Discrete ill–posed problems

Gérard MEURANT

October, 2008
1 Introduction to ill-posed problems
2 Tikhonov regularization
3 The L–curve criterion
4 Generalized cross-validation
5 Comparisons of methods
Introduction to ill–posed problems

We speak of a discrete ill–posed problem (DIP) when the solution is sensitive to perturbations of the data

Example:

\[
A = \begin{pmatrix} 0.15 & 0.1 \\ 0.16 & 0.1 \\ 2.02 & 1.3 \end{pmatrix}, \quad c + \Delta c = A \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.01 \\ -0.032 \\ 0.01 \end{pmatrix}
\]

The solution of the perturbed least squares problem (rounded to 4 decimals) using the QR factorization of \(A\) is

\[
x_{QR} = \begin{pmatrix} -2.9977 \\ 7.2179 \end{pmatrix}
\]

Why is it so?
The SVD of $A$ is

$$U = \begin{pmatrix} -0.0746 & 0.7588 & -0.6470 \\
-0.0781 & -0.6513 & -0.7548 \\
-0.9942 & -0.0058 & 0.1078 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 2.4163 & 0 \\
0 & 0.0038 \\
0 & 0 \end{pmatrix}$$

$$V = \begin{pmatrix} -0.8409 & -0.5412 \\
-0.5412 & 0.8409 \end{pmatrix}$$

The component $(u^2)^T \Delta c/\sigma_2$ ($u^2$ being the second column of $U$) corresponding to the smallest nonzero singular value is large being 6.2161

This gives the large change in the solution
\[ Ax \approx c = \bar{c} - e \]

where \( A \) is a matrix of dimension \( m \times n \), \( m \geq n \) and the right hand side \( \bar{c} \) is contaminated by a (generally) unknown noise vector \( e \)

- The standard solution of the least squares problem
  \[ \min ||c - Ax|| \] (even using backward stable methods like QR)
  may give a vector \( x \) severely contaminated by noise
- This may seem hopeless
- The solution is to modify the problem by regularization
- We have to find a balance between obtaining a problem that we can solve reliably and obtaining a solution which is not too far from the solution without noise
Tikhonov regularization

Replace the LS problem by

$$\min_x \{ \| c - Ax \|^2 + \mu \| x \|^2 \}$$

where $\mu \geq 0$ is a regularization parameter to be chosen.

For some problems (particularly in image restoration) it is better to consider

$$\min_x \{ \| c - Ax \|^2 + \mu \| Lx \|^2 \}$$

where $L$ is typically the discretization of a derivative operator of first or second order.

The solution $x_\mu$ of the problem solves the linear system

$$(A^T A + \mu I)x = A^T c$$
The main problem is to choose $\mu$

- If $\mu$ is too small the solution is contaminated by the noise in the right hand side
- If $\mu$ is too large the solution is a poor approximation of the original problem
- Many methods have been devised for choosing $\mu$
- Most of these methods lead to the evaluation of bilinear forms with different matrices
Some methods for choosing $\mu$

- Morozov’s discrepancy principle
  Ask for the norm of the residual to be equal to the norm of the noise vector
  \[ \| c - A(A^TA + \mu I)^{-1}A^Tc \| = \| e \| \]

- The Gfrerer/Raus method
  \[ \mu^3 c^T (AA^T + \mu I)^{-3} c = \| e \|^2 \]

- The quasi–optimality criterion
  \[ \min \left[ \mu^2 c^T A(A^TA + \mu I)^{-4} A^T c \right] \]
The L–curve criterion

- plot the curve \((\|x_\mu\|, \|b - Ax_\mu\|)\) obtained by varying the value of \(\mu \in [0, \infty)\) in log–log scale
- In most cases this curve is shaped as an “L”
- \textbf{Lawson and Hanson} proposed to choose the value \(\mu_L\) corresponding to the “corner” of the L–curve (the point of maximal curvature (see also Hansen; Hansen and O’Leary))
- This is done to have a balance between \(\mu\) being too small and the solution contaminated by the noise, and \(\mu\) being too large giving a poor approximation of the solution. The “vertex” of the L–curve gives an average value between these two extremes
An example of L–curve

The L-curve for the Baart problem, $m = n = 100$, noise $= 10^{-3}$
How to locate the corner of the L–curve?

see Hansen and al.

- Easy if we know the SVD of $A$
- Otherwise compute points on the L–curve and use interpolation
- However, computing a point on the L–curve is expensive
- Alternative, L–ribbon approximation (Calvetti, Golub and Reichel)
The L–ribbon

\[
\|x_\mu\|^2 = c^T A (A^T A + \mu I)^{-2} A^T c
\]

and

\[
\|c - Ax_\mu\|^2 = c^T c + c^T A (A^T A + \mu I)^{-1} A^T A (A^T A + \mu I)^{-1} A^T c
\]
\[
-2c^T A (A^T A + \mu I)^{-1} A^T c
\]

By denoting \(K = A^T A\) and \(d = A^T c\)

\[
\|c - Ax_\mu\|^2 = c^T c + d^T K (K + \mu I)^{-2} d - 2d^T (K + \mu I)^{-1} d
\]
Define

\[ \phi_1(t) = (t + \mu)^{-2} \]
\[ \phi_2(t) = t(t + \mu)^{-2} - 2(t + \mu)^{-1} \]

we are interested in \( s_i = d^T \phi_i(K)d, \ i = 1, 2 \)

We can obtain bounds using the Golub–Kahan bidiagonalization algorithm

At iteration \( k \), the algorithm computes a Jacobi matrix \( J_k = B_k^T B_k \) and the Gauss rule gives

\[ I_G^k(\phi_i) = \|d\|^2 (e_1)^T \phi_i(J_k)e_1 \]

We can also use the Gauss–Radau rule with a prescribed node \( a = 0 \)

\[ I_{GR}^k(\phi_i) = \|d\|^2 (e_1)^T \phi_i(\hat{J}_k)e_1 \]
\( \hat{J}_k = \hat{B}_k^T \hat{B}_k \) where \( \hat{B}_k \) is obtained from \( B_k \) by setting the last diagonal element \( \delta_k = 0 \)

**Theorem**

\[
I_k^G(\phi_1) \leq s_1 \leq I_k^{GR}(\phi_1)
\]

where

\[
I_k^G(\phi_1) = \|d\|^2(e^1)^T(B_k^T B_k + \mu I)^{-2}e^1
\]
\[
I_k^{GR}(\phi_1) = \|d\|^2(e^1)^T(\hat{B}_k^T \hat{B}_k + \mu I)^{-2}e^1
\]

\[
I_k^{GR}(\phi_2) \leq s_2 \leq I_k^G(\phi_2)
\]

where

\[
I_k^G(\phi_2) = \|d\|^2[(e^1)^T B_k^T B_k(B_k^T B_k + \mu I)^{-2}e^1 - 2(e^1)^T(B_k^T B_k + \mu I)^{-1}e^1]
\]
\[
I_k^{GR}(\phi_2) = \|d\|^2[(e^1)^T \hat{B}_k^T \hat{B}_k(\hat{B}_k^T \hat{B}_k + \mu I)^{-2}e^1 - 2(e^1)^T(\hat{B}_k^T \hat{B}_k + \mu I)^{-1}e^1]
\]
\[ x^- (\mu) = \sqrt{I_k^G(\phi_1)}, \quad x^+ (\mu) = \sqrt{I_k^{GR}(\phi_1)} \]

\[ y^- (\mu) = \sqrt{c^T c + I_k^{GR}(\phi_2)}, \quad y^+ (\mu) = \sqrt{c^T c + I_k^G(\phi_2)} \]

For a given value of \( \mu > 0 \) the bounds are

\[ x^- (\mu) \leq \| x_\mu \| \leq x^+ (\mu), \quad y^- (\mu) \leq \| c - Ax_\mu \| \leq y^+ (\mu) \]

Calvetti, Golub and Reichel defined the L–ribbon as the union of rectangles for all \( \mu > 0 \)

\[ \bigcup_{\mu > 0} \{ \{ x(\mu), y(\mu) \} : x^- (\mu) \leq x(\mu) \leq x^+ (\mu), \ y^- (\mu) \leq y(\mu) \leq y^+ (\mu) \} \]

Then, we have to select a point (a value of \( \mu \)) inside the L–ribbon
The L–curvature

Another possibility is to obtain bounds of the curvature (in log–log scale) and to look for the maximum

\[ C_\mu = 2 \frac{\rho''\eta' - \rho'\eta''}{((\rho')^2 + (\eta')^2)^{3/2}} \]

where \( I \) denotes differentiation with respect to \( \mu \) and

\[ \rho(\mu) = \frac{1}{2} \log \| c - Ax_\mu \| = \log \mu^2 c^T \phi(AA^T)c \]
\[ \eta(\mu) = \frac{1}{2} \log \| x_\mu \| = \log c^T A\phi(A^TA)A^Tc \]

where \( \phi(t) = (t + \mu)^{-2} \)
The first derivatives can be computed as

$$\rho'(\mu) = \frac{c^T A(A^T A + \mu I)^{-3} A^T c}{\mu c^T (A A^T + \mu I)^{-2} c}$$

$$\eta'(\mu) = -\frac{c^T A(A^T A + \mu I)^{-3} A^T c}{c^T A(A^T A + \mu I)^{-2} A^T c}$$

The numerator is more complicated

$$\rho' \eta'' - \rho'' \eta' = \left(\frac{c^T A(A^T A + \mu I)^{-3} A^T c}{\mu c^T (A A^T + \mu I)^{-2} c \cdot c^T A(A^T A + \mu I)^{-2} A^T c} \right)^2$$

$$\left(\frac{c^T (A A^T + \mu I)^{-2} c \cdot c^T A(A^T A + \mu I)^{-2} A^T c}{c^T (A A^T + \mu I)^{-2} c \cdot c^T A(A^T A + \mu I)^{-2} A^T c} \right)$$

$$+ 2 \mu c^T (A A^T + \mu I)^{-3} c \cdot c^T A(A^T A + \mu I)^{-2} A^T c$$

$$- 2 \mu c^T (A A^T + \mu I)^{-2} c \cdot c^T A(A^T A + \mu I)^{-3} A^T c$$
Locating the corner of the L–curve

- Using the SVD (Hansen): 1c
- Pruning algorithm (Hansen, Jensen and Rodriguez): 1p
- Rotating the L–curve (GM): 1c1
- Finding an interval where $\log \| x_\mu \|$ and $\log \| c - Ax_\mu \|$ are almost constant (GM): 1c2
L-curve algorithms, Baart problem, $n = 100$

<table>
<thead>
<tr>
<th>noise</th>
<th>meth</th>
<th>$\mu$</th>
<th>$|c - Ax|$</th>
<th>$|x - x_0|$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-3}$</td>
<td>opt</td>
<td>$2.4990 \times 10^{-8}$</td>
<td>$9.8720 \times 10^{-4}$</td>
<td>$1.5080 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>lc</td>
<td>$4.5414 \times 10^{-9}$</td>
<td>$9.8524 \times 10^{-4}$</td>
<td>$1.6030 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>lp</td>
<td>$8.2364 \times 10^{-9}$</td>
<td>$9.8545 \times 10^{-4}$</td>
<td>$1.5454 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>lc1</td>
<td>$6.3232 \times 10^{-9}$</td>
<td>$9.8534 \times 10^{-4}$</td>
<td>$1.5669 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>lc2</td>
<td>$5.8203 \times 10^{-12}$</td>
<td>$9.8463 \times 10^{-4}$</td>
<td>$4.1492 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$4.1297 \times 10^{-8}$</td>
<td>$9.8996 \times 10^{-4}$</td>
<td>$1.5153 \times 10^{-1}$</td>
</tr>
</tbody>
</table>
L-curve algorithms, Phillips problem, $n = 200$

<table>
<thead>
<tr>
<th>noise</th>
<th>meth</th>
<th>$\mu$</th>
<th>$|c - Ax|$</th>
<th>$|x - x_0|$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-3}$</td>
<td>opt</td>
<td>$8.5392 \times 10^{-7}$</td>
<td>$9.9864 \times 10^{-4}$</td>
<td>$7.3711 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>lc</td>
<td>$7.1966 \times 10^{-10}$</td>
<td>$8.5111 \times 10^{-4}$</td>
<td>$5.3762 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>lp</td>
<td>$4.5729 \times 10^{-10}$</td>
<td>$8.3869 \times 10^{-4}$</td>
<td>$6.8849 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>lc1</td>
<td>$3.6084 \times 10^{-10}$</td>
<td>$8.3172 \times 10^{-4}$</td>
<td>$7.8603 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>lc2</td>
<td>$1.0250 \times 10^{-9}$</td>
<td>$8.6013 \times 10^{-4}$</td>
<td>$4.4563 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>2.9147 \times 10^{-7}</td>
<td>$9.7098 \times 10^{-4}$</td>
<td>$1.3595 \times 10^{-2}$</td>
<td></td>
</tr>
</tbody>
</table>
## L-ribbon

<table>
<thead>
<tr>
<th>Ex</th>
<th>noise</th>
<th>nb it</th>
<th>$\mu$</th>
<th>nb it no reorth.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baart</td>
<td>$10^{-7}$</td>
<td>11</td>
<td>$6.0889 \times 10^{-17}$</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>9</td>
<td>$6.1717 \times 10^{-13}$</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>$10^{-3}$</td>
<td>8</td>
<td>$6.3232 \times 10^{-9}$</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>$10^{-1}$</td>
<td>6</td>
<td>$7.2928 \times 10^{-5}$</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>5</td>
<td>$3.260 \times 10^{-2}$</td>
<td>5</td>
</tr>
</tbody>
</table>
Generalized cross-validation

GCV comes from statistics (Golub, Heath and Wahba)

The regularized problem is written as

$$\min \{ ||c - Ax||^2 + m\mu ||x||^2 \}$$

where \( \mu \geq 0 \) is the regularization parameter and the matrix \( A \) is \( m \) by \( n \)

The GCV estimate of the parameter \( \mu \) is the minimizer of

$$G(\mu) = \frac{\frac{1}{m}(I - A(A^TA + m\mu I)^{-1}A^T)c||^2}{\left(\frac{1}{m}\text{tr}(I - A(A^TA + m\mu I)^{-1}A^T)\right)^2}$$
If we know the SVD of $A$ and $m \geq n$ this can be computed as

$$G(\nu) = \frac{m \left\{ \sum_{i=1}^{r} d_i^2 \left( \frac{\nu}{\sigma_i^2 + \nu} \right)^2 + \sum_{i=r+1}^{m} d_i^2 \right\}}{[m - n + \sum_{i=1}^{r} \frac{\nu}{\sigma_i^2 + \nu}]^2}$$

where $\nu = m \mu$

- $G$ is almost constant when $\nu$ is very small or large, at least in log–log scale
- When $\nu \to \infty$, $G(\nu) \to \|c\|^2 / m$
- When $\nu \to 0$ the situation is different wether $m = n$ or not
An example of GCV function

GCV function for the Baart problem, $m = n = 100$, noise = $10^{-3}$
GCV functions for the Bäär problem, $m = n = 100$ for different noise levels
The main problem is that the GCV function is usually quite flat near the minimum.

For large problems we cannot use the SVD:

- First we approximate the trace in the denominator $\rightarrow \tilde{G}$
- Then using the Golub–Kahan bidiagonalization algorithms we can obtain bounds of all the terms in $\tilde{G}$
- Finally we have to locate the minimum of the lower and/or upper bounds.
Proposition

Let $B$ be a symmetric matrix of order $n$ with $\text{tr}(B) \neq 0$.
Let $\mathcal{Z}$ be a discrete random variable with values $1$ and $-1$ with equal probability $0.5$ and let $z$ be a vector of $n$ independent samples from $\mathcal{Z}$.
Then $z^T B z$ is an unbiased estimator of $\text{tr}(B)$.

$$E(z^T B z) = \text{tr}(B)$$

$$\text{var}(z^T B z) = 2 \sum_{i \neq j} b_{i,j}^2$$

where $E(\cdot)$ denotes the expected value and $\text{var}$ denotes the variance.
For GCV we just use one vector $z$. 
$G$ (plain) and $\tilde{G}$ (dotted) functions for the Baart problem,

$m = n = 100$, noise $= 10^{-3}$
$G (\nu)$, $G_t (\nu)$

$G$ (plain) and $\tilde{G}$ (dotted) functions for the Baart problem,

$m = n = 100$, $noise = 10^{-1}$
The Golub and Von Matt algorithm

Let \( s_z(\nu) = z^T(A^T A + \nu l)^{-1}z \), where \( z \) is a random vector. Using Gauss and Gauss–Radau we can obtain

\[
g_z(\nu) \leq s_z(\nu) \leq r_z(\nu)
\]

We can also bound \( s_c^{(p)}(\nu) = c^T A(A^T A + \nu l)^{-p} A^T c, \quad p = 1, 2 \) satisfying

\[
g_c^{(p)}(\nu) \leq s_c^{(p)}(\nu) \leq r_c^{(p)}(\nu)
\]

We want to compute approximations of

\[
\tilde{G}(\mu) = m \frac{c^T c - s_c^{(-1)}(\nu) - \nu s_c^{(-2)}(\nu)}{(m - n + \nu s_z(\nu))^2}
\]
We define

\[ L_0(\nu) = m \frac{c^T c - r_c^{(-1)}(\nu) - \nu r_c^{(-2)}(\nu)}{(m - n + \nu r_z(\nu))^2} \]

\[ U_0(\nu) = m \frac{c^T c - g_c^{(-1)}(\nu) - \nu g_c^{(-2)}(\nu)}{(m - n + \nu g_z(\nu))^2} \]

These quantities \( L_0 \) and \( U_0 \) are lower and upper bounds for the estimate of \( G(\mu) \).

We can also compute estimates of the derivatives of \( L_0 \) and \( U_0 \).

These bounds improve with the number of Lanczos iterations.
They first do $k_{min} = \lceil 3 \log \min(m, n) \rceil$ Lanczos iterations.

Then the global minimizer $\hat{\nu}$ of $U_0(\nu)$ is computed.

If one can find a $\nu$ such that $0 < \nu < \hat{\nu}$ and $L_0(\nu) > L_0(\hat{\nu})$, the algorithm stops and return $\hat{\nu}$.

Otherwise, the algorithm executes one more Lanczos iteration and repeats the convergence test.

Von Matt computed the minimum of the upper bound:

- By sampling the function on 100 points with an exponential distribution.
- If the neighbors of the minimum do not have the same values, he looked at the derivative and sought for a local minimum in either the left or right interval depending on the sign of the derivative.
- The local minimum is found by using bisection.
The upper bound does not have the right asymptotic behavior when \( m = n \) and \( \nu \to 0 \)

\[ G \text{ (plain) and } \tilde{G} \text{ (dashed) functions and upper bounds for the Baart problem, } m = n = 100, \text{ noise } = 10^{-3} \]
To obtain a better behavior we add a term $\|c\|^2$ to the denominator.

$G$ (plain) and $\tilde{G}$ (dashed) functions and upper bounds for the Baart problem, $m = n = 100$, noise $= 10^{-3}$
Optimization of the algorithm

- We choose a (small) value of $\nu$ (denoted as $\nu_0$)
- When
\[
\left| \frac{U_k^0(\nu_0) - U_{k-1}^0(\nu_0)}{U_{k-1}^0(\nu_0)} \right| \leq \epsilon_0
\]
we start computing the minimum of the upper bound

The algorithm for finding the minimum is modified as follows

- We work in log–log scale and compute only a minimizer of the upper bound
- We evaluate the numerator of the approximation by computing the SVD of $B_k$ once per iteration
- We compute 50 samples of the function on a regular mesh
- We locate the minimum, say the point $k$, we then compute again 50 samples in the interval $[k-1 \ k+1]$
We use the Von Matt algorithm for computing a local minimum in this interval.

After locating a minimum $\nu_k$ with a value of the upper bound $U^0_k$ at iteration $k$, the stopping criteria is

$$\left| \frac{\nu_k - \nu_{k-1}}{\nu_{k-1}} \right| + \left| \frac{U^0_k - U^0_{k-1}}{U^0_{k-1}} \right| \leq \epsilon$$
## GCV algorithms, Baart problem

<table>
<thead>
<tr>
<th>noise</th>
<th>$\mu$</th>
<th>$|c - Ax|$</th>
<th>$|x - x_0|$</th>
<th>t (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-7}$</td>
<td>$9.6482 , 10^{-15}$</td>
<td>$9.8049 , 10^{-8}$</td>
<td>$5.9424 , 10^{-2}$</td>
<td>0.38</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>$9.7587 , 10^{-12}$</td>
<td>$9.8566 , 10^{-6}$</td>
<td>$6.5951 , 10^{-2}$</td>
<td>0.18</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>$1.2018 , 10^{-8}$</td>
<td>$9.8573 , 10^{-4}$</td>
<td>$1.5239 , 10^{-1}$</td>
<td>0.16</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>$1.0336 , 10^{-7}$</td>
<td>$9.8730 , 10^{-2}$</td>
<td>$1.6614$</td>
<td>--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>noise</th>
<th>$\mu$</th>
<th>$|c - Ax|$</th>
<th>$|x - x_0|$</th>
<th>t (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-7}$</td>
<td>$1.0706 , 10^{-14}$</td>
<td>$9.8058 , 10^{-8}$</td>
<td>$5.9519 , 10^{-2}$</td>
<td>0.18</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>$1.0581 , 10^{-11}$</td>
<td>$9.8588 , 10^{-6}$</td>
<td>$6.5957 , 10^{-2}$</td>
<td>0.27</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>$1.3077 , 10^{-8}$</td>
<td>$9.8582 , 10^{-4}$</td>
<td>$1.5205 , 10^{-1}$</td>
<td>0.14</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>$1.1104 , 10^{-7}$</td>
<td>$9.8736 , 10^{-2}$</td>
<td>$1.6227$</td>
<td>--</td>
</tr>
</tbody>
</table>
GCV algorithms, Phillips problem

<table>
<thead>
<tr>
<th>noise</th>
<th>$\mu$</th>
<th>$|c - Ax|$</th>
<th>$|x - x_0|$</th>
<th>t (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>vm</td>
<td>$10^{-7}$</td>
<td>$8.7929 \times 10^{-11}$</td>
<td>$9.0162 \times 10^{-8}$</td>
<td>$2.2391 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>$4.5432 \times 10^{-9}$</td>
<td>$9.0825 \times 10^{-6}$</td>
<td>$2.2620 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$10^{-3}$</td>
<td>$4.3674 \times 10^{-7}$</td>
<td>$9.7826 \times 10^{-4}$</td>
<td>$1.0057 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$10^{-1}$</td>
<td>$3.8320 \times 10^{-5}$</td>
<td>$9.8962 \times 10^{-2}$</td>
<td>$9.3139 \times 10^{-2}$</td>
</tr>
<tr>
<td>gm-opt</td>
<td>$10^{-7}$</td>
<td>$1.6343 \times 10^{-10}$</td>
<td>$1.1260 \times 10^{-7}$</td>
<td>$2.2163 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>$5.3835 \times 10^{-9}$</td>
<td>$9.1722 \times 10^{-6}$</td>
<td>$2.1174 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$10^{-3}$</td>
<td>$4.1814 \times 10^{-7}$</td>
<td>$9.7737 \times 10^{-4}$</td>
<td>$1.0375 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$10^{-1}$</td>
<td>$4.1875 \times 10^{-5}$</td>
<td>$9.9016 \times 10^{-2}$</td>
<td>$9.0659 \times 10^{-2}$</td>
</tr>
</tbody>
</table>
Comparisons of methods

Baart problem, $n = 100$

<table>
<thead>
<tr>
<th>noise</th>
<th>meth</th>
<th>$\mu$</th>
<th>$|c - Ax|$</th>
<th>$|x - x_0|$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-3}$</td>
<td>$\mu$ opt</td>
<td>$2.7826 \times 10^{-8}$</td>
<td>$2.3501 \times 10^{-3}$</td>
<td>$1.5084 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$vm$</td>
<td>$1.2018 \times 10^{-8}$</td>
<td>$9.8573 \times 10^{-4}$</td>
<td>$1.5239 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$gm$-opt</td>
<td>$1.3077 \times 10^{-8}$</td>
<td>$9.8582 \times 10^{-4}$</td>
<td>$1.5205 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$gcv$</td>
<td>$9.4870 \times 10^{-9}$</td>
<td>$9.8554 \times 10^{-4}$</td>
<td>$1.5362 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$disc$</td>
<td>$8.4260 \times 10^{-8}$</td>
<td>$1.0000 \times 10^{-3}$</td>
<td>$1.5556 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$gr$</td>
<td>$1.7047 \times 10^{-7}$</td>
<td>$1.0235 \times 10^{-3}$</td>
<td>$1.6373 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$lc$</td>
<td>$4.5414 \times 10^{-9}$</td>
<td>$9.8524 \times 10^{-4}$</td>
<td>$1.6028 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$qo$</td>
<td>$1.2586 \times 10^{-8}$</td>
<td>$9.8450 \times 10^{-4}$</td>
<td>$6.6072 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>L-rib</td>
<td>$6.3232 \times 10^{-9}$</td>
<td>$9.8534 \times 10^{-4}$</td>
<td>$1.5669 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>L-cur</td>
<td>$5.8220 \times 10^{-9}$</td>
<td>$9.8531 \times 10^{-4}$</td>
<td>$1.5749 \times 10^{-1}$</td>
</tr>
</tbody>
</table>
Solutions for the Baart problem, $m = n = 100$, $noise = 10^{-3}$, solid=unperturbed solution, dashed=$vm$, dot-dashed=$gm$-opt
Phillips problem, $n = 200$

<table>
<thead>
<tr>
<th>noise</th>
<th>meth</th>
<th>$\mu$</th>
<th>$|c - Ax|$</th>
<th>$|x - x_0|$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-5}$</td>
<td>$\mu$ opt</td>
<td>$1.3725 \times 10^{-7}$</td>
<td>$2.9505 \times 10^{-14}$</td>
<td>$1.6641 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>vm</td>
<td>$4.5432 \times 10^{-9}$</td>
<td>$9.0825 \times 10^{-6}$</td>
<td>$2.2620 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>gm-opt</td>
<td>$5.3835 \times 10^{-9}$</td>
<td>$9.1722 \times 10^{-6}$</td>
<td>$2.1174 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>gcv</td>
<td>$3.1203 \times 10^{-9}$</td>
<td>$8.9283 \times 10^{-6}$</td>
<td>$2.6499 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>disc</td>
<td>$1.2107 \times 10^{-8}$</td>
<td>$1.0000 \times 10^{-5}$</td>
<td>$1.6873 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>gr</td>
<td>$4.1876 \times 10^{-8}$</td>
<td>$1.5784 \times 10^{-5}$</td>
<td>$1.9344 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>lc</td>
<td>$3.6731 \times 10^{-14}$</td>
<td>$2.4301 \times 10^{-6}$</td>
<td>$7.9811 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>qo</td>
<td>$1.5710 \times 10^{-8}$</td>
<td>$1.0542 \times 10^{-5}$</td>
<td>$1.6463 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>L-rib</td>
<td>$2.6269 \times 10^{-14}$</td>
<td>$2.2118 \times 10^{-6}$</td>
<td>$8.9457 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>L-cur</td>
<td>$4.7952 \times 10^{-14}$</td>
<td>$2.6093 \times 10^{-6}$</td>
<td>$7.2750 \times 10^{-1}$</td>
</tr>
</tbody>
</table>


G.H. Golub and U. von Matt, Generalized cross-validation for large scale problems, in Recent advances in total least squares techniques and errors in variable modeling, S. van Huffel ed., SIAM, (1997), pp 139–148


