**SeReMpy library**

**External libraries**

SeReMpy uses the following Python libraries:

* NumPy 1.19.2
* SciPy 1.6.2
* matplotlib 3.4.1

**Data**

The folder Data contains multiple synthetic datasets. All the data are synthetic and covered by the MIT license included with the code.

**Output**

The folder Output contains sample output files of the proposed scripts.

**Description of modules**

Functions included in the RockPhysics.py module

|  |  |
| --- | --- |
| *Function* | *Description* |
| DensityModel | linear porosity-density relation to compute density |
| MatrixFluidModel | Voigt-Reuss-Hill averages to compute the elastic moduli and density of the solid and fluid phases |
| GassmannModel | Gassmann's equations to compute the elastic moduli of the fluid-saturated rock |
| VelocityDefinitions | definitions of P-wave and S-wave velocity |
| LinearizedRockPhysicsModel | linear rock physics model based on a multilinear regression to compute P-wave and S-wave velocity and density |
| WyllieModel | Wyllie equation to compute P-wave velocity |
| RaymerModel | Raymer’s equation to compute P-wave velocity |
| SoftsandModel | Dvorkin's soft sand model to compute P-wave and S-wave velocity |
| StiffsandModel | Dvorkin's stiff sand model to compute P-wave and S-wave velocity |
| SphericalInclusionModel | inclusion model for spherical pores to compute P-wave and S-wave velocity |
| BerrymanInclusionModel | Berryman's inclusion model for prolate and oblate pores to compute P-wave and S-wave velocity |

Functions included in the Geostats.py module

|  |  |
| --- | --- |
| *Function* | *Description* |
| ExpCov | exponential spatial covariance model |
| GauCov | Gaussian spatial covariance model |
| SphCov | spherical spatial covariance model |
| SpatialCovariance1D | 1D spatial covariance function according to one of the available models (exponential, Gaussian, and spherical); |
| RadialCorrLength | radial correlation length for 2D spatial covariance functions |
| SpatialCovariance2D | 2D spatial covariance function according to one of the available models (exponential, Gaussian, and spherical); |
| SimpleKriging | simple kriging interpolation at a given location based on a set of measurements |
| OrdinaryKriging | ordinary kriging interpolation at a given location based on a set of measurements |
| IndicatorKriging | indicator kriging interpolation at a given location based on a set of measurements |
| GaussianSimulation | Gaussian simulation to generate a sample of a random variable at a given location based on a set of measurements |
| RandDisc | Simulation of a discrete random variable with given probability mass function |
| SeqGaussianSimulation | Sequential Gaussian Simulation method to generate spatially correlated realizations of a continuous random variable based on a set of measurements given a set of location coordinates |
| SeqIndicatorSimulation | Sequential Indicator Simulation method to generate spatially correlated realizations of a discrete random variable based on a set of measurements given a set of location coordinates |
| CorrelatedSimulation | sampling approach to simulate spatially correlated stochastic (1D) realizations of multiple random variables given a set of location coordinates |
| MarkovChainSimulation | sampling approach to simulate multiple 1-dimensional realizations of a discrete random variable based on a stationary first-order Markov chain |

Functions included in the Inversion.py module

|  |  |
| --- | --- |
| *Function* | *Description* |
| RickerWavelet | Ricker wavelet with given dominant frequency |
| AkiRichardsCoefficientsMatrix | Aki Richards coefficient matrix |
| DifferentialMatrix | differential matrix for discrete differentiation |
| WaveletMatrix | wavelet Toeplitz matrix for discrete convolution |
| SeismicModel | synthetic seismic data according to a linearized seismic model based on the convolution of a wavelet and the linearized approximation of Zoeppritz equations |
| SeismicInversion | posterior distribution of elastic properties according to the Bayesian linearized AVO inversion |
| RockPhysicsLinGaussInversion | posterior distribution of petrophysical properties conditioned on elastic properties assuming a Gaussian distribution and a linear rock physics model |
| RockPhysicsLinGaussMixInversion | posterior distribution of petrophysical properties conditioned on elastic properties assuming a Gaussian mixture distribution and a linear rock physics model |
| RockPhysicsGaussInversion | posterior distribution of petrophysical properties conditioned on elastic properties assuming a Gaussian distribution estimated from a training dataset |
| RockPhysicsGaussMixInversion | posterior distribution of petrophysical properties conditioned on elastic properties assuming a Gaussian mixture distribution estimated from a training dataset |
| RockPhysicsKDEInversion | posterior distribution of petrophysical properties conditioned on elastic properties assuming a non-parametric distribution estimated from a training dataset using kernel density estimation |
| EpanechnikovKernel | Epanechnikov kernel used in kernel density estimation |
| EnsembleSmootherMDA | updated model realizations of the model variables conditioned on seismic data using the ES-MDA method |
| LogitBounded | logit transformation for bounded properties |
| InvLogitBounded | inverse logit transformation for bounded properties |

Functions included in the Facies.py module

|  |  |
| --- | --- |
| *Function* | *Description* |
| BayesGaussFaciesClass | Bayesian facies classification assuming a multivariate Gaussian distribution of the continuous properties |
| BayesKDEFaciesClass | Bayesian facies classification assuming a multivariate non-parametric distribution of the continuous properties |
| ConfusionMatrix | classification confusion matrix |