

<b>FINE MODEL CONVERGED: NO</b> <b>COARSE MODEL CONVERGED: NO</b>	<b>FINE MODEL CONVERGED: NO</b> <b>COARSE MODEL CONVERGED: YES</b>	<b>FINE MODEL CONVERGED: YES</b> <b>COARSE MODEL CONVERGED: YES</b>
<pre>// given: new coarse cplData from solvers H(x) measureConvergence(coarseModel) → false _coarseModelOptimizationActive ← true _MMPostProcessing.performPP(cplData)   _coarseOptimization.optimize(cplData, q_k)   RegisterSolution() // store coupling data for conv. measure and extrp // update time and iteration counts</pre>	<pre>// given: new coarse cplData from solvers H(x) measureConvergence(coarseModel) → true _coarseModelOptimizationActive ← false _doOnlySolverEvaluation ← true</pre>	<pre>// given: new coarse cplData from solvers H(x) measureConvergence(coarseModel) → true _coarseModelOptimizationActive ← false _doOnlySolverEvaluation ← true</pre>
<pre>// fluid solver: readBlockScalarData(cplReadData) mapDown(cplReadData) evaluateCoarseModel() mapUp(cplWriteData) writeBlockScalarData(cplWriteData) // to coarse cplData IDs writeBlockScalarData(cplWriteData) // to fine cplData IDS  // solid solver: readBlockScalarData(cplReadData) mapDown(cplReadData) evaluateCoarseModel() mapUp(cplWriteData) writeBlockScalarData(cplWriteData) // to coarse cplData IDs</pre>	<pre>// fluid solver: readBlockScalarData(cplReadData) evaluateFineModel() writeBlockScalarData(cplWriteData) // to fine cplData IDs  // solid solver: readBlockScalarData(cplReadData) evaluateFineModel() writeBlockScalarData(cplWriteData) // to coarse cplData IDs</pre>	<pre>// fluid solver: readBlockScalarData(cplReadData) evaluateFineModel() writeBlockScalarData(cplWriteData) // to fine cplData IDs  // solid solver: readBlockScalarData(cplReadData) evaluateFineModel() writeBlockScalarData(cplWriteData) // to coarse cplData IDs</pre>
	<pre>// given: new fine cplData from solvers H(x) measureConvergence(fineModel) → false _doOnlySolverEvaluation ← false _MMPostProcessing.performPP(cplData)   _coarseOptimization.iterationsConverged()   UpdateDifferenceMatrices() // F and C   updateCoarseModelDesignSpecification(q_k)   _coarseModelOptimizationActive ← true   _coarseOptimization.optimize(cplData, q_k)   RegisterSolution() // store coupling data for conv. measure and extrp // update time and iteration counts</pre>	<pre>// given: new fine cplData from solvers H(x) measureConvergence(fineModel) → true _doOnlySolverEvaluation ← false _MMPostProcessing.iterationsConverged(cplData)   _coarseOptimization.iterationsConverged()   updateDifferenceMatrices() // F and C   _coarseModelOptimizationActive ← true  // store coupling data for conv. measure and extrp // update time and iteration counts</pre>