Parcels design overview

Classes in blue; methods in green

Not all methods and classes are shown

Field

`Field` objects hold the data of the hydrodynamic fields, stored as 4-dimensional (lon, lat, depth, time) dask or numpy arrays.

.from_netcdf()

`FieldSet.from_netcdf` is the method used to read hydrodynamic fields in NetCDF data using info of `Fields` in dictionaries.

.from_xarray()

`FieldSet.from_xarray` can directly parse `xarray` objects into Parcels.

.from_list()

`ParticleSet.from_list` is one of the methods used to define the starting positions of `Particles`.

Particle

`Particle` objects contain the position and other variables of each particle in the `ParticleSet`.

FieldSet

`FieldSet` objects are collections of `Fields` or `FieldLists`. At least a U and V `Field` are required for Parcels to work.

ParticleSet

`ParticleSet` objects are the main objects in Parcels. They contain a `FieldSet` and a list of `Particles`.
The `.from_list`, `.execute` and `.show` are the most important methods defined on `ParticleSets`.

.execute()

`ParticleSet.execute` is the method used to actually compute the evolution of particles by executing `Kernel` objects.

.show()

`ParticleSet.show` is the method used to plot particle positions, optionally overlayed on a `Field`.

Kernel

`Kernels` are little snippets of code that get run when a `ParticleSet` is executed. Parcels comes with some built-in kernels like 4th order Runge-Kutta advection, but it is very easy to create custom kernels. Multiple kernels can be concatenated with the `+` operator.

oceanparcels.org