{E}_k) V_k = U_{k+1} M_k, \dots$

inexact LSQR (iLSQR)

$q_k = \arg\min_{q \in \mathcal{R}(U_{k+1} M_k)} \|q - r_0\|$

equivalently

$s_k = \arg\min_{s \in \mathbb{R}^k} \|M_k s - \beta e_1\|$

does not minimize the true residual!

equivalently

$(M_k^T M_k) s_k = M_k^T (\beta e_1)$

Transitioning from exact to inexact linear system solvers

Inspired by: [Simoncini and Szyld, *SIMAX*, 2003]

$x_k = x_0 + z_k = x_0 + V_k s_k$

GKB: $AV_k = U_{k+1}\bar{B}_k, \dots$

LSQR

$q_k = \arg\min_{q \in \mathcal{R}(U_{k+1}\bar{B}_k) = \mathcal{R}(AV_k)} \|q - r_0\|$

equivalently

$s_k = \arg\min_{s \in \mathbb{R}^k} \|\bar{B}_k s - \beta e_1\|$

equivalently

$(\bar{B}_k^T \bar{B}_k) s_k = \bar{B}_k^T (\beta e_1)$

equivalently, **CGLS**

$V_k^T (A^T A) V_k s_k = V_k^T A^T r_0 = \bar{B}_k^T \beta e_1$

equivalently

$q_k \in \mathcal{R}(V_{k+1}\bar{T}_k),\ A^T r_0 - q_k \perp \mathcal{R}(V_k)$

iGK: $(A + \mathcal{E}_k)V_k = U_{k+1}M_k, \dots$

inexact LSQR (iLSQR)

$q_k = \arg\min_{q \in \mathcal{R}(U_{k+1}M_k)} \|q - r_0\|$

equivalently

$s_k = \arg\min_{s \in \mathbb{R}^k} \|M_k s - \beta e_1\|$

does not minimize the true residual!

equivalently

$(M_k^T M_k) s_k = M_k^T (\beta e_1)$

inexact CGLS (iCGLS)

$q_k \in \mathcal{R}(V_{k+1}\widehat{H}_k),\ (A + \mathcal{F}_{k+1})^T r_0 - q_k \perp \mathcal{R}(V_k)$

not orthogonal to the true NE residual!

equivalently

$V_k^T (\widehat{A} + \widehat{\mathcal{E}}_k) V_k s_k = V_k^T (A + \mathcal{F}_{k+1})^T r_0$

equivalently

$\bar{L}_k^T M_k s_k = [\bar{L}_k]_{1,1} \beta e_1$

Transitioning from inexact linear system solvers to inexact hybrid solvers

Recall, iGK: $(A+\mathcal{E}_k)V_k = U_{k+1}M_k$, $(A+\mathcal{F}_{k+1})^T U_{k+1} = V_{k+1}L_{k+1}^T$
$x_{\lambda,k} = x_0 + z_{\lambda,k} = x_0 + V_k s_{\lambda,k}$
λ fixed

inexact LSQR (iLSQR)

$$q_k = \arg\min_{q\in\mathcal{R}(U_{k+1}M_k)} \|q - r_0\|$$

inexact hybrid LSQR (hybrid-iLSQR)

$$q_{\lambda,k} = \arg\min_{q\in\mathcal{R}(W_{\lambda,k})} \left\| q - \begin{bmatrix} r_0 \\ 0 \end{bmatrix} \right\|, \quad W_{\lambda,k} = \left[(U_{k+1}M_k)^T, \lambda(V_k)^T\right]^T$$

inexact CGLS (iCGLS)

$$q_k \in \mathcal{R}(V_{k+1}\bar{T}_k),\ A^T r_0 - q_k \perp \mathcal{R}(V_k)$$

inexact hybrid CGLS (hybrid-iCGLS)

$$q_{\lambda,k} \in \mathcal{R}(W_{\lambda,k}) = \mathcal{R}(V_{k+1}(\widehat{H}_k + \lambda^2\bar{I})),$$
$$(A+\mathcal{F}_{k+1})^T r_0 - q_{\lambda,k} \perp \mathcal{R}(V_k)$$

Transitioning from inexact linear system solvers to inexact hybrid solvers

Recall, iGK: $(A+\mathcal{E}_k)V_k = U_{k+1}M_k$, $(A+\mathcal{F}_{k+1})^T U_{k+1} = V_{k+1}L_{k+1}^T$
$x_{\lambda,k} = x_0 + z_{\lambda,k} = x_0 + V_k s_{\lambda,k}$
λ fixed

inexact LSQR (iLSQR)

$$q_k = \arg\min_{q\in\mathcal{R}(U_{k+1}M_k)} \|q - r_0\|$$

inexact hybrid LSQR (hybrid-iLSQR)

$$q_{\lambda,k} = \arg\min_{q\in\mathcal{R}(W_{\lambda,k})} \left\| q - \begin{bmatrix} r_0 \\ 0 \end{bmatrix} \right\|, \quad W_{\lambda,k} = \left[(U_{k+1}M_k)^T, \lambda(V_k)^T\right]^T$$

equivalently

$$\begin{aligned} s_{\lambda,k} &= \arg\min_{s\in\mathbb{R}^k} \|M_k s - \beta e_1\|^2 + \lambda^2\|s\|^2 \\ &= (M_k^T M_k + \lambda^2 I)^{-1} M_k^T(\beta e_1) \end{aligned}$$

inexact CGLS (iCGLS)

$$q_k \in \mathcal{R}(V_{k+1}\bar{T}_k),\ A^T r_0 - q_k \perp \mathcal{R}(V_k)$$

inexact hybrid CGLS (hybrid-iCGLS)

$$q_{\lambda,k} \in \mathcal{R}(W_{\lambda,k}) = \mathcal{R}(V_{k+1}(\widehat{H}_k + \lambda^2\bar{I})), \quad (A+\mathcal{F}_{k+1})^T r_0 - q_{\lambda,k} \perp \mathcal{R}(V_k)$$

equivalently

$$(\bar{L}_k^T M_k + \lambda^2 I)s_{\lambda,k} = [\bar{L}_k]_{1,1}\beta e_1$$

Transitioning from inexact linear system solvers to inexact hybrid solvers

Recall, iGK: $(A+\mathcal{E}_k)V_k = U_{k+1}M_k$, $(A+\mathcal{F}_{k+1})^T U_{k+1} = V_{k+1}L_{k+1}^T$

$x_{\lambda,k} = x_0 + z_{\lambda,k} = x_0 + V_k s_{\lambda,k}$

λ fixed: shift-invariance only under some conditions!

inexact LSQR (iLSQR)

$$q_k = \arg\min_{q\in\mathcal{R}(U_{k+1}M_k)} \|q - r_0\|$$

inexact hybrid LSQR (hybrid-iLSQR)

$$q_{\lambda,k} = \arg\min_{q\in\mathcal{R}(W_{\lambda,k})} \left\| q - \begin{bmatrix} r_0 \\ 0 \end{bmatrix} \right\|,$$

$$W_{\lambda,k} = \left[(U_{k+1}M_k)^T, \lambda(V_k)^T\right]^T$$

equivalently

$$\begin{aligned} s_{\lambda,k} &= \arg\min_{s\in\mathbb{R}^k} \|M_k s - \beta e_1\|^2 + \lambda^2\|s\|^2 \\ &= (M_k^T M_k + \lambda^2 I)^{-1} M_k^T(\beta e_1) \end{aligned}$$

inexact CGLS (iCGLS)

$$q_k \in \mathcal{R}(V_{k+1}\bar{T}_k),\ A^T r_0 - q_k \perp \mathcal{R}(V_k)$$

inexact hybrid CGLS (hybrid-iCGLS)

$$q_{\lambda,k} \in \mathcal{R}(W_{\lambda,k}) = \mathcal{R}(V_{k+1}(\widehat{H}_k + \lambda^2\bar{I})),$$
$$(A+\mathcal{F}_{k+1})^T r_0 - q_{\lambda,k} \perp \mathcal{R}(V_k)$$

equivalently

$$(\bar{L}_k^T M_k + \lambda^2 I)s_{\lambda,k} = [\bar{L}_k]_{1,1}\beta e_1$$

When are inexact solvers 'meaningful'?

When are inexact solvers 'meaningful'?

Inspired by: [Simoncini and Szyld, *SIMAX*, 2003]

Depends on the relations between exact (i.e., r^e, $r^e_{\lambda,k}$) and inexact (i.e., r, $r_{\lambda,k}$) residuals, keeping in mind that:

- there is ill-posedness: r^e may not be small
- there is regularization: $r^e_{\lambda,k}$ may not be small
  (i.e., $\left\| r^e_{\lambda,k} \right\| = \left(\|Az_{\text{true}} - r^e_0\|^2 + \lambda^2 \|z_{\text{true}}\|^2 \right)^{1/2} = \left(\|e\|^2 + \lambda^2 \|z_{\text{true}}\|^2 \right)^{1/2}$)

When are inexact solvers 'meaningful'?

Inspired by: [Simoncini and Szyld, *SIMAX*, 2003]

Depends on the relations between exact (i.e., r^e, $r^e_{\lambda,k}$) and inexact (i.e., r, $r_{\lambda,k}$) residuals, keeping in mind that:

- there is ill-posedness: r^e may not be small
- there is regularization: $r^e_{\lambda,k}$ may not be small
  (i.e., $\left\|r^e_{\lambda,k}\right\| = \left(\|Az_{\text{true}} - r^e_0\|^2 + \lambda^2\|z_{\text{true}}\|^2\right)^{1/2} = \left(\|e\|^2 + \lambda^2\|z_{\text{true}}\|^2\right)^{1/2}$)

Focusing on:

- **iLSQR**: $\|r^e_k\| \leq \|r_k\| + \|E_0x_0\| + \sum_{l=1}^k \|E_l\| \, |[s_k]_l|$
- **hybrid-iLSQR**, fixed λ

$$\left\|r^e_{\lambda,k}\right\| \leq \|r_{\lambda,k}\| + \|E_0x_0\| + \sum_{l=1}^{k} \|E_l\| |[s_{\lambda,k}]_l|$$

'A priori' bounds, ϵ desired accuracy:

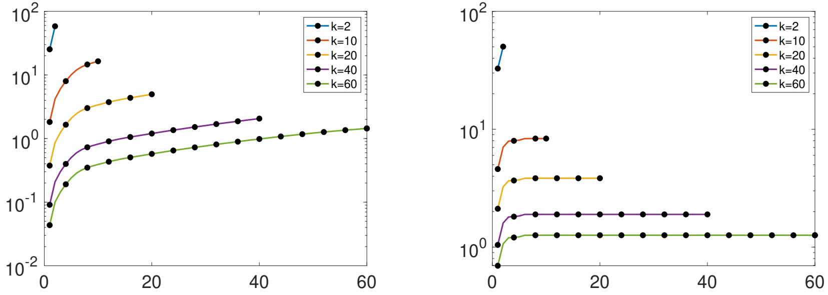
- **iLSQR**: $\|E_j\| \leq \dfrac{\sigma_k(M_k)}{k} \frac{1}{\|r_{j-1}\|}\epsilon$, $j = 1, \ldots, k$
- **hybrid-iLSQR**, fixed λ

$$\|E_j\| \leq \frac{(\sigma_k(M_k^T M_k + \lambda^2 I))^{1/2}}{k} \frac{1}{\|r_{\lambda,j-1}\|}\epsilon, \quad j = 1, \ldots, k$$

An illustration

`satellite` blind deblurring example, with $\lambda = 0.5$

$$\frac{\sigma_k(M_k)}{k}\frac{1}{\|r_{j-1}\|} \text{ vs. } j \qquad \frac{(\sigma_k(M_k^T M_k + \lambda^2 I))^{1/2}}{k}\frac{1}{\|r_{\lambda,j-1}\|} \text{ vs. } j$$



Recap on separable NLLS and VarPro

[Golub and Pereyra, *Inverse Problems*, 2003] [Chung and Nagy, *SISC*, 2010]

- Problem to be solved

$$z_\lambda = \arg\min_{z\in\mathbb{R}^n, y\in\mathbb{R}^p} g(z,y)\,,\ \text{where}\ \begin{array}{lcl} g(z,y) &=& \|F(z,y)\|^2 \\ F(z,y) &=& \widetilde{A}_\lambda(y)z - \widetilde{r}_0 \\ \widetilde{A}_\lambda(y) &=& [A^T(y), \lambda I]^T,\ \widetilde{r}_0 = [r_0^T, 0^T]^T \\ x_\lambda &=& x_0 + z_\lambda \end{array}$$

- Consider the reduced cost functional

$$h(y) := g(z_\lambda(y), y)\,, \quad \text{where} \quad \begin{array}{lcl} z_\lambda(y) &=& \arg\min_{z\in\mathbb{R}^n} g(z,y) \\ &=& (A^T(y)A(y) + \lambda^2 I)^{-1} A^T(y) r_0 \end{array}$$

Take $x_\lambda(y) = x_0 + z_\lambda(y)$

- Apply Gauss-Newton to minimize the reduced cost functional

$$y_l = y_{l-1} + \gamma_l d_{l-1} \quad (\text{setting the steplength } \gamma_l)$$

Note that

$$d_{l-1} = \arg\min_d \|\widehat{J}_h d - r_{l-1}\|,\ J_h = \begin{bmatrix} d(A(y)z_\lambda)/dy \\ 0 \end{bmatrix} = \begin{bmatrix} \widehat{J}_h \\ 0 \end{bmatrix},\ J_h^T F(z_\lambda, y) = \nabla_y g(z_\lambda, y)$$

(computationally convenient analytical expression of $d(A(y)z_\lambda)/dy$ for blind deblurring)

Towards an iGK-based algorithm for separable nonlinear least squares problems

[Chung and Nagy, *SISC*, 2010]

Algorithm Variable projection with Gauss-Newton and *hybrid LSQR* solver

```
Choose initial guesses x_0 and y_0
for l = 1, 2, ... until a stopping criterion is satisfied do
    for k = 1, 2, ... until a stopping criterion is satisfied do
        Expand K_k(A(y_{l-1})^T A(y_{l-1}), A(y_{l-1})^T r_0) using GKB
        Compute x_{λ,k} solving the projected problem with adaptive choice of λ
    end for
    Compute the residual r_l = b - A(y_{l-1}) x_{λ,k}
    Compute d_l = arg min_d ||Ĵ_h d - r_l||
    Update y_l = y_{l-1} + γ_l d_l (setting the steplength γ_l)
    Update x_0
end for
```

Towards an iGK-based algorithm for separable nonlinear least squares problems

Algorithm Variable projection with Gauss-Newton and *hybrid LSQR* solver

Choose initial guesses x_0 and y_0
~~**for** $l = 1, 2, \dots$ until a stopping criterion is satisfied **do**~~
  for $k = 1, 2, \dots$ until a stopping criterion is satisfied **do**
    Expand $\mathcal{K}_k(A(y_{l-1})^T A(y_{l-1}), A(y_{l-1})^T r_0)$ using GKB
    Compute $x_{\lambda,k}$ solving the projected problem with adaptive choice of λ
  end for
  Compute the residual $r_l = b - A(y_{l-1})x_{\lambda,k}$
  Compute $d_l = \arg\min_d \|\widehat{J}_h d - r_l\|$
  Update $y_l = y_{l-1} + \gamma_l d_l$ (setting the steplength γ_l)
  Update x_0
~~**end for**~~

Towards an iGK-based algorithm for separable nonlinear least squares problems

Algorithm Variable projection with Gauss-Newton and *hybrid-iLSQR* solver

Choose initial guesses x_0 and y_0

for $k = 1, 2, \dots$ **do**
  Expand the approximation subspace $\mathcal{R}(V_k)$ using $A(y_{k-1})$ and iGK
  Compute $x_{\lambda,k}$ solving the projected problem with adaptive choice of λ
  Compute the residual $r_k = b - A(y_{k-1})x_{\lambda,k}$
  Compute $d_k = \arg\min_d \|\widehat{J}_h d - r_k\|$
  Update $y_k = y_{k-1} + \gamma_k d_k$ (setting the steplength γ_k)
end for

Towards an iGK-based algorithm for separable nonlinear least squares problems

Algorithm Variable projection with Gauss-Newton and *hybrid-iLSQR* solver

1: Choose initial guesses x_0 and y_0; set an accuracy ε
2: **for** $l = 1, 2, \dots$ until a stopping criterion is satisfied **do**
3: **for** $k = 1, 2, \dots$ until inexactness exceeds the bound ε **do**
4: Expand the approximation subspace $\mathcal{R}(V_k)$ using $A(y_{k-1})$ and iGK
5: Compute $x_{\lambda,k}$ solving the projected problem with adaptive choice of λ
6: Compute the residual $r_k = b - A(y_{k-1})x_{\lambda,k}$
7: Compute $d_k = \arg\min_d \|\widehat{J}_h d - r_k\|$
8: Update $y_k = y_{k-1} + \gamma_k d_k$ (setting the steplength γ_k)
9: **end for**
10: Update x_0; take $y_0 = y_k$
11: **end for**

A few details

- Defining inexactness, with some pragmatism:

consider as exact matrix the latest computed approximation of $A(y)$, i.e.,

after $j-1$ iterations, $A(y_{i-1}) = A(y_{j-1}) + E_i^j$, where $E_i^j := A(y_{i-1}) - A(y_{j-1})$,

iGK being expressed as

$$
\begin{aligned}
(A(y_{j-1}) + \mathcal{E}_j^j)V_j &= U_{j+1}M_j, & \mathcal{E}_j^j &= \textstyle\sum_{i=1}^j E_i^j v_i v_i^T \\
(A(y_{j-1}) + \mathcal{F}_{j+1}^j)^T U_{j+1} &= V_{j+1}L_{j+1}^T, & \mathcal{F}_{j+1}^j &= \textstyle\sum_{i=1}^{j+1} u_i u_i^T E_{i-1}^j
\end{aligned}
$$

- Setting the Gauss-Newton stepsize:

$$
y_j = y_{j-1} + \gamma_j d_{j-1}\,, \quad \text{where} \quad \gamma_j = \arg\min_{\gamma\geq 0} g(z_\lambda(y_{j-1}), y_{j-1} + \gamma d_{j-1})
$$

We get

$$
\|\widetilde{A}_\lambda(y_j)z_{\lambda,j+1} - \widetilde{r}_0\| \quad \leq \quad \|\widetilde{A}_\lambda(y_{j-1})z_{\lambda,j} - \widetilde{r}_0\| + 2\widetilde{\varepsilon}\,,
$$

instead of

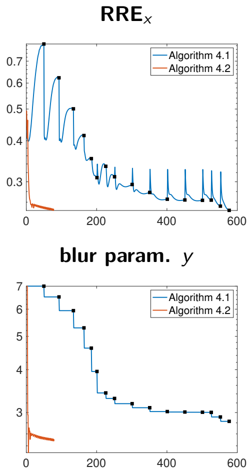
$$
\|\widetilde{A}_\lambda(y_j)z_{\lambda,j+1} - \widetilde{r}_0\| \quad \leq \quad \|\widetilde{A}_\lambda(y_{j-1})z_{\lambda,j} - \widetilde{r}_0\|
$$

An illustration

`satellite` blind deblurring example, with $y_{\text{true}} = [2.5, 2.5, 0]^T$, $\lambda = 0.5$

An illustration

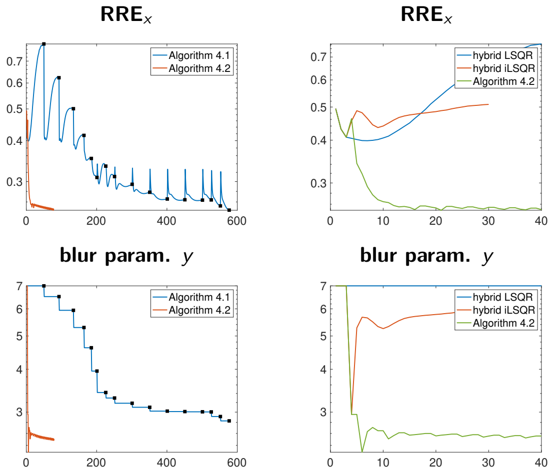
satellite blind deblurring example, with $y_{\text{true}} = [2.5, 2.5, 0]^T$, $\lambda = 0.5$



Algorithm 4.1: [Chung and Nagy, *SISC*, 2010]; Algorithm 4.2: new solver

An illustration

satellite blind deblurring example, with $y_{\text{true}} = [2.5, 2.5, 0]^T$, $\lambda = 0.5$



Algorithm 4.1: [Chung and Nagy, *SISC*, 2010]; Algorithm 4.2: new solver

An illustration

Hybrid-iLSQR
(it. 30, RRE_x 0.5819)

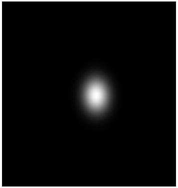
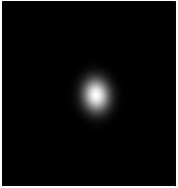
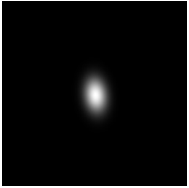
Algorithm 4.1
(it. 577, RRE_x 0.2454)

Algorithm 4.2
(it. 79, RRE_x 0.2474)

Algorithm 4.1: [Chung and Nagy, *SISC*, 2010]; Algorithm 4.2: new solver

Another example

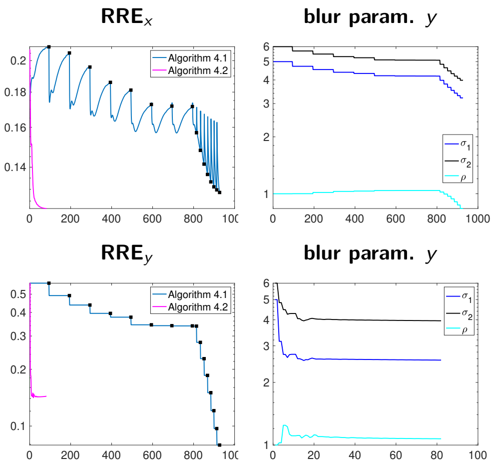
cameraman blind deblurring example, with $y_{\text{true}} = [3, 4, 0.5]^T$, $y_0 = [5, 6, 1]^T$

exact	**Algorithm 4.1**	**Algorithm 4.2**
	(it. 927, RRE_x 0.1286)	(it. 82, RRE_x 0.1219)
		
	(it. 927, RRE_y 0.0679)	(it. 82, RRE_y 0.1438)
		

Algorithm 4.1: [Chung and Nagy, *SISC*, 2010]; Algorithm 4.2: new solver

Another example

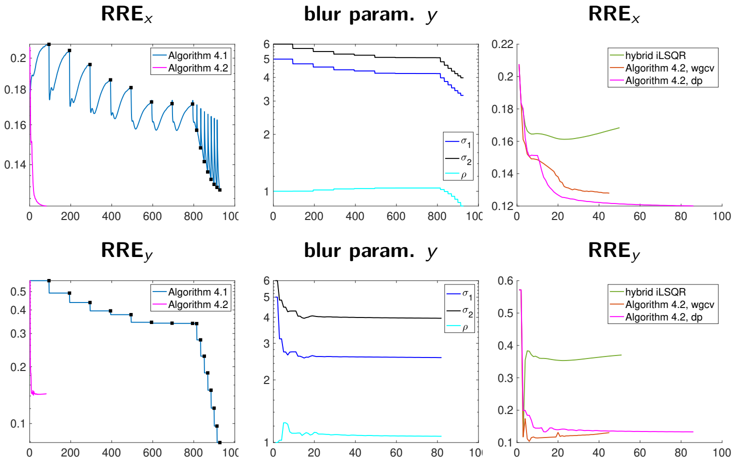
`cameraman` blind deblurring example, with $y_{\text{true}} = [3, 4, 0.5]^T$, $y_0 = [5, 6, 1]^T$



Algorithm 4.1: [Chung and Nagy, *SISC*, 2010]; Algorithm 4.2: new solver

Another example

cameraman blind deblurring example, with $y_{\text{true}} = [3, 4, 0.5]^T$, $y_0 = [5, 6, 1]^T$



Algorithm 4.1: [Chung and Nagy, *SISC*, 2010]; Algorithm 4.2: new solver

Summary and Outlook

Summary and Outlook

- The story so far:
  - introduced the new (hybrid) iLSQR and iCGLS methods
  - applications to separable NLLS problems arising in blind deblurring, handled with a variable projection approach

Summary and Outlook

- The story so far:
  - introduced the new (hybrid) iLSQR and iCGLS methods
  - applications to separable NLLS problems arising in blind deblurring, handled with a variable projection approach
- Looking ahead:
  - inexact solvers other than iLSQR and iCGLS
    methods other than standard formTikhonov
  - nonlinear separable inverse problems other than blind deblurring (MRI, CT, radar, superresolution, instrumental calibration, ML tasks)

Summary and Outlook

- The story so far:
  - introduced the new (hybrid) iLSQR and iCGLS methods
  - applications to separable NLLS problems arising in blind deblurring, handled with a variable projection approach
- Looking ahead:
  - inexact solvers other than iLSQR and iCGLS
    methods other than standard formTikhonov
  - nonlinear separable inverse problems other than blind deblurring (MRI, CT, radar, superresolution, instrumental calibration, ML tasks)

Thanks for your attention!

Silvia Gazzola and Malena Sabaté Landman
Regularization by inexact Krylov methods
with applications to blind deblurring
SIAM J. Matrix Anal. Appl. 42, 2021