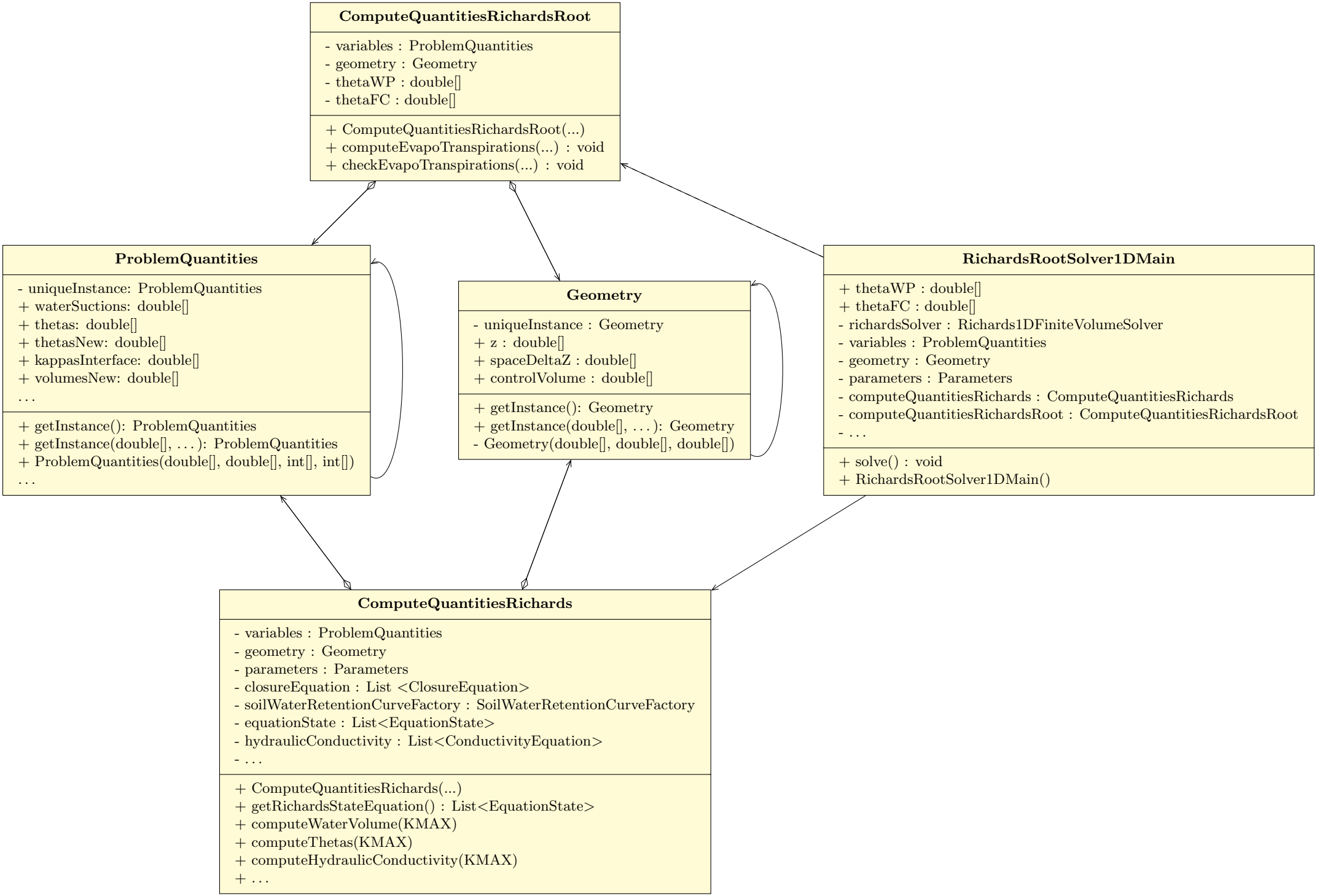
ComputeQuantitiesRichardsRoot
- variables : ProblemQuantities
- geometry : Geometry
- thetaWP : double[]
- thetaFC : double[]
+ ComputeQuantitiesRichardsRoot(...)
+ computeEvapoTranspirations(...) : void
+ checkEvapoTranspirations(...) : void
ProblemQuantities
- uniqueInstance: ProblemQuantities
+ waterSuctions: double[]
+ thetas: double[]
+ thetasNew: double[]
+ kappasInterface: double[]
+ volumesNew: double[]
...
+ getInstance(): ProblemQuantities
+ getInstance(double[], ...): ProblemQuantities
+ ProblemQuantities(double[], double[], int[], int[])
...
Geometry
- uniqueInstance : Geometry
+ z : double[]
+ spaceDeltaZ : double[]
+ controlVolume : double[]
+ getInstance(): Geometry
+ getInstance(double[], ...): Geometry
- Geometry(double[], double[], double[])
RichardsRootSolver1DMain
+ thetaWP : double[]
+ thetaFC : double[]
- richardsSolver : Richards1DFiniteVolumeSolver
- variables : ProblemQuantities
- geometry : Geometry
- parameters : Parameters
- computeQuantitiesRichards : ComputeQuantitiesRichards
- computeQuantitiesRichardsRoot : ComputeQuantitiesRichardsRoot
- ...
+ solve() : void
+ RichardsRootSolver1DMain()
ComputeQuantitiesRichards
- variables : ProblemQuantities
- geometry : Geometry
- parameters : Parameters
- closureEquation : List <ClosureEquation>
- soilWaterRetentionCurveFactory : SoilWaterRetentionCurveFactory
- equationState : List<EquationState>
- hydraulicConductivity : List<ConductivityEquation>
- ...
+ ComputeQuantitiesRichards(...)
+ getRichardsStateEquation() : List<EquationState>
+ computeWaterVolume(KMAX)
+ computeThetas(KMAX)
+ computeHydraulicConductivity(KMAX)
+ ...