— 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	
`compute_residual`	Print the residual after convergence	`l2`, `l1`, `linfty`	
`push_prefix`	Prepend a prefix for all following options (use `-hpddm_pop_prefix` when done)		
`reuse_preconditioner`	Do not factorize again the local matrices when solving subsequent systems	Boolean	
`operator_spd`	Assume the operator is symmetric 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`, `richardson`, `none`	`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`
`richardson_damping_factor`	Damping factor using in Richardson iterations	Numeric	1.0
`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_estimate_nu`	Estimate the number of eigenvalues below a threshold using the inertia of the stencil	Numeric	
`geneo_force_uniformity`	Ensure that the number of local eigenvectors is the same for all subdomains	`min`, `max`	

When using multilevel methods, there are additional options, that are all prefixed by `-hpddm_level_`N`_`, with $N > 1$.

Keyword	*Description*	*Possible values*	*Default*
`level_([2-9]|[1-9]\d+)_p`	Number of main processes	Integer	1
`level_([2-9]|[1-9]\d+)_distribution`	Distribution of coarse right-hand sides and solution vectors	`centralized`, `sol`	`centralized`
`level_([2-9]|[1-9]\d+)_topology`	Distribution of the main processes	`0`, `1`, `2`	`0`

`level_([2-9]|[1-9]\d+)_assembly_hierarchy`	Hierarchy used for the assembly of the coarse operator	Integer	
`level_([2-9]|[1-9]\d+)_aggregate_size`	Number of main processes per MPI sub-communicators	Integer	`p`
`level_([2-9]|[1-9]\d+)_dump_matrix`	Save the coarse operator to disk	String	
`level_([2-9]|[1-9]\d+)_exclude`	Exclude the main processes from the domain decomposition	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 or 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|1[013]|2[1457])`	Integer control parameters of MKL PARDISO for the subdomain solvers	Integer

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

Keyword	*Description*	*Possible values*
`mumps_icntl_([678]|1[234]|2[34789]|35)`	Integer control parameters	Integer
`mumps_cntl_([123457])`	Real control parameters	Numeric

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

Keyword	*Description*	*Possible values*	*Default*
`hypre_solver`	Solver used by *hypre* to solve coarse linear systems	`fgmres`, `pcg`, `amg`	`fgmres`
`hypre_tol`	Relative convergence tolerance	Numeric	10^{-12}
`hypre_max_it`	Maximum number of iterations	Integer	500
`hypre_gmres_restart`	Maximum number of Arnoldi vectors generated per cycle when using FGMRES	Integer	100
`boomeramg_num_sweeps`	Number of sweeps	Integer	1
`boomeramg_max_levels`	Maximum number of multigrid levels	Integer	10
`boomeramg_coarsen_type`	Parallel coarsening algorithm	Integer	6
`boomeramg_relax_type`	Smoother	Integer	3
`boomeramg_interp_type`	Parallel interpolation operator	Integer	0

When using ARPACK as an eigensolver, there is an additional option.

Keyword	*Description*	*Possible values*
`arpack_ncv`	Number of Lanczos basis vectors generated in one iteration	Integer

References

For the keyword `krylov_method`:

- value `gmres`, see Y. Saad and M. H. Schultz. "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems". In: *SIAM Journal on Scientific and Statistical Computing* 7.3 (1986), pp. 856–869,
- value `bgmres`, see M. H. Gutknecht. "Block Krylov space methods for linear systems with multiple right-hand sides: an introduction". In: *Modern Mathematical Models, Methods and Algorithms for Real World Systems*. Ed. by A. Siddiqui, I. Duff, and O. Christensen. 2006, pp. 420–447,
- value `cg`, see M. R. Hestenes and E. Stiefel. "Methods of conjugate gradients for solving linear systems". In: *Journal of Research of the National Bureau of Standards* 49.6 (1952), pp. 409–436,
- value `bcg`, see D. P. O'Leary. "The block conjugate gradient algorithm and related methods". In: *Linear Algebra and its Applications* 29 (1980), pp. 293–322,
- value `gcrodr`, see M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti. "Recycling Krylov subspaces for sequences of linear systems". In: *SIAM Journal on Scientific Computing* 28.5 (2006), pp. 1651–1674,
- value `bgcrodr`, see P. Jolivet and P.-H. Tournier. "Block iterative methods and recycling for improved scalability of linear solvers". In: *Proceedings of the 2016 International Conference for High Performance Computing, Networking, Storage and Analysis*. SC16. IEEE. 2016,
- value `bfbcg`, see H. Ji and Y. Li. "A breakdown-free block conjugate gradient method". In: *BIT Numerical Mathematics* 57.2 (2017), pp. 379–403,
- value `richardson`, see `https://en.wikipedia.org/wiki/Modified_Richardson_iteration`.

For the keyword `variant`, value `flexible`, see Y. Saad. "A flexible inner–outer preconditioned GMRES algorithm". In: *SIAM Journal on Scientific Computing* 14.2 (1993), pp. 461–469.

For the keyword `qr`:

- value `cholqr`, see A. Stathopoulos and K. Wu. "A block orthogonalization procedure with constant synchronization requirements". In: *SIAM Journal on Scientific Computing* 23.6 (2002), pp. 2165–2182,
- value `cgs`, see Algorithm 3 on page 3 of V. Hernández, J. E. Román, A. Tomás, and V. Vidal. *Orthogonalization routines in SLEPc*. Tech. rep. URL: `http://slepc.upv.es/documentation/reports/str1.pdf`,
- value `mgs`, see Algorithm 4 on page 4.

For the keyword `deflation_tol`, see section 12 of M. H. Gutknecht. "Block Krylov space methods for linear systems with multiple right-hand sides: an introduction". In: *Modern Mathematical Models, Methods and Algorithms for Real World Systems*. Ed. by A. Siddiqui, I. Duff, and O. Christensen. 2006, pp. 420–447.

For the keywords `geneo_nu`, `geneo_threshold`, `p`, and `topology` see respectively eq. (8), eq. (9), section 3.1.1, and figure 5 of P. Jolivet, F. Hecht, F. Nataf, and C. Prud'homme. "Scalable domain decomposition preconditioners for heterogeneous elliptic problems". In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. SC13. ACM. 2013.

For the keyword `schwarz_method`:

- value `ras`, see X.-C. Cai and M. Sarkis. "Restricted additive Schwarz preconditioner for general sparse linear systems". In: *SIAM Journal on Scientific Computing* 21.2 (1999), pp. 792–797,
- values `oras` and `soras`, see R. Haferssas, P. Jolivet, and F. Nataf. "An additive Schwarz method type theory for Lions's algorithm and a symmetrized optimized restricted additive Schwarz method". In: *SIAM Journal on Scientific Computing* 39.4 (2017), A1345–A1365,
- value `asm`, see eq. (1.30) section 1.4 of V. Dolean, P. Jolivet, and F. Nataf. *An introduction to domain decomposition methods: algorithms, theory, and parallel implementation*. Vol. 144. SIAM, 2015,
- value `osm`, see M. J. Gander. "Optimized Schwarz methods". In: *SIAM Journal on Numerical Analysis* 44.2 (2006), pp. 699–731.

For the keyword `schwarz_coarse_correction`:

- value `deflated`, see eq. (13) section 2.3.3 of J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga. "Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods". In: *Journal of Scientific Computing* 39.3 (2009), pp. 340–370,
- value `additive`, see eq. (7) section 2.3.1,
- value `balanced`, see the first unnumbered equation of section 2.3.4.

For the keyword `substructuring_scaling`:

- value `multiplicity`, see the first bullet point section 3.2.1 of P. Gosselet and C. Rey. "Non-overlapping domain decomposition methods in structural mechanics". In: *Archives of Computational Methods in Engineering* 13.4 (2006), pp. 515–572,
- value `stiffness`, see the second bullet point,
- value `coefficient`, see the third bullet point.