Python Documentation

version

July 20, 2022

Contents

OceanTracker

Fast particle tracking in unstructured grids

OceanTracker is a fast extendable code for particle tracking in unstructured grids [1].

OceanTracker's speed enables millions of particles to be simulated in unstructured grids. This significantly increases the range of particle behaviours that can be modeled and the quality of statistics derived from the particles. To eliminate the need to store and wade through the analysis of vast volumes of recorded particle tracks, the code has the ability to calculate statistics on the fly, such as heat maps and connectivity between regions.

OceanTracker code is highly flexible and extendable by the user, whether run by a new user with a text file of parameters, or by an expert adding their specialised code for novel particle behaviours or statistics, to the computational pipe line.

Features

- Fast and extendable offline native grid particle tracking for unstructured grids [1]
- Calculate the tracks of millions of particles
- Native grid particle tracking for SCHISM like grids, which preserves the resolution of Slayer and LSC vertical grids
- Builds heat maps on the fly, without recording particle tracks; plus inside polygon statistics computed on the fly
- Backward and forward in time particle tracking
- Shoreline stranding of particles by the tide and resuspension from the bottom
- 2D and 3D particle tracking, with option to run 3D as 2D

Architecture

- Implemented in Python
- Driven by parameters in JSON or YAML file, or in code from a Python dictionary
- Tools to read output, plus plot animations
- Highly customizable at parameter level
- Extendable to create novel particle behaviours, eg. vertical migration of plankton
- Can run particle tracking cases in parallel to further improve computational speed

CAWTHRON INSTITUTE

About

Background

Lagrangian particle tracking, is an important tool in quantifying bio-physical transports in the ocean. Particle tracking in the unstructured grids typically used in coastal regions is computationally slow, limiting the number of particles and ranges of behaviours that can be modeled.

OceanTracker was created to be 100s of times faster than an existing freely available particle tracker when using unstructured grids. More significantly, OceanTracker computational speed is simmilar to that achieved when particle tracking on a regular grid (Vennell et. al.)

This makes it possible to routinely calculate the trajectories of millions of particles. Allowing large number of particles allows much better estimates of dispersion and transport statistics, particularly when the probability of connection is low but the consequences are significant, e.g. the spread of invasive species. It also enables wider exploration of parameter sensitivity and particles' bio-physical behaviours to provide more robust results.

The speed increases result largely from exploiting history and reuse within the spatial interpolation of the hydrodynamic model's output. Using multiple computer cores further increased the speed to track a given number of particles.

In addition to speed the internal architecture of OceanTracker makes it easy for the user to customise and extend.

Architecture

- Fully driven by parameters in JSON/YAML file or dictionary
- **Highly flexible architecture enabling:**
    - user implemented approaches to core classes, core classes can be replaced via string name in parameter dictionary, eg. user spatial interpolator
    - **user developed:**
        - custom particle properties derived from other properties though inheritance
        - augment particle velocity given by water_velocity read from hindcast, eg. particle fall velocity
        - modify particle trajectories, eg. resupensension.
- Automated processes to add user developed particle proprieties, velocity, trajectory modifiers, etc , to calculation and output chain.
- Reduce memory requirement in 'compact_mode', which only retains active particles, eg. those young enough to be of interest.
- Uses numba code for fast in-place operations on particle properties and read fields based on set of indices arrays.
- **All core and optional classes can be changed or added to list as parameter string using class_name as a string, eg optional particle distance travelled to date calculation:**
  `{'class_name':'oceantracker.particle_properties.distance_travelled.DistanceTravelled'}`
  which will automatically writes distance data to tracks output file. Class name can be an OceanTracker version or a users supplied class or version of a class.

Demos

Minimal example

Example to show the minmum required to run oceanTracker, using code to both run and plot, or from parameter file. For this 3D hindcast, grey particles are stranded on the bottom, green ones are stranded on the shore by the tide.

Run and plot using code:

../../../demos/minimal_example.py

```
# minimal_example.py
#-------------------
from oceantracker import main

# create parameters as a dictionary
params= {'shared_params' : {'output_file_base' :'minimal_example',
                            'root_output_dir':'output'},
    'reader': { 'class_name': 'oceantracker.reader.schism_reader.SCHSIMreaderNCDF',
                'input_dir': 'demo_hindcast',
                'file_mask': 'demoHindcastSchism3D.nc',
                },
    'base_case_params' : { 'solver': {'n_sub_steps': 12}, #not required but runs 5min steps
                'particle_release_groups':
                      [{'points': [[1595000, 5483300, -2],[1596000, 5487200, -2] ], # two 3D
                        'pulse_size': 10, 'release_interval': 3600}
                       ]
```

```
                            }
        }

# run OceanTracker to give track output files
runInfo_file_name, has_errors = main.run(params)

# output now in folder output/minimal_example

# below is optional code for plotting
#-------------------------------------
from oceantracker.post_processing.plotting.plot_tracks import animate_particles, plot_tracks
from oceantracker.post_processing.read_output_files.load_output_files import get_case_info_f

# find case_info_file name, to used to locate output for the caseInfo file
case_info_file_name = get_case_info_file_from_run_file(runInfo_file_name)

# read particle tracks for plotting
track_data = load_particle_track_vars(case_info_file_name)

# plot tracks
anim = plot_tracks(track_data, axis_lims=[1591000, 1601500, 5478500, 5491000],
                           plot_file_name='output\\minimal_example.jpeg')

# animate particles
anim = animate_particles(track_data, axis_lims=[1591000, 1601500, 5478500, 5491000],
                         title='Minimal example of OceanTracker with 3D point release',
                         movie_file='output\\minimal_example.mp4')
```

Run using parameter file:

```
python run_oceantracker.py --param_file minimal_example.yaml
```

../../../demos/demo_yaml/minimal_example.yaml

```yaml
base_case_params:
  particle_release_groups:
  - points:
    - - 1595000
      - 5483300
      - -2
    - - 1596000
      - 5487200
      - -2
    pulse_size: 10
    release_interval: 3600
  solver:
    n_sub_steps: 12
reader:
  class_name: oceantracker.reader.schism_reader.SCHSIMreaderNCDF
  file_mask: demoHindcastSchism3D.nc
  input_dir: demo_hindcast
shared_params:
  output_file_base: minimal_example
  root_output_dir: output
```

Output is written to directory given set by shared_parameters, ie. a dir named

```
['root_output_dir']/['output_file_base']
```

in this example output is in

```
output/minimal_example
```

Particle properties

Particle properties can easily add to computation via parameters, eg decaying concentrations, by listing in particle_properties parameters. Standard properties include, status, tide, water_depth and particle age.

Decaying particle

Decaying particle property used to size and colour particles. `decay_time_scale` parameter = 3.5 hours.

- `particle_properties.age_decay.AgeDecay`
- demo60_SCHISM_3D_decaying_particle.py

../../../demos/demo_code/demo60_SCHISM_3D_decaying_particle.py

```
# demo60_SCHISM_3D_decaying_particle.py
#-------------------------------------
import oceantracker.main as main
from oceantracker.util import json_util
params = json_util.read_JSON("..\demo_json\demo60_SCHISM_3D_decaying_particle.json")

runInfo_file_name, has_errors = main.run(params)

# output is now in output/demo60_SCHISM_3D_decaying_particle

# below only required for plotting
from oceantracker.post_processing.read_output_files.load_output_files import load_particle_t
from oceantracker.post_processing.plotting.plot_tracks import animate_particles

output_file= "output\demo60_SCHISM_3D_decaying_particle"

case_info_file_name = get_case_info_file_from_run_file(runInfo_file_name)

track_data = load_particle_track_vars(case_info_file_name, var_list=['tide', 'water_depth',

animate_particles(track_data, axis_lims=[1591000, 1601500, 5478500, 5491000],
                            heading='SCHISIM reader, 3D, decaying particles, decay time 3.5
                            colour_using_data=track_data['C'], part_color_map='hot_r',
                            size_using_data=track_data['C'],
                            vmax=1.0,
                            movie_file=output_file + '.mp4' if output_file is not None else
                            fps=24,
                            interval=20)
```

../../../demos/demo_yaml/demo60_SCHISM_3D_decaying_particle.yaml

```
base_case_params:
  dispersion:
    A_H: 0.2
    A_V: 0.001
  particle_properties:
  - class_name: oceantracker.particle_properties.age_decay.AgeDecay
    decay_time_scale: 25714.28571428571
    name: C
  particle_release_groups:
  - maximum_age: 17280.000000000004
    points:
    - - 1594500
      - 5487000
      - -1
    - - 1594500
      - 5483000
      - -1
    - - 1598000
```

```
      - 5486100
      - -1
    pulse_size: 1
    release_interval: 150.0
  run_params: {}
  solver:
    n_sub_steps: 24
reader:
  class_name: oceantracker.reader.schism_reader.SCHSIMreaderNCDF
  depth_average: true
  field_variables:
    water_temperature: temp
  file_mask: demoHindcastSchism3D.nc
  input_dir: E:/H_Local_drive/ParticleTracking/oceantracker02/demos\demo_hindcast
shared_params:
  compact_mode: true
  debug: true
  output_file_base: demo60_SCHISM_3D_decaying_particle
  root_output_dir: output
```

Polygon aware particles

Particles with additional inside polygon property, with optional logging of polygon entry and exit events

- class: `particle_properties.inside_polygons.InsidePolygonsNonOverlapping2D`
- class: `event_loggers.log_polygon_entry_and_exit.LogPolygonEntryAndExit`

../../../demos/demo_code/demo07_inside_polygon_events.py

```
# demo07_inside_polygon_events.py
#---------------------------------------
import oceantracker.main as main
from oceantracker.util import json_util
params = json_util.read_JSON("..\demo_json\demo07_inside_polygon_events.json")

runInfo_file_name, has_errors = main.run(params)

# output is now in output/demo07_inside_polygon_events

# below only required for plotting
from oceantracker.post_processing.read_output_files.load_output_files import load_particle_t
from oceantracker.post_processing.plotting.plot_tracks import animate_particles
from matplotlib import colors

output_file= "output\demo07_inside_polygon_events"

case_info_file_name = get_case_info_file_from_run_file(runInfo_file_name)
caseInfo = read_case_info_file(case_info_file_name)
track_data = load_particle_track_vars(case_info_file_name, var_list=['event_polygon'])

cmap = colors.ListedColormap(['b', 'm', 'y'])
animate_particles(track_data, colour_using_data=track_data['event_polygon'],
                                part_color_map=cmap,
                                axis_lims=[1591000, 1601500, 5478500, 5491000],
                                heading='Event logger, polygon aware particles',
                                vmin=-1,
                                vmax=1,
                                interval=70,
                                movie_file=output_file + '.mp4' if output_file is not None e
                                fps=15,
                                polygon_list_to_plot=caseInfo['full_params']['event_loggers'
```

../../../demos/demo_yaml/demo07_inside_polygon_events.yaml

```yaml
base_case_params:
  dispersion:
    A_H: 0.1
  event_loggers:
  - class_name: oceantracker.event_loggers.log_polygon_entry_and_exit.LogPolygonEntryAndExit
    particle_prop_to_write_list:
    - ID
    - x
    - IDrelease_group
    - status
    - age
    polygon_list:
    - points:
      - - 1592682.1237
        - 5489972.7479
      - - 1593604.1667
        - 5490275.5488
      - - 1593886.4247
        - 5489464.0424
      - - 1592917.3387
        - 5489000.0
      - - 1592300.0
        - 5489000.0
      - - 1592682.1237
        - 5489972.7479
      user_polygon_name: A
    - points:
      - - 1597682.1237
        - 5489972.7479
      - - 1598604.1667
        - 5490275.5488
      - - 1598886.4247
        - 5489464.0424
      - - 1597917.3387
        - 5487000
      - - 1597300
        - 5487000
      - - 1597682.1237
        - 5489972.7479
      user_polygon_name: B
  particle_properties:
  - class_name: oceantracker.particle_properties.age_decay.AgeDecay
    decay_time_scale: 86400.0
  particle_release_groups:
  - points:
    - - 1594500
      - 5490000
    - - 1598000
      - 5488500
    pulse_size: 10
    release_interval: 10800
  run_params:
    description: test of notes
  solver:
    n_sub_steps: 6
  tracks_writer:
    turn_on_write_particle_properties_list:
    - n_cell
reader:
```

```
  class_name: oceantracker.reader.generic_unstructured_reader.GenericUnstructuredReader
  dimension_map:
    node: nodes
    time: time
  field_variables:
    tide: tide
    water_depth: depth
    water_velocity:
    - east_vel
    - north_vel
  file_mask: demoHindcast2D*.nc
  grid_variables:
    time: time_sec
    triangles: tri
    x:
    - east
    - north
  input_dir: E:/H_Local_drive/ParticleTracking/oceantracker02/demos\demo_hindcast
  isodate_of_hindcast_time_zero: '2020-06-01'
  search_sub_dirs: true
  time_buffer_size: 24
shared_params:
  add_date_to_run_output_dir: false
  compact_mode: true
  output_file_base: demo07_inside_polygon_events
  root_output_dir: output
```

Particle Status

Particles coloured by their status property. Status can be one of following strings

```
['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead', 'outside_open_bo
undary', 'frozen', 'stranded_by_tide', 'on_bottom', 'moving']
```

../../../demos/demo_code/demo02_animation.py

```
# demo02_animation.py
#----------------------------------------
import oceantracker.main as main
from oceantracker.util import json_util
params = json_util.read_JSON("..\demo_json\demo02_animation.json")

runInfo_file_name, has_errors = main.run(params)

# output is now in output/demo02_animation

# below only required for plotting
from oceantracker.post_processing.read_output_files.load_output_files import load_particle_t
from oceantracker.post_processing.plotting.plot_tracks import animate_particles

output_file= "output\demo02_animation"

case_info_file_name = get_case_info_file_from_run_file(runInfo_file_name)

track_data = load_particle_track_vars(case_info_file_name, fraction_to_read=0.9)

animate_particles(track_data, axis_lims=[1591000, 1601500, 5478500, 5491000],
                              heading='3 hourly point and polygon releases with tidal strandin
                              release_group=None,
                              movie_file=output_file + '.mp4' if output_file is not None else
                              fps=15, back_ground_depth=True)
```

../../../demos/demo_yaml/demo02_animation.yaml

```yaml
base_case_params:
  dispersion:
    A_H: 0.1
  particle_properties:
  - class_name: oceantracker.particle_properties.age_decay.AgeDecay
    decay_time_scale: 86400.0
    initial_value: 20.0
    name: Oxygen
  - class_name: oceantracker.particle_properties.distance_travelled.DistanceTravelled
  particle_release_groups:
  - points:
    - - 1594500
      - 5483000
    - - 1598000
      - 5486100
    pulse_size: 10
    release_interval: 10800
  - class_name: oceantracker.particle_release_groups.polygon_release.PolygonRelease
    points:
    - - 1597682.1237
      - 5489972.7479
    - - 1598604.1667
      - 5490275.5488
    - - 1598886.4247
      - 5489464.0424
    - - 1597917.3387
      - 5489000
    - - 1597300
      - 5489000
    - - 1597682.1237
      - 5489972.7479
    pulse_size: 10
    release_interval: 10800
  run_params:
    description: test of notes
  solver:
    n_sub_steps: 6
  tracks_writer:
    turn_on_write_particle_properties_list:
    - n_cell
reader:
  class_name: oceantracker.reader.generic_unstructured_reader.GenericUnstructuredReader
  dimension_map:
    node: nodes
    time: time
  field_variables:
    tide: tide
    water_depth: depth
    water_velocity:
    - east_vel
    - north_vel
  file_mask: demoHindcast2D*.nc
  grid_variables:
    time: time_sec
    triangles: tri
    x:
    - east
    - north
  input_dir: E:/H_Local_drive/ParticleTracking/oceantracker02/demos\demo_hindcast
```

```
  isodate_of_hindcast_time_zero: '2020-06-01'
  search_sub_dirs: true
  time_buffer_size: 24
shared_params:
  add_date_to_run_output_dir: false
  compact_mode: true
  output_file_base: demo02_animation
  root_output_dir: output
```

Trajectory modifiers

Trajectory modifiers classes change the path of particles, examples from standard trajectory_modifiers classes below.

Particle behaviour

Particles with a random fraction temporarily frozen on a polygon shaped reef.

- class: `trajectory_modifiers.settlement_in_polygon.SettleInPolygon`

../../../demos/demo_code/demo06_reefstranding.py

```
# demo06_reefstranding.py
#----------------------------------------
import oceantracker.main as main
from oceantracker.util import json_util
params = json_util.read_JSON("..\demo_json\demo06_reefstranding.json")

runInfo_file_name, has_errors = main.run(params)

# output is now in output/demo06_reefstranding

# below only required for plotting
from oceantracker.post_processing.read_output_files.load_output_files import load_particle_t
from oceantracker.post_processing.plotting.plot_tracks import animate_particles

output_file= "output\demo06_reefstranding"

case_info_file_name = get_case_info_file_from_run_file(runInfo_file_name)

track_data = load_particle_track_vars(case_info_file_name)

animate_particles(track_data, axis_lims=[1591000, 1601500, 5478500, 5491000],
                            heading='Trajectory Modifer example, particles liking a reef',
                            release_group=None,
                            movie_file=output_file + '.mp4' if output_file is not None else
                            fps=15, back_ground_depth=True)
```

../../../demos/demo_yaml/demo06_reefstranding.yaml

```
base_case_params:
  dispersion:
    A_H: 0.1
  particle_properties:
  - class_name: oceantracker.particle_properties.age_decay.AgeDecay
    decay_time_scale: 86400.0
    initial_value: 20.0
    name: Oxygen
  - class_name: oceantracker.particle_properties.distance_travelled.DistanceTravelled
  particle_release_groups:
  - points:
    - - 1594500
```

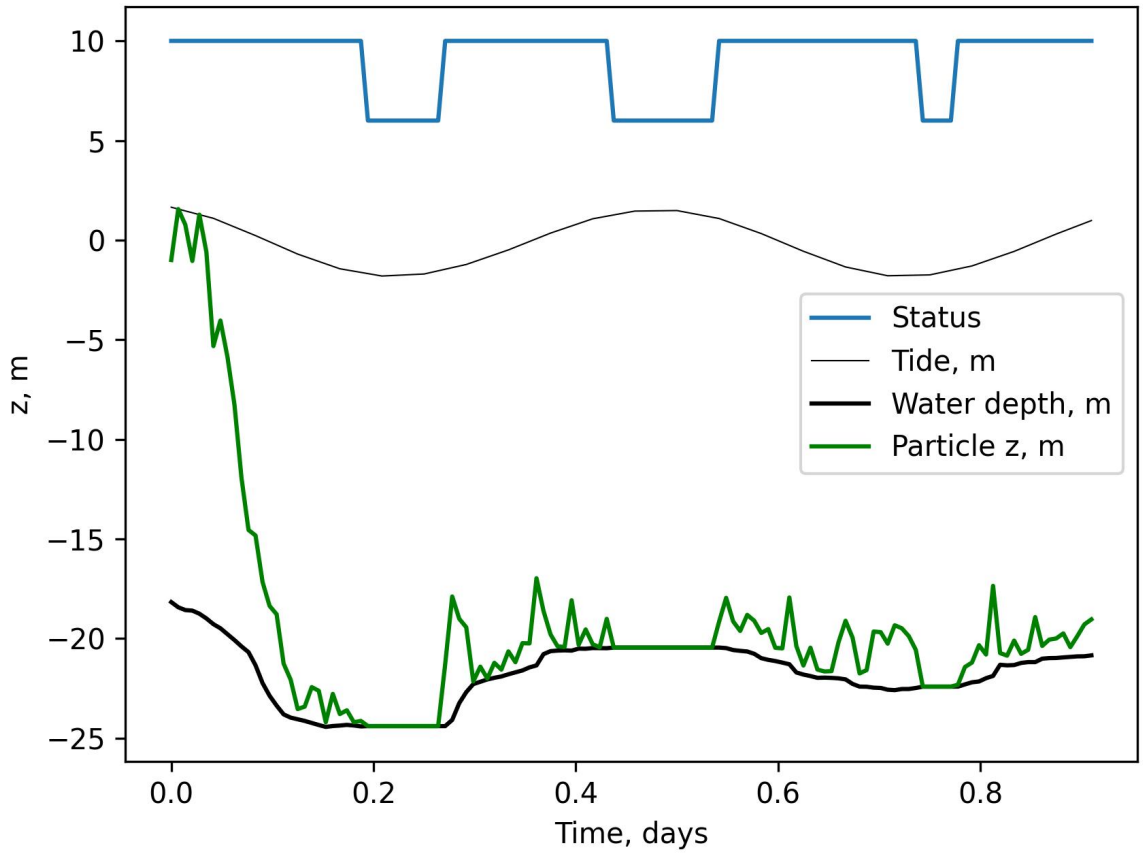
```
        - 5482700
      - - 1598000
        - 5486100
      - - 1595500
        - 5489700
      pulse_size: 10
      release_interval: 10800
    - class_name: oceantracker.particle_release_groups.polygon_release.PolygonRelease
      points:
      - - 1597682.1237
        - 5489972.7479
      - - 1598604.1667
        - 5490275.5488
      - - 1598886.4247
        - 5489464.0424
      - - 1597917.3387
        - 5489000
      - - 1597300
        - 5489000
      - - 1597682.1237
        - 5489972.7479
      pulse_size: 1
      release_interval: 0
    run_params:
      description: test of notes
    solver:
      n_sub_steps: 6
    tracks_writer:
      turn_on_write_particle_properties_list:
      - n_cell
    trajectory_modifiers:
    - class_name: oceantracker.trajectory_modifiers.settle_in_polygon.SettleInPolygon
      polygon:
        points:
        - - 1597682.1237
          - 5489972.7479
        - - 1598604.1667
          - 5490275.5488
        - - 1598886.4247
          - 5489464.0424
        - - 1597917.3387
          - 5489000
        - - 1597300
          - 5489000
        - - 1597682.1237
          - 5489972.7479
      probability_of_settlement: 0.1
      settlement_duration: 10800.0
  reader:
    class_name: oceantracker.reader.generic_unstructured_reader.GenericUnstructuredReader
    dimension_map:
      node: nodes
      time: time
    field_variables:
      tide: tide
      water_depth: depth
      water_velocity:
      - east_vel
      - north_vel
    file_mask: demoHindcast2D*.nc
```

```
    grid_variables:
      time: time_sec
      triangles: tri
      x:
      - east
      - north
    input_dir: E:/H_Local_drive/ParticleTracking/oceantracker02/demos\demo_hindcast
    isodate_of_hindcast_time_zero: '2020-06-01'
    search_sub_dirs: true
    time_buffer_size: 24
  shared_params:
    add_date_to_run_output_dir: false
    backtracking: true
    compact_mode: true
    output_file_base: demo06_reefstranding
    root_output_dir: output
```

Resuspension

Particles with fall velocity and resuspension based on critical friction velocity.

- class `trajectory_modifiers.resuspension.BasicResuspension`



Vertical slice showing one example of a falling particle and resuspension, with particle on bottom during low flows around low and high tides. Blue line is particle status, 10= moving, 6 = on the bottom.

../../../demos/demo_code/demo58_bottomBounce.py

```
# demo58_bottomBounce.py
#---------------------------------------
import oceantracker.main as main
```

```
from oceantracker.util import json_util
params = json_util.read_JSON("..\demo_json\demo58_bottomBounce.json")

runInfo_file_name, has_errors = main.run(params)

# output is now in output/demo58_bottomBounce

# below only required for plotting
from oceantracker.post_processing.read_output_files.load_output_files import load_particle_t
from oceantracker.post_processing.plotting.plot_tracks import animate_particles
from oceantracker.post_processing.plotting.plot_vertical_tracks import plot_path_in_vertical

output_file= "output\demo58_bottomBounce"

case_info_file_name = get_case_info_file_from_run_file(runInfo_file_name)

track_data = load_particle_track_vars(case_info_file_name, var_list=['tide', 'water_depth'])

plot_path_in_vertical_section(track_data,  title= 'fall velocity, always resuspend ',
                                        plot_file_name=output_file +  '_section.jpeg' if outpu

plot_relative_height(track_data, title='fall velocity, always resuspend ')
plot_relative_height(track_data, title='fall velocity, always resuspend ', bottom=False)

animate_particles(track_data, axis_lims=[1591000, 1601500, 5478500, 5491000], heading='verti
```

../../../demos/demo_yaml/demo58_bottomBounce.yaml

```
base_case_params:
  dispersion:
    A_H: 0.05
    A_V: 0.002
  event_loggers:
  - class_name: oceantracker.event_loggers.log_polygon_entry_and_exit.LogPolygonEntryAndExit
    particle_prop_to_write_list:
    - ID
    - x
    - IDrelease_group
    - status
    - age
    polygon_list:
    - points:
      - - 1592682.1237
        - 5489972.7479
      - - 1593604.1667
        - 5490275.5488
      - - 1593886.4247
        - 5489464.0424
      - - 1592917.3387
        - 5489000.0
      - - 1592300.0
        - 5489000.0
      - - 1592682.1237
        - 5489972.7479
      user_polygon_name: A
    - points:
      - - 1597682.1237
        - 5489972.7479
      - - 1598604.1667
        - 5490275.5488
      - - 1598886.4247
```

```
        - 5489464.0424
      - - 1597917.3387
        - 5487000
      - - 1597300
        - 5487000
      - - 1597682.1237
        - 5489972.7479
      user_polygon_name: B
  particle_properties:
  - class_name: oceantracker.particle_properties.age_decay.AgeDecay
    decay_time_scale: 86400.0
  - class_name: oceantracker.particle_properties.friction_velocity.FrictionVelocity
  particle_release_groups:
  - points:
    - - 1593000.0
      - 5486000.0
      - -1
    pulse_size: 10
    release_interval: 0
  particle_statistics:
  - calculation_interval: 3600
    class_name: oceantracker.particle_statistics.gridded_statistics.GriddedStats2D_timeBased
    grid_size:
    - 120
    - 121
    particle_property_list:
    - water_depth
  run_params: {}
  solver:
    n_sub_steps: 6
  trajectory_modifiers:
  - class_name: oceantracker.trajectory_modifiers.resuspension.BasicResuspension
  velocity_modifiers:
  - class_name: oceantracker.velocity_modifiers.terminal_velocity.AddTerminalVelocity
    mean: -0.002
reader:
  class_name: oceantracker.reader.schism_reader.SCHSIMreaderNCDF
  depth_average: false
  field_variables:
    water_temperature: temp
  field_variables_to_depth_average:
  - water_velocity
  - water_depth
  - salt
  - water_temperature
  file_mask: demoHindcastSchism3D.nc
  input_dir: E:/H_Local_drive/ParticleTracking/oceantracker02/demos\demo_hindcast
shared_params:
  backtracking: false
  compact_mode: true
  debug: true
  output_file_base: demo58_bottomBounce
  root_output_dir: output
```

Splitting particles

Particles with splitting in two every 6 hours and a 5% chance of dying every 6 hours.

- class: `trajectory_modifiers.split_particles.SplitParticles`
- class: `trajectory_modifiers.cull_particles.CullParticles`

../../../demos/demo_code/demo08_particle_splitting.py

```
# demo08_particle_splitting.py
#--------------------------------------
import oceantracker.main as main
from oceantracker.util import json_util
params = json_util.read_JSON("..\demo_json\demo08_particle_splitting.json")

runInfo_file_name, has_errors = main.run(params)

# output is now in output/demo08_particle_splitting

# below only required for plotting
from oceantracker.post_processing.read_output_files.load_output_files import load_particle_t
from oceantracker.post_processing.plotting.plot_tracks import animate_particles

output_file= "output\demo08_particle_splitting"

case_info_file_name = get_case_info_file_from_run_file(runInfo_file_name)

track_data = load_particle_track_vars(case_info_file_name)

animate_particles(track_data, axis_lims=[1591000, 1601500, 5478500, 5491000],
                              heading='Split moving particles in two and culling 5%  every 6 h
                              release_group=0,
                              min_status=-2,
                              movie_file=output_file + '.mp4' if output_file is not None else
                              fps=15, back_ground_depth=True)
```

../../../demos/demo_yaml/demo08_particle_splitting.yaml

```
base_case_params:
  dispersion:
    A_H: 0.1
  event_loggers:
  - class_name: oceantracker.event_loggers.log_polygon_entry_and_exit.LogPolygonEntryAndExit
    particle_prop_to_write_list:
    - ID
    - x
    - IDrelease_group
    - status
    - age
    polygon_list:
    - points:
      - - 1592682.1237
        - 5489972.7479
      - - 1593604.1667
        - 5490275.5488
      - - 1593886.4247
        - 5489464.0424
      - - 1592917.3387
        - 5489000.0
      - - 1592300.0
        - 5489000.0
      - - 1592682.1237
        - 5489972.7479
      user_polygon_name: A
    - points:
      - - 1597682.1237
        - 5489972.7479
      - - 1598604.1667
        - 5490275.5488
```

```
        - - 1598886.4247
          - 5489464.0424
        - - 1597917.3387
          - 5487000
        - - 1597300
          - 5487000
        - - 1597682.1237
          - 5489972.7479
        user_polygon_name: B
    particle_properties:
    - class_name: oceantracker.particle_properties.age_decay.AgeDecay
      decay_time_scale: 86400.0
    particle_release_groups:
    - points:
      - - 1594500
        - 5483500
      - - 1594500
        - 5486500
      pulse_size: 1
      release_interval: 0
    run_params:
      description: test of notes
      particle_buffer_size: 5000
      retain_culled_part_locations: true
    solver:
      n_sub_steps: 6
    tracks_writer:
      turn_on_write_particle_properties_list:
      - n_cell
    trajectory_modifiers:
    - class_name: oceantracker.trajectory_modifiers.split_particles.SplitParticles
      split_status_greater_than: frozen
      splitting_interval: 21600
    - class_name: oceantracker.trajectory_modifiers.cull_particles.CullParticles
      cull_interval: 21600
      cull_status_greater_than: dead
      probability_of_culling: 0.05
  reader:
    class_name: oceantracker.reader.generic_unstructured_reader.GenericUnstructuredReader
    dimension_map:
      node: nodes
      time: time
    field_variables:
      tide: tide
      water_depth: depth
      water_velocity:
      - east_vel
      - north_vel
    file_mask: demoHindcast2D*.nc
    grid_variables:
      time: time_sec
      triangles: tri
      x:
      - east
      - north
    input_dir: E:/H_Local_drive/ParticleTracking/oceantracker02/demos\demo_hindcast
    isodate_of_hindcast_time_zero: '2020-06-01'
    search_sub_dirs: true
    time_buffer_size: 24
  shared_params:
```

```
  add_date_to_run_output_dir: false
  compact_mode: false
  output_file_base: demo08_particle_splitting
  root_output_dir: output
```

Particle Statistics

Heat maps

Heatmaps built on the fly with no particle tracks recorded. Options for both time and aged based gridded heatmaps

- class: `particle_statistics.gridded_statistics.GriddedStats2D_timeBased`
- class: `particle_statistics.gridded_statistics.GriddedStats2D_agedBased`

along with counts of particles inside polygons

- class: `particle_statistics.polygon_statistics.PolygonStats2D_timeBased`
- class: `particle_statistics.polygon_statistics.PolygonStats2D_ageBased`

../../../demos/demo_code/demo03_heatmaps.py

```
# demo03_heatmaps.py
#---------------------------------------
import oceantracker.main as main
from oceantracker.util import json_util
params = json_util.read_JSON("..\demo_json\demo03_heatmaps.json")

runInfo_file_name, has_errors = main.run(params)

# output is now in output/demo03_heatmaps

# below only required for plotting
from oceantracker.post_processing.read_output_files.load_output_files import load_stats_file
from oceantracker.post_processing.plotting.plot_heat_maps import plot_heat_map, animate_heat

output_file= "output\demo03_heatmaps"

case_info_file_name = get_case_info_file_from_run_file(runInfo_file_name)

stats_data = load_stats_file(case_info_file_name, var_list=['water_depth'], nsequence=1)
axis_lims = [1591000, 1601500, 5478500, 5491000]
animate_heat_map(stats_data, axis_lims=axis_lims,
                                heading='Particle count heatmaps built on the fly, no tracks
                                movie_file=output_file + '.mp4' if output_file is not None e
                                fps=7)
plot_heat_map(stats_data, axis_lims=axis_lims, var='water_depth', heading='Water depth built
                              plot_file_name=output_file + '_water_depth.jpeg' if output_file
s = load_stats_file(case_info_file_name, nsequence=2, var_list=['water_depth'])  # test poly
```

../../../demos/demo_yaml/demo03_heatmaps.yaml

```
base_case_params:
  dispersion:
    A_H: 0.1
  particle_properties:
  - class_name: oceantracker.particle_properties.age_decay.AgeDecay
    decay_time_scale: 86400.0
    initial_value: 20.0
    name: Oxygen
  - class_name: oceantracker.particle_properties.distance_travelled.DistanceTravelled
  particle_release_groups:
  - points:
```

```
      - - 1596000
        - 5486000
      pulse_size: 2000
      release_interval: 7200
      release_radius: 100.0
    - points:
      - - 1596000
        - 5490000
      pulse_size: 2000
      release_interval: 7200
  particle_statistics:
  - calculation_interval: 1800
    class_name: oceantracker.particle_statistics.gridded_statistics.GriddedStats2D_timeBased
    grid_size:
    - 220
    - 221
    particle_property_list:
    - water_depth
  - calculation_interval: 1800
    class_name: oceantracker.particle_statistics.polygon_statistics.PolygonStats2D_timeBased
    count_status_equal_to: moving
    particle_property_list:
    - water_depth
    polygon_list:
    - points:
      - - 1597682.1237
        - 5489972.7479
      - - 1598604.1667
        - 5490275.5488
      - - 1598886.4247
        - 5489464.0424
      - - 1597917.3387
        - 5489000
      - - 1597300
        - 5489000
      - - 1597682.1237
        - 5489972.7479
  run_params:
    description: test of notes
    duration: 259200
    write_tracks: false
  solver:
    n_sub_steps: 2
  tracks_writer:
    turn_on_write_particle_properties_list:
    - n_cell
reader:
  class_name: oceantracker.reader.generic_unstructured_reader.GenericUnstructuredReader
  dimension_map:
    node: nodes
    time: time
  field_variables:
    tide: tide
    water_depth: depth
    water_velocity:
    - east_vel
    - north_vel
  file_mask: demoHindcast2D*.nc
  grid_variables:
    time: time_sec
```

```
    triangles: tri
    x:
    - east
    - north
  input_dir: E:/H_Local_drive/ParticleTracking/oceantracker02/demos\demo_hindcast
  isodate_of_hindcast_time_zero: '2020-06-01'
  search_sub_dirs: true
  time_buffer_size: 24
shared_params:
  add_date_to_run_output_dir: false
  output_file_base: demo03_heatmaps
  root_output_dir: output
```

These demo plots and animations

- were created with standard postprocessing.plot methods
- views of parameters and code
- their code and parameters are in demos dir

Users guide

Installing

Requirements

Python version >= 3.7

../../../requirements.txt

```
numpy  < 1.22
matplotlib>=3.3.4
netcdf4>=1.5.6
setuptools>=52.0.0
psutil>=5.8.0
numba >= 0.5
pyproj>=3.0
scipy>=1.6.1
pyyaml >= 5.3.1
python-dateutil >= 2.8
```

Linux

1. Clone repository

   ```
   git clone https://gitlab.com/cinst/oceantracker02.git
   ```

2. **Working in a Virtual environment**

   Change dir to repository dir eg. oceantracker

3. Create virtual environment

   ```
   python3 -m venv venv
   ```

4. Activate venv

   ```
   source ./venv/bin/activate
   ```

5. Install packages in venv

   ```
   python setup.py develop pip install -r ./requirements.txt
   ```

6. Deactivate environment

   ```
   deactivate
   ```

Windows

Install Anaconda and create a conda virtual envioment with this, TDDO more details coming.

Running OceanTracker

Ways to run

OceanTracker can be run in two ways.

1. From command line using json or yaml parameterfile eg.

   ```
   python run_oceantracker.py --param_file my_params.json
   ```

   or

   ```
   python run_oceantracker.py --param_file my_params.yaml
   ```

   Then write code to do post processing and/or plotting, using oceantracker.post_processing.* or with user's own tools.

2. From python code by:

   - **creating a parameter dictionary by either:**
     - building a parameter dictionary in code, or
     - reading a json or yaml parameter file
   - passing the parameter dictionary to main.run() method
   - adding optional code for post processing and/or plotting using oceantracker.post_processing.*

Minimal example

Example to show the minmum required to run oceanTracker, using code to both run and plot, or from parameter file. For this 3D hindcast, grey particles are stranded on the bottom, green ones are stranded on the shore by the tide.

Run and plot using code:

../../../demos/minimal_example.py

```
# minimal_example.py
#-------------------
from oceantracker import main

# create parameters as a dictionary
params= {'shared_params' : {'output_file_base' :'minimal_example',
                            'root_output_dir':'output'},
    'reader': { 'class_name': 'oceantracker.reader.schism_reader.SCHSIMreaderNCDF',
                'input_dir': 'demo_hindcast',
                'file_mask': 'demoHindcastSchism3D.nc',
                },
    'base_case_params' : { 'solver': {'n_sub_steps': 12}, #not required but runs 5min steps
                'particle_release_groups':
                      [{'points': [[1595000, 5483300, -2],[1596000, 5487200, -2] ], # two 3D
                        'pulse_size': 10, 'release_interval': 3600}
                       ]
                           }
        }

# run OceanTracker to give track output files
runInfo_file_name, has_errors = main.run(params)

# output now in folder output/minimal_example

# below is optional code for plotting
```

```
#-------------------------------------
from oceantracker.post_processing.plotting.plot_tracks import animate_particles, plot_tracks
from oceantracker.post_processing.read_output_files.load_output_files import get_case_info_f

# find case_info_file name, to used to locate output for the caseInfo file
case_info_file_name = get_case_info_file_from_run_file(runInfo_file_name)

# read particle tracks for plotting
track_data = load_particle_track_vars(case_info_file_name)

# plot tracks
anim = plot_tracks(track_data, axis_lims=[1591000, 1601500, 5478500, 5491000],
                           plot_file_name='output\\minimal_example.jpeg')

# animate particles
anim = animate_particles(track_data, axis_lims=[1591000, 1601500, 5478500, 5491000],
                         title='Minimal example of OceanTracker with 3D point release',
                         movie_file='output\\minimal_example.mp4')
```

Run using parameter file:

```
python run_oceantracker.py --param_file minimal_example.yaml
```

../../../demos/demo_yaml/minimal_example.yaml

```
base_case_params:
  particle_release_groups:
  - points:
    - - 1595000
      - 5483300
      - -2
    - - 1596000
      - 5487200
      - -2
    pulse_size: 10
    release_interval: 3600
  solver:
    n_sub_steps: 12
reader:
  class_name: oceantracker.reader.schism_reader.SCHSIMreaderNCDF
  file_mask: demoHindcastSchism3D.nc
  input_dir: demo_hindcast
shared_params:
  output_file_base: minimal_example
  root_output_dir: output
```

Output is written to directory given set by shared_parameters, ie. a dir named

```
['root_output_dir']/['output_file_base']
```

in this example output is in

```
output/minimal_example
```

Run from parameter file

Using run_oceantracker.py, eg.

```
python run_oceantracker.py --param_file YAMLinputFiles\demo01_plot_tracks.yaml --inpu
t_dir demohindcast --root_output_dir output\demo01 --duration 3600
```

Command line parameters of run_oceantracker.py can override input and output dirs in the parameter file. Usage

```
python run_oceantracker.py --param_file  ./demos/demo02_animation.json   ( + options below)
```

```
usage: run_oceantracker.py [-h] [--param_file PARAM_FILE]
                           [--input_dir INPUT_DIR]
                           [--root_output_dir ROOT_OUTPUT_DIR]
                           [--processors PROCESSORS] [--replicates REPLICATES]
                           [--duration DURATION] [--cases CASES] [-debug]

optional arguments:
  -h, --help            show this help message and exit
  --param_file PARAM_FILE
                        json or yaml file of input parameters
  --input_dir INPUT_DIR
                        overrides dir for hindcast files given in param file
  --root_output_dir ROOT_OUTPUT_DIR
                        overrides root output dir given in param file
  --processors PROCESSORS
                        overrides number of processors in param file
  --replicates REPLICATES
                        overrides number of case replicates given in param
                        file
  --duration DURATION   in seconds, overrides model duration in seconds of all
                        of cases, useful in testing
  --cases CASES         only runs first "cases" of the case_list, useful in
                        testing
  -debug                gives better error information, but runs slower, eg
                        checks Numba array bounds
```

Example of parameters as yaml file , where indenting with spaces (not tabs) creates nested dictionaries.

../../../demos/demo_yaml/demo55_SCHISM_3D_fall_velocity.yaml

```yaml
base_case_params:
  dispersion:
    A_H: 0.2
    A_V: 0.001
  event_loggers:
  - class_name: oceantracker.event_loggers.log_polygon_entry_and_exit.LogPolygonEntryAndExit
    particle_prop_to_write_list:
    - ID
    - x
    - IDrelease_group
    - status
    - age
    polygon_list:
    - points:
      - - 1592682.1237
        - 5489972.7479
      - - 1593604.1667
        - 5490275.5488
      - - 1593886.4247
        - 5489464.0424
      - - 1592917.3387
        - 5489000.0
      - - 1592300.0
        - 5489000.0
      - - 1592682.1237
        - 5489972.7479
      user_polygon_name: A
    - points:
      - - 1597682.1237
        - 5489972.7479
      - - 1598604.1667
```

```
          - 5490275.5488
        - - 1598886.4247
          - 5489464.0424
        - - 1597917.3387
          - 5487000
        - - 1597300
          - 5487000
        - - 1597682.1237
          - 5489972.7479
        user_polygon_name: B
    particle_properties:
    - class_name: oceantracker.particle_properties.age_decay.AgeDecay
      decay_time_scale: 86400.0
    particle_release_groups:
    - points:
      - - 1594500
        - 5487000
        - -1
      - - 1594500
        - 5483000
        - -1
      - - 1598000
        - 5486100
        - -1
      pulse_size: 10
      release_interval: 3600
    - class_name: oceantracker.particle_release_groups.polygon_release.PolygonRelease
      points:
      - - 1597682.1237
        - 5489972.7479
      - - 1598604.1667
        - 5490275.5488
      - - 1598886.4247
        - 5489464.0424
      - - 1597917.3387
        - 5489000
      - - 1597300
        - 5489000
      - - 1597682.1237
        - 5489972.7479
      pulse_size: 10
      release_interval: 3600
      z_max: -4
      z_min: -2
    particle_statistics:
    - calculation_interval: 3600
      class_name: oceantracker.particle_statistics.gridded_statistics.GriddedStats2D_timeBased
      grid_size:
      - 120
      - 121
      particle_property_list:
      - water_depth
    run_params: {}
    solver:
      n_sub_steps: 6
    velocity_modifiers:
    - class_name: oceantracker.velocity_modifiers.terminal_velocity.AddTerminalVelocity
      mean: -0.0005
  reader:
    class_name: oceantracker.reader.schism_reader.SCHSIMreaderNCDF
```

```
  depth_average: false
  field_variables:
    water_temperature: temp
  field_variables_to_depth_average:
  - water_velocity
  - water_depth
  - salt
  - water_temperature
  file_mask: demoHindcastSchism3D.nc
  input_dir: E:/H_Local_drive/ParticleTracking/oceantracker02/demos\demo_hindcast
shared_params:
  debug: true
  output_file_base: demo55_SCHISM_3D_fall_velocity
  root_output_dir: output
```

Example demos\JSONinputFiles\demo01_plot_tracks.json

../../../demos/demo_json/demo01_plot_tracks.json

```
{"shared_params": {"output_file_base": "demo01_plot_tracks", "add_date_to_run_output_dir": f
```

Run from code

Below is example of building a dictionary of parameters, running particle tracking and plotting:

../../../demos/run_from_code_demo.py

```
# run oceantracker direct from code from dictionary built in code
# make polygons staggered to south west, by appending polygon release groups
import numpy as np
from main import run
from oceantracker.post_processing.read_output_files import load_output_files
from oceantracker.post_processing.plotting.plot_tracks import plot_tracks

params={
'shared_params' :{
                  'output_file_base' :'demo1000_runFromCodeDemo',
                  'root_output_dir': 'output', 'debug': True,
                  'backtracking': False},
 'reader':  { "class_name": 'oceantracker.reader.generic_unstructured_reader.GenericUnstruct
              'input_dir': 'demo_hindcast',
              'file_mask': 'demoHindcast2D*.nc',
              'search_sub_dirs': True,
              'dimension_map': {'time': 'time', 'z': None,'node': 'nodes'},
                'grid_variables': {'time': 'time_sec',
                                   'x': [ 'east', 'north'],
                                  'triangles': 'tri'},
                'field_variables' :{'water_depth' : 'depth', 'water_velocity': ['east_vel','
                'time_buffer_size': 24, 'isodate_of_hindcast_time_zero': '2000-01-01'} ,
 'base_case_params' : {
    'run_params' : {},
    'dispersion': {'A_H': 1.},
    'solver': {'n_sub_steps': 12},
        'particle_release_groups': [],
                                  }
 }

poly_points=np.asarray([[1597682.1237, 5489972.7479],
                        [1598604.1667, 5490275.5488],
                        [1598886.4247, 5489464.0424],
                        [1597917.3387, 5489000],
                        [1597300, 5489000], [1597682.1237, 5489972.7479]])
```

```
# make polygons staggered to south west, by appending polygon release groups
for n in range(4):
    points=poly_points+np.asarray([[-1050*n,-2100*n]])
    params['base_case_params']['particle_release_groups'].append({'class_name': 'oceantracke
                                                                  'points': points.tolist(),

runInfo_file_name, has_errors = run(params)
caseInfoFile = load_output_files.get_case_info_file_from_run_file(runInfo_file_name)
plot_tracks.animate_particles(caseInfoFile, axis_lims=[1590800, 1601800, 5478700, 5491200],
                              title='Run from code demo, staggered polygons', back_ground_
```

Parameters overview

OceanTracker is structured to run one more “cases” of particle tracking, each with their own set of parameters. The parameters are a python dictionary of key-value pairs (parameters in JSON or Yaml file are converted to a python dictionary). Thus key-value pairs of parameters setup the computations. The top level parameter dictionary, may contain other nested parameter dictionaries, or lists of other parameter dictionaries.

Top level params

The top level parameter dictionary has the structure:

```
{   'shared_params': { 'root_output_dir': 'my_output', .... } # common to all cases
    'reader': {'class_name':,.... } ,           # shared hindcast reader
    'base_case_params': {},                     # user given defaults for all cases
    'case_list': [],                            # list of individual case parameters
 }
```

- ‘shared_params’ are parameters used for all cases
- ‘reader’ are parameters for the hindcast reader, which reads variables from a set of hindcast files and is shared by all cases, eg input_dir and file_mask
- ‘base_case_params’ are parameters common to all cases,
- ‘case_list’ are parameters unique to each individual case which will also overwrite any also given in base_case_params.

For single case parameter tracking, the user only needs to set ‘base_case_params’. OceanTracker will automatically create a case with these parameters. A case list allows multiple cases to be run at the same time. Theses case will be run in in parallel if shared_param ‘processors’ > 1, otherwise they will run in serial.

Case level params

Both ‘base_case_params’ and those within the ‘case_list’ have the exactly the same parameter structure.

```
{    'run_params' : {}
    # core classes
   'solver': {},
   'field_group_manager':{} ,
   'interpolator': {} ,
   'particle_group_manager': {},
   'tracks_writer':   {},
   'dispersion':      {},
   'particle_release_groups':  [],  # required
 # below are optional user classes held in named lists
   'fields':          [],  # prop calculated from other fields  on reading
   'particle_properties':     [], # user added particle properties, eg DistanceTraveled
   'velocity_modifiers':      [], # user added velocity effects, eg TerminalVelocity
   'trajectory_modifiers':    [], # change particle paths, eg. re-suspension
   'particle_statistics': [],# heat map inside polygon statitics calculated on the fly
   'event_loggers':           [], # writes events files ,eg PolygonEntryExit
```

```
    'particle_concentrations': [], # writes concentration of particles within triangles of th
                                   # and other properties calculated on the fly.   files ,eg
    }
```

- ‘run_params’ are case specific parameters, eg ‘duration’
- the next block are core classes parameters, eg set turbulent eddy viscosity in random walk ‘dispersion’
- ‘particle_release_groups’ is required. It is a list of parameters detailing where and when particles are released from points or within a polygon. The list allows particles from each release group to be followed separately, eg. counts in statistics are separated by release group.
- The following are optional lists of parameters for classes which create new particle properties, calculate statistics, or modfiy how the particles move

Defaults

All parameters, at top level or lower level, or within lists, are dictionary like, ie. key-value pairs. A value which is a dictionary is treated as a nested parameter dictionary of key-value pairs.

The values at the ends of any top level dictionary or nested parameter dictionary have specified default values associated with their key, which are used if none is provide by the user.

Checking

Any user given values are checked to ensure they are the required data type, are in the range of acceptable values, etc. For a few keys, a value must be supplied by the user (eg. input_dir of hindcast files). If these are not given an “is_required error” will be generated.

The user is warned of any unexpected keys found within the parameters, ie those not in the default parameter dictionary.

Full details of defaults are given in ….??

class_name parameter

To give flexibility the classes which make up the particle tracking computational pipeline are added via their class_name parameter given as a string, eg.

`"class_name": "oceantracker.particle_release_groups.polygon_release.PolygonRelease"`

which is used to import the class. This enables users to freely add the capabilities needed. In this case there might be releases from several different polygons, each with their individual release rates etc, each added by their parameters in the particle_release_groups list.

Adding via a string also allows users to create their own variants of the classes, which inherit most of their functionality from their parent class, but are tweaked to have behaviours or functionality to suit the user. These new particle properties are added to the computational pipeline with their parameters. Provided they are in the current working dir or accessible via the python path, these user added classes, along with all other classes, are imported using their class_name string.

An example below is the class which adds the ‘AgeDecay’ particle property. The important method is its update() which calculates the current amount of the property remaining based on the core particle property ‘age’ and exponential decay. Age is part of the same computational pipeline as AgeDecay, like a particle’s location property ‘x’, another core property. Core particle properties are updated before those added by the user.

The mechanics of accessing AgeDecay’s data and writing it to the output file are done with methods inherited from the ParticleProperty base class `oceantracker.particle_properties._base_properties.ParticleProperty`, all overseen by the core ParticleGroupManager class.

Once the ‘AgeDecay’ class is imported, its parameters are merged with the defaults, it is then initilaized and added to the computational pipeline by the ParticleGroupManager. Then it will automatically:

- be updated each time step
- be written to the output file along with ‘x’ and other properties (unless its ‘write’ parameter is False)

- if requested, have statistics such as heat maps automatically calculated and written to output files on the fly
- its values can be access by other particle properties, which may depend on it (eg. age_decay particle property depends on the 'age' property)

../../../oceantracker/particle_properties/age_decay.py

```
from oceantracker.particle_properties._base_properties import ParticleProperty
import numpy as np
from oceantracker.util.parameter_checking import ParamDictValueChecker as PVC
class AgeDecay(ParticleProperty):

    def __init__(self):
        super().__init__()
        self.add_default_params({'name': PVC('age_decay', str) ,
                                 'description': PVC( 'decaying particle based on age', str),
                                 'initial_value': PVC(1., float),
                                 'decay_time_scale': PVC( 1.*3600*24, float)})

    def check_requirements(self):
        msg_list = self.check_class_required_fields_properties_grid_vars_and_3D(required_pro
        return msg_list


    def initial_value_at_birth(self, new_part_IDs):
        self.set_values(self.params['initial_value'], new_part_IDs) # sets this properties v

    def update(self,active):
        # update decay prop each time step
        age = self.shared_info.classes['particle_properties']['age'].get_values(active)
        val = self.params['initial_value']*np.exp(-np.abs(age) / self.params['decay_time_sca
        self.set_values(val,active)
```

Output files

Output location

Output is written to directory given by shared_parameters, ie. a dir named

`['root_output_dir']/['output_file_base']`

eg. `output/minimal_example`

Note: To make it easier to keep output from different runs separate, the date can optionally be added to the dir name by setting shared_params 'add_date_to_run_output_dir' to be True

File names

All files start with the given parameter ['output_file_base'] ie. files have form

`['root_output_dir']/['output_file_base']/['output_file_base']*.*`

eg. `output/minimal_example/minimal_example*.*`

If more than one case is run or there are replicate runs requested, files are tagged by case number and replicate number, eg.

`output/minimal_example/minimal_example_C010R02*.*`

File types and roles

_runInfo.json

Records overall information about the run, such as

- run start and ends times
- **all output file names used to locate them to read for post processing, eg.**
    - the dir which has the output “run_output_dir”
    - name of hindcast grid file netCDF to be used for plotting
    - a list of file names of the _caseInfo.json files, which hold the details of each case
- copy of user supplied parameters
- over all performance across all cases
- ocean tracker version
- basic info on computer running code

_runLog.txt

Has the screen output associated with main.rum(), ie lines starting with “M” on the screen. This covers setup progress, parameter checking, ….

_caseInfo.json

Holds useful details about each case run

- case run start and ends times
- hindcast details, eg, start end time step, …
- **all output file names to locate them to read for post processing, eg.**
    - the dir which has the output “run_output_dir”
    - name of hindcast grid file netCDF to be used for plotting
    - particle track output files
    - lists of file names of output files from the outputs on each particle_statistics, particle_concentrations, event_loggers etc, add to computational pile line
- full set of parameters used with all defaults
- over all timing performance with in some sections of
- **part information**
    - map from namt to numerical values of particle_status_flags which as recorded in the output
- basic info on computer running code

_caseLog.txt

Has the screen output associated with each individual case, starting with `P001:` etc.

_grid.nc

The hindcast grid as netCDF file, used for plotting.

_tracks.nc

The recorded particle tracks and particle propeties in a netCDF file.

stats??_nnn.nc

Output as netCDF file, from each class added to list of particle_statistics, nnn is the sequence number as added to list, eg. `??_stats_gridded_age_001.nc`

_events_nnn.nc

Output as netCDF file, from each class added to list of event_loggers, nnn is the sequence number as added to list, eg. `??_events_001.nc`

_concentrations_nnn.nc

Output as netCDF file, from each class added to list of particle_concentrations, nnn is the sequence number as added to list, eg. `??_concentrations_001.nc`

Post processing

todo more work here!

Reading output files

Read NetCDF files methods

`post_processing.read_output_files.read_ncdf_output_files` module

Read case_info, NetCDF and grid methods

`post_processing.read_output_files.load_output_files` module

Plotting

`post_processing.plotting` module

Parameter details

Links to details of parameter default values, acceptable values etc.

> **Note**
>
> Lots more to add here!!

Top level parameters

shared_params

> **Warning**
>
> Lots more to add here and work on layout!!

Parameters:

- `add_date_to_run_output_dir`: ***<optional>***

    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `backtracking`**: *<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `compact_mode`**: *<optional>***

  Description: - Periodically discard dead particles from memory, eg. those too old to be be of interest, if used track output file also has a compact format
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `debug`**: *<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `max_duration`**: *<optional>***
    - type: `<class 'float'>`
    - default: `1e+20`
- `max_warnings`**: *<optional>***
    - type: `<class 'int'>`
    - default: `50`
    - min: `0`
- `multiprocessing_case_start_delay`**: *<optional>***
    - type: `<class 'float'>`
    - default: `0.0`
    - min: `0.0`
- `numba_function_cache_size`**: *<optional>***
    - type: `<class 'int'>`
    - default: `512`
    - min: `128`
- `output_file_base`**: *<optional>***
    - type: `<class 'str'>`
    - default: `default_output_file_base`
- `processors`**: *<optional>***

Description: - number of processors used, if > 1 then cases in the case_list run in parallel

- type: `<class 'int'>`
- default: `1`
- min: `1`

- `replicates`**: *<optional>***

  Description: - number of replicates of each case to run, allows running larger particle numbers for each case in less time if running in parallel

  - type: `<class 'int'>`
  - default: `1`
  - min: `1`

- `root_output_dir`**: *<optional>***

  Description: - base dir for all ouput files

  - type: `<class 'str'>`
  - default: `default_root_output_dir`

- `use_numpy_random_seed`**: *<optional>***

  Description: - Makes results reproducible, only use for testing developments give the same results!

  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`

- `write_grid`**: *<optional>***

  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

- `write_output_files`**: *<optional>***

  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

reader

Module: oceantracker.reader

GenericUnstructuredReader

Class: oceantracker.reader.generic_unstructured_reader.GenericUnstructuredReader

File: oceantracker/reader/generic_unstructured_reader.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`**: *<optional>***

  - type: `<class 'str'>`
  - default: `None`

- `cords_in_lat_long`**: *<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `depth_average`**: *<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`

dimension_map: still working on display of default params of type <class ‘dict’>

field_variables: still working on display of default params of type <class ‘dict’>

field_variables_to_depth_average: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

<isrequired>

- type: `<class 'str'>`
- default: `None`

grid_variables: still working on display of default params of type <class ‘dict’>

- `input_dir`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `isodate_of_hindcast_time_zero`**: *<optional>***
    - type: `iso8601date`
    - default: `1970-01-01`
- `max_numb_files_to_load`**: *<optional>***
    - type: `<class 'int'>`
    - default: `10000000`
    - min: `1`
- `minimum_total_water_depth`**: *<optional>***

  Description: - Min. water depth used to decide if stranded by tide and which are dry cells to block particles from entering
    - type: `<class 'float'>`
    - default: `0.25`
    - min: `0.05`
- `name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `search_sub_dirs`**: *<optional>***

- type: `<class 'bool'>`
- default: `False`
- possible_values: `[True, False]`

- `time_buffer_size`**: *<optional>***
  - type: `<class 'int'>`
  - default: `48`
  - min: `2`
- `time_zone`**: *<optional>***
  - type: `<class 'int'>`
  - default: `None`
  - min: `-12`
  - max: `23`

SCHSIMreaderNCDF

Class: oceantracker.reader.schism_reader.SCHSIMreaderNCDF

File: oceantracker/reader/schism_reader.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `cords_in_lat_long`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `depth_average`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

dimension_map: still working on display of default params of type <class ‘dict’>

field_variables: still working on display of default params of type <class ‘dict’>

field_variables_to_depth_average: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

<isrequired>

- type: `<class 'str'>`
- default: `None`

grid_variables: still working on display of default params of type <class 'dict'>

- `hgrid_file_name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `input_dir`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `isodate_of_hindcast_time_zero`**: *<optional>***
    - type: `iso8601date`
    - default: `1970-01-01`
- `max_numb_files_to_load`**: *<optional>***
    - type: `<class 'int'>`
    - default: `10000000`
    - min: `1`
- `minimum_total_water_depth`**: *<optional>***

  Description: - Min. water depth used to decide if stranded by tide and which are dry cells to block particles from entering
    - type: `<class 'float'>`
    - default: `0.25`
    - min: `0.05`
- `name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `search_sub_dirs`**: *<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `time_buffer_size`**: *<optional>***
    - type: `<class 'int'>`
    - default: `48`
    - min: `2`
- `time_zone`**: *<optional>***
    - type: `<class 'int'>`
    - default: `None`
    - min: `-12`
    - max: `23`
- `water_velocity_depth_average`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`

BaseReader

Class: oceantracker.reader._base_reader.BaseReader

File: oceantracker/reader/_base_reader.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `cords_in_lat_long`**: *<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `depth_average`**: *<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`

dimension_map: still working on display of default params of type <class 'dict'>

field_variables: still working on display of default params of type <class 'dict'>

field_variables_to_depth_average: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

<isrequired>

- type: `<class 'str'>`
- default: `None`

grid_variables: still working on display of default params of type <class 'dict'>

- `input_dir`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `isodate_of_hindcast_time_zero`**: *<optional>***
    - type: `iso8601date`
    - default: `1970-01-01`
- `max_numb_files_to_load`**: *<optional>***
    - type: `<class 'int'>`
    - default: `10000000`
    - min: `1`

- `minimum_total_water_depth`: ***<optional>***

  Description: - Min. water depth used to decide if stranded by tide and which are dry cells to block particles from entering

  - type: `<class 'float'>`
  - default: `0.25`
  - min: `0.05`
- `name`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `search_sub_dirs`: ***<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `time_buffer_size`: ***<optional>***
  - type: `<class 'int'>`
  - default: `48`
  - min: `2`
- `time_zone`: ***<optional>***
  - type: `<class 'int'>`
  - default: `None`
  - min: `-12`
  - max: `23`

Case parameters

run_params

Warning

Lots more to add here and work on layout!!

Parameters:

- `case_output_file_tag`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `description`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `duration`: ***<optional>***

- type: `<class 'float'>`
- default: `1e+300`

- `open_boundary_type`**: *<optional>***
  - type: `<class 'int'>`
  - default: `0`
  - min: `0`
  - max: `1`
- `particle_buffer_size`**: *<optional>***
  - type: `<class 'int'>`
  - default: `None`
  - min: `1`
- `retain_culled_part_locations`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `write_tracks`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`
- `z0`**: *<optional>***
  - type: `<class 'float'>`
  - default: `0.005`
  - min: `0.0001`

Core classes

dispersion

Module: oceantracker.dispersion

RandomWalk

Class: oceantracker.dispersion.random_walk.RandomWalk

File: oceantracker/dispersion/random_walk.py

Default internal name: `"random_walk"`

Description:

Parameters:

- `A_H`**: *<optional>***
  - type: `<class 'float'>`
  - default: `1.0`
  - min: `0.0`

- `A_V`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.001`
    - min: `0.0`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `is3D`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `random_walk`

BaseTrajectoryModifer

Class: oceantracker.dispersion._base_dispersion.BaseTrajectoryModifer

File: oceantracker/dispersion/_base_dispersion.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `is3D`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

field_group_manager

Module: oceantracker.field_group_manager

FieldGroupManager

Class: oceantracker.field_group_manager.field_group_manager.FieldGroupManager

File: oceantracker/field_group_manager/field_group_manager.py

Default internal name: `"field_group_manager"`

Description:

Parameters:

- `class_name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `field_group_manager`

interpolator

Module: oceantracker.interpolator

InterpTriangularNativeGrid_Slayer_and_LSCgrid

Class: oceantracker.interpolator.interp_triangle_native_grid.InterpTriangularNativeGrid_Slayer_and_LSCgrid

File: oceantracker/interpolator/interp_triangle_native_grid.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `bc_walk_tol`**: *<optional>***
    - type: `<class 'float'>`
    - default: `1e-06`
    - min: `0.0`
- `class_name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`

- `max_search_steps`: ***<optional>***
    - type: `<class 'int'>`
    - default: `100`
    - min: `1`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

BaseInterp

Class: oceantracker.interpolator._base_interp.BaseInterp

File: oceantracker/interpolator/_base_interp.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

particle_group_manager

Module: oceantracker.particle_group_manager

ParticleGroupManager

Class: oceantracker.particle_group_manager.particle_group_manager.ParticleGroupManager

File: oceantracker/particle_group_manager/particle_group_manager.py

Default internal name: `"particle_group_manager"`

Description:

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***

    - type: `<class 'str'>`
    - default: `None`
- `max_age`**: *<optional>***
    - type: `<class 'float'>`
    - default: `100000000000.0`
    - min: `0.0`
- `name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `particle_group_manager`

solver

Module: oceantracker.solver

Solver

Class: oceantracker.solver.solver.Solver

File: oceantracker/solver/solver.py

Default internal name: `"solver"`

Description:

Parameters:

- `RK_order`**: *<optional>***
    - type: `<class 'int'>`
    - default: `4`
    - possible_values: `[1, 2, 4]`
- `class_name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `n_sub_steps`**: *<optional>***
    - type: `<class 'int'>`
    - default: `1`
    - min: `1`
- `name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `solver`
- `screen_output_step_count`**: *<optional>***
    - type: `<class 'int'>`
    - default: `1`

tracks_writer

Module: oceantracker.tracks_writer

FlatTrackWriter

Class: oceantracker.tracks_writer.track_writer_compact.FlatTrackWriter

File: oceantracker/tracks_writer/track_writer_compact.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `NCDF_time_chunk`**: *<optional>***
    - type: `<class 'int'>`
    - default: `24`
    - min: `1`
- `case_output_file_tag`**: *<optional>***
    - type: `<class 'str'>`
    - default: `tracks`
- `class_name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `output_step_count`**: *<optional>***
    - type: `<class 'int'>`
    - default: `1`
    - min: `1`

turn_off_write_particle_properties_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

turn_on_write_particle_properties_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

RectangularTrackWriter

Class: oceantracker.tracks_writer.track_writer_retangular.RectangularTrackWriter

File: oceantracker/tracks_writer/track_writer_retangular.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `NCDF_time_chunk`: ***<optional>***
    - type: `<class 'int'>`
    - default: `24`
    - min: `1`
- `case_output_file_tag`: ***<optional>***
    - type: `<class 'str'>`
    - default: `tracks`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `output_step_count`: ***<optional>***
    - type: `<class 'int'>`
    - default: `1`
    - min: `1`

turn_off_write_particle_properties_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

turn_on_write_particle_properties_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

BaseWriter

Class: oceantracker.tracks_writer._base_tracks_writer.BaseWriter

File: oceantracker/tracks_writer/_base_tracks_writer.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `case_output_file_tag`: ***<optional>***
    - type: `<class 'str'>`
    - default: `tracks`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `output_step_count`: ***<optional>***
    - type: `<class 'int'>`
    - default: `1`
    - min: `1`

turn_off_write_particle_properties_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

turn_on_write_particle_properties_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

User added classes

event_loggers

Module: oceantracker.event_loggers

LogPolygonEntryAndExit

Class: oceantracker.event_loggers.log_polygon_entry_and_exit.LogPolygonEntryAndExit

File: oceantracker/event_loggers/log_polygon_entry_and_exit.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `case_output_file_tag`: ***<optional>***
    - type: `<class 'str'>`
    - default: `inside_polygon_events`
- `chunk_size`: ***<optional>***
    - type: `<class 'int'>`
    - default: `5000`
    - min: `1`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

particle_prop_to_write_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

polygon_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `write`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

BaseEventLogger

Class: oceantracker.event_loggers._base_event_loggers.BaseEventLogger

File: oceantracker/event_loggers/_base_event_loggers.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `case_output_file_tag`: ***<optional>***
    - type: `<class 'str'>`
    - default: `event_logger`
- `chunk_size`: ***<optional>***
    - type: `<class 'int'>`
    - default: `5000`
    - min: `1`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

particle_prop_to_write_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `write`: ***<optional>***
    - type: `<class 'bool'>`

- default: `True`
- possible_values: `[True, False]`

fields

Module: oceantracker.fields

TotalWaterDepth

Class: oceantracker.fields.total_water_depth.TotalWaterDepth

File: oceantracker/fields/total_water_depth.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `create_particle_property_with_same_name`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `dtype`**: *<optional>***
  - type: `<class 'numpy.dtype'>`
  - default: `<class 'numpy.float64'>`
- `is3D_in_file`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `is_time_varying`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

<isrequired>

- type: `<class 'str'>`
- default: `None`

- `num_components`**: *<optional>***
  - type: `<class 'int'>`

- default: None
- `reader_variable_list`: ***<optional>***
  - type: <class 'str'>
  - default: None
- `requires_depth_averaging`: ***<optional>***
  - type: <class 'bool'>
  - default: False
  - possible_values: [True, False]

BaseField

Class: oceantracker.fields._base_field.BaseField

File: oceantracker/fields/_base_field.py

Default internal name: "not given in defaults"

Description:

Parameters:

- `class_name`: ***<optional>***
  - type: <class 'str'>
  - default: None
- `create_particle_property_with_same_name`: ***<optional>***
  - type: <class 'bool'>
  - default: True
  - possible_values: [True, False]
- `description`: ***<optional>***
  - type: <class 'str'>
  - default: None
- `dtype`: ***<optional>***
  - type: <class 'numpy.dtype'>
  - default: <class 'numpy.float64'>
- `is3D_in_file`: ***<optional>***
  - type: <class 'bool'>
  - default: False
  - possible_values: [True, False]
- `is_time_varying`: ***<optional>***
  - type: <class 'bool'>
  - default: True
  - possible_values: [True, False]

<isrequired>

- type: <class 'str'>
- default: None

- `num_components`: ***<optional>***
    - type: `<class 'int'>`
    - default: `None`
- `reader_variable_list`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `requires_depth_averaging`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`

UserFieldBase

Class: oceantracker.fields._base_field.UserFieldBase

File: oceantracker/fields/_base_field.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `create_particle_property_with_same_name`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `dtype`: ***<optional>***
    - type: `<class 'numpy.dtype'>`
    - default: `<class 'numpy.float64'>`
- `is3D_in_file`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `is_time_varying`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

<isrequired>

- type: `<class 'str'>`
- default: `None`

- `num_components`**: *<optional>***
  - type: `<class 'int'>`
  - default: `None`
- `reader_variable_list`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `requires_depth_averaging`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`

particle_concentrations

Module: oceantracker.particle_concentrations

ParticleConcentrations2D

Class: oceantracker.particle_concentrations.particle_concentrations.ParticleConcentrations2D

File: oceantracker/particle_concentrations/particle_concentrations.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `calculation_interval`**: *<optional>***
  - type: `<class 'float'>`
  - default: `1`
  - min: `1`
- `case_output_file_tag`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

<isrequired>

- type: `<class 'str'>`
- default: `None`

- `count_status_equal_to`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`**: *<optional>***

- type: `<class 'str'>`
- default: `None`

- `name`**: *<optional>***

- type: `<class 'str'>`
- default: `None`

- `only_update_concentrations_on_write`**: *<optional>***

- type: `<class 'bool'>`
- default: `False`
- possible_values: `[True, False]`

- `output_step_count`**: *<optional>***

- type: `<class 'int'>`
- default: `1`
- min: `1`

particle_properties_to_track: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `release_group_to_track`**: *<optional>***

- type: `<class 'int'>`
- default: `None`
- min: `0`

- `write`**: *<optional>***

- type: `<class 'bool'>`
- default: `True`
- possible_values: `[True, False]`

BaseTriangleProperties

Class: oceantracker.particle_concentrations._base_user_triangle_properties.BaseTriangleProperties

File: oceantracker/particle_concentrations/_base_user_triangle_properties.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `calculation_interval`**: *<optional>***

- type: `<class 'float'>`
- default: `1`
- min: `1`

- `case_output_file_tag`**: *<optional>***

- type: `<class 'str'>`
- default: `None`

<isrequired>

- type: `<class 'str'>`

- default: None

- count_status_equal_to: ***<optional>***
  - type: <class 'str'>
  - default: None
  - possible_values: dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',
- description: ***<optional>***
  - type: <class 'str'>
  - default: None
- name: ***<optional>***
  - type: <class 'str'>
  - default: None
- only_update_concentrations_on_write: ***<optional>***
  - type: <class 'bool'>
  - default: False
  - possible_values: [True, False]
- output_step_count: ***<optional>***
  - type: <class 'int'>
  - default: 1
  - min: 1

particle_properties_to_track: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- release_group_to_track: ***<optional>***
  - type: <class 'int'>
  - default: None
  - min: 0
- write: ***<optional>***
  - type: <class 'bool'>
  - default: True
  - possible_values: [True, False]

particle_properties

Module: oceantracker.particle_properties

AgeDecay

Class: oceantracker.particle_properties.age_decay.AgeDecay

File: oceantracker/particle_properties/age_decay.py

Default internal name: "age_decay"

Description: decaying particle based on age

Parameters:

- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `decay_time_scale`**: *<optional>***
  - type: `<class 'float'>`
  - default: `86400.0`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `decaying particle based on age`
- `dtype`**: *<optional>***
  - type: `<class 'type'>`
  - default: `<class 'numpy.float64'>`
  - possible_values: `[<class 'numpy.float32'>, <class 'numpy.float64'>, <class 'numpy.int8'>, <clas`
- `initial_value`**: *<optional>***
  - type: `<class 'float'>`
  - default: `1.0`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `age_decay`
- `prop_dim3`**: *<optional>***
  - type: `<class 'int'>`
  - default: `1`
  - min: `1`
- `time_varying`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`
- `type`**: *<optional>***

  Description: - particle property
  - type: `<class 'str'>`
  - default: `user`
  - possible_values: `['manual_update', 'from_fields', 'user']`
- `update`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`
- `vector_dim`**: *<optional>***

    - type: `<class 'int'>`
    - default: `1`
    - min: `1`
- `write`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

DistanceTravelled

Class: oceantracker.particle_properties.distance_travelled.DistanceTravelled

File: oceantracker/particle_properties/distance_travelled.py

Default internal name: `"distance_travelled"`

Description:

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `dtype`: ***<optional>***
    - type: `<class 'type'>`
    - default: `<class 'numpy.float64'>`
    - possible_values: `[<class 'numpy.float32'>, <class 'numpy.float64'>, <class 'numpy.int8'>, <class`
- `initial_value`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `distance_travelled`
- `prop_dim3`: ***<optional>***
    - type: `<class 'int'>`
    - default: `1`
    - min: `1`
- `time_varying`: ***<optional>***
    - type: `<class 'bool'>`
    - possible_values: `[True, False]`
    - default: `True`

- `type`: ***<optional>***

  Description: - particle property

  - type: `<class 'str'>`
  - default: `user`
  - possible_values: `['manual_update', 'from_fields', 'user']`
- `update`: ***<optional>***

  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`
- `vector_dim`: ***<optional>***

  - type: `<class 'int'>`
  - default: `1`
  - min: `1`
- `write`: ***<optional>***

  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

FrictionVelocity

Class: oceantracker.particle_properties.friction_velocity.FrictionVelocity

File: oceantracker/particle_properties/friction_velocity.py

Default internal name: `"friction_velocity"`

Description:

Parameters:

- `class_name`: ***<optional>***

  - type: `<class 'str'>`
  - default: `None`
- `description`: ***<optional>***

  - type: `<class 'str'>`
  - default: `None`
- `dtype`: ***<optional>***

  - type: `<class 'type'>`
  - default: `<class 'numpy.float64'>`
  - possible_values:
    `[<class 'numpy.float32'>, <class 'numpy.float64'>, <class 'numpy.int8'>, <clas`
- `initial_value`: ***<optional>***

  - type: `<class 'float'>`
  - default: `0.0`
- `name`: ***<optional>***

- type: `<class 'str'>`
- default: `friction_velocity`

- `prop_dim3`**: *<optional>***
  - type: `<class 'int'>`
  - default: `1`
  - min: `1`
- `time_varying`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`
- `type`**: *<optional>***
  Description: - particle property
  - type: `<class 'str'>`
  - default: `user`
  - possible_values: `['manual_update', 'from_fields', 'user']`
- `update`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`
- `vector_dim`**: *<optional>***
  - type: `<class 'int'>`
  - default: `1`
  - min: `1`
- `write`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

InsidePolygonsNonOverlapping2D

Class: oceantracker.particle_properties.inside_polygons.InsidePolygonsNonOverlapping2D

File: oceantracker/particle_properties/inside_polygons.py

Default internal name: `"inside_polygons_non_overlapping"`

Description: index of polygon particle is in side

Parameters:

- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `description`**: *<optional>***
  - type: `<class 'str'>`

- default: `index of polygon particle is in side`

- `dtype`**: *<optional>***
  - type: `<class 'type'>`
  - default: `<class 'numpy.int32'>`
- `initial_value`**: *<optional>***
  - type: `<class 'int'>`
  - default: `-1`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `inside_polygons_non_overlapping`

polygon_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `prop_dim3`**: *<optional>***
  - type: `<class 'int'>`
  - default: `1`
  - min: `1`
- `time_varying`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`
- `type`**: *<optional>***

  Description: - particle property
  - type: `<class 'str'>`
  - default: `user`
  - possible_values: `['manual_update', 'from_fields', 'user']`
- `update`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`
- `vector_dim`**: *<optional>***
  - type: `<class 'int'>`
  - default: `1`
  - min: `1`
- `write`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

ParticleParameterFromNormalDistribution

Class: oceantracker.particle_properties.particle_parameter_from_normal_distribution.ParticleParameterFromNormalDistribution

File: oceantracker/particle_properties/particle_parameter_from_normal_distribution.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `dtype`**: *<optional>***
  - type: `<class 'type'>`
  - default: `<class 'numpy.float64'>`
  - possible_values: `[<class 'numpy.float32'>, <class 'numpy.float64'>, <class 'numpy.int8'>, <clas`
- `initial_value`**: *<optional>***
  - type: `(<class 'int'>, <class 'float'>, <class 'bool'>)`
  - default: `0.0`

<isrequired>

  - type: `<class 'float'>`
  - default: `0.0`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `prop_dim3`**: *<optional>***
  - type: `<class 'int'>`
  - default: `1`
  - min: `1`
- `time_varying`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `type`**: *<optional>***

  Description: - particle property
  - type: `<class 'str'>`
  - default: `user`
  - possible_values: `['manual_update', 'from_fields', 'user']`
- `update`**: *<optional>***

- type: `<class 'bool'>`
- default: `True`
- possible_values: `[True, False]`

<isrequired>

- type: `<class 'float'>`
- default: `0.0`
- min: `0.0`

- `vector_dim`**: *<optional>***
  - type: `<class 'int'>`
  - default: `1`
  - min: `1`
- `write`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

BasePropertyInfo

Class: oceantracker.particle_properties._base_properties.BasePropertyInfo

File: oceantracker/particle_properties/_base_properties.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `dtype`**: *<optional>***
  - type: `<class 'type'>`
  - default: `<class 'numpy.float64'>`
  - possible_values: `[<class 'numpy.float32'>, <class 'numpy.float64'>, <class 'numpy.int8'>, <class`
- `initial_value`**: *<optional>***
  - type: `(<class 'int'>, <class 'float'>, <class 'bool'>)`
  - default: `0.0`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

- `prop_dim3`: ***<optional>***
    - type: `<class 'int'>`
    - default: `1`
    - min: `1`
- `time_varying`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`
- `update`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`
- `vector_dim`: ***<optional>***
    - type: `<class 'int'>`
    - default: `1`
    - min: `1`
- `write`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

ParticleProperty

Class: oceantracker.particle_properties._base_properties.ParticleProperty

File: oceantracker/particle_properties/_base_properties.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `dtype`: ***<optional>***
    - type: `<class 'type'>`
    - default: `<class 'numpy.float64'>`
    - possible_values: `[<class 'numpy.float32'>, <class 'numpy.float64'>, <class 'numpy.int8'>, <clas`
- `initial_value`: ***<optional>***

        - type: (<class 'int'>, <class 'float'>, <class 'bool'>)
        - default: 0.0
- name: ***<optional>***
        - type: <class 'str'>
        - default: None
- prop_dim3: ***<optional>***
        - type: <class 'int'>
        - default: 1
        - min: 1
- time_varying: ***<optional>***
        - type: <class 'bool'>
        - default: True
        - possible_values: [True, False]
- type: ***<optional>***

    Description: - particle property
        - type: <class 'str'>
        - default: user
        - possible_values: ['manual_update', 'from_fields', 'user']
- update: ***<optional>***
        - type: <class 'bool'>
        - default: True
        - possible_values: [True, False]
- vector_dim: ***<optional>***
        - type: <class 'int'>
        - default: 1
        - min: 1
- write: ***<optional>***
        - type: <class 'bool'>
        - default: True
        - possible_values: [True, False]

TimeVaryingInfo

Class: oceantracker.particle_properties._base_properties.TimeVaryingInfo

File: oceantracker/particle_properties/_base_properties.py

Default internal name: "not given in defaults"

Description:

Parameters:

- class_name: ***<optional>***

        - type: `<class 'str'>`
        - default: `None`
- `description`: ***<optional>***
        - type: `<class 'str'>`
        - default: `None`
- `dtype`: ***<optional>***
        - type: `<class 'type'>`
        - default: `<class 'numpy.float64'>`
        - possible_values: `[<class 'numpy.float32'>, <class 'numpy.float64'>, <class 'numpy.int8'>, <clas`
- `initial_value`: ***<optional>***
        - type: `(<class 'int'>, <class 'float'>, <class 'bool'>)`
        - default: `0.0`
- `name`: ***<optional>***
        - type: `<class 'str'>`
        - default: `None`
- `prop_dim3`: ***<optional>***
        - type: `<class 'int'>`
        - default: `1`
        - min: `1`
- `time_varying`: ***<optional>***
        - type: `<class 'bool'>`
        - default: `True`
        - possible_values: `[True, False]`
- `update`: ***<optional>***
        - type: `<class 'bool'>`
        - default: `True`
        - possible_values: `[True, False]`
- `vector_dim`: ***<optional>***
        - type: `<class 'int'>`
        - default: `1`
        - min: `1`
- `write`: ***<optional>***
        - type: `<class 'bool'>`
        - default: `True`
        - possible_values: `[True, False]`

particle_release_groups

Module: oceantracker.particle_release_groups

PointRelease

Class: oceantracker.particle_release_groups.point_release.PointRelease

File: oceantracker/particle_release_groups/point_release.py

Default internal name: `"not given in defaults"`

Description: Release particles at 1 or more given locations, releasing pulse_size particles every release_interval

Parameters:

- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `Release particles at 1 or more given locations, releasing pulse_size particles`
- `maximum_age`**: *<optional>***
  - type: `<class 'float'>`
  - default: `1e+32`
  - min: `1.0`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

<isrequired>

Description: - List of points where particles are released

- type: `vector`
- default: `[]`

- `pulse_size`**: *<optional>***
  - type: `<class 'int'>`
  - default: `1`
  - min: `1`
- `release_radius`**: *<optional>***
  - type: `<class 'float'>`
  - default: `1e+32`
  - min: `0`
- `release_interval`**: *<optional>***
  - type: `<class 'float'>`
  - default: `0.0`
  - min: `0.0`
- `release_radius`**: *<optional>***
  - type: `<class 'float'>`
  - default: `0.0`
  - min: `0.0`

- `release_start_date`: ***<optional>***
    - type: `iso8601date`
    - default: `None`
- `release_z`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`

user_particle_property_parameters: still working on display of default params of type <class 'dict'>

- `user_release_group_ID`: ***<optional>***
    - type: `<class 'int'>`
    - default: `0`
- `user_release_group_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

PolygonRelease

Class: oceantracker.particle_release_groups.polygon_release.PolygonRelease

File: oceantracker/particle_release_groups/polygon_release.py

Default internal name: `"not given in defaults"`

Description: Release particles at random locations within given polygon. Points chosen are always inside the domain and inside wet cells.

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default:
      `Release particles at random locations within given polygon. Points chosen are a`
- `maximum_age`: ***<optional>***
    - type: `<class 'float'>`
    - default: `1e+32`
    - min: `1.0`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

<isrequired>

- type: `vector`
- default: `[]`

        - list_contains_type: `<class 'float'>`
    - `pulse_size`**: *<optional>***
        - type: `<class 'int'>`
        - default: `1`
        - min: `1`
    - `release_duration`**: *<optional>***
        - type: `<class 'float'>`
        - default: `1e+32`
        - min: `0`
    - `release_interval`**: *<optional>***
        - type: `<class 'float'>`
        - default: `0.0`
        - min: `0.0`
    - `release_start_date`**: *<optional>***
        - type: `iso8601date`
        - default: `None`
    - `release_z`**: *<optional>***
        - type: `<class 'float'>`
        - default: `0.0`

user_particle_property_parameters: still working on display of default params of type <class ‘dict’>

    - `user_polygonID`**: *<optional>***
        - type: `<class 'int'>`
        - default: `0`
        - min: `0`
    - `user_polygon_name`**: *<optional>***
        - type: `<class 'str'>`
        - default: `None`
    - `user_release_group_ID`**: *<optional>***
        - type: `<class 'int'>`
        - default: `0`
    - `user_release_group_name`**: *<optional>***
        - type: `<class 'str'>`
        - default: `None`
    - `z_max`**: *<optional>***
        - type: `<class 'float'>`
        - default: `0.0`
    - `z_min`**: *<optional>***
        - default: `0.0`
        - type: `<class 'float'>`

PolygonReleaseWaterDepthRange

Class:
oceantracker.particle_release_groups.polygon_release_water_depth_range.PolygonReleaseWaterDepthRange

File: oceantracker/particle_release_groups/polygon_release_water_depth_range.py

Default internal name: `"not given in defaults"`

Description: Release particles at random locations within given polygon. Points chosen are always inside the domain and inside wet cells.

Parameters:

- `class_name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default:
      `Release particles at random locations within given polygon. Points chosen are a`
- `max_water_depth`**: *<optional>***
    - type: `<class 'float'>`
    - default: `1e+37`
- `maximum_age`**: *<optional>***
    - type: `<class 'float'>`
    - default: `1e+32`
    - min: `1.0`
- `min_water_depth`**: *<optional>***
    - type: `<class 'float'>`
    - default: `-1e+37`
- `name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`

<isrequired>

- type: `vector`
- default: `[]`
- list_contains_type: `<class 'float'>`

- `pulse_size`**: *<optional>***
    - type: `<class 'int'>`
    - default: `1`
    - min: `1`
- `release_duration`**: *<optional>***
    - type: `<class 'float'>`
    - default: `1e+32`
    - min: `0`

- `release_interval`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`
    - min: `0.0`
- `release_start_date`: ***<optional>***
    - type: `iso8601date`
    - default: `None`
- `release_z`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`

user_particle_property_parameters: still working on display of default params of type <class 'dict'>

- `user_polygonID`: ***<optional>***
    - type: `<class 'int'>`
    - default: `0`
    - min: `0`
- `user_polygon_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `user_release_group_ID`: ***<optional>***
    - type: `<class 'int'>`
    - default: `0`
- `user_release_group_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `z_max`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`
- `z_min`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`

particle_statistics

Module: oceantracker.particle_statistics

GriddedStats2D_agedBased

Class: oceantracker.particle_statistics.gridded_statistics.GriddedStats2D_agedBased

File: oceantracker/particle_statistics/gridded_statistics.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `age_bin_size`: ***<optional>***
  - type: `<class 'float'>`
  - default: `86400.0`
- `calculation_interval`: ***<optional>***

  Description: - time in sec, between calculating statistics
  - type: `<class 'float'>`
  - default: `3600.0`
- `case_output_file_tag`: ***<optional>***
  - type: `<class 'str'>`
  - default: `stats_gridded_age`
- `class_name`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `count_status_equal_to`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `count_status_greater_than`: ***<optional>***
  - type: `<class 'str'>`
  - default: `dead`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`

grid_center: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_size: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_span: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `max_age_to_bin`: ***<optional>***
  - type: `<class 'float'>`
  - default: `2592000.0`
- `min_age_to_bin`: ***<optional>***
  - type: `<class 'float'>`
  - default: `0.0`

- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

particle_property_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `release_group_centered_grids`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `write`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

GriddedStats2D_timeBased

Class: oceantracker.particle_statistics.gridded_statistics.GriddedStats2D_timeBased

File: oceantracker/particle_statistics/gridded_statistics.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `calculation_interval`: ***<optional>***
  Description: - time in sec, between calculating statistics
    - type: `<class 'float'>`
    - default: `3600.0`
- `case_output_file_tag`: ***<optional>***
    - type: `<class 'str'>`
    - default: `stats_gridded_time`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `count_status_equal_to`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
    - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `count_status_greater_than`: ***<optional>***
    - type: `<class 'str'>`
    - default: `dead`

- possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`

- `description`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`

grid_center: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_size: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_span: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `name`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`

particle_property_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `release_group_centered_grids`: ***<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `write`: ***<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

CorePolygonMethods

Class: oceantracker.particle_statistics.polygon_statistics.CorePolygonMethods

File: oceantracker/particle_statistics/polygon_statistics.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `description`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `name`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`

polygon_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

PolygonStats2D_ageBased

Class: oceantracker.particle_statistics.polygon_statistics.PolygonStats2D_ageBased

File: oceantracker/particle_statistics/polygon_statistics.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `age_bin_size`**: *<optional>***
  - type: `<class 'float'>`
  - default: `86400.0`
- `calculation_interval`**: *<optional>***

  Description: - time in sec, between calculating statistics
  - type: `<class 'float'>`
  - default: `3600.0`
- `case_output_file_tag`**: *<optional>***
  - type: `<class 'str'>`
  - default: `stats_polygon_age`
- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `count_status_equal_to`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `count_status_greater_than`**: *<optional>***
  - type: `<class 'str'>`
  - default: `dead`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

grid_center: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

grid_size: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

grid_span: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `max_age_to_bin`: ***<optional>***
    - type: `<class 'float'>`
    - default: `2592000.0`
- `min_age_to_bin`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

particle_property_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

polygon_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `release_group_centered_grids`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `write`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

PolygonStats2D_timeBased

Class: oceantracker.particle_statistics.polygon_statistics.PolygonStats2D_timeBased

File: oceantracker/particle_statistics/polygon_statistics.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `calculation_interval`: ***<optional>***
  Description: - time in sec, between calculating statistics
    - type: `<class 'float'>`
    - default: `3600.0`
- `case_output_file_tag`: ***<optional>***
    - type: `<class 'str'>`
    - default: `stats_polygon_time`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `count_status_equal_to`: ***<optional>***

- type: `<class 'str'>`
- default: `None`
- possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`

- `count_status_greater_than`**: *<optional>***
  - type: `<class 'str'>`
  - default: `dead`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

grid_center: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_size: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_span: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

particle_property_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

polygon_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `release_group_centered_grids`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `write`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

GriddedStats2D_ageBasedTopBottom

Class: oceantracker.particle_statistics.statisics_in_top_or_bottom_layer.GriddedStats2D_ageBasedTopBottom

File: oceantracker/particle_statistics/statisics_in_top_or_bottom_layer.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `age_bin_size`**: *<optional>***
    - type: `<class 'float'>`
    - default: `86400.0`
- `calculation_interval`**: *<optional>***

  Description: - time in sec, between calculating statistics
    - type: `<class 'float'>`
    - default: `3600.0`
- `case_output_file_tag`**: *<optional>***
    - type: `<class 'str'>`
    - default: `stats_gridded_age_layer`
- `class_name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `count_status_equal_to`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
    - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `count_status_greater_than`**: *<optional>***
    - type: `<class 'str'>`
    - default: `dead`
    - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`

grid_center: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

grid_size: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

grid_span: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `layer_thick_ness`**: *<optional>***
    - type: `<class 'float'>`
    - default: `0.0`
    - min: `0.0`
- `max_age_to_bin`**: *<optional>***
    - type: `<class 'float'>`
    - default: `2592000.0`
- `max_status`**: *<optional>***

- type: `<class 'str'>`
- default: `moving`
- possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`

- `min_age_to_bin`**: *<optional>***
  - type: `<class 'float'>`
  - default: `0.0`
- `min_status`**: *<optional>***
  - type: `<class 'str'>`
  - default: `frozen`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

particle_property_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `release_group_centered_grids`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `top_layer`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`
- `write`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

GriddedStats2D_timeBasedTopBottom

Class: oceantracker.particle_statistics.statisics_in_top_or_bottom_layer.GriddedStats2D_timeBasedTopBottom

File: oceantracker/particle_statistics/statisics_in_top_or_bottom_layer.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `calculation_interval`**: *<optional>***
  Description: - time in sec, between calculating statistics
  - type: `<class 'float'>`

- default: 3600.0

- case_output_file_tag: ***<optional>***
  - type: <class 'str'>
  - default: stats_gridded_time_layer
- class_name: ***<optional>***
  - type: <class 'str'>
  - default: None
- count_status_equal_to: ***<optional>***
  - type: <class 'str'>
  - default: None
  - possible_values: dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',
- count_status_greater_than: ***<optional>***
  - type: <class 'str'>
  - default: dead
  - possible_values: dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',
- description: ***<optional>***
  - type: <class 'str'>
  - default: None

grid_center: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

grid_size: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

grid_span: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- layer_thick_ness: ***<optional>***
  - type: <class 'float'>
  - default: 0.0
  - min: 0.0
- max_status: ***<optional>***
  - type: <class 'str'>
  - default: moving
  - possible_values: dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',
- min_status: ***<optional>***
  - type: <class 'str'>
  - default: frozen
  - possible_values: dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',

- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

particle_property_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `release_group_centered_grids`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `top_layer`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`
- `write`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

PolygonStats2D_ageBasedTopBottom

Class: oceantracker.particle_statistics.statisics_in_top_or_bottom_layer.PolygonStats2D_ageBasedTopBottom

File: oceantracker/particle_statistics/statisics_in_top_or_bottom_layer.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `age_bin_size`: ***<optional>***
    - type: `<class 'float'>`
    - default: `86400.0`
- `calculation_interval`: ***<optional>***

    Description: - time in sec, between calculating statistics
    - type: `<class 'float'>`
    - default: `3600.0`
- `case_output_file_tag`: ***<optional>***
    - type: `<class 'str'>`
    - default: `stats_polygon_age_depth_layer`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `count_status_equal_to`: ***<optional>***

- type: `<class 'str'>`
- default: `None`
- possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`

- `count_status_greater_than`**: *<optional>***
  - type: `<class 'str'>`
  - default: `dead`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

grid_center: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_size: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_span: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `layer_thick_ness`**: *<optional>***
  - type: `<class 'float'>`
  - default: `0.0`
  - min: `0.0`
- `max_age_to_bin`**: *<optional>***
  - type: `<class 'float'>`
  - default: `2592000.0`
- `max_status`**: *<optional>***
  - type: `<class 'str'>`
  - default: `moving`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `min_age_to_bin`**: *<optional>***
  - type: `<class 'float'>`
  - default: `0.0`
- `min_status`**: *<optional>***
  - type: `<class 'str'>`
  - default: `frozen`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

particle_property_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

polygon_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `release_group_centered_grids`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `top_layer`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`
- `write`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

PolygonStats2D_timeBasedTopBottom

Class: oceantracker.particle_statistics.statisics_in_top_or_bottom_layer.PolygonStats2D_timeBasedTopBottom

File: oceantracker/particle_statistics/statisics_in_top_or_bottom_layer.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `calculation_interval`: ***<optional>***
  Description: - time in sec, between calculating statistics
    - type: `<class 'float'>`
    - default: `3600.0`
- `case_output_file_tag`: ***<optional>***
    - type: `<class 'str'>`
    - default: `stats_polygon_time_depth_layer`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `count_status_equal_to`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
    - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `count_status_greater_than`: ***<optional>***
    - type: `<class 'str'>`
    - default: `dead`

- possible_values: dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',
- description: ***<optional>***
  - type: <class 'str'>
  - default: None

grid_center: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

grid_size: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

grid_span: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- layer_thick_ness: ***<optional>***
  - type: <class 'float'>
  - default: 0.0
  - min: 0.0
- max_status: ***<optional>***
  - type: <class 'str'>
  - default: moving
  - possible_values: dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',
- min_status: ***<optional>***
  - type: <class 'str'>
  - default: frozen
  - possible_values: dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',
- name: ***<optional>***
  - type: <class 'str'>
  - default: None

particle_property_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

polygon_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- release_group_centered_grids: ***<optional>***
  - type: <class 'bool'>
  - default: False
  - possible_values: [True, False]
- top_layer: ***<optional>***
  - type: <class 'bool'>
  - default: True
  - possible_values: [True, False]
- write: ***<optional>***

- type: `<class 'bool'>`
- default: `True`
- possible_values: `[True, False]`

TopBottomLayerStats

Class: oceantracker.particle_statistics.statisics_in_top_or_bottom_layer.TopBottomLayerStats

File: oceantracker/particle_statistics/statisics_in_top_or_bottom_layer.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `layer_thick_ness`**: *<optional>***
  - type: `<class 'float'>`
  - default: `0.0`
  - min: `0.0`
- `max_status`**: *<optional>***
  - type: `<class 'str'>`
  - default: `moving`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `min_status`**: *<optional>***
  - type: `<class 'str'>`
  - default: `frozen`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `top_layer`**: *<optional>***
  - default: `True`
  - type: `<class 'bool'>`
  - possible_values: `[True, False]`

GriddedStats2D_ageBasedDepthRange

Class: oceantracker.particle_statistics.statisics_in_water_depth_range.GriddedStats2D_ageBasedDepthRange

File: oceantracker/particle_statistics/statisics_in_water_depth_range.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `age_bin_size`**: *<optional>***
  - type: `<class 'float'>`
  - default: `86400.0`
- `calculation_interval`**: *<optional>***

  Description: - time in sec, between calculating statistics
  - type: `<class 'float'>`
  - default: `3600.0`
- `case_output_file_tag`**: *<optional>***
  - type: `<class 'str'>`
  - default: `stats_gridded_age_depth_range`
- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `count_status_equal_to`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `count_status_greater_than`**: *<optional>***
  - type: `<class 'str'>`
  - default: `dead`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

grid_center: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_size: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_span: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `max_age_to_bin`**: *<optional>***
  - type: `<class 'float'>`
  - default: `2592000.0`
- `max_water_depth`**: *<optional>***
  - type: `<class 'float'>`
  - default: `1000000000.0`

- `min_age_to_bin`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`
- `min_depth`: ***<optional>***
    - type: `<class 'float'>`
    - default: `-1000000000.0`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

particle_property_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `release_group_centered_grids`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `write`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

GriddedStats2D_timeBasedDepthRange

Class: oceantracker.particle_statistics.statisics_in_water_depth_range.GriddedStats2D_timeBasedDepthRange

File: oceantracker/particle_statistics/statisics_in_water_depth_range.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `calculation_interval`: ***<optional>***
  Description: - time in sec, between calculating statistics
    - type: `<class 'float'>`
    - default: `3600.0`
- `case_output_file_tag`: ***<optional>***
    - type: `<class 'str'>`
    - default: `stats_gridded_time_depth_range`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `count_status_equal_to`: ***<optional>***
    - type: `<class 'str'>`

- default: `None`
- possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`

- `count_status_greater_than`**: *<optional>***
  - type: `<class 'str'>`
  - default: `dead`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

grid_center: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

grid_size: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

grid_span: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `max_water_depth`**: *<optional>***
  - type: `<class 'float'>`
  - default: `1000000000.0`
- `min_depth`**: *<optional>***
  - type: `<class 'float'>`
  - default: `-1000000000.0`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

particle_property_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `release_group_centered_grids`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `write`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

PolygonStats2D_ageBasedDepthRange

Class: oceantracker.particle_statistics.statisics_in_water_depth_range.PolygonStats2D_ageBasedDepthRange

File: oceantracker/particle_statistics/statisics_in_water_depth_range.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `age_bin_size`: ***<optional>***
  - type: `<class 'float'>`
  - default: `86400.0`
- `calculation_interval`: ***<optional>***

  Description: - time in sec, between calculating statistics
  - type: `<class 'float'>`
  - default: `3600.0`
- `case_output_file_tag`: ***<optional>***
  - type: `<class 'str'>`
  - default: `stats_polygon_age_depth_range`
- `class_name`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `count_status_equal_to`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `count_status_greater_than`: ***<optional>***
  - type: `<class 'str'>`
  - default: `dead`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`

grid_center: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_size: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_span: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `max_age_to_bin`: ***<optional>***
  - type: `<class 'float'>`
  - default: `2592000.0`
- `max_water_depth`: ***<optional>***
  - type: `<class 'float'>`

- default: `1000000000.0`

- `min_age_to_bin`: ***<optional>***
  - type: `<class 'float'>`
  - default: `0.0`
- `min_depth`: ***<optional>***
  - type: `<class 'float'>`
  - default: `-1000000000.0`
- `name`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`

particle_property_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

polygon_list: still working on display of default params of type <class 'oceantracker.util.parameter_checking.ParameterListChecker'>

- `release_group_centered_grids`: ***<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `write`: ***<optional>***
  - type: `<class 'bool'>`
  - default: `True`
  - possible_values: `[True, False]`

PolygonStats2D_timeBasedDepthRange

Class: oceantracker.particle_statistics.statisics_in_water_depth_range.PolygonStats2D_timeBasedDepthRange

File: oceantracker/particle_statistics/statisics_in_water_depth_range.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `calculation_interval`: ***<optional>***
  Description: - time in sec, between calculating statistics
  - type: `<class 'float'>`
  - default: `3600.0`
- `case_output_file_tag`: ***<optional>***
  - type: `<class 'str'>`
  - default: `stats_polygon_time_depth_range`
- `class_name`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`

- `count_status_equal_to`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
    - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `count_status_greater_than`**: *<optional>***
    - type: `<class 'str'>`
    - default: `dead`
    - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`

grid_center: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_size: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

grid_span: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `max_water_depth`**: *<optional>***
    - type: `<class 'float'>`
    - default: `1000000000.0`
- `min_depth`**: *<optional>***
    - type: `<class 'float'>`
    - default: `-1000000000.0`
- `name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`

particle_property_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

polygon_list: still working on display of default params of type <class ‘oceantracker.util.parameter_checking.ParameterListChecker’>

- `release_group_centered_grids`**: *<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `write`**: *<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

WaterDepthRangeStats

Class: oceantracker.particle_statistics.statisics_in_water_depth_range.WaterDepthRangeStats

File: oceantracker/particle_statistics/statisics_in_water_depth_range.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `max_water_depth`: ***<optional>***
    - type: `<class 'float'>`
    - default: `1000000000.0`
- `min_depth`: ***<optional>***
    - type: `<class 'float'>`
    - default: `-1000000000.0`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

BaseParticleLocationStats

Class: oceantracker.particle_statistics._base_location_stats.BaseParticleLocationStats

File: oceantracker/particle_statistics/_base_location_stats.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `calculation_interval`: ***<optional>***
    - type: `<class 'float'>`
    - default: `86400.0`
- `case_output_file_tag`: ***<optional>***
    - type: `<class 'str'>`
    - default: `stats_base`
- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`

- `count_status_equal_to`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
    - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `count_status_greater_than`: ***<optional>***
    - type: `<class 'str'>`
    - default: `dead`
    - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `write`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `True`
    - possible_values: `[True, False]`

time_varying_info

Module: oceantracker.time_varying_info

trajectory_modifiers

Module: oceantracker.trajectory_modifiers

CullParticles

Class: oceantracker.trajectory_modifiers.cull_particles.CullParticles

File: oceantracker/trajectory_modifiers/cull_particles.py

Default internal name: `"ParticleKill"`

Description:

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `cull_interval`: ***<optional>***
    - type: `<class 'float'>`
    - default: `86400`
    - min: `0`

- `cull_status_equal_to`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
    - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `cull_status_greater_than`: ***<optional>***
    - type: `<class 'str'>`
    - default: `dead`
    - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `ParticleKill`
- `probability_of_culling`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.1`
    - min: `0`
    - max: `1.0`
- `requires_3D`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`

BasicResuspension

Class: oceantracker.trajectory_modifiers.resuspension.BasicResuspension

File: oceantracker/trajectory_modifiers/resuspension.py

Default internal name: `"BasicResuspension"`

Description:

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `critical_friction_velocity`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`
    - min: `0.0`

- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `BasicResuspension`
- `requires_3D`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`

SettleInPolygon

Class: oceantracker.trajectory_modifiers.settle_in_polygon.SettleInPolygon

File: oceantracker/trajectory_modifiers/settle_in_polygon.py

Default internal name: `"settle_in_polygon"`

Description:

Parameters:

- `class_name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`: ***<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`: ***<optional>***
    - type: `<class 'str'>`
    - default: `settle_in_polygon`

polygon: still working on display of default params of type <class ‘dict’>

- `probability_of_settlement`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`
- `requires_3D`: ***<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`
- `settlement_duration`: ***<optional>***
    - type: `<class 'float'>`
    - default: `0.0`

SplitParticles

Class: oceantracker.trajectory_modifiers.split_particles.SplitParticles

File: oceantracker/trajectory_modifiers/split_particles.py

Default internal name: `"particle_splitting"`

Description:

Parameters:

- `class_name`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `description`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `name`: ***<optional>***
  - type: `<class 'str'>`
  - default: `particle_splitting`
- `probability_of_splitting`: ***<optional>***
  - type: `<class 'float'>`
  - default: `1.0`
  - min: `0.0`
  - max: `1.0`
- `requires_3D`: ***<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `split_status_equal_to`: ***<optional>***
  - type: `<class 'str'>`
  - default: `None`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `split_status_greater_than`: ***<optional>***
  - type: `<class 'str'>`
  - default: `dead`
  - possible_values: `dict_keys(['unknown', 'bad_cord', 'cell_search_failed', 'notReleased', 'dead',`
- `splitting_interval`: ***<optional>***
  - type: `<class 'float'>`
  - default: `3600`
  - min: `1`

TrajectoryModifiersBase

Class: oceantracker.trajectory_modifiers._base_trajectory_modifers.TrajectoryModifiersBase

File: oceantracker/trajectory_modifiers/_base_trajectory_modifers.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `requires_3D`**: *<optional>***
    - type: `<class 'bool'>`
    - default: `False`
    - possible_values: `[True, False]`

velocity_modifiers

Module: oceantracker.velocity_modifiers

AddTerminalVelocity

Class: oceantracker.velocity_modifiers.terminal_velocity.AddTerminalVelocity

File: oceantracker/velocity_modifiers/terminal_velocity.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `description`**: *<optional>***
    - type: `<class 'str'>`
    - default: `None`
- `is3D`**: *<optional>***
    - type: `<class 'bool'>`

- default: `False`
- possible_values: `[True, False]`

- `mean`**: *<optional>***
  - type: `<class 'float'>`
  - default: `0.0`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `variance`**: *<optional>***
  - type: `<class 'float'>`
  - default: `0.0`
  - min: `0.0`

VelocityModiferBase

Class: oceantracker.velocity_modifiers._base_velocity_modifer.VelocityModiferBase

File: oceantracker/velocity_modifiers/_base_velocity_modifer.py

Default internal name: `"not given in defaults"`

Description:

Parameters:

- `class_name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `description`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`
- `is3D`**: *<optional>***
  - type: `<class 'bool'>`
  - default: `False`
  - possible_values: `[True, False]`
- `name`**: *<optional>***
  - type: `<class 'str'>`
  - default: `None`

Documentation

Internal code structure

todo lots to add

Setup workflow

Computational workflow

shared_info

Conventions

To add

Future additions

- support structured grids, eg. ROMS
- option for particle locations to working natively in lat/log
- support short class_name

Change log

Version 0.2.774 20/7/22

New features

1. polygon release only releases into wet cells, not just those inside domain
2. added pages giving full most of default parameters for each class to doc

Changes

1. Restructured to move all core classes up one level and delete core dir
2. **changes to make dir names and class names match parameter names**
    - folders interpolators now interpolator, affects class imports
    - folders readers now reader, affects user class imports
    - particle_velocity and velocity_modifiers param now velocity_modifiers
    - internally interp is now interpolator

Bug fixes

1. reintroduced a lost feature, that blocked movement of particles into dry cells

Version 0.2.772 11/7/22

1. Name changes for split and cull classes and module names
2. oceantracker_main is now just main and running is now just main.run(params)
3. move input_dir param from shared_params to a reader param

Version 0.2.768 01/7/2022

1. fixed bug in calculating depth average velocity, which meant it was zero and resupension would not happen for non zero critcal frict vel
2. created _base_reader and simplified reader as basis for making a structured grid reader

Version 0.2.760, 28/6/2022

1. bug fix: where velocity modifiers were not being used after restructure, eg terminal velocity
2. added open boundary condition, die on exit, for schism if hgrid file is available
3. split post_processing into two sub folders, plotting and readoutputfiles, ploting is now slit into subfiles, eg plot_tracks
4. plot_tracks, fraction_to_plot, has moved to reading of output data to become load_particle_track_vars(.., fraction_to_read=0.1)
5. particle status flags 'stranded_Bytide' is now 'stranded_by_tide', 'stranded_onBottom' is 'on_bottom', values also changed, 6. 'stranded_by_tide': 3, 'on_bottom': 6, to make it easier to set hierarchy of movement (this affects split status greater than a given value and "count_status_equal_to"),
6. To make it easier for user and future proof, status flags are now passed by name, not value, possible names are ['unknown', 'notReleased', 'bad', 'outside_domain', 'dead', 'frozen', 'stranded_by_tide', 'on_bottom', 'moving']
7. rebuilt tidal stranding to be based on total waterdepth < min_depth, code relating to dry cells, that was used for stranding, deleted
8. merged calaculate velocity by add_modifiers into solver core particle_velocity class now gone
9. sharedinfo.class_interators_dict is now sharedinfo.class_list_interators
10. brought field class, eg friction velocity, into line with initialize from parameters, as for all other classes
11. total water_depth feild added automatically, using zlevels if available, otherwise tide and water depth (as schism "tide" is not always top zlevel in dry cells)
12. added shared_params['use_numpy_random_seed'] boolean, for testing only!
13. (not yet working in linux) ?? short version of long class names eg, 'class_name': 'oceantracker.particle_release_groups.polygon_release.PolygonRelease', with the oceantracker package can optionally use class name only, eg PolygonRelease, this requires
14. added ability for any class to add the fields or particle properties they need to operate
15. particle and other numba utilities have move to util subfolder of pariticle_properties, as have field util etc…

Version 0.2.751, 22/6/2022

1. Addition of triangle based concentrations fields required more uniform way for coding users to cite all classes by name, (as already done for particle and fields) , so as to use their values in altering particle behaviour, so class referencing and iteration are now split - All classes can now be accessed by name through self.shared_info.classes , eg self.shared_info.classes['solver'] or self.shared_info.classes['particle_properties']['x'], - Classes which don't require a name and none is give generic name "unnamed001" or unnamed002 etc based on the sequence they are added in parameters - The ability to iterate over sets of classes and sub sets of these classes is now separated to dicts contained in self.shared_info.class_interators, eg to iterate over different types of particle properties
2. Case numbering/sequence numbering/file names numbering, eg for class lists , eg stats, events, are now more intuitive 1 base, so first stats file has index 001, not 000, and plotting needs to use nsequence = 1 to get the first
3. Plotting heatmaps and concertation fields, can now gourad shade concentration fields, which requires as conversion from face to node values in the code
4. Param key 'user_onfly_particle_statistics' is now 'particle_statistics'
5. Param key 'user_derived_fields' is now 'fields'
6. "user" tags of folders and params were not needed from user perspective, so all are now gone
7. Added load_output_files.get_case_info_files_from_dir(dir_name) to load all case files in folder, with None for any missing cases, optionally can select one case, with first case is case=1

8. Note run_output_folder is deted are start of run, but using shared_param “add date to folder name” will persevere todays work in a folder tagged with date
9. All file and module names are lower case (to avoid issues where linux is not always case sensitive, as is git which is case insensitive to file names by default, but python is case sensitive) and Classes are camel case which is a python convention
10. Almost a full check on params is now done on start up before cases are spawned
11. Error/warning handling and recording mechanics have been rewritten from scratch
12. Plotting: animate_particles and plot_tracks now have fraction_to_plot, which only plots a randomly chosen fraction of the tracks

1(1, 2) Vennell, R., Scheel, M., Weppe, S., Knight, B. and Smeaton, M., 2021.Fast lagrangian particle tracking in unstructured ocean model grids. Ocean Dynamics , 71(4), pp.423-437.

Last updated: 2022-07-17