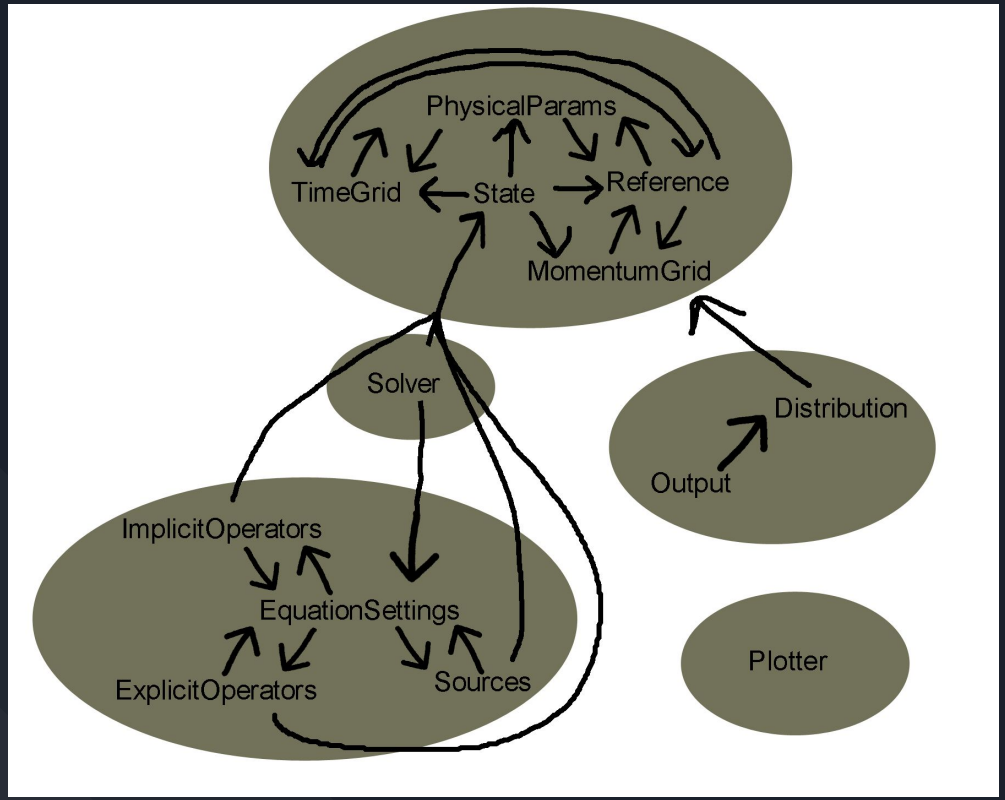
CODE in a nutshell

Overview


PhysicalParams
TimeGrid
State
Reference
MomentumGrid
Solver
Distribution
Output
ImplicitOperators
EquationSettings
ExplicitOperators
Sources
Plotter

Class: SmartLUSolver

Pointed by	Points to	Does
None	State EquationSettings	• Function calls relevant to execute the program • Initializing and saves data • Reinitialization and timestepping

Class: EquationSettings

Pointed by	Points to	Does
Solvers Operators Sources	State	• Determines what operators to use and initializes objects thereof

Class: Operators (Implicit/Explicit)

Pointed by	Points to	Does
EquationSettings	State EquationSettings	• Calculate the matrix relevant for the operators name

Class: Sources

Pointed by	Points to	Does
EquationSettings	State EquationSettings	• Calculate the vector relevant for the source's name

Class: State

Pointed by	Points to	Does
Solvers Operators EquationSettings	TimeGrid MomentumGrid PhysicalParameters Reference	• Collects objects which describes the discretization of the differential equation and its parameters such as temperature

Class: PhysicalParameters

Pointed by	Points to	Does
State TimeGrid Reference	TimeGrid Reference	• Handles the initialization of physical parameters and their interpolation to the TimeGrid

Class: TimeGrid

Pointed by	Points to	Does
State Reference PhysicalParameters	Reference PhysicalParameters	• Contains the time grid where we step in our differenatial equation

Class: MomentumGrid

Pointed by	Points to	Does
State Reference	Reference	• Contains the discretization in momentum space and relevant operators such as differential and integrational

Class: Reference

Pointed by	Points to	Does
State	MomentumGrid	• Contains the normalization of the differential equation and functions to renormalize
MomentumGrid	TimeGrid	
TimeGrid	PhysicalParameters	
PhysicalParameters		

Class: Output

Pointed by	Points to	Does
	Distribution	• Saves data from takeTimeSteps of SmartLUSolver • Saves distributions • Right now very few things are calculated (but the code is there but unchecked)

Class: Distribution

Pointed by	Points to	Does
Output	MomentumGrid	• Saves the distribution function

Class: Plotter

Pointed by

Points to

Does

- Takes Distributions and other parameters to its functions and plots the data
- Should be able to take output and plot relevant data from there but is not implemented