0.1 Description of SolveInverseProblemWithTikhonovSVD

The module **SolveInverseProblemWithTikhonovSVD** applies the Tikhonov regularization to an existing forward model. The Tikhonov regularization minimizes the following functional:

$$x_0 \quad = \quad \text{argmin } \|Ax - b\|_2^2 + \lambda^2 \|Lx\|_2^2 \tag{1}$$

$$\tag{2}$$

where A is the transfer matrix, and b is the vector of observed data. λ is the regularization parameter.

SolveInverseProblemWithTikhonovSVD achieves the same Tikhonov regularization as another module, SolveInverseProblemWithTikhonv, but in a different way. Here, the input A and L are replaced by four matrices U, S, V, X, which are the generalized singular value decomposition of the matrix pair (A, L):

$$A = U\ (diag(s_1))\ X^{-1},\ \ L = V\ (diag(s_2))\ X^{-1} \tag{3}$$

Here S contains the generalized singular values, and has two columns: $S = [s_1, s_2]$.

SolveInverseProblemWithTikhonovSVD has five matrix inputs and three matrix outputs. The five inputs (from left to right) are U, S, V, X, b. The three outputs are 1) "InverseSoln", which gives the solution vector x_0; 2)"RegParam", which gives the regularization parameter λ; and 3) "RegInverseMat", which gives a pseudo-invese of the matrix A.

To use this module, users are responsible to carry out the general SVD for the matrix pair (A, L).

The UI of the **SolveInverseProblemWithTikhonovSVD** module provides three methods to choose λ: 1) directly typing a single value; 2) choosing a value by moving the slider (the range of the slider and the increments are pre-defined inside the code); and 3) determine the value by means of the L-curve method. The range of regularization parameters used for the slider and to obtain the L-curve is user defined.

The network "**TikhonovSVD_ExampleNetwork.srn**" gives an example of how to use this module. The input data for this network is given by "TestData_for_TikhonovSVD.mat". Most variables in the matlab file is self-explanatory. Some variables need explanation. x_0 is a "ground truth" solution, and $b = Ax_0$ is the exact result at observation points. RHS is obtained by perturbing b by a small amount, simulating the real situation that observed data are contaminated by some noise.