README: GEOSPIN3D MATLAB PACKAGE

In this document, we provide a detailed description of the GeosPIn3D MATLAB package for petrophysical inversion.

The GeosPIn3D package includes functions for the geophysical forward model, the geostatistical simulations and the petrophysical inversion.

The geophysical forward model functions include:

* *AkiRichardsCoefficientsMatrix.m*: It computes the Aki Richards coefficients matrix for the seismic forward model;
* *DifferentialMatrix.m:* It computes the differential matrix for the seismic forward model;
* *ForwardGeopModel1D.m* : It computes the seismic response in 1D using the seismic convolutional model;
* *SeismicModel1D.m*: It computes the seismic response in 1D from velocity using the seismic convolutional model;
* *WaveletMatrix.m*: It computes the wavelet matrix for the seismic forward model.

The geostatistical simulation functions include:

* *CorrelationFunction3D.m*: It computes the spatial correlation function in 3D;
* *ProbFieldSimulation3D.m*: It computes geostatistical realizations in 3D using the PFS method;
* *SpatialCovariance.m*: It computes the vertical correlation function in 1D.

The petrophysical inversion functions include:

* *BayesPetroInversion3D\_GMM.m*: It computes the petrophysical properties according to the Bayesian inversion method with Gaussian mixture PDFs;
* *BayesPetroInversion3D\_KDE.m*: It computes the petrophysical properties according to the Bayesian inversion method with non-parametric PDFs;
* *GeosPetroInversion3D.m*: It computes the petrophysical properties according to the geostatistical inversion based on the ES-MDA method.
* *RockPhysicsGMM.m*: It computes the conditional distribution of petrophysical properties given elastic properties using a Gaussian mixture PDF;
* *RockPhysicsKDE.m*: It computes the conditional distribution of petrophysical properties given elastic properties using a non-parametric PDF;
* *SeismicInversion3D.m*: It computes the conditional distribution of elastic properties given seismic data using the Bayesian linearized AVO inversion approach.

Examples

The GeosPIn3D package includes three drivers to demonstrate how to apply the inversion codes:

* *DriverBayesPetroInversion3D\_GMM.*m: It demonstrates the Bayesian petrophysical inversion using a Gaussian mixture PDF for the rock physics component;
* *DriverBayesPetroInversion3D\_KDE.*m: It demonstrates the Bayesian petrophysical inversion using a non-parametric PDF, estimated using KDE, for the rock physics component;
* *DriverGeosPetroInversion3D.m:* It demonstrates the geostatistical inversion using ES-MDA with PFS geostatistical realizations.

The three scripts are tested on the same dataset, SeismicData3D.mat, that is stored in the subfolder Data3D. The dataset includes two real seismic lines, inline number 39 and crossline number 34, whereas the other traces in the 3D volume are synthetic.

We first define the input data and the seismic parameters as

%% Data

% load test dataset

load Data3D/SeismicData3D.mat

% selection of lines to display

interx = 39; intery = 34;

%% Initial parameters

% number of samples (petrophysical properties)

nm = size(near,3)+1;

% number of lines

nxl = size(near,1);

nil = size(near,2);

% number of variables

nv = 3;

% reflection angles

theta = [15, 30, 45];

ntheta = length(theta);

% time vector

TimeSeis = squeeze(Z(1,1,:));

% time sampling

dt = TimeSeis(2)-TimeSeis(1);

% error variance

varerr = 10^-3;

sigmaerr = varerr\*eye(ntheta\*(nm-1));

%% Wavelet

% wavelet parameters

freq = 45;

ntw = 64;

[wavelet, tw] = RickerWavelet(freq, dt, ntw);

Here, the MATLAB workspace SeismicData3D.mat contains 6 variables: three angles stacks, near, mid and far, and their coordinates X, Y, and Z (defined at each node of the seismic grid) where X represents the inline, Y the crossline and Z the two-way travel time. The input parameters include: nm is the number of samples of the unknown petrophysical model at each trace, nxl and nil represent the number of crosslines and inlines, nv is the number of variables, ntheta is the number of angles, TimeSeis is the seismic time vector, dt is the sampling interval, varerr is the variance of the error, sigmaerr is the error covariance diagonal matrix. The seismic wavelet is defined by a discretized Ricker wavelet with dominant frequency freq and number of samples ntw.

In both Bayesian petrophysical inversions, using Gaussian mixture or non-parametric PDFs, we define a prior model for the elastic properties, a rock physics likelihood function, and an error model.

%% Prior model

vppriormean = 4;

vspriormean = 2.4;

rhopriormean = 2.3;

Vpprior = vppriormean\*ones(nxl,nil,nm);

Vsprior = vspriormean\*ones(nxl,nil,nm);

Rhoprior = rhopriormean\*ones(nxl,nil,nm);

%% Spatial correlation matrix

corrlength = 5\*dt;

sigma0 = [ 0.0034 0.0037 0.0014

0.0037 0.0042 0.0012

0.0014 0.0012 0.0015];

sigmaprior = SpatialCovariance(corrlength, dt, nm, sigma0);

%% Rock physics parameters

% training dataset

load RockPhysicsTrain.mat

np = size(petrotrain,2);

elastrain = [VpTrain VsTrain RhoTrain];

nd = size(elastrain,2);

faciestrain = ones(size(PhiTrain));

faciestrain(PhiTrain>mean(PhiTrain(:)))=2;

nf = max(unique(faciestrain));

%% GMM

% grid petrophysical properties

ndiscr = 30;

phigrid = linspace(0.01, 0.4, ndiscr)';

claygrid = linspace(0, 0.8, ndiscr)';

swgrid = linspace(0, 1, ndiscr)';

% domain elastic properties

ndiscr = 25;

vpgrid = linspace(3.2, 4.6, ndiscr)';

vsgrid = linspace(2, 3, ndiscr)';

rhogrid = linspace(2, 2.6, ndiscr)';

% Error

rpsigmaerr = 10^-2\*eye(nd,nd);

In the proposed example scripts, we use a constant mean model for P- and S-wave velocity and density whereas the spatial covariance matrix is defined as the Kronecker product of the spatially invariant component and a 1D vertical correlation matrix associated with a spatial correlation function. The rock physics likelihood is estimated from a training dataset containing elastic and petrophysical properties generated using a statistical rock physics model. The Gaussian mixture implementation requires a facies vector for the training dataset. The non-parametric implementation requires the definition of the kernel bandwidth. Both algorithms require the discretization of the domains of the model properties (Table 3) for the calculation of the integral form of the posterior distribution. The inversion is then performed by running the code BayesPetroInversion3D\_GMM.m or BayesPetroInversion3D\_KDE.m.

In the geostatistical inversion the prior model is defined in the petrophysical domain, and it is represented by a set of geostatistical realizations:

%% Prior model

% prior mean

phipriormean = 0.2;

claypriormean = 0.23;

swpriormean = 0.6;

phiprior = phipriormean\*ones(nxl,nil,nm);

clayprior = claypriormean\*ones(nxl,nil,nm);

swprior = swpriormean\*ones(nxl,nil,nm);

%% Spatial correlation matrix

corrpetro = [ 1 -0.6 -0.5

-0.6 1 0.2

-0.5 0.2 1];

stdpetro = [0.035 0.055 0.09];

%% Rock physics parameters

% training dataset

load RockPhysicsTrain.mat

petrotrain = [PhiTrain ClayTrain SwTrain];

elastrain = [VpTrain VsTrain RhoTrain];

%% ESMDA inversion

nsim = 500;

niter = 4;

vertcorr = 20;

horcorr = 25;

For the geostatistical inversion, we also define the number of geostatistical realizations in the ensemble (nsim), the number of iterations of the ES-MDA (niter), and the lateral and vertical correlation ranges of the geostatistical simulations (horcorr and vertcorr). The inversion is then performed by running the code GeosPetroInversion3D.m.