Per Christian Hansen, DTU Compute

Exercises – Intro to Discrete Inverse Problems

Preparation

Download the *Regularization Tools* MATLAB software from

http://www2.imm.dtu.dk/~pch/Regutools ,

unpack the zip file, and place all the functions in a directory. Start MATLAB and go to the directory. Have fun with the exercises!

1. SVD Analysis of the Gravity Surveying Problem

The purpose of this exercise is to illustrate how a simple inverse problem is discretized by means of the midpoint quadrature rule. The exercise also illustrates that the coefficient matrix for the resulting system of linear equations is very ill conditioned, and that the solution is highly sensitive to errors.

We use the geophysical model problem from the slides. For a problem with n data points and also n unknowns, and using $s_i = (i-\frac{1}{2})/n$ and $t_j = (j-\frac{1}{2})/n$ for the collocation and quadrature points, derive the following formula for the matrix elements

$$a_{ij} = \frac{d}{n}\left(d^2 + ((i-j)/n))^2\right)^{-3/2}, \qquad i,j = 1,\ldots,n,$$

and explain why the matrix A is symmetric.

This is implemented in the function `gravity` in *Regularization Tools*. Use this function (e.g., for $n = 32$) to generate the matrix A, the right-hand side $b = A\,x$ and the exact solution x for two different choices – one corresponding to a smooth solution (constant and linear functions are not allowed), and one corresponding to a solution with a jump discontinuity. You can use examples 1 and 3 in `gravity`. Notice that the right-hand side is always smooth. Study experimentally how the right-hand side varies with the depth d.

Now we consider the condition number of A (MATLAB: `cond(A)`). Keep n fixed and study how the condition number varies with d. Can you intuitively explain the observed behavior.

Try to solve the problem by computing the "naive" solution $x = A^{-1}b$ (MATLAB: `x = A\b`) for $n = 32$ and $d = 0.25$. First solve the problem with a noise-free right-hand side; then try again with perturbed right-hand sides $b+e$ in which a very small amount of Gaussian white noise e has been added. How large can you make the noise (as measured by $\|e\|_2$) before the inverted noise starts to dominate the computed solution? Start with $\|e\|_2 = 10^{-10}$.

2. TSVD Solutions

Use the function `shaw` from *Regularization Tools* to generate the test problem for $n = 60$, and add Gaussian white noise e computed as `0.001*randn(n,1)` – corresponding to the standard deviation $\eta = 0.001$. Then use the calls

```
[U,s,V] = csvd(A); [X,resnrm,solnrm] = tsvd(U,s,V,b,1:15);
```

to compute the TSVD solutions for truncation parameters $k = 1, 2, \ldots, 15$, along with the residual and solution norms. The function `csvd` from *Regularization Tools* computes a "thin SVD" instead of the "full SVD" computed by MATLAB's `svd` function, and it stores the singular values in a vector. The function `tsvd` needs this form of the SVD as input.

Inspect the TSVD solutions x_k, stored as the columns of the matrix `X`; what value of k gives the best approximate solution? What is the norm of the corresponding residual, and can you relate this norm to the norm of the error vector e? Why are the norms of x_k and x^{exact} almost identical for the optimal k? Hint: think about the behavior of the SVD components.

3. Tikhonov Solutions via SVD

The purpose of this exercise is to illustrate Tikhonov regularization of the second derivative test problem. Use the call `[A,b,x] = deriv2(n,3)` function to generate the test problem with $n = 32$. Then use `[U,s,V] = csvd(A)` to compute the SVD of A, and inspect the singular values.

Add a bit of noise to the right-hand side, e.g., `1e-3*randn(size(b))` corresponding to noise with standard deviation $\eta = 0.001$. This noise is certainly not visible when plotting the right-hand side vector, but very significant with respect to the naive solution.

Now use MATLAB's `logspace` function to generate a number of different regularization parameters λ in the range 10^{-3} to 1, compute the corresponding filter factors $\varphi_i^{[\lambda]}$ by means of `fil_fac`, as well as the corresponding Tikhonov solution x_λ by means of

```
[X,res_norm,sol_norm] = tikhonov(U,s,V,b,lambda)
```

For each λ, plot both the filter factors and the solution, and comment on your results.

4. From Over-Smoothing to Under-Smoothing

The purpose of this exercise is to illustrate how the Tikhonov solution x_λ, its norm $\|x_\lambda\|_2$, and the residual norm $\|A\,x_\lambda - b\|_2$ change as λ goes from large values (over-smoothing) to small values (under-smoothing). Use the `shaw` test problem from *Regularization Tools* with $n = 32$, and add Gaussian white noise with standard deviation $\eta = 10^{-3}$ to the right-hand side.

Use `lambda = logspace(1,-5,20)` to generate 20 logarithmically distributed values of λ from 10^{-5} to 10. Then use `csvd` to compute the SVD of A and use

```
[X,res_norm,sol_norm] = tikhonov(U,s,V,b,lambda)
```

to compute the 20 corresponding Tikhnonov solutions x_λ, stored as columns of the matrix `X`. Inspect the columns of the matrix `X` (e.g., by means of `mesh` or `surf`) in order to study the progression of the regularized solution x_λ as λ varies from over-smoothing to under-smoothing.

Use `loglog` to plot the L-curve, i.e., the solution norm $\|x_\lambda\|_2$ versus the residual norm $\|A\,x_\lambda - b\|_2$, for the chosen values of λ. Explain how the behavior of the L-curve is related to the behavior of the regularized solutions.

For the given λ-values in `lambda` (or perhaps more values in the same interval) compute the error norm $\|x^{\text{exact}} - x_\lambda\|_2$ and plot it versus λ. Determine the optimum value of λ – the one that leads to the smallest error norm – and locate the position of the corresponding solution on the L-curve. Is it near the "corner"?

5. The GCV and L-curve Parameter-Choice Methods

This exercise illustrates the use of the GCV and L-curve methods for choosing the regularization parameter, and we compare these methods experimentally. As part of this comparison we investigate how robust – or reliable – the methods are, i.e., how often they produce a regularization parameter close to the optimal one. We use the `shaw` test problem with $n = 64$ and the parameter-choice functions `gcv` and `l_curve` from *Regularization Tools*.

Plot the GCV function for, say, 10 different perturbations with the same $\eta = 0.01$, and note the general behavior of the GCV function. Is the minimum always at the transition region between the flat part and the more vertical part?

Use the L-curve criterion to compute the regularization parameter, for the same perturbations as above. Does the regularization parameter computed by means of `l_curve` always correspond to a solution near the corner?

If you are not completely out of energy at this time, try the experiments again, this time with the discrepancy principle; use the function `discrep` from *Regularization Tools*.