PCGA parameter document ver 0.1

In this document, I list parameters used in PCGA ver 0.1. Because of on-going updates, the parameters listed below would change in PCGA ver 0.2. I note some expected changes in ver 0.2 for interface improvement and consistency.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | | | |
| Input parameters for initialization | | | | |
| forward\_model(s, parallelization, ncores) | (function) wrapper for forward model obs = f(s). See a template python file in each example for more information | | | |
| s\_init | (numpy.ndarray) m by 1 initial solution for Gauss-Newton method. In theory, the choice of s\_init does not affect the estimation while total number of iterations/number of forward model runs depend on s\_init | | | |
| pts | (numpy) m by dimension (e.g. m by 3 for 3D problem) spatial grid coordinates  Note: this input is required for inverse problem in unstructured grid and plotting. I think this is not required in the initialization and may be removed in the next version. | | | |
| params | (dictionary) auxiliary forward model parameters | | | |
| s\_true | (numpy.ndarray) [optional] m by 1 true solution for synthetic problems. Can be used for generating synthetic observations and measuring inversion performance by providing RMSE of estimates | | | |
| obs | (numpy.ndarray) [optional] n measurements for inversion. If obs is not given but s\_true or obs\_true is provided, it will generate synthetic noisy data for inversion to proceed | | | |
| obs\_true | (numpy.ndarray) [optional] true observation (without noise) for synthetic problem | | | |
| X | (numpy.ndarray) [optional] introduce user-defined drift/trend. If not given, it uses a constant unknown prior mean. See the parameter drift | | | |
| PCGA parameters | | | | |
| m | (int) number of unknows (grid points) | | | |
| n | (int) number of observations | | | |
| precision | Machine precision for finite difference (sqrt(precision) will be used for finite difference interval), default: 1e-8 | | | |
| parallel | (boolean) support parallelization | | | |
|  | n\_cores | (int) If parallel = True, use n\_cores for parallel executions. Default: number of maximum available cores in users’ machine | | |
| kernel | (function) prior covariance kernel | | | |
| prior\_std | (int) prior standard deviation | | | |
| prior\_cov\_scale | (numpy.float) prior covariance scale (correlation) parameters | | | |
| n\_pc | (int) number of principle components | | | |
|  |  | | | |
| matvec  ‘Dense’  ‘FFT’  ‘Hmatrix’, ‘FMM’ | (str) fast matrix vector multiplication method (‘Dense’, ‘FFT’, ‘Hmatrix’, ‘FMM’)  Note:  Currently supports dense matrix-vector multiplication for small-size problem and FFT for structured grid. For unstructured grid, one should decompose the covariance matrix externally and provide its eigenvalues and eigenvectors (priord and priorU) explicitly.  Hmatrix and FMM will be included to PCGA in ver0.2 | | | |
| Q | | (numpy.ndarray) covariance matrix for dense matrix operation | |
| N | | (int) dimension of structured grids | |
| min, xmax | | (int,int) min and max coordinates of structured grids in x, y, z | |
| pts | | (numpy.ndarray) spatial grid coordinates | |
| priord | | (numpy.ndarray) n\_pc eigenvalues of prior covariance | |
| priorU | | (numpy.ndarray) m by n\_pc eigenvectors of prior covariance | |
| drift | (str) drift/trend introduced in prior. (‘constant’, ‘linear’) default: constant  Note: X | | | |
| R | (float) measurement error level, scalar for data with the same error level and vector for heterogeneous data sets | | | |
| maxiter | (int) maximum iteration of Gauss-Newton method | | | |
| restol | (float) relative tolerance for stopping Gauss-Newton iteration | | | |
| post\_cov | (Boolean, str) calculation of posterior covariance.  True: currently same as ‘diag’. A full posterior covariance matrix characterization method will be included in the future version  ‘diag’: compute only diagonal component of posterior covariance, i.e., posterior variance.  False: do not compute posterior covariance | | | |
| direct | (boolean) whether user uses a direct matrix solver in inversion, default: False | | | |
| precond | (boolean) whether user uses a preconditioner for iterative matrix solver (minres and gmres) in inversion, default: False | | | |
| objeval | (boolean) if true, perform accurate and expensive objective function evaluation, default: False (for approximate objective function evaluation) | | | |
| LM | (boolean) whether user uses Levenberg Marquart method, default: False | | | |
| alphamax\_LM | | | (float) maximum multiplier/damping factor for LM method. PCGA uses a LM method by introducing a multiplier to the error matrix for dampening, default: 1000 |
| nopts\_LM | | | (int) number of LM steps (equally spaced damping factors assigned between 1 and alphamax\_LM) used in inversion, default: n\_cores |
| LM\_smin, LM\_smax | | | (float, NoneType) min and max value of allowed LM producing intermediate solutions. If Default: None, None |
| linesearch | (boolean) whether user uses linesearch method, default: False | | | |
| nopts\_LS | | | (int) number of linesearch points (equally spaced points between previous solution and current solution), default: n\_cores |
| JacSave | (boolean) save Jacobian-matrix products (HX, HQ, Hs) in the current iteration as explicit variables for user investigation, default: False | | | |
| iter\_save | (boolean) if True, it will produce text files (shat[#iter].txt, simulobs[#iter].txt) for intermediate solution and its simulate result in each iteration, default: false | | | |
| verbose | (boolean) if True, it will print out intermediate inversion information default: False | | | |
| forward\_model\_verbose | (boolean) if True, it will print out forward model messages, default: False | | | |
| Output | | | | |
| s\_hat | (numpy.ndarray) m by 1 best estimate | | | |
| simul\_obs | (numpy.ndarray) n by 1 reproduced observation from the best estimate | | | |
| post\_diagv | (numpy.ndarray) m by 1 posterior variance | | | |
| iter\_best | (int) iteration number in which the best estimate is obtained | | | |