Homework #3

Simulation and solution of a linear inverse problem.

1. Create a script that uses the `np.random.normal()` method to create a column vector d of `N=1000` realizations of a normally-distributed random variable with mean `d_bar=4` and standard deviation `sigma=2`. Plot the data and count the number of instances where d>2 sigma_d. Is this about the number you expected?

2. Assume model parameters $\boldsymbol{m}^{true}$. Based on these model prameters generate synthetic data $\boldsymbol{d}^{true}$, invert the data to obtain a model estimation, $\boldsymbol{m}^{est}$. Does the estimated model (the solution) agree with the original model parameters? Add noise of variable color and strength to the data and discuss how much $\boldsymbol{m}^{est}$ differs from $\boldsymbol{m}^{true}$. Plot the synthetic data without and with noise and the predicted data for the estimated model.

   The model we consider is a third order polynomial (cubic equation) given by $y=m_1+xm_2+x^2m_3+x^3m_4$. We use `m_true=[1, -1, 0.4, -0.03]`. Use a data vector with length of N=100.

   a.) Create Gaussian noise with a standard deviation of `sigma=2` (also try other standard deviations). Plot (i) the synthetic data, (ii) the synthetic data with noise added including predicted data based on estimated model, (iii) the data resolution matrix, (iv) the model resolution matrices.

   b.) Also invert the synthetic data using linear and quadratic models. Use `sigma=2` but also try other standard deviations (e.g. sigma=0.1). Which is your preferred model for these two sigma? Pretend you don't know that the data were created using a cubic model (N.B.: Using an f-test we can determine whether the data warrant the model with higher complexity, see f).

   c.) Use only the first 4 elements of the data vector. Estimate the model and create the same plots as for a (including the data and model resolution matrices). How did the characteristics of the inverse problem change?

   d.) Now consider a model with 6 model parameters $y=m_1+xm_2+xm_3+xm_4+x^2m_5+x^3m_6$. Create the design matrix and estimate the model parameters. What type of inverse problem is this?

   e.) Same as a), but consider non-Gaussian noise. It can be obtained by adding a constant offset to part of the data and by sampling from a double exponential distribution. Estimate the model parameters using the cubic model. What is the pitfall of this approach? (voluntarily: also consider lognormal and power law distributions)

   f.) (Voluntarily except for InSAR students. Repeat 3.2b using Gaussian noise of different strength as described by different standard deviations. Use the F-test to determine whether the data can be explained using a linear model or require the more complex quadratic and cubic models which have more model parameters.)

3. For your (linear or non-linear) inverse problem of interest, create a Jupyter notebook that handles the forward problem, i.e. generates synthetic data based on a priori-selected model. We will use this function in the methodology to be developed for non-linear inverse problems later in the class. Include a function to plot the data in the way you are used.

InSAR students or volunteers:

4. Make yourself familiar with uncorrelated and correlated noise in time series. Consider Gaussian noise, lognormal noise, red noise and power law noise. What are the probability distributions and covariance matrices used to generate this noise (see solutions Juptyer notebook)

[3. for Giacomo: Create a Jupyter notebook for magma source modelling. First create a python script to generate synthetic data in MintPy HDF5 format and GPS data based on a magma source model (penny-shaped crack magma source, but also allow for Yang source (ellipsoid), McTigue and Mogi source. Add noise (Gaussian and power-law noise). Sample from the data (homogeneous sampling and quadtree). Invert the data using the simulated annealing and neighborhood algorithms (we will discuss them later in the class). For a penny-shaped crack model as input, for how much added noise you can't distinguish anymore between the penny-shaped crack and the Mogi model? Create the synthetic data using the geometryRadar.h5 and start with Campi Flegreii and Sierra Negra. Make sure you can display using MintPy functions. Once it works replace synthetic data using real data].

[3. for Emirhan: Create a Jupyter notebook to estimate the dem error of persistent scatterer InSAR data. Use a typical perpendicular baseline history from Miami. Consider pixels at 20, 50, 100, 150 and 200 meters elevation and create a synthetic phase history due to the topography. Retrieve the elevation through linear inversion. Add noise to the phase history and investigate how the estimated elevation depends on the magnitude of the added noise. Next add decaying displacements to the phase history given by a linear and a quadratic model. The model vector now has 2 and 3 parameters, respectively. Given the variance in the perpendicular baseline, we want to understand how a long a time series is required to robustly estimate the elevation.

Next we consider the non-linear problem using the wrapped phases. This should eliminate any bias form phase-unwrapping errors. Wrap and then plot the phase histories onbtained above and plot them.

Also retrieve the demError estimation routines from MintPy. There are options for estimating using the phase and phase velocity for estimating the temporal model. Also extract the relevant code from SARvey].