AdaptInfo

Parameters for AdaptInfo(`[name]`,`nComponents`)

keyword	data type	default	description
`(name)->start time`	**double**	`[0.0]`	Initial time
`(name)->timestep`	**double**	`[0.0]`	Time step size to be used
`(name)->end time`	**double**	`[1.0]`	Final time
`(name)->max iteration`	**int**	`[-1]`	maximal allowed number of iterations of the adaptive procedure; if maxIteration ≤ 0, no iteration bound is used
`(name)->max timestep iteration`	**int**	`[30]`	Maximal number of iterations for choosing a timestep
`(name)->max time iteration`	**int**	`[30]`	Maximal number of time iterations
`(name)->min timestep`	**double**	`[0.0]`	Minimal step size
`(name)->max timestep`	**double**	`[1.0]`	Maximal step size
`(name)->number of timesteps`	**int**	`[0]`	Per default this value is 0 and not used. If it is set to a non-zero value, the computation of the stationary problem is done nTimesteps times with a fixed timestep.

Parameters for AdaptInfo for each component separately. `(name):=(name)[`i`]` for $i = 0, \ldots,$`nComponents`-1

keyword	data type	default	description
`(name)[i]->tolerance`	**double**	`[0.0]`	Tolerance for the (absolute or relative) error
`(name)[i]->time tolerance`	**double**	`[0.0]`	Time tolerance.
`(name)[i]->coarsen allowed`	**int {0,1}**	`[0]`	true if coarsening is allowed, false otherwise.
`(name)[i]->refinement allowed`	**int {0,1}**	`[1]`	true if refinement is allowed, false otherwise.
`(name)[i]->refine bisections`	**int**	`[1]`	parameter to tell the marking strategy how many bisections should be performed when an element is marked for refinement; usually the value is 1 or DIM
`(name)[i]->coarsen bisections`	**int**	`[1]`	parameter to tell the marking strategy how many bisections should be undone when an element is marked for coarsening; usually the value is 1 or DIM
`(name)[i]->sum factor`	**double**	`[1.0]`	factors to combine max and integral time estimate
`(name)[i]->max factor`	**double**	`[0.0]`	factors to combine max and integral time estimate

AdaptInstationary

`(name)` ist the first argument of the constructor: AdaptInstationary(`(name)`,...)

keyword	data type	default	description
`(name)->strategy`	**int**	`[0]`	Strategy for choosing one timestep: strategy 0: Explicit strategy, strategy 1: Implicit strategy.

(name)->time delta 1	**double**	[0.7071]	Parameter δ_1 used in time step reduction
(name)->time delta 2	**double**	[1.4142]	Parameter δ_2 used in time step enlargement
(name)->info	**int**	[10]	Info level (from AdaptBase)
(name)->break when stable	**int**	[0]	If this parameter is 1 and the instationary problem is stable, hence the number of solver iterations to solve the problem is zero, the adaption loop will stop.
(name)->time adaptivity debug mode	**bool**	[0]	In debug mode, the adapt loop will print information about timestep decreasing and increasing.
(name)->queue->runtime	**int**	[-1]	Runtime of the queue (of the servers batch system) in seconds. If the problem runs on a computer/server without a time limited queue, the value is -1.
(name)->queue->serialization filename	**string**	[__serialized_problem.ser]	Name of the file used to automatically serialize the problem.

AdaptStationary

(name) ist the first argument of the constructor: AdaptStationary((name),...)

keyword	data type	default	description
(name)->info	**int**	[10]	Info level (from AdaptBase)

Estimator

Global Estimator Parameters

keyword	data type	default	description
(estimator)->error norm	**enum {**NO_NORM, L2_NORM, H1_NORM**}**	[NO_NORM]	Used norm

Parameters for the **RecoveryEstimator**

keyword	data type	default	description
(estimator)->rec method	**int {0,1,2}**	[0]	Recovery method: 0: superconvergent patch recovery (discrete ZZ), 1: local L2-averaging (continuous ZZ-recovery), 2: simple averaging
(estimator)->rel error	**int {0,1}**	[0]	
(estimator)->C	**double**	[1.0]	

Parameters for the **ResidualEstimator**

keyword	data type	default	description

(estimator)->C0	**double**	[0.0]
(estimator)->C1	**double**	[0.0]
(estimator)->C2	**double**	[0.0]
(estimator)->C3	**double**	[0.0]

Parameters for the **SimpleResidualEstimator**

keyword	data type	default	description
(estimator)->C0	**double**	[0.0]	
(estimator)->C1	**double**	[0.0]	

FileWriter

Parameters for Data output. Typically the label ist (name):=(problem-name)->output

keyword	data type	default	description
(problem-name)->output->filename	**string**	[]	Used filename prefix.
(problem-name)->output->AMDiS format	**bool**	[0]	0: Don't write AMDiS files; 1: Write AMDiS files.
(problem-name)->output->AMDiS mesh ext	**string**	[.mesh]	AMDiS mesh-file extension.
(problem-name)->output->AMDiS data ext	**string**	[.dat]	AMDiS solution-file extension.
(problem-name)->output->ParaView format	**bool**	[0]	0: Don't write ParaView files; 1: Write ParaView files.
(problem-name)->output->ParaView vector format	**bool**	[0]	0: Don't write ParaView vector files; 1: Write ParaView vector files.
(problem-name)->output->ParaView animation	**bool**	[0]	0: Don't write ParaView animation file; 1: Write ParaView animation file.
(problem-name)->output->ParaView ext	**string**	[.vtu]	VTK file extension.
(problem-name)->output->Periodic format	**bool**	[0]	0: Don't write periodic files; 1: Write periodic files.
(problem-name)->output->Periodic ext	**string**	[.per]	Periodic file extension.
(problem-name)->output->PNG format	**bool**	[0]	0: Don't write png files; 1: Write png image files.
(problem-name)->output->PNG type	**int**	[0]	0: Gray color picture; 1: RGB picture.
(problem-name)->output->append index	**int**	[0]	0: Don't append time index to filename prefix, 1: Append time index to filename prefix.
(problem-name)->output->index length	**int**	[5]	Total length of appended time index.
(problem-name)->output->index decimals	**int**	[3]	Number of decimals in time index.
(problem-name)->output->write every i-th timestep	**int**	[1]	Timestep modulo: write only every tsModulo-th timestep!
(problem-name)->output->Povray format	**bool**	[0]	0: Don't write Povray scripts; 1: Write Povray scripts
(problem-name)->output->Povray template	**string**	[]	name of the template file that will be prepended to all created *.pov files
(problem-name)->output->Povray camera location	**string**	[]	camera position for povray script files
(problem-name)->output->Povray camera look_at	**string**	[]	orientation for camera in povray script files
(problem-name)->output->DOF format	**bool**	[0]	0: Don't write DOF files; 1: Write DOF files
(problem-name)->output->ARH format	**bool**	[0]	0: Don't write ARH files; 1: Write ARH files

(problem-name)->output->compression	**enum** {gz, gzip, bz2, bzip2}	[]

HL_SignedDistTraverse

Reinitialization class HL_SignedDistTraverse((name),...) with (name) as first argument in the constructor.

keyword	data type	default	description
(name)->tolerance	**double**	[]	Tolerance for Hopf-Lax update iteration loop.
(name)->maximal number of iteration steps	**int**	[]	Maximal number of mesh iterations for Hopf-Lax update.
(name)->Gauss-Seidel iteration	**bool**	[]	Indicates whether Gauss-Seidel or Jacobi iteration is used. 0: Jacobi, 1: Gauss-Seidel
(name)->infinity value	**double**	[]	Initialization value "inifinity" for non-boundary vertices. Must be > 1000
(name)->boundary initialization	**int**	[]	Define boundary initialization strategy. 0: BoundaryElementLevelSetDist, 1: BoundaryElementTopDist, 2: BoundaryElementEdgeDist, 3: BoundaryElementNormalDist

Marker

Global Marker Parameters

keyword	data type	default	description
(marker)->strategy	**int {0-4}**	[0]	0..no marker, 1..GRMarker, 2..MSMarker, 3..ESMarker, 4..GERSMarker
(marker)->p	**int**	[2]	power in estimator norm
(marker)->info	**int**	[10]	Info level
(marker)->max refinement level	**int**	[-1]	Maximal level of all elements
(marker)->min refinement level	**int**	[-1]	Minimal level of all elements

Parameters for the **ESMarker** (Equidistribution strategy [**?**])

keyword	data type	default	description
(marker)->ESTheta	**double**	[0.9]	Marking parameter θ
(marker)->ESThetaC	**double**	[0.2]	Marking parameter θ_C

Parameters for the **GERSMarker** (Guaranteed error reduction strategy [**?**])

keyword	data type	default	description
(marker)->GERSThetaStar	**double**	[0.6]	Marking parameter θ^*
(marker)->GERSNu	**double**	[0.1]	Marking parameter ν

`(marker)->GERSThetaC`	**double**	`[0.1]`	Marking parameter θ_C

Parameters for the **GRMarker** (Global refinement strategy)

keyword	data type	default	description

Parameters for the **MSMarker** (Maximum strategy)

keyword	data type	default	description
`(marker)->MSGamma`	**double**	`[0.5]`	Marking parameter γ
`(marker)->MSGammaC`	**double**	`[0.1]`	Marking parameter γ_C

Mesh

Mesh(`(name)`,·)

keyword	data type	default	description
`(name)->macro file name`	**string**	`[]`	Filename for the macrofile
`(name)->value file name`	**string**	`[]`	Filename for value file.
`(name)->periodic file`	**string**	`[]`	filename for periodic file
`(name)->check`	**int**	`[1]`	Check the mesh structure
`(name)->preserve coarse dofs`	**bool**	`[0]`	When an element is refined, not all dofs of the coarse element must be part of the new elements. An example are centered dofs when using higher lagrange basis functions. The midpoint dof of the parents element is not a dof of the both children elements. Therefore, the dof can be deleted. In some situation, e.g., when using multigrid techniques, it can be necessary to store this coarse dofs. Then this variable must be set to true. If false, the not required coarse dofs will be deleted.
`(name)->macro weights`	**string**	`[]`	file of file that contains weight number for each element. The higher the weigh, the more this element will be refined and thus more processors share this element.
`(name)->global refinements`	**int**	`[0]`	Number of global refinements

Meshdistributor

The label `(name)` represents the string that is passed to the constructor of MeshDistributor.

keyword	data type	default	description

(name)->repartitioning	**bool**	[0]	En/disables repartitioning of the macro mesh, when derivation of mean number of DOFs exceeds a threshold value.
(name)->repartition ith change	**int**	[20]	Sets number of mesh changes to wait before threshold check for repartitioning will be performed.
(name)->partitioner	**enum** {parmetis, zoltan, simple}	[parmetis]	Defines the external tool that performs the partitioning of the, e.g. by graph-partitioning. *simple* does not change the initial partitioning, i.e., its a random distribution of the macro elements to the processors.
(name)->box partitioning	**bool**	[0]	If the macro mesh is globally refined from macro.stand.2d or macro.stand.3d, than the partitioner may compute the partitioning not based on triangled or tetrahedron, but on (composed) rectangles or boxes. Till now this is implemented only for 3D and Zoltan partitioner.
(name)->log main rank	**bool**	[0]	If set to *true*, stdout output will be printed only by the main rank 0. Otherwise, all ranks stdout output will be created.
(name)->pre refine	**int**	[-1]	If set to -1, the number of pre refinements for the macro mesh will be calculated for the given number of processors. This value can be overwritten by setting a value >= 0.
(name)->output->serialization filename	**string**	[]	Name of the parallel serialization file. If at least one stationary problem is serialized, this parameter must be set.
(name)->input->serialization filename	**string**	[]	Name of the parallel deserialization file. If at least one stationary problem is deserialized, this parameter must be set.
(name)->debug output dir	**string**	[]	Path name where debug data should be written to.
(name)->write parallel debug file	**bool**	[0]	If set to *true*, the parallelization will create for each rank a file with the name "mpi-dbg-{rank-no}.dat". This files contain information about all DOF indices in ranks domain. They may be useful for debugging or some postprocessing steps.

OEMSolver

Standard parameters for OEMSolver

keyword	data type	default	description
(solver)->left precon	**enum** {diag,ilu,ic,no}	[no]	left preconditioner
(solver)->right precon	**enum** {diag,ilu,ic,no}	[no]	right preconditioner
(solver)->ell	**int**	[1]	additional solver parameter
(solver)->tolerance	**double**	[0.0]	Solver tolerance norm(r).
(solver)->relative tolerance	**double**	[0.0]	Relative solver tolerance norm(r)/norm(r0).
(solver)->max iteration	**int**	[1000]	maximal number of iterations.
(solver)->print cycle	**int**	[100]	Print cycle, after how many iterations the residuum norm is logged.
(solver)->info	**int**	[0]	info level during solving the system.

Parameters for the direct sparse LU-solver **Umfpack**

keyword	data type	default	description
(solver)->store symbolic	**int**	[0]	
(solver)->symmetric strategy	**int**	[0]	
(solver)->alloc init	**double**	[0.7]	

ProblemStat

Standard ProblemStat((name)). First construktor argument is (name)

keyword	data type	default	description
(name)->components	**int**	[-1]	Number of problem components (must be set)
(name)->input->read serialization	**int**	[0]	
(name)->input->serialization with adaptinfo	**int**	[0]	
(name)->input->serialization filename	**string**	[]	
(name)->mesh	**string**	[]	Name of the mesh
(name)->dim	**int**	[0]	problem dimension
(name)->refinement set[i]	**int**	[-1]	$i = 0, \ldots,$nComponents,
(name)->polynomial degree[i]	**int**	[1]	$i = 0, \ldots,$nComponents,
(name)->solver	**enum** {cg, cgs, bicg, bicgstab, bicgstab2, bicgstab_ell, qmr, tfqmr, gmres, idr_s, minres, (umfpack), 0}	[0]	iterative/direct solver for the linear system
(name)->estimator[i]	**enum** {residual, simple-residual, recovery, 0}	[0]	$i = 0, \ldots,$nComponents, estimator type for each components
(name)->output->write serialization	**int**	[0]	write serialization files

RosenbrockAdaptInstationary

RosenbrockAdaptInstationary

keyword	data type	default	description
(name)->rosenbrock method	**enum** {ros2, rowda3, ros3p, rodasp}	[]	Rosenbrock method that should be used.
(name)->fix first timesteps	**int**	[0]	If greater than 0, than for the first given number of timesteps the timestep will be not changed and is set to the very first one.

`(name)->rosenbrock->timestep study`	**bool**	`[0]`	If true, the first timestep is calculated with different timesteps. This is usually used to make a study how the time error estimator behavous for different timesteps.
`(name)->rosenbrock->timestep study steps`	**bool**	`[0]`	If dbgTimestepStudy is set to true, then this array contains the timesteps for which the first timestep should be calculated.