— HPDDM —

https://github.com/hpddm/hpddm

All keywords must be prefixed by `-hpddm_`. If a value is specified in the column "Default", this value is used when the corresponding option is not set by the user. When no default value is specified but the corresponding option is set by the user, the option is true (represented internally by 1). If the option is not set, its value is false (represented internally by 0). Options highlighted in red should be reserved to expert users.

Keyword	Description	Possible values	Default
`help`	Display available options	Anything	
`version`	Display information about HPDDM	Anything	
`config_file`	Load options from a file saved on disk	String	
`tol`	Relative decrease in residual norm to reach in order to stop iterative methods	Numeric	10^{-6}
`max_it`	Maximum number of iterations of iterative methods	Integer	100
`verbosity`	Level of output (higher means more displayed information)	Integer	
`reuse_preconditioner`	Do not factorize again the local matrices when solving subsequent systems	Boolean	
`local_operators_not_spd`	Assume local operators are not positive definite	Boolean	
`orthogonalization`	Method used to orthogonalize a vector against an orthogonal basis	`cgs`, `mgs`	`cgs`
`dump_matri(ces|x_[[:digit:]]+)`	Save either one or all local matrices to disk	String	
`dump_eigenvectors(_[[:digit:]]+)?`	Save either one or all local eigenvectors to disk	String	
`krylov_method`	Type of iterative method used to solve linear systems	`gmres`, `bgmres`, `cg`, `bcg`, `gcrodr`, `bgcrodr`, `bfbcg`	`gmres`
`enlarge_krylov_subspace`	Split the initial right-hand side into multiple vectors	Integer	1
`gmres_restart`	Maximum number of Arnoldi vectors generated per cycle	Integer	40
`variant`	Left, right, or variable preconditioning	`left`, `right`, `flexible`	`right`
`qr`	Method used to perform distributed QR factorizations	`cholqr`, `cgs`, `mgs`	`cholqr`
`deflation_tol`	Tolerance when deflating right-hand sides inside block methods	Numeric	
`recycle`	Number of harmonic Ritz vectors to compute	Integer	
`recycle_same_system`	Assume the system is the same as the one for which Ritz vectors have been computed	Boolean	
`recycle_strategy`	Generalized eigenvalue problem to solve for recycling	`A`, `B`	`A`
`recycle_target`	Criterion to select harmonic Ritz vectors	`SM`, `LM`, `SR`, `LR`, `SI`, `LI`	`SM`
`eigensolver_tol`	Tolerance for computing eigenvectors by ARPACK or LAPACK	Numeric	10^{-6}
`geneo_nu`	Number of local eigenvectors to compute for adaptive methods	Integer	20
`geneo_threshold`	Threshold for selecting local eigenvectors for adaptive methods	Numeric	
`geneo_force_uniformity`	Ensure that the number of local eigenvectors is the same for all subdomains	`min`, `max`	
`master_p`	Number of master processes	Integer	1
`master_distribution`	Distribution of coarse right-hand sides and solution vectors	`centralized`, `sol`, `sol_and_rhs`	`centralized`
`master_topology`	Distribution of the master processes	`0`, `1`, `2`	`0`
`master_assembly_hierarchy`	Hierarchy used for the assembly of the coarse operator	Integer	
`master_aggregate_sizes`	Number of master processes per MPI sub-communicators	Integer	`master_p`
`master_dump_matrix`	Save the coarse operator to disk	String	
`master_exclude`	Exclude the master processes from the domain decomposition	Boolean	
`master_not_spd`	Assume the coarse operator is symmetric (instead of symmetric positive definite)	Boolean	

When using Schwarz methods, there are additional options.

Keyword	Description	Possible values	Default
`schwarz_method`	Type of Schwarz preconditioner used to solve linear systems	`ras`, `oras`, `soras`, `asm`, `osm`, `none`	`ras`
`schwarz_coarse_correction`	Type of coarse correction used in two-level methods	`deflated`, `additive`, `balanced`	

When using substructuring methods, there is an additional option.

Keyword	Description	Possible values	Default
`substructuring_scaling`	Scaling used in the definition of the Schur complement preconditioner	`multiplicity`, `stiffness`, `coefficient`	`multiplicity`

When using MKL PARDISO as a subdomain solver (resp. coarse operator solver), there are additional options, cf. `https://software.intel.com/en-us/node/470298` (resp. `https://software.intel.com/en-us/node/590089`).

Keyword	Description	Possible values
`mkl_pardiso_iparm_(2|8|1[013]|2[147])`	Integer control parameters of MKL PARDISO for the subdomain solvers	Integer
`master_mkl_pardiso_iparm_(2|1[013]|2[17])`	Integer control parameters of MKL PARDISO for the coarse operator solver	Integer

When using MUMPS as a subdomain solver (resp. coarse operator solver), there are additional options, cf. `http://mumps.enseeiht.fr/index.php?page=doc`.

Keyword	Description	Possible values
`mumps_icntl_([6-9]|[1-3][0-9]|40)`	Integer control parameters of MUMPS for the subdomain solvers	Integer
`master_mumps_icntl_([6-9]|[1-3][0-9]|40)`	Integer control parameters of MUMPS for the coarse operator solver	Integer

When using *hypre* as a coarse operator solver, there are additional options, cf. `http://acts.nersc.gov/hypre/#Documentation`.

Keyword	Description	Possible values	Default
`master_hypre_solver`	Solver used by *hypre* to solve coarse linear systems	`fgmres`, `pcg`, `amg`	`fgmres`
`master_hypre_tol`	Relative convergence tolerance	Numeric	10^{-12}
`master_hypre_max_it`	Maximum number of iterations	Integer	500
`master_hypre_gmres_restart`	Maximum number of Arnoldi vectors generated per cycle when using FlexGMRES	Integer	100
`master_boomeramg_num_sweeps`	Number of sweeps	Integer	1
`master_boomeramg_max_levels`	Maximum number of multigrid levels	Integer	10
`master_boomeramg_coarsen_type`	Parallel coarsening algorithm	Integer	6
`master_boomeramg_relax_type`	Smoother	Integer	3
`master_boomeramg_interp_type`	Parallel interpolation operator	Integer	0