Numerical Linear Algebra, Homework III
Regularization and ill-posed problems

Thomas Mach, Karl Meerbergen, Raf Vandebril, Marc Van Barel

November 4, 2015

1 Theoretical background

The concept of ill-posed problems goes back to Hadamard in the beginning of the 20th century. Hadamard essentially defined a problem to be *ill-posed* if the solution is not unique or if it is not a continuous function of the data, i.e., if small perturbation of the data can cause large perturbation of the solution.

The classical example of an ill-posed problem is a Fredholm integral equation of the first kind with a square integrable kernel

$$\int_a^b K(s,t)f(t)dt = g(s), \quad c \leq s \leq d, \tag{1}$$

where the right-hand side g and the kernel K are given, and where f is an unknown solution.

Certain finite-dimensional discrete problems have properties very similar to those of ill-posed problems, such as being highly sensitive to high frequency perturbations. We can be more precise with this characterization for linear systems of equations

$$Ax = b, \quad A \in \mathbb{R}^{n\times n}. \tag{2}$$

We say that this is a discrete ill-posed problem if both of the following criteria are satisfied:

1. the singular values of A decay gradually to zero
2. the ratio between the largest and the smallest nonzero singular values is large.

Criterion 2 implies that the matrix A is ill-conditioned, i.e., that the solution is potentially very sensitive to perturbations; criterion 1 implies that there is no "nearby" problem with a well-conditioned coefficient matrix and with well-determined numerical rank.

An important aspect of discrete ill-posed problems is that the ill-conditioning of the problem does not mean that a meaningful approximate solution cannot be computed. Rather, the ill-conditioning implies that standard methods in numerical linear algebra for solving $Ax = b$, such as LU, Cholesky or QR factorization, cannot be used in a straightforward manner to compute such a solution. Instead, more sophisticated methods must be applied in order to ensure computation of a meaningful solution. This is the essential goal of regularization methods.

The primary difficulty with the discrete ill-posed problem (2) is that they are essentially under determined due to the cluster of small singular values of A. Hence, it is necessary to

incorporate further information about the desired solution in order to stabilize the problem and to single out a useful and stable solution. This is the purpose of *regularization*.

The dominating approach to regularization of discrete ill-posed problems is to require that the 2-norm of the solution be small. An initial estimate x^* of the solution may also be included in the side constraint. Hence, the side constraint involves minimization of the quantity

$$\Omega(x) = \|L(x - x^*)\|_2. \tag{3}$$

Here the matrix L is typically either the identity matrix I_n or a $p \times n$ discrete approximation of the $(n-p)$-th derivative operator, in which case L is a banded matrix with full row rank.

When the side constraint $\Omega(x)$ is introduced, one must give up the requirement $Ax = b$ in the linear system (2) and instead seek a solution that provides a fair balance between minimizing $\Omega(x)$ and minimizing the residual norm $\|Ax - b\|_2$. The underlying idea is that a regularized solution with small norm and a suitably small residual norm is not too far from the desired, unknown solution to the unperturbed problem underlying the given problem.

1.1 Tikhonov regularization

The idea of Tikhonov regularaization is to define the regularized solution x_λ as the minimizer of the following weighted combination of the residual norm and the side constraint

$$x_\lambda = \operatorname{argmin}\{\|Ax - b\|_2^2 + \lambda^2 \|L(x - x^*)\|_2^2\},$$

where the *regularization parameter* λ controls the weight given to minimization of the side constraint relative to minimization of the residual norm. A large λ (strong regularization) favors a small solution norm at the cost of large residual norm, while a small λ has the opposite effect. With λ one also controls the sensitivity of the regularized solution x_λ to perturbations in A and b, and the perturbation bound is proportional to λ^{-1}. Numerical methods for actually computing λ will be discussed later.

1.2 SVD and related regularization

Let $A \in \mathbb{R}^{m \times n}$. Then the SVD of A is a decomposition of the form

$$A = U \Sigma V^T = \sum_{i=1}^{n} u_i \sigma_i v_i^T, \tag{4}$$

where $U = [u_1, \ldots, u_n]$ and $V = [v_1, \ldots, v_n]$ are matrices with orthonormal columns and $\Sigma = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\}$ has non-negative diagonal elements appearing in non-increasing order.

For matrices A being discretizations of some operators with good smoothness we can mention one of the characteristic features of their SVD. It is that the left and right singular vectors u_i and v_i tend to have more sign changes in their elements as σ_i decreases.

To see how the SVD gives insight into the ill-conditioning of A, consider the following relations which follow directly from (4):

$$\left.\begin{array}{rl} Av_i & = \sigma_i u_i \\ \|Av_i\|_2 & = \sigma_i \end{array}\right\} \quad i = 1, \ldots, n.$$

We see that a small singular value σ_i compared to $\|A\|_2 = \sigma_1$ means that there exists a certain linear combination of the columns of A, characterized by the elements of the right singular

vector v_i, such that $\|Av_i\|_2$ is small. In other words, one or more small σ_i implies that A is nearly rank-deficient, and the vectors v_i, associated with the small σ_i, are numerical null-vectors of A.

The characteristic feature of the SVD promises that as σ_i decreases, the singular vectors u_i and v_i become more and more oscillatory. Consider now the mapping Ax of an arbitrary vector x. Using the SVD we get $x = \sum_{i=1}^{n}(v_i^T x)v_i$ and

$$Ax = \sum_{i=1}^{n} \sigma_i (v_i^T x) u_i.$$

This clearly shows the smoothing effect of A: due to multiplication with σ_i the high-frequency components of x are more damped in Ax than low-frequency components. Moreover, the inverse problem of computing x from $Ax = b$ must have the opposite effect: it amplifies the high-frequency oscillations in the right-hand side b.

2 Discrete Picard condition and filter factors

Suppose A was constructed by discretization of an integral equation (1). As we have seen in the previous section, the multiplication by A has a smoothing effect on x. This corresponds to the smoothing effect of an integration of $f(t)$ with a square integrable kernel $K(t, s)$ in the continuous case. Solving an integral equation leads to amplifying oscillations in the right-hand side $g(s)$. Thus to get a smooth solution $f(t)$ the input function $g(s)$ should be smooth enough to survive the inversion to f. The mathematical formulation of this smoothness criterion is called the Picard condition.

For discrete ill-posed problems one can formulate the Picard condition as follows. In a real-world application the right-hand side b is always contaminated by various types of errors. Hence we can write it as

$$b = \hat{b} + e,$$

where e are the errors and $\hat{b}$ is the unperturbed right-hand side. Both $\hat{b}$ and the corresponding unperturbed solution $\hat{x}$ represent the underlying unperturbed (and unknown) problem. Now, if we want to be able to compute a regularized solution x_{REG} from the given b such that x_{REG} approximates the exact solution $\hat{x}$, then it is shown that the corresponding exact right-hand side $\hat{b}$ must satisfy the following criterion: *the Fourier coefficients $|u_i^T \hat{b}|$ on the average decay to zero faster than the singular values σ_i.*

Assume now for simplicity that A has no exact zero singular values. Using the SVD, it is easy to show that the least-squares solution of the linear system $Ax = b$ is given by the equation

$$x_{LSQ} = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i. \tag{5}$$

This relation illustrates the difficulties with the standard solution to $Ax = b$. Since the Fourier coefficients $|u_i^T b|$ corresponding to the smaller singular values σ_i do not decay as fast as singular values, the solution x_{LSQ} is dominated by terms in the sum corresponding to the smallest σ_i. As a consequence, the solution x_{LSQ} has many sign changes and thus appears completely random.

With this analysis in mind, we can see that the purpose of a regularization method is to dampen or to filter out the contributions to the solution corresponding to the small singular values. Hence, we will require that a regularization method produces a regularized solution x_{REG}

that, for $x^* = 0$ and $L = I_n$, can be written as follows

$$x_{REG} = \sum_{i=1}^{n} f_i \frac{u_i^T b}{\sigma_i} v_i. \tag{6}$$

Here the numbers f_i are *filter factors* for the particular regularization method.

For Tikhonov regularization the filter factors are $f_i = \sigma_i^2/(\sigma_i^2 + \lambda^2)$, and the filtering takes place for $\sigma_i < \lambda$. In particular, this shows that discrete ill-posed problems are essentially unregularized by Tikhonov's method for $\lambda \ll \sigma_n$.

2.1 The L-curve

Perhaps the most convenient graphical tool for analysis of discrete ill-posed problems is a plot –for all valid regularization parameters– of the norm $\|Lx_{REG}\|_2$ of the regularized solution versus the corresponding residual norm $\|Ax_{REG} - b\|_2$. In this way, the L-curve clearly displays the compromise between minimization of these two quantities, which is the heart of any regularization method. The use of such plots in connection with ill-conditioned problems goes back to Miller and Lawson & Hanson.

For discrete ill-posed problems it turns out that the L-curve, when plotted in log-log scale, almost always has a characteristic L-shaped appearance (hence its name) with a distinct corner separating the vertical and horizontal parts of the curve. To see why it is so, we notice that if $\hat{x}$ denotes the exact, unregularized solution corresponding to the exact right-hand side $\hat{b}$, then the error $x_{REG} - \hat{x}$ in the regularized solution consists of two components, namely, a perturbation error from the error e in the given right-hand side b, and a regularization error due to regularization of the error-free component $\hat{b}$ in the right-hand side.

We can substantiate this by means of the relations for the regularized solution x_{REG} in terms of filter factors. Let $L = I_n$. Equation (6) yields the following expression for the error in x_{REG}:

$$x_{REG} - \hat{x} = \sum_{i=1}^{n} f_i \frac{u_i^T e}{\sigma_i} v_i + \sum_{i=1}^{n} (f_i - 1) \frac{u_i^T \hat{b}}{\sigma_i} v_i. \tag{7}$$

Here, the first term is the *perturbation error* due to the perturbation e, and the second term is the *regularization error* caused by regularization of the unperturbed solution $\hat{b}$ of the right-hand side. When only little regularization is introduced, most of the filter factors f_i are approximately one and the error $x_{REG} - \hat{x}$ is dominated by the perturbation error. On the other hand, with plenty of regularization most filter factors are small, $f_i \ll 1$, and the regularization error dominates.

For a given fixed right-hand side $b = \hat{b} + e$, there is obviously an optimal regularization parameter that balances the two types of errors defined above. An essential feature of the L-curve is that this optimal regularization parameter (defined in the above sense) is not far from the regularization parameter that corresponds to the L-curve corner.

3 Regularization methods

In this section we briefly review the regularization methods that you have to implement. The first two methods are direct, the last one is iterative. We choose $x^* = 0$ and $L = I_n$ in the definition (3) of the side constraint.

3.1 Tikhonov regularization

The regularized solution x_λ is the solution to the following least squares problem

$$\min \left\| \begin{bmatrix} A \\ \lambda I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2, \tag{8}$$

The most efficient algorithm for numerical treatment of Tikhonov's method is to transform the matrix A into a bidiagonal matrix B by means of left and right orthogonal transformations,

$$A = UBV^T,$$

and finally solve the resulting sparse problem with a banded B for $V^T x_\lambda$. Another way to implement Tikhonov's method is to use the SVD of A and filter factors (6) directly.

3.2 TSVD

A fundamental observation regarding the above mentioned methods is that they circumvent the ill-conditioning of A by introducing a new problem with a well-conditioned coefficient matrix $\begin{bmatrix} A \\ \lambda I \end{bmatrix}$ with full rank. A different way to treat the ill-conditioning of A is to derive a new problem with a well-conditioned *rank deficient* coefficient matrix. A fundamental result about rank deficient matrices is the Eckart-Young theorem. It states that the closest rank-k approximation A_k to A (measured in 2-norm) is obtained by truncating the SVD expansion at k, i.e., A_k is given by

$$A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T, \quad k \leq n.$$

The truncated SVD (TSVD) regularization method is based on this observation in that one solves the problem

$$\min \|x\|_2 \quad \text{subject to} \quad \min \|A_k x - b\|_2.$$

The solution to this problem is given by

$$x_k = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} v_i.$$

Let us note that the TSVD solution x_k is the only solution that has no component in the numerical null-space of A, spanned by columns of V with numbers from $k+1$ to n.

Instead of using filter factors 0 and 1 as in TSVD, one can introduce a smoother cut-off by means of filter factors f_i defined as

$$f_i = \frac{\sigma_i}{\sigma_i + \lambda},$$

thus getting damped SVD (DSVD). The new filter factors decay slower than Tikhonov filter factors and thus introduce less filtering.
Note: You have to avoid multiple computations of the SVD in your programs!

3.3 Conjugate gradients

The conjugate gradient algorithm is a well-known method for solving sparse systems of equations with a symmetric positive definite coefficient matrix. In connection with ill-posed problems, let us apply the CG algorithm to the unregularized normal equations $A^TAx = A^Tb$ (implemented such that A^TA is not formed). CG quickly solves the problems in the components associated with the large singular values (cf. multigrid). The small singular values are not solved as quickly. This is also related to the fact that the eigenvalues of A^TA are the squares of the singular values of A which even enhances this effect. In our applications, the components with small singular values are associated with highly oscillating singular vectors, so the CG method smooths the solution. The number of iterations plays the role of the regularization parameter.

Suppose that A has a cluster of large singular values (or that the singular values decay gradually to zero but (a) there is a good separation between large singular values and (b) the discrete Picard condition is satisfied for the unperturbed component of right-hand side). Then the same holds for the eigenvalues of A^TA. To explain the regularizing effect of the CG recall that the Ritz values, corresponding to largest eigenvalues of such A^TA, converge faster. One can show (by means of the Ritz polynomial) that filter factors, corresponding to almost converged eigenvalues, are close to one and filter factors for other eigenvalues decay as σ_i^2 for $\sigma_i < \theta_k$ (θ_k is the square root of the smallest converged eigenvalue).

CG minimizes the error in the A^TA norm, so the largest singular values are filtered out of the solution quicker, the small singular values need more time or remain in the error, and this is what we want. This smoothing effect is also used to use CG as a smoother in multigrid. To put the CG method in a common framework we may notice that the solution x_k after k CG steps can be defined as

$$\min \|Ax - b\|_2 \quad \text{subject to} \quad x \in \mathcal{K}_k(A^TA, A^Tb), \tag{9}$$

where $\mathcal{K}_k(A^TA, A^Tb)$ is a Krylov space associated with the normal equations. However, even the best implementation of the normal-equation CG algorithm suffers from loss of accuracy due to implicit use of the cross-product matrix A^TA.

You should be extremely careful with CG since the convergence can be delayed due to round-off errors in finite precision arithmetic.

4 Methods for choosing the regularization parameter

You have to implement two methods for choosing the regularization parameter.

4.1 L-curve

The L-curve criterion is already known to you from the previous sections. (For experienced programmers:) For a continuous regularization parameter λ one may compute the curvature of the curve $(\log \|Ax_\lambda - b\|_2, \log \|x_\lambda\|_2)$ (with λ as a parameter) and seek the point with maximum curvature, which is defined as L-curve corner. If the regularization parameter is discrete, then you have to use a discrete analogue of the curvature. (For instance: use a discrete difference quotient for computing the curvature **or** approximate the discrete L-curve by splines, find the corner on the spline and finally the point on the discrete L-curve closest to the one on the spline).

If the last procedure seems too hard to implement, you may use some kind of user-interactive procedure (plot the L-curve, let the user choose the optimal λ by `ginput`). However in this case we prefer a separate procedure for exploring the L-curve.

4.2 Generalized cross-validation

GCV is based on the philosophy that if an arbitrary element b_i of the right-hand side b is left out, then the corresponding regularized solution should predict this observation well, and the choice of regularization parameter should be independent of an orthogonal transformation of b. This leads to choosing the regularization parameter, which minimizes the GCV function

$$G = \frac{\|Ax_{REG} - b\|_2^2}{(\text{trace}(I_m - AA^I))^2},$$

where A^I is a matrix which produces the regularized solution x_{REG} when multiplied with b, i.e. $x_{REG} = A^I b$. Note that G is defined for both continuous and discrete regularization parameters. You can use the Lanczos bidiagonalization algorithm to compute the denominator in $O(n)$ operations.

The filter factors could be used to evaluate the denominator by means of the (very simple) expression of type trace $= F(m, n) - \Sigma f_i$. F is some (possibly linear) combination of m and n. You may get an exact form of this expression and use it in your program.

You can use the `fminbnd` procedure for numerical minimization of the function G.

5 Tasks

Download from the website of the course the mat-file (`reg.mat`) with a pregenerated matrices A and perturbed right-hand sides b. The perturbation of b is of the form $b = b_{EXACT} + \alpha \cdot$ randn(size(b)). The exact right hand side as well as α is not known to you.

Load your file in `Matlab`. Apply different methods to solve the (ill-posed) problem $Ax = b$. What are the possibilities to compare the results coming from different methods? For each matrix choose the method that gives the best result. Try to predict some properties of the solution. Should it be very smooth? very oscillating?

For every problem plot the L-curve and find the position of the 'optimal' parameter on the curve. Plot also the singular values and the discrete Picard condition. (If you are experienced in programming, you may try to write the procedure for computing the optimal value on the L-curve *without* looking at the curve. This is not easy.)

Show how trace$(I_m - AA^I)$ can be computed easy and cheap (for filter factors f_i, TSVD, and CG).

Write your results in a structured way on paper or in a file. Send all the programs to `thomas.mach@cs.kuleuven.be`.

DEADLINE: Friday, 20 November 2015