# WP7 – Quick guide for tool adopters

This is an internal MyCOAST document aimed at providing MyCOAST partners with a quick overview of the principle requirements they must meet in order to adopt the tool to work with their system.

## Action 3. Pollution

### Tool 1 - MyCoastLCS

MyCoastLCS has been developed to enable the visualisation, identification and quantification of pollution hotspots of non-reactive, buoyant, slow diffusing and short lived substances. These assumptions can approximate the behaviour of plastic debris as well as sewage waste over the time scale of a few days. The tools specifically calculates the spatial distribution and time evolution of Finite Time Lyapunov Exponent (FTLE) fields, extracts Lagrangian Coherent Structures (LCS) and estimates spatially discretised time evolving concentrations and residence times.

The tools have been developed by Plymouth Marine Laboratory (PML) and Universidade de Santiago de Compostela (USC). The system comprises of a python/cython package that includes a lagrangian particle simulator (Pylag), the MyCoastLLCS tool and a series of scripts to generate initial seeding locations for PyLag and to plot the results and compile a set of fixed graphs into a html and/or pdf file.

We provide PyLag to enable the generation of lagrangian trajectories required to calculate the Finite Time Lyapunov Exponents and pathways and concentrations of point source pollutants. However, MyCoastLCS tool can be used with other lagrangian models provided they produce a netcdf file with variables that include particle positions at discrete time intervals in 3 dimensions and associated particle ids. MyCoastLCS has so far been tested with LAGAR and PyLag.

*Requirements*

1. Data requirements:
   1. The users need to have access to outputs from a forecast system. Currently this is restricted to products available through the CMEMS platform, FVCOM based systems, NEMO based systems and soon ROMS based systems. Partner’s systems can be used provided they adhere to WP6 model output standards.
2. System requirements:
   1. A multi-core Linux system (stand alone or HPC type) is recommended for running Pylag within a parallel environment due to the high computational requirements for running 105 particles. The MyCoastLCS tool doesn’t require multi-core capabilities. The tools have been tested in PML’s local HPC (720 cpu, intel compilers), and several multi-core linux systems with Fedora versions ranging from 27 to 30.
3. Software requirements:
   1. MPI passage module, C++ compiler, bash terminal (or equivalent), python 3 and conda (optional). The tools’ dependencies on specific python packages and versions are all handled automatically during the installation process (i.e. Pylag-tools, PyFVCOM, scypy).

How to get the code

To gain access to PML gitlab server where PyLag and MyCoastLCS are hosted contact Ricardo Torres ([rito@pml.ac.uk](mailto:rito@pml.ac.uk)) or James Clark ([jcl@pml.ac.uk](mailto:jcl@pml.ac.uk)).

1. PyLag:

Precompiled binaries for 64bit Linux are accessible through a conda installation archived in a private conda channel. Detailed instructions can be found here (<https://pylag.readthedocs.io/en/latest/install/installation.html#installation-using-conda>)

The original code is maintained here <https://github.com/jimc101/PyLag>

1. MyCoastLCS:

Precompiled binaries are distributed with PyLag through a conda private channel (see above).

The code can be accessed through pmlmodelling’s github (<https://github.com/pmlmodelling/MyCoastLCS/>)

How to install the code

1. PyLag can be installed through conda or pip and instructions are included with the downloaded package.
2. MyCoastLCS can be installed through conda with PyLag or as a site package with python. Instructions are included with the downloaded package.

How to run the code

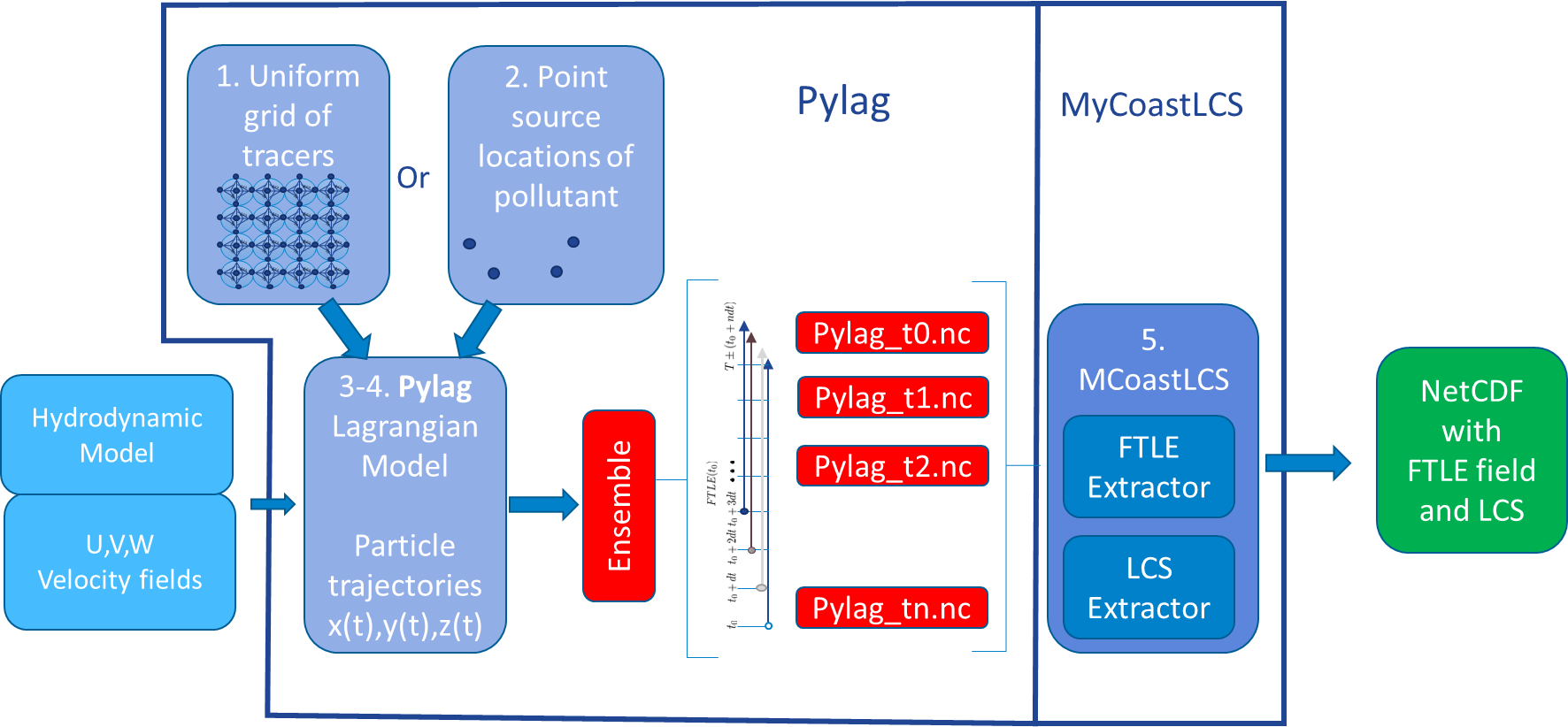
At the time of writing the tools are run via command line instructions from within a bash terminal (or equivalent). Once all the tools are installed, two configuration files are needed (examples provided by PML): one for the PyLag simulations and one for MyCoastLCS setup and plotting. Other configurable templates include an HPC queue submission script (if using an HPC system to run PyLag, PML can help if needed) and the location of rivers and/or Waste Water source points (Waste Water Treatment Plants and Surface Drainage Outfalls) as a shapefile or csv file.

The steps a user would follow (as outlined in Figure 1) include:

1. Generate a dense regularly spaced set of initial conditions. A python script is provided for this purpose.
2. Generate a set of initial positions for point source locations of pollutants (Figure 2: left). The python script provided as an example considers 3 types of sources points: rivers, waste water treatment plants and surface drainage outfalls. Ideally, the source concentrations would vary depending on river flow, water treatment plant capacity and rain fall. These dependencies are implemented in our example for surface drainage outfalls. In such a case, it is also necessary to generate a set of ‘trigger points’ (e.g. the minimum rainfall threshold that determines an overspill at a surface drainage outfall site) for each of the point source locations of pollutants, and a weighting for each site.
3. Generate master configuration file for PyLag for point source release of pollutants to specifying type of simulation (forward integration), duration of simulation (dependent of typical longevity of pollutants), total length of simulation (depends on user’s operational system forecast window), location of pollution point sources among others.
4. Generate master configuration file for PyLag for FTLE, to specify type of simulation (backward integration for identifying attractors to dispersal), duration of simulation for FTLE integration (i.e. 6-12 hours), total length of simulation (depends on user’s operational system forecast window), and location points of initial conditions among others.
5. Run Pylag with output from your forecast model of choice to generate the files to be used by MycastLCS and plotting scripts (Figure 2: right). While PyLag reads netcdf files that can be accessed from a local folder or a THREDDS or ERDDAP server via OpendAP it is recommended that the files are downloaded first for performance benefits.
6. Run MyCoastLCS to obtain FTLE and LCS fields (Figure 3). The tool can optionally be used to estimate concentrations and residence times that can be used to in unison with the point source location trajectories to generate a qualitative water quality indicator (Figure 4). This requires validation of the model by determining relationships between pollutant concentrations in environmental samples and model outputs.

A detailed manual for both PyLag and MyCoastLCS will be included with the code as Sphynx documentation.

Jupyter Notebooks will be provided by PML to illustrate a typical workflow, and an example of the implementation of the MyCoastLCS tool in a Galician estuary can be access through <http://thredds-gfnl.usc.es/thredds/catalog/MYCOASTLCS/catalog.html>



**Figure 1 Example of operational setup to generate potential pollutant trajectories and FTLE and LCS fields**

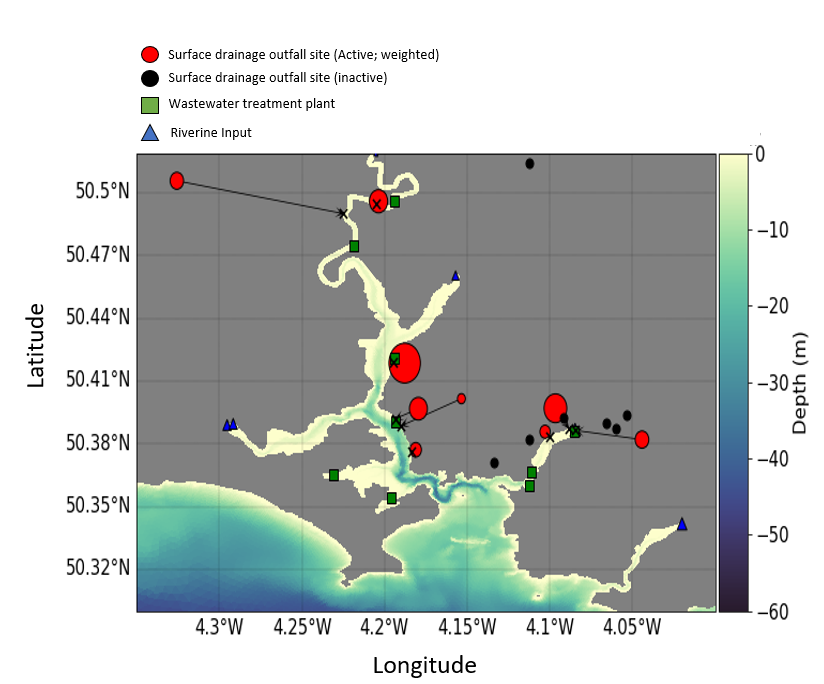
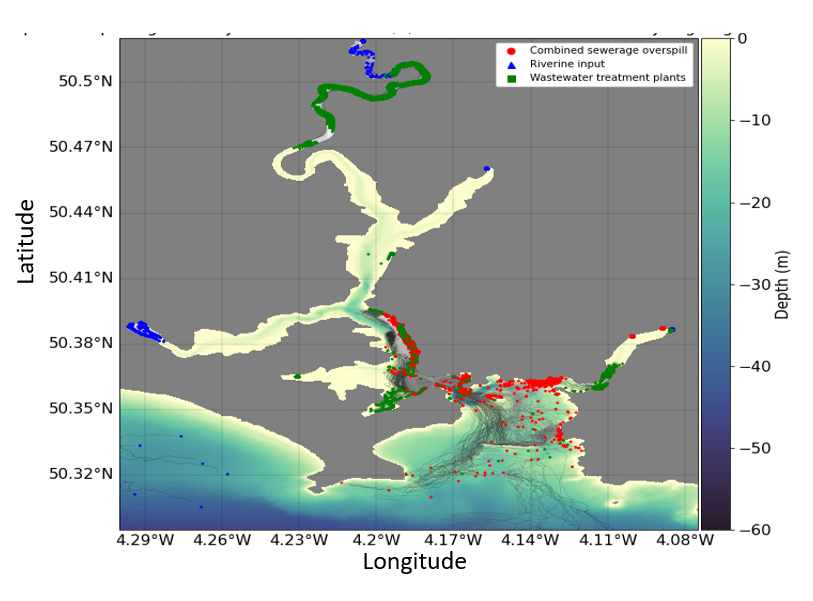
** **

Figure 2 Example of initial positions and 12 hour particle trajectories originating in the rivers, waste water plants and active surface drainage outfalls in the Tamar estuary domain.

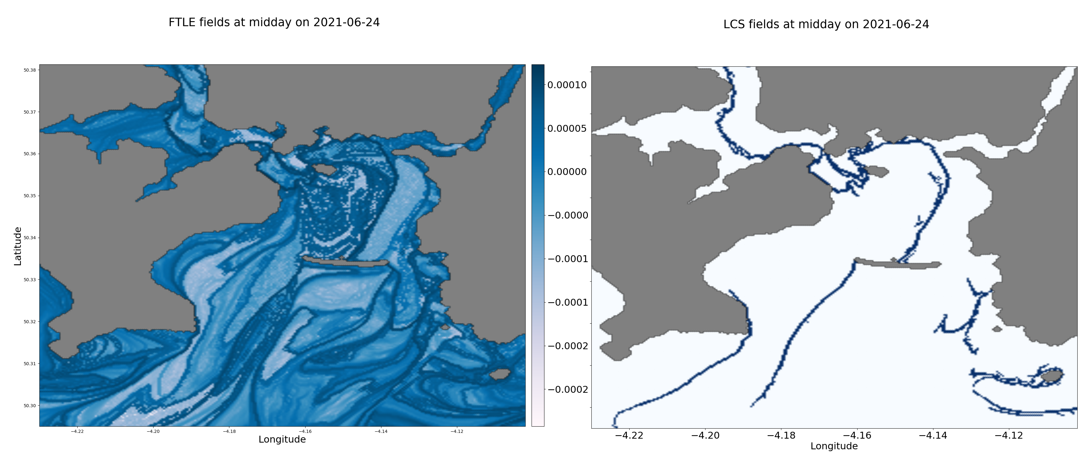


Figure 3 FTLE and LCS fields after 12-hour integration of backward trajectories highlighting areas of potential accumulation of positively buoyant pollutants. The darker blues in the FTLE fields indicate stronger attractors, and the LCS fields are the ridges of the FTLEs

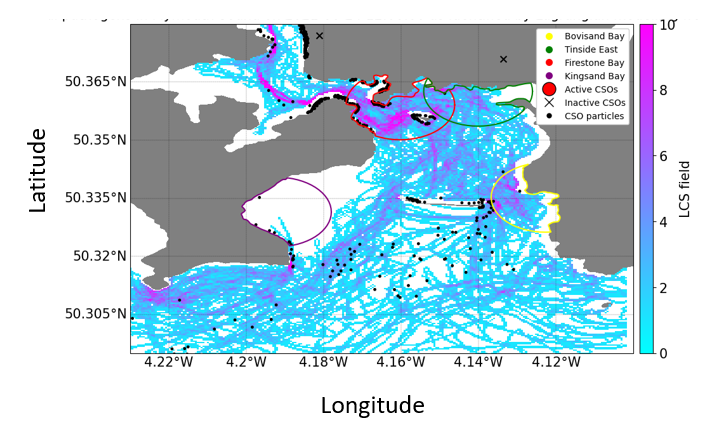


Figure 4 Locations of particles released from active surface drainage outfall sites and cumulative LCS fields over a 24-hour period. High number of particles and purple LCS fields indicate high potential for pollution risk in the bathing area (high likelihood of particles reaching the bathing areas from surface drainage outfalls and high likelihood of accumulation by hydrodynamics).