Regularization Using a Parameterized Trust Region Subproblem, TRS

Oleg Grodzevich and Henry Wolkowicz

hwolkowicz@uwaterloo.ca

Department of Combinatorics and Optimization

University of Waterloo

1 Outline

- regularization can be reformulated as a TRS, if an appropriate/correct TR radius $\bar{\varepsilon}$ can be found
- each step of an efficient TRS algorithm finds an optimal solution of TRS, $x(\varepsilon)$, for a corresponding TR radius $\|x(\varepsilon)\| \leq \varepsilon$
- our new method for regularization of ill-conditioned problems makes use of the *L-curve* maximum curvature criterion
- uses derivative information while solving TRS to efficiently change the TR radius ε and move along points of the L-curve to get to the point of maximum curvature
- MATLAB code for the algorithm is tested and compared to the conjugate gradient least squares, CGLS

2 Background

Regularization: find approximate solutions for linear least-squares problems

(2.1) **LLS** $\min_x \|Gx - d\|_2 ,$

$G, n \times n$ is: singular or ill-conditioned forward operator
d is: observed data

with noise η:

$$Gx = Gx_{\text{true}} + \eta = d = d_{\text{true}} + \eta.$$

remarkable fact: for many applications, a small amount of noise η can result in a solution $x = G^{\dagger} d$ that has no relation to x_{true}

LLS; (Hadamard) Ill-Posed

- LLS (2.1) from discretizations of linear equations $Tx = d$
- T compact linear operator; unbounded inverse
- x not continuous function of the data d (contaminated by noise)
- Sample important applications:
  - CAT, PET (computer assisted tomography)
  - training of neural networks
  - numerical differentiation
  - interior-point algorithms
  - many more ...(surveys: Engl, Neumaier; book: Groetsch)

Regularization (TRS, Tikhonov)

AIM: *find generalized solutions* stable under small changes in d

TRS Approach:

(2.2) **TRS**
$$\begin{array}{ll} \min & \|Gx-d\|_2^2 \\ \text{subject to} & \|x\|_2^2 \le \varepsilon^2 \end{array}$$

Tikhonov Regularization:

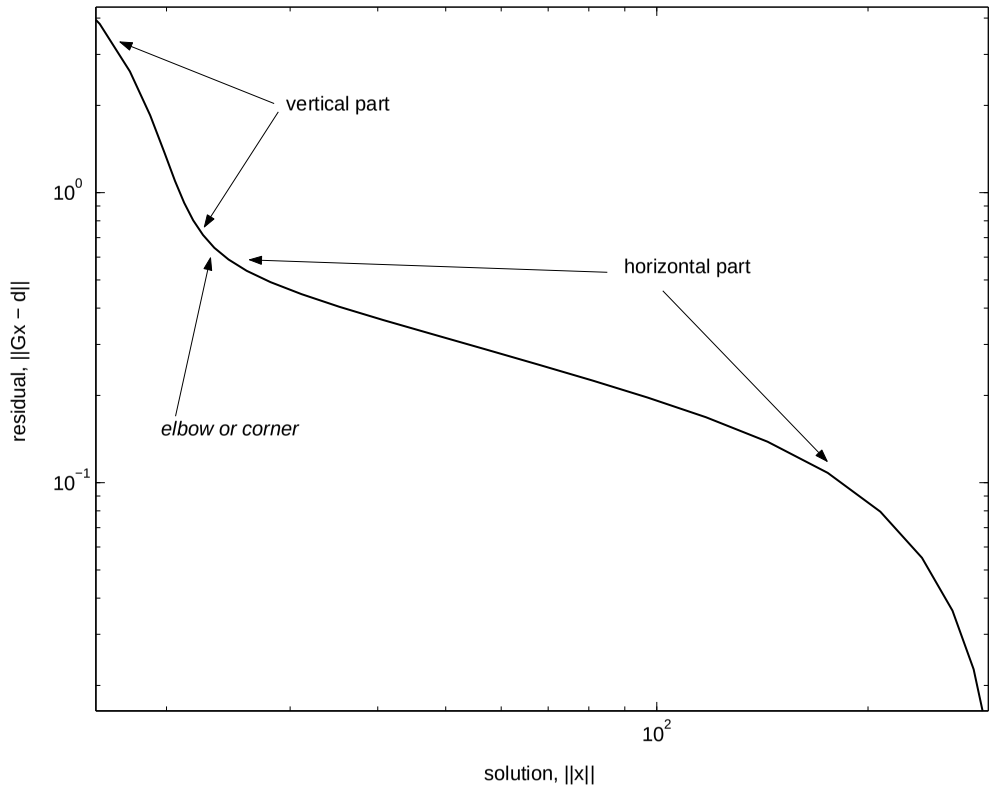
(2.3) **TikhReg** $(G^TG+\alpha^2 I)x_\alpha = G^Td$

Result: larger residual error $\|Gx-d\|_2$ but smaller propagated error in $\|x\|_2$.

L-Curve (Point of Max Curvature)

From TRS: Find the point of max curvature/elbow

$$\mathcal{L}(G,d) = \{(\log(\varepsilon), \log \|Gx(\varepsilon) - d\|_2) : \varepsilon > 0,\ x(\varepsilon) \text{ optimal for } TRS\}$$



TRS

reformulate TRS (2.2)

$$(TRS) \qquad \mu_\varepsilon := \mu(A, a, \varepsilon) := \begin{array}{ll} \min & q(x) := x^T Ax - 2a^T x \\ \text{subject to} & \|x\|_2^2 \leq \varepsilon^2, \end{array}$$

where:

$A = G^T G$ is $n \times n$ symmetric (nonsingular, ill-cond.)

$a = G^T d \in \mathbb{R}^n$

$\varepsilon > 0$

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

define λ^* optimal Lagrange multiplier

$x(0) = A^{-1}a = G^{-1}d$ is unconstrained optimum

Singular Values for Tikhonov Regularization

SVD: $G = USV^T, \quad U^TU = I, \quad V^TV = I, S = \text{Diag}\,(\sigma_i)$
(Lagrange multiplier argument shows equivalence between choosing correct α and correct TRS radius ε)

$$(G^TG + \alpha^2 I)x_\alpha = G^Td$$

$$\begin{aligned}(VSU^TUSV^T + \alpha^2 I)x_\alpha &= VSU^Td \\ V^Tx_\alpha &= (S^2 + \alpha^2 I)^{-1}SU^Td.\end{aligned}$$

Tikhonov filter factors $f_i = \frac{\sigma_i^2}{\sigma_i^2+\alpha^2}$

$$x_\alpha = V(S^2 + \alpha^2 I)^{-1}SU^Td = \sum_{i=1}^n f_i \frac{U_{:i}^T d}{\sigma_i} V_{:i}$$
$$d - Gx_\alpha = d - USV^Tx_\alpha = U(I - S(S^2 + \alpha^2 I)^{-1}S)U^Td,$$

Singular Values for Tikhonov regularization ...

$$\|x_\alpha\|_2^2 = \sum_{i=1}^n f_i^2 \left(\frac{U_{:i}^T d}{\sigma_i}\right)^2$$

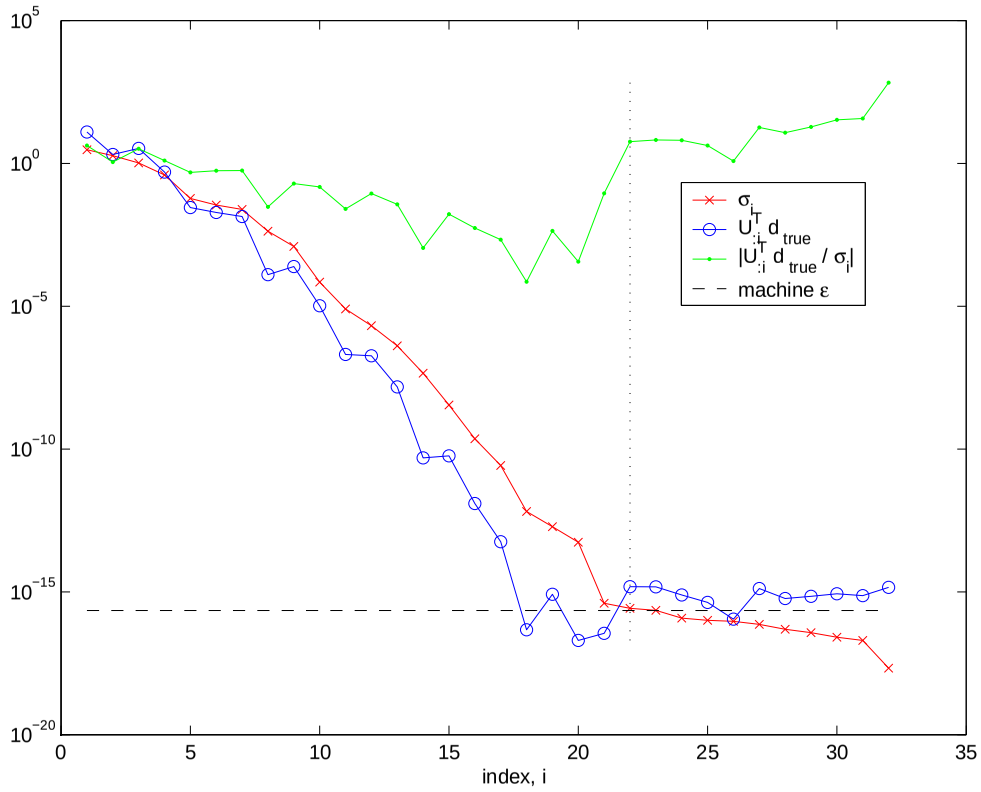
$$\|Gx_\alpha - d\|_2^2 = \sum_{i=1}^n (1-f_i)^2 \left(U_{:i}^T d\right)^2.$$

With: G invertible, $\alpha = 0$, uncorrelated noise η

$$\|x_0\|_2^2 = \sum_{i=1}^n \left(\frac{U_{:i}^T d_{\text{true}}}{\sigma_i} + \frac{U_{:i}^T \eta}{\sigma_i}\right)^2.$$

Discrete Picard Condition

the Fourier coefficients $|U_{:i}^T d_{\text{true}}|$ decay faster than the σ_i



norms of: LLSS is $\sim 10^5$; true solution ~ 10

Curvature of the L-curve

$$\eta = \|x_\varepsilon\|_2^2, \quad \hat{\eta} = \log \eta; \qquad \rho = \|Gx_\varepsilon - d\|_2^2 = \mu_\varepsilon + d^T d, \quad \hat{\rho} = \log \rho$$

curvature of L-curve

$$\begin{aligned} \kappa_\varepsilon &= 2\frac{\hat{\rho}'\hat{\eta}'' - \hat{\rho}''\hat{\eta}'}{((\hat{\rho}')^2 + (\hat{\eta}')^2)^{3/2}} \\ &= \varepsilon^2 \mu_\varepsilon \left(2\varepsilon^2 \lambda^{*2} - 2\mu_\varepsilon \lambda^* - \varepsilon \mu_\varepsilon \left(\frac{\partial \lambda^*}{\partial \varepsilon}\right)\right)\left(\varepsilon^4 \lambda^{*2} + \mu_\varepsilon^2\right)^{-3/2} \end{aligned}$$

$\frac{\partial \lambda^*}{\partial \varepsilon} = \varepsilon / (a^T (A - \lambda^* I)^{-3} a)$ expensive

Curvature Estimation and Gauss Quadrature

Denominator of $\frac{\partial \lambda^*}{\partial \varepsilon} = \varepsilon / (a^T (A - \lambda^* I)^{-3} a)$, is expensive
Use (ref. Golub and Von Matt): find upper and lower bounds

$$l_p(\alpha) \leq \nu_p(\alpha) = d^T G (G^T G + \alpha I)^p G^T d \leq u_p(\alpha)$$

where α is a positive scalar and p is a negative integer
(here: $\alpha = -\lambda^*, p = -3, G^T G = A, G^T d = a$)

- bounds from Lanczos Bidiagonalization on G (with restarts);
- accuracy can be increased as needed

Trust Region Subproblem, TRS

$$(TRS) \qquad \mu_\varepsilon := \mu(A, a, \varepsilon) := \begin{array}{c} \min \\ \text{subject to} \end{array} \begin{array}{l} q(x) := x^T Ax - 2a^T x \\ \|x\|_2^2 \leq \varepsilon^2, \end{array}$$

Characterization of Optimality:

$$\left.\begin{array}{c} (A - \lambda^* I)x^* = a, \\ A - \lambda^* I \succeq 0, \lambda^* \leq 0 \end{array}\right\} \quad \text{dual feasibility}$$

$$\|x^*\|^2 \leq \varepsilon^2 \quad \text{primal feasibility}$$

$$\lambda^*(\|x^*\|^2 - \varepsilon^2) = 0 \quad \text{complementary slackness}$$

For our applications: $\lambda^* < 0 \leq \lambda_1(A)$.
Therefore, $\|x^*\| = \varepsilon$ and the *easy case* holds. (Though *near* hard case can hold.)

Exploiting Optimality Conditions

For each $\bar{\lambda} < 0 \leq \lambda_1(A)$ (implies $A - \bar{\lambda} I \succ 0$)

$$x(\bar{\lambda}) := (A - \bar{\lambda} I)^{-1} a$$

solves TRS with $\varepsilon = \|x(\bar{\lambda})\|$, i.e.

$$(\log(\varepsilon), \log \|Gx(\varepsilon) - d\|)$$ is point on L-curve

we use parameter λ to steer to elbow on L-curve
(while exploiting sparsity)

Derivatives of: λ^*, μ

implicit differentiation on $\|(A-\lambda^* I)^{-1}a\|^2 - \varepsilon^2 = 0$:

$$2\left(\frac{\partial \lambda^*}{\partial \varepsilon}\right) a^T (A-\lambda^* I)^{-3} a = 2\varepsilon$$

i.e.

$$\frac{\partial \lambda^*}{\partial \varepsilon} = \frac{\varepsilon}{a^T (A-\lambda^* I)^{-3} a}.$$

$$\mu_\varepsilon = (x^*)^T A x^* - 2a^T x^* = a^T (A-\lambda^* I)^{-1} a + \lambda^* \varepsilon^2.$$

$$\frac{\partial \mu_\varepsilon}{\partial \varepsilon} = a^T (A-\lambda^* I)^{-2} a \left(-\frac{\partial \lambda^*}{\partial \varepsilon}\right) + \left(\frac{\partial \lambda^*}{\partial \varepsilon}\right) \varepsilon^2 + 2\lambda^* \varepsilon = 2\lambda^* \varepsilon$$

$$\frac{\partial^2 \mu_\varepsilon}{\partial \varepsilon^2} = 2\left(\lambda^* + \varepsilon \frac{\partial \lambda^*}{\partial \varepsilon}\right)$$

Exploit strong Lagrangian duality

$$L(x,\lambda) = x^T A x - 2a^T x + \lambda(\varepsilon^2 - x^T x), \quad \lambda \leq 0 \quad \text{Lagrangian}$$

$$\mu_\varepsilon = \min_x \max_{\lambda \leq 0} L(x,\lambda) = \max_{\lambda \leq 0} \min_x L(x,\lambda) \quad \text{strong duality}$$

Define: $D(t) = \begin{pmatrix} t & -a^T \\ -a & A \end{pmatrix}$, symmetric, $(n+1) \times (n+1)$

$$k(t) = (\varepsilon^2 + 1)\lambda_{\min}(D(t)) - t$$

Then: $\boxed{\mu_\varepsilon = \max_t k(t)}$ unconstr. dual - concave max.

$$k'(t) = (\varepsilon^2 + 1)y_0^2 - 1, \quad \text{normalized eigenvector } \begin{pmatrix} y_0 \\ x \end{pmatrix}$$

Four Parameters

t – control parameter in $k(t), D(t)$

ε – $= \|x(\varepsilon)\|_2$, trust-region radius

α – Tikhonov regularization parameter

λ – optimal Lagrange multiplier for TRS

Relationships (isotonic):

$$\begin{aligned} -\infty &< \lambda = \lambda_1(D(t)) = -\alpha^2 &&\leq 0 \\ 0 &< t = \lambda + d^T G(G^T G - \lambda I)^{-1} G^T d &&\leq \|d\|_2^2 \\ 0 &< \varepsilon = \left\|(G^T G - \lambda I)^{-1} G^T d\right\|_2 &&\leq \left\|G^{-1} d\right\|_2 \end{aligned}$$

Upper bound corresponds to the LLSS

Geometry of Elbow

Define

$$l_r(\varepsilon) := \log(\|Gx(\varepsilon) - d\|_2), \quad l_x(\varepsilon) := \log(\|x(\varepsilon)\|_2)$$

$$\begin{aligned} \frac{\partial(l_r(\varepsilon))}{\partial(l_x(\varepsilon))} &= \frac{1}{2}\frac{\partial(\log(\mu_\varepsilon + d^T d))/\partial(\varepsilon)}{\partial(\log(\varepsilon))/\partial(\varepsilon)} \\ &= \frac{1}{2}\frac{\mu'_\varepsilon \varepsilon}{\mu_\varepsilon + d^T d} \\ &= \frac{\varepsilon^2 \lambda_\varepsilon}{\mu_\varepsilon + d^T d} \end{aligned}$$

large negative number as we approach the *elbow* from the left

negative close to zero if we are at the plateau right of the *elbow*

Initial L-curve point

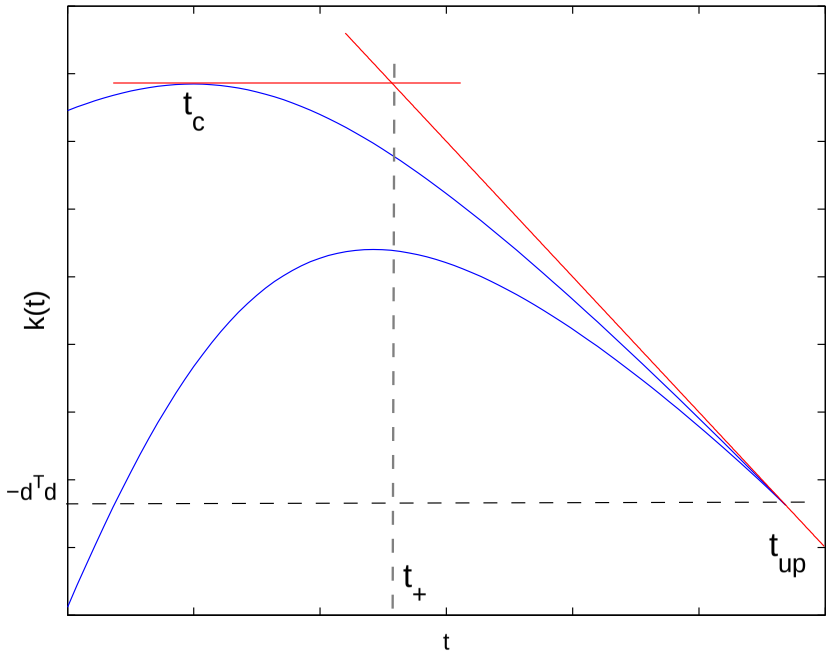
- We start to the left of the elbow;
- each iterate increases the value of t to locate the elbow
- exploit relationships between different parameters

e.g. use initial point $\lambda = -\sigma_n(G)^2$ (lower bound on optimal λ^*)

or use small enough value $t = \frac{d^T d}{2}$

small changes in t result in large changes in ε on the horizontal part to the right (plateau) of the elbow
Conversely, large changes in t result in small changes in ε when on the vertical part to the left of the elbow.

Triangle Interpolation for t_+



(str. concave) $k(t) = (\varepsilon_c^2 + 1)\lambda_{\min}(D(t)) - t;$ $\qquad \mu_\varepsilon = \max_t k(t)$
$k'(t_c) = 0$, $k'(t_{up}) = -1$, independent of ε

$$t_+ = t_c - (\varepsilon^2 + 1)\lambda_1(D(t_c)), \quad \varepsilon_+ = \|x(t_+)\|$$

New $k(t)$ curve for t_+ lies below old

Trust-Region Based Regularization: Initialization

1 compute *largest* singluar value σ_n of G

2 compute initial bidiagonalization (γ, δ) of G using Lanczos Bidiagonalization; use d as starting vector.

3 $t_{low} = 0$

4 $t_{up} = d^T d$

5 $\lambda_{low} = -\sigma_n^2$

6 $\lambda_{up} = 0$

7 $\varepsilon_{up} = -1$

8 $\kappa_{low}^{previous} = \infty$

9 $\kappa_{up}^{previous} = \infty$

10 $\lambda = \lambda_{low}$

11 find starting L-curve point parameters $[t, x, k]$

Trust-Region Based Regularization: Main Loop

1 **while** $\lambda < \lambda_{up} - 10^{-10}$ **do**
2 ♯ *calculate slope of L-curve and* $\frac{\partial \lambda}{\partial t}$
3 $\varepsilon^2 = x^T x, res^2 = k + d^T d$
4 $L_{slope} = \lambda \varepsilon^2 / res^2$
5 $\frac{\partial \lambda}{\partial t} = (1 + \varepsilon^2)^{-1}$
6 save current point to the solutions history
7 $t_{low} = t, \lambda_{low} = \lambda$
8 $[\kappa_{low}, \kappa_{up}] =$ **curvature**$(\varepsilon, res^2, \lambda)$
9 ♯ *termination criteria*
10 **while** *curvature value is not certain* **do**
11 **if** $\kappa_{low} > \kappa_{up}^{previous}$ **then**
12 **DONE**, proceed to the final solution refinement
13 **end**
14 **if** $\kappa_{up} < \kappa_{low}^{previous}$ **then**
15 $\kappa_{low}^{previous} = \kappa_{low}$
16 $\kappa_{up}^{previous} = \kappa_{up}$
17 curvature value is now specified, **break**
18 **else**
19 update bidiagonalization (γ, δ) of G to improve precision

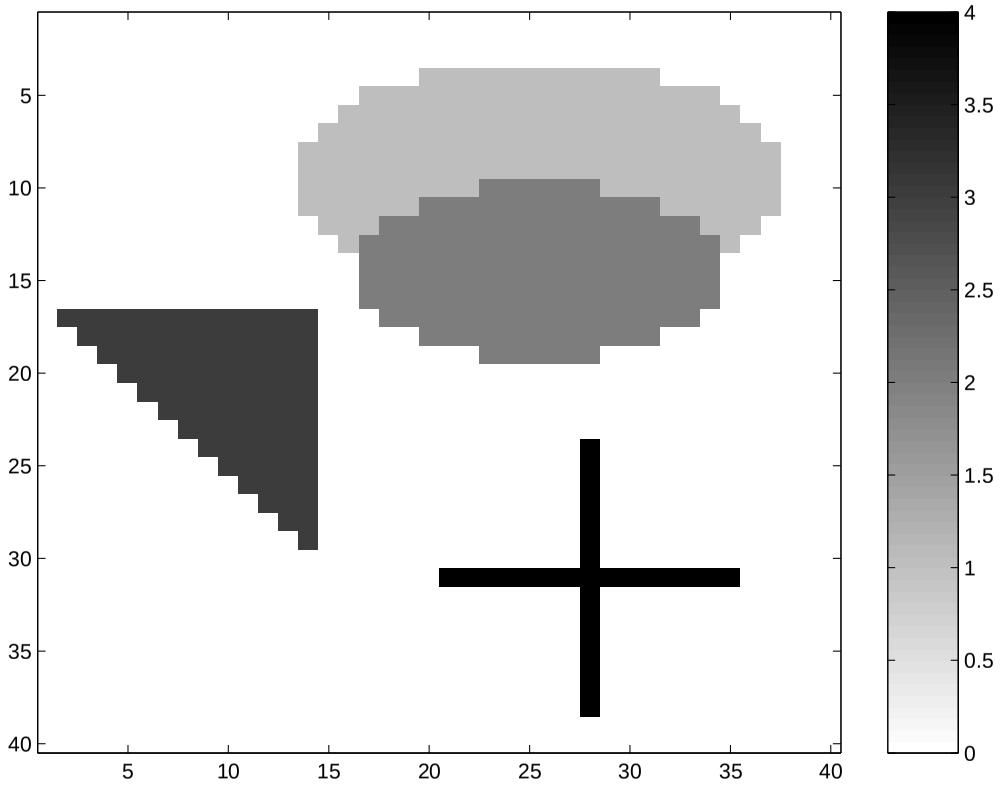
Trust-Region Based Regularization: Main Loop cont 1...

```
1  while λ < λ_up − 10^{−10} do
2      ♯ calculate ....
3      ♯ termination .....
4      while curvature value is not certain do
5          if κ_low > κ_up^previous then
6              DONE, proceed to the final solution refinement
7          end
8          if κ_up < κ_low^previous then
9              κ_low^previous = κ_low
10             κ_up^previous = κ_up
11             curvature value is now specified, break
12         else
13             update bidiagonalization (γ, δ) of G to improve precision
14             [κ_low, κ_up] = curvature(ε, res^2, λ)
15             recalculate bounds on κ^previous
16         end
17     end
18     ♯ update t
19         ε_target = ε
```

Trust-Region Based Regularization: Main Loop cont 2...

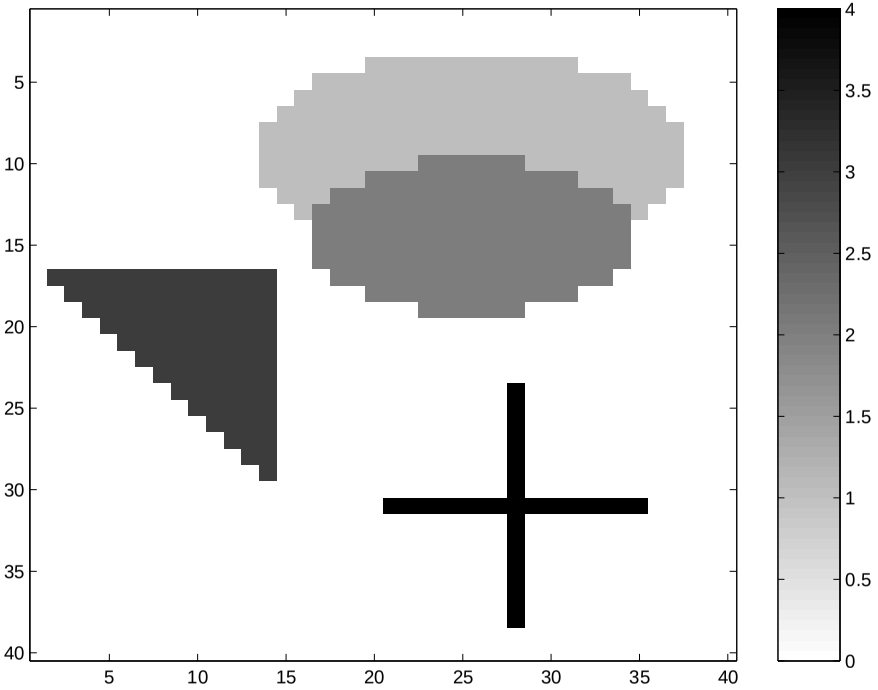
```
1 while λ < λ_up − 10^−10 do
2 |   ♯ calculate ....
3 |   ♯ update t
4 |     ε_target = ε
5 |     perform triangle interpolation on the k(t) to get an estimated t for ε_target
6 |     find next L-curve parameters [t, x, k] using (??)
7 end
```

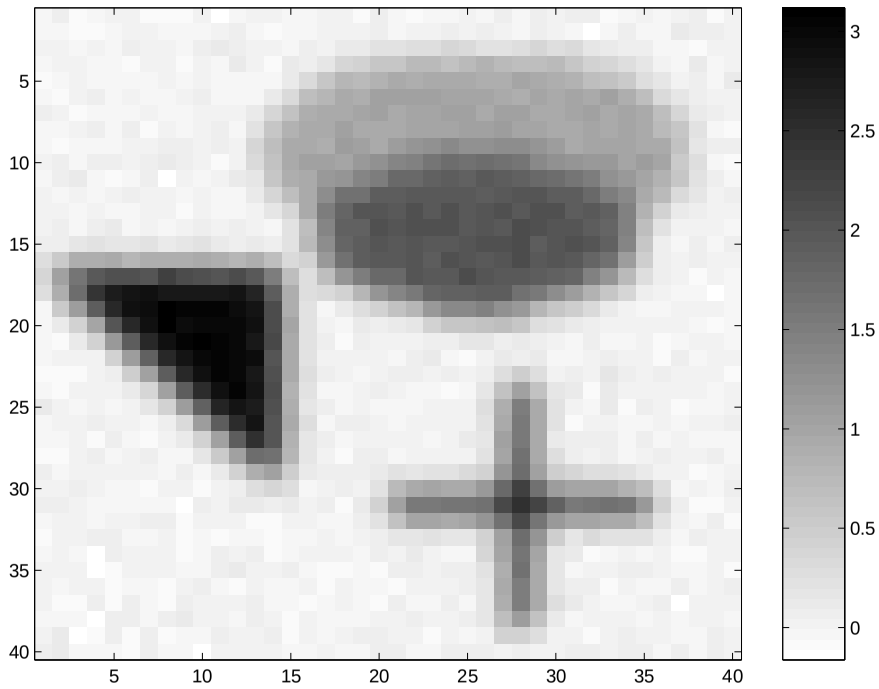
Image Deblurring Example
Original Picture



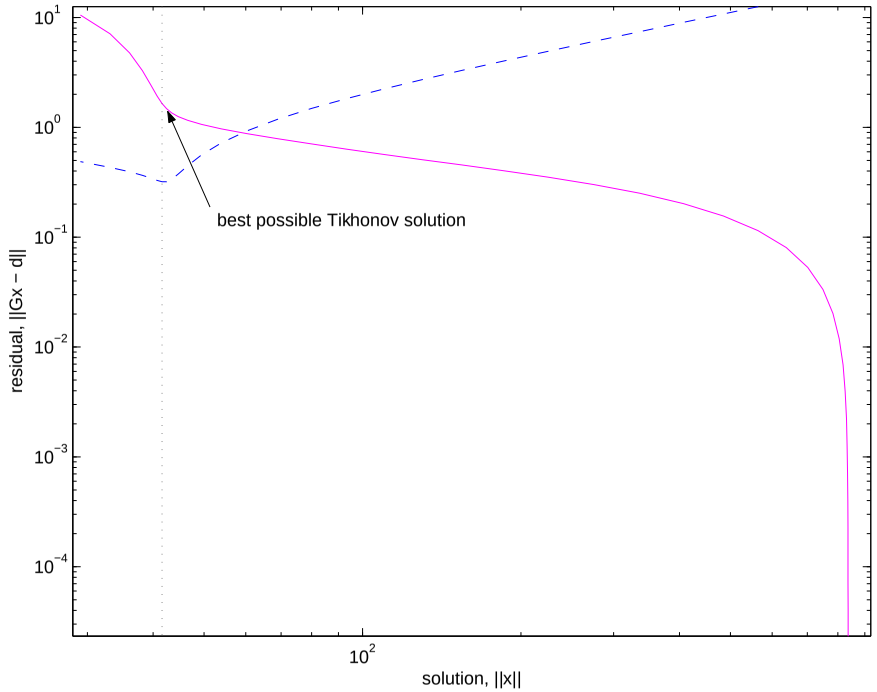
(ref. Hansen MATLAB package - use blur command)

Original Picture and Blurred with Added Noise





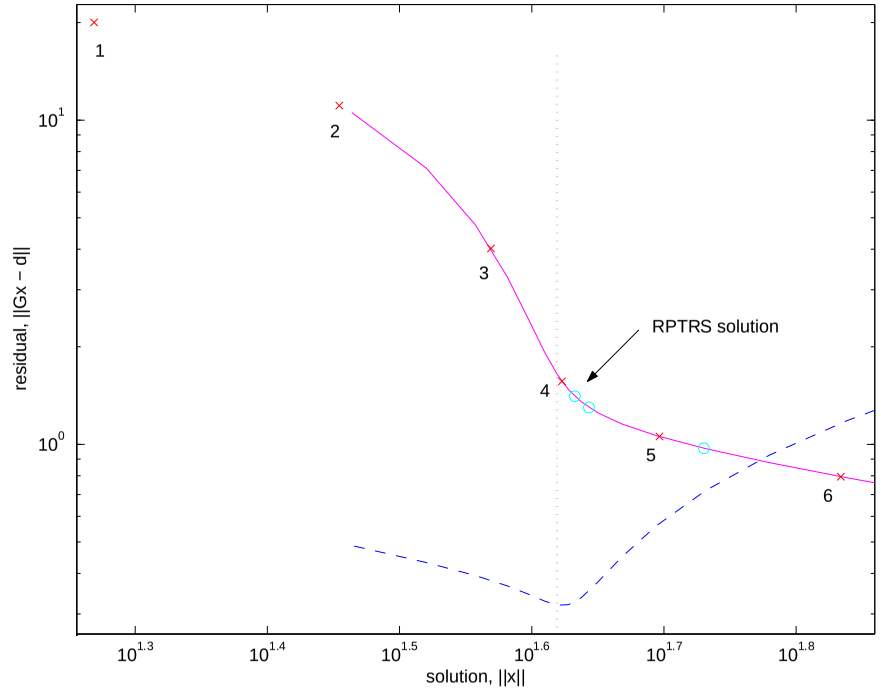
Corresponding L-curve



dashed line shows relative accuracy $\frac{\|x_{\mathrm{true}} - x\|_2}{\|x_{\mathrm{true}}\|_2}$

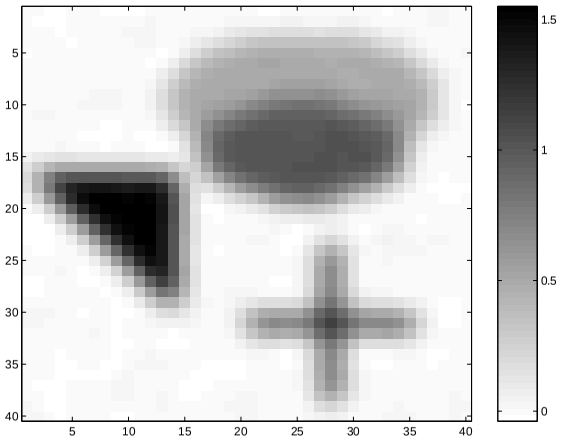
minimum - best possible Tikhonov regularization point

L-curve with RPTRS points

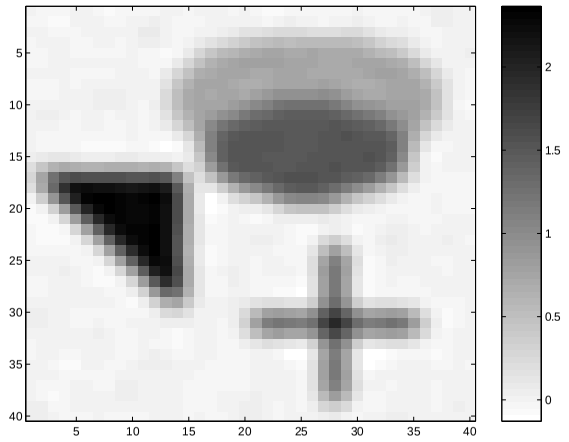


x (cross) visited during the main loop
circles final refinement steps
Final point - close to best Tikhonov solution!

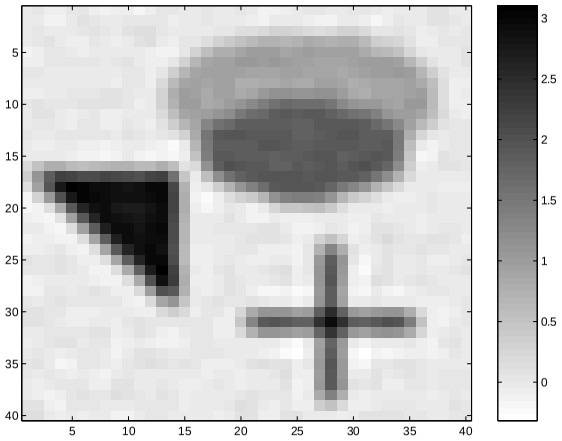
Points 1-4



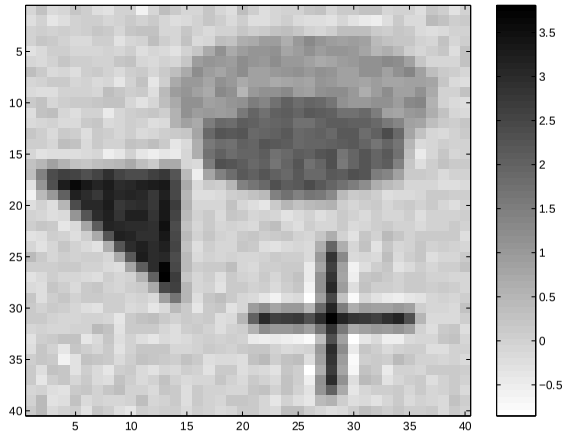
t = 652.166, rel.acc. = 65.39



t = 994.155, rel.acc. = 49.63

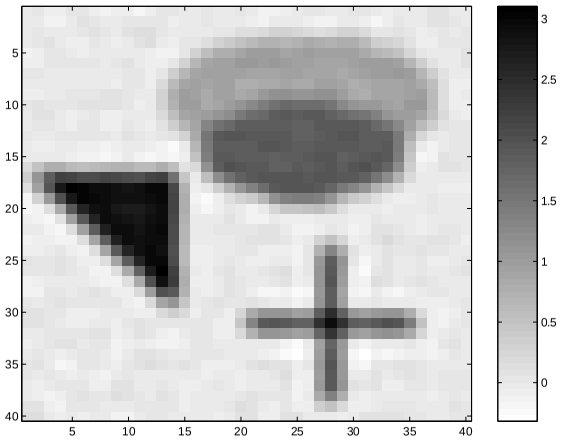


t = 1271.46, rel.acc. = 38.07

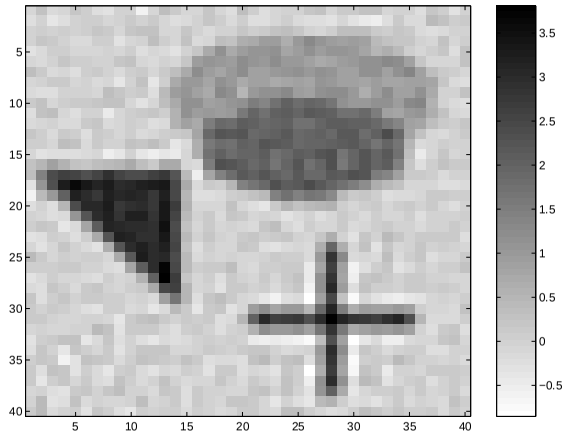


t = 1378.38, rel.acc. = 31.82

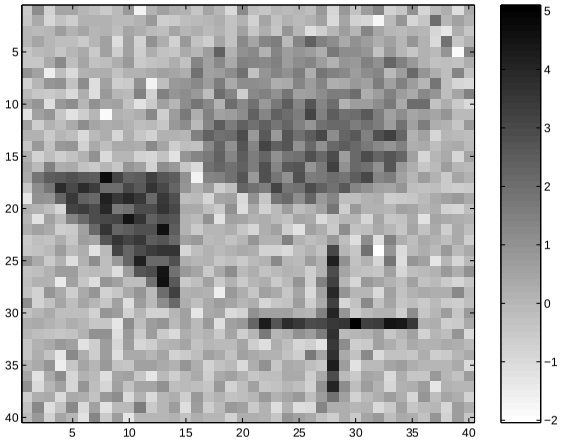
Points 3-6



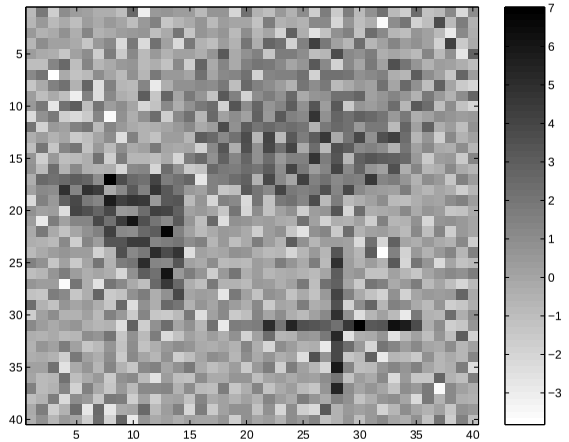
t = 1271.46, rel.acc. = 38.07



t = 1378.38, rel.acc. = 31.82

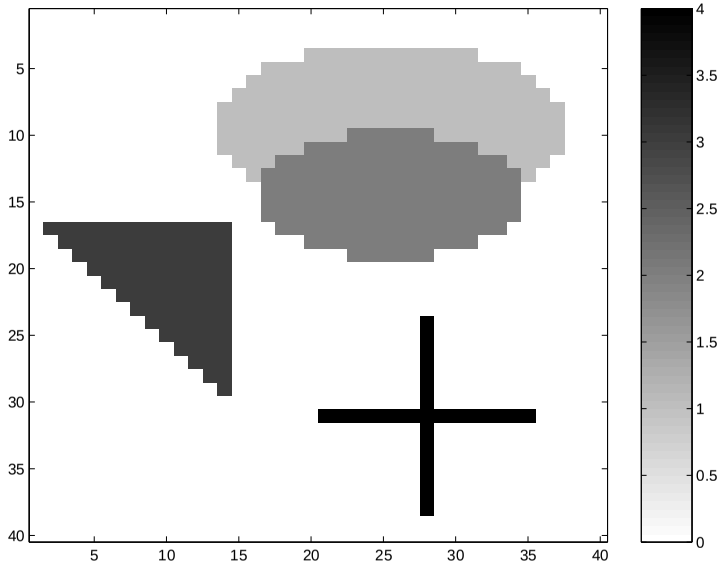


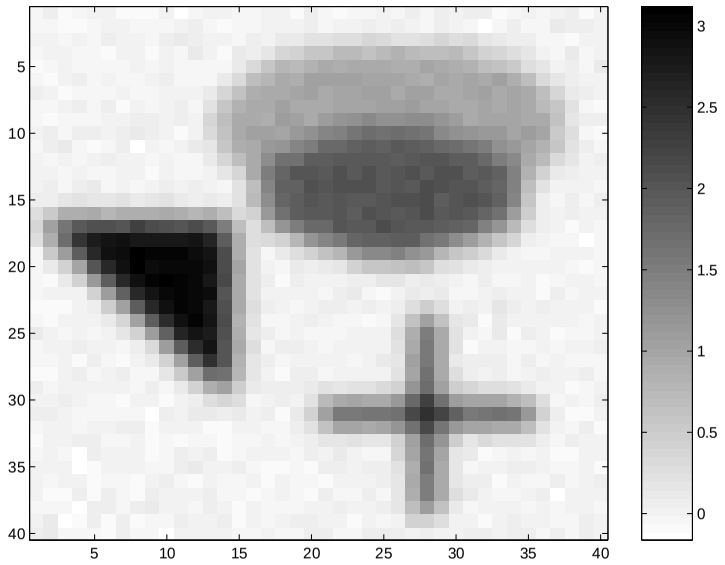
t = 1392.12, rel.acc. = 57.14

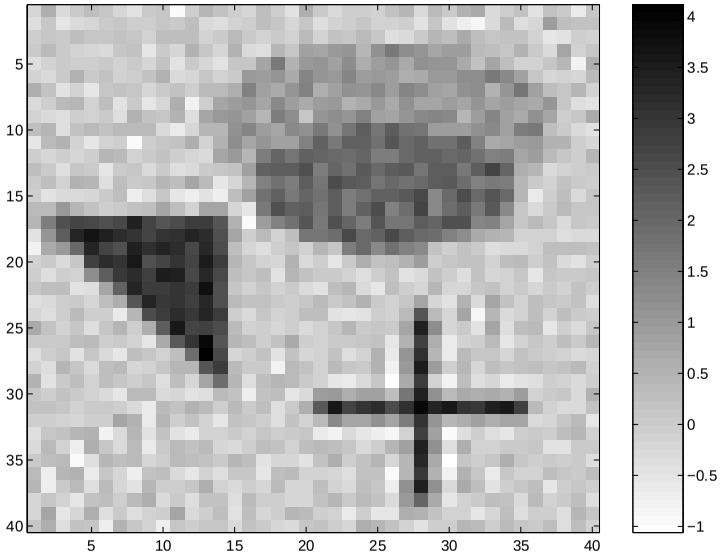


t = 1393.45, rel.acc. = 116.29

Original; Blurred; RPTRS Solution

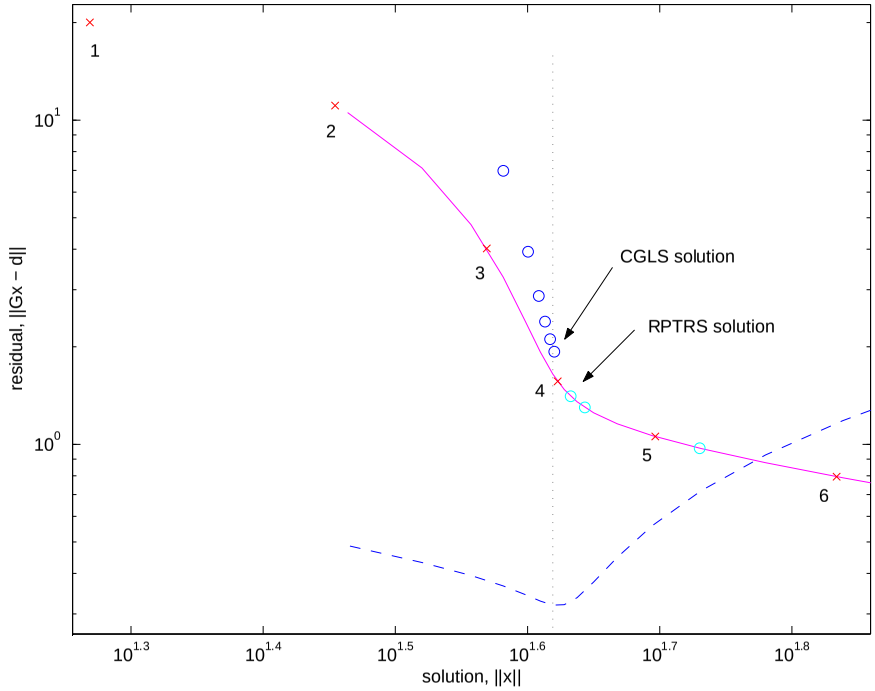




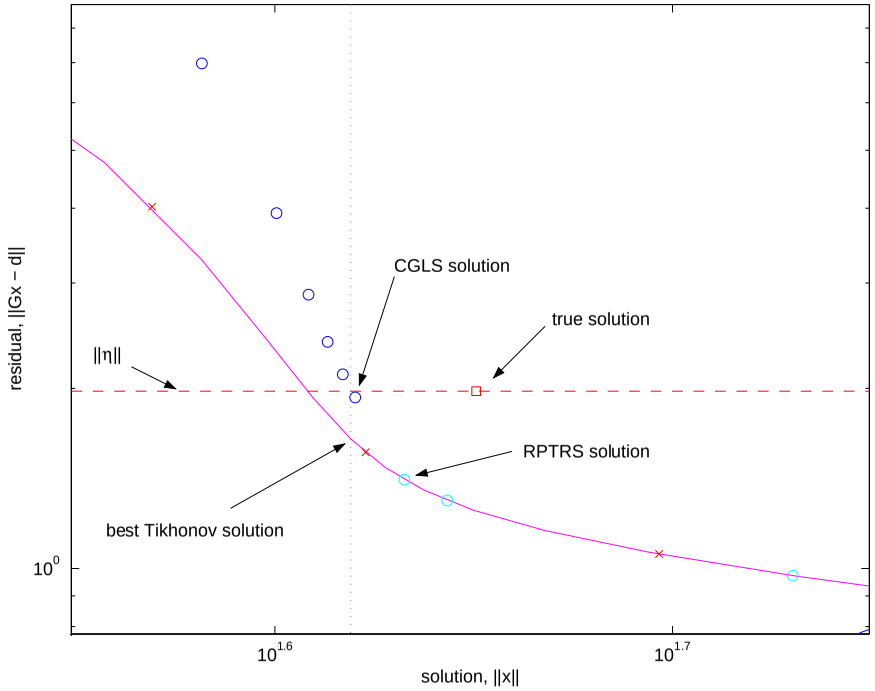


t = 1388.32, rel.acc. = 35.36

L-curve with CGLS and RPTRS



L-curve with CGLS



CGLS, RPTRS with best Tikhonov

Conclusion

- Used ideas from RW algorithm for TRS

- new algorithm efficiently finds point of maximum curvature on the L-curve for regularization of ill-conditioned problems $Gx = d$

- takes advantage of: each iteration of RW algorithm corresponds to a point on the L-curve; cost is approx. ONE TRS solve

- Advantage over CGLS approach when the norm of the error is not known