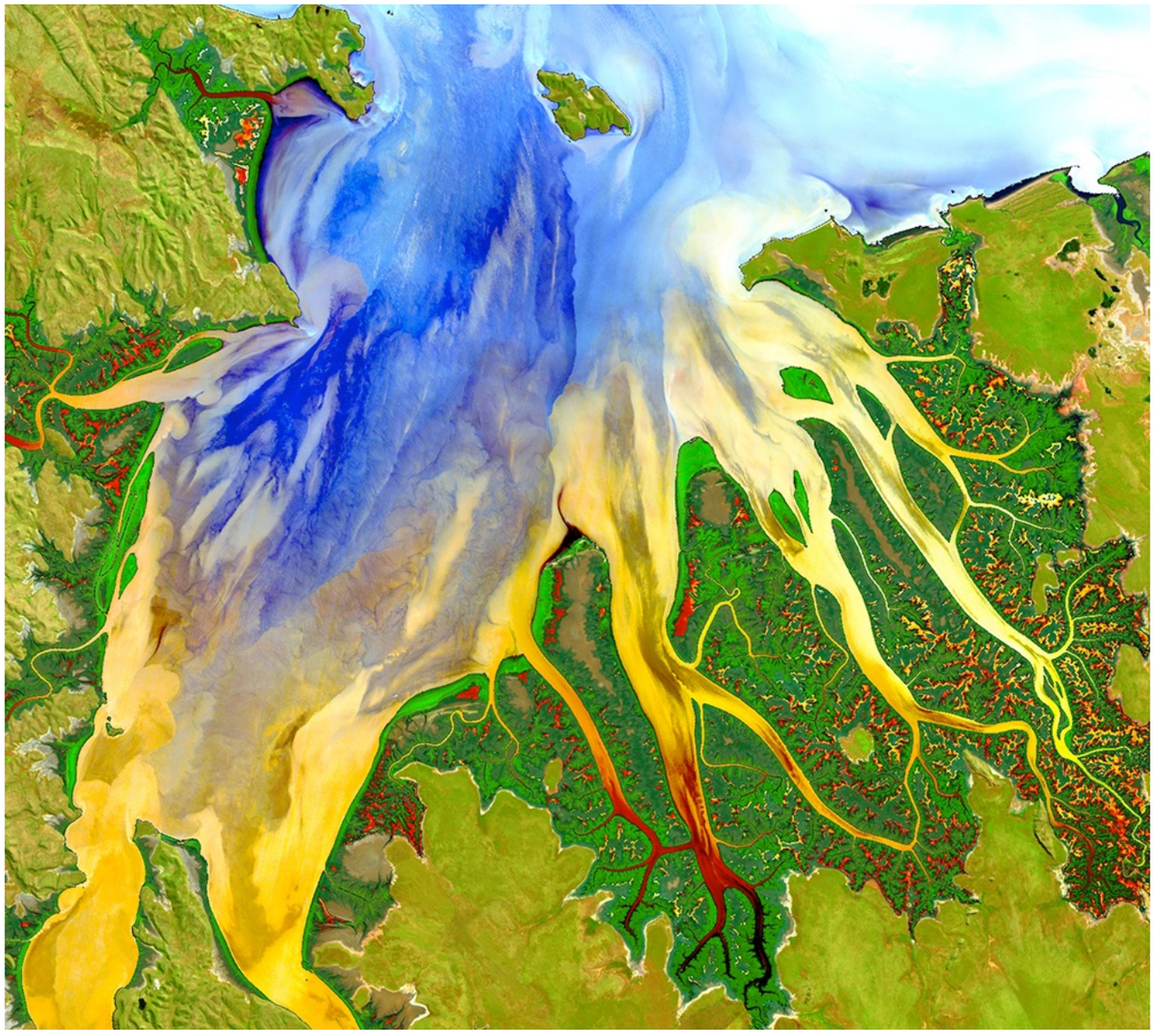
MARS 5001
Coastal Processes & Systems



MODELING PART

Setting up the environment

Installation on your Mac or PC

If you want to install the environment on your own computer you will need to download the Docker Toolbox which is an installer to quickly and easily install and setup a Docker environment on your computer.

Kitematic

Once it's installed you will need to log to the Docker Hub through the **Kitematic** application (which is shipped within the Docker Toolbox installation). For a guide on how to use **Kitematic** follow this link.

To connect to **Kitematic** and the **DockerHub** you will be required to get a login access. We have made one for you:
- name **usydgeos** (not required but in case: usydgeos@gmail.com)
- password **madsenf09**

Once the **Kitematic** window is active, search for the lab Docker image called **mars5001** from the DockerHub and click create. It will emulate a virtual machine with the IPython notebooks that we will use during these labs.

When the download has finished click on the right icon (square and arrow sign) on the right panel facing the Web Preview. It will fire-up a webpage which contains everything we need.

Using DigitalOcean

The labs have also been deployed via **DockerCloud** and **DigitalOcean**. To access these nodes open your web browser (preferably **firefox** or **chrome**) and get one of the links below.

You will need to be the only one using a given node number otherwise you will end-up modifying someone else notebook or your notebook might be modified! Also it will decrease the performance of the node and execution time could become really long!

Node 1 ocean-model-1-1.05994529.cont.dockerapp.io:32768

Node 2 ocean-model-1-10.bdbf6ca7.cont.dockerapp.io:32768

Node 3 ocean-model-1-2.220ec93c.cont.dockerapp.io:32768

Node 4 ocean-model-1-3.113c65da.cont.dockerapp.io:32768

Node 5 ocean-model-1-4.d3fbc0ca.cont.dockerapp.io:32768

Node 6 ocean-model-1-5.d3753d91.cont.dockerapp.io:32768

Node 7 ocean-model-1-7.197177ac.cont.dockerapp.io:32768

Node 8 ocean-model-1-8.3243a7ec.cont.dockerapp.io:32768

Node 9 ocean-model-1-9.46e50a9d.cont.dockerapp.io:32768

Node 10 ocean-model-2-1.e680f63d.cont.dockerapp.io:32768

Node 11 ocean-model-2-10.7a11cf2c.cont.dockerapp.io:32768

Node 12 ocean-model-2-2.5aae59a3.cont.dockerapp.io:32768

Node 13 ocean-model-2-3.d612d3eb.cont.dockerapp.io:32768

Node 14 ocean-model-2-5.6a2016fb.cont.dockerapp.io:32768

Node 15 ocean-model-2-6.6978100b.cont.dockerapp.io:32768

Node 16 ocean-model-2-7.99c7950a.cont.dockerapp.io:32768

Node 17 ocean-model-2-8.f42062c2.cont.dockerapp.io:32768

Node 18 ocean-model-2-9.0317bc97.cont.dockerapp.io:32768

LAB - DATA QUERY

Ocean Data Analysis

Instructions

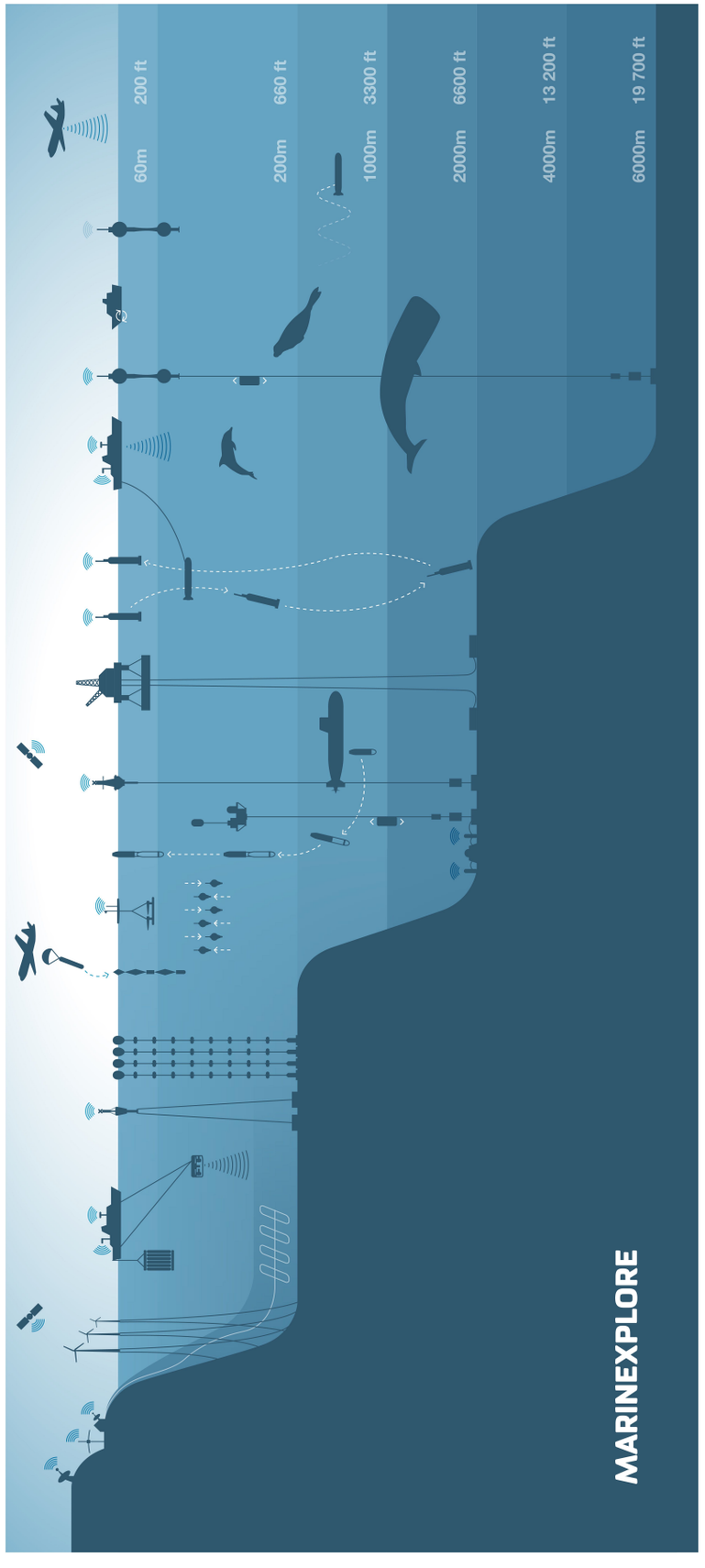
The goals of this first part are:

- to help you get comfortable with IPython notebooks. (We will use it throughout our modelling pracs for this UoS.
- to show you some examples of dataset coming from different systems: buoys, radars, models...
- to familiarize you with the different open data services and the most common data formats.

Through the notes and the notebooks you will see questions. You will need to provide solution for each of them.

Here we will introduce several **IPython** libraries such as **Matplotlib**, **Basemap** or **pandas** as well as common data format libraires (like **netCDF**). These libraries can be used to create maps, plot marine datasets but also to analyse them. If you follow along the associated ipython notebooks, you'll build the basic knowledge of scientific data analyze that will be useful in your future job as engineer, consultant or researcher.

The goal is not to press button here but to make sense of what you are doing! If at any point you need help, don't be shy and ask for help.



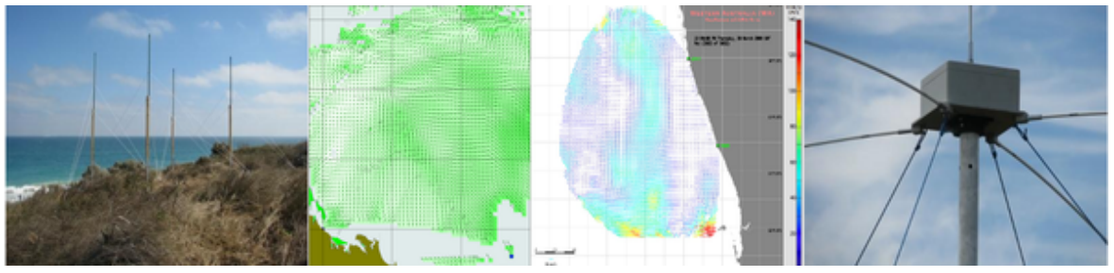
Overview

Access to quality data is essential for the understanding of marine processes. Over the last 10 to 5 years several Ocean Data portals have emerged and are routinely used by Scientific organizations, Research agencies and the industry to better understand the complexity of the Ocean and its interactions with Climate and Life. These portals facilitate seamless access to marine data/services and promote the exchange and dissemination of marine data and services.

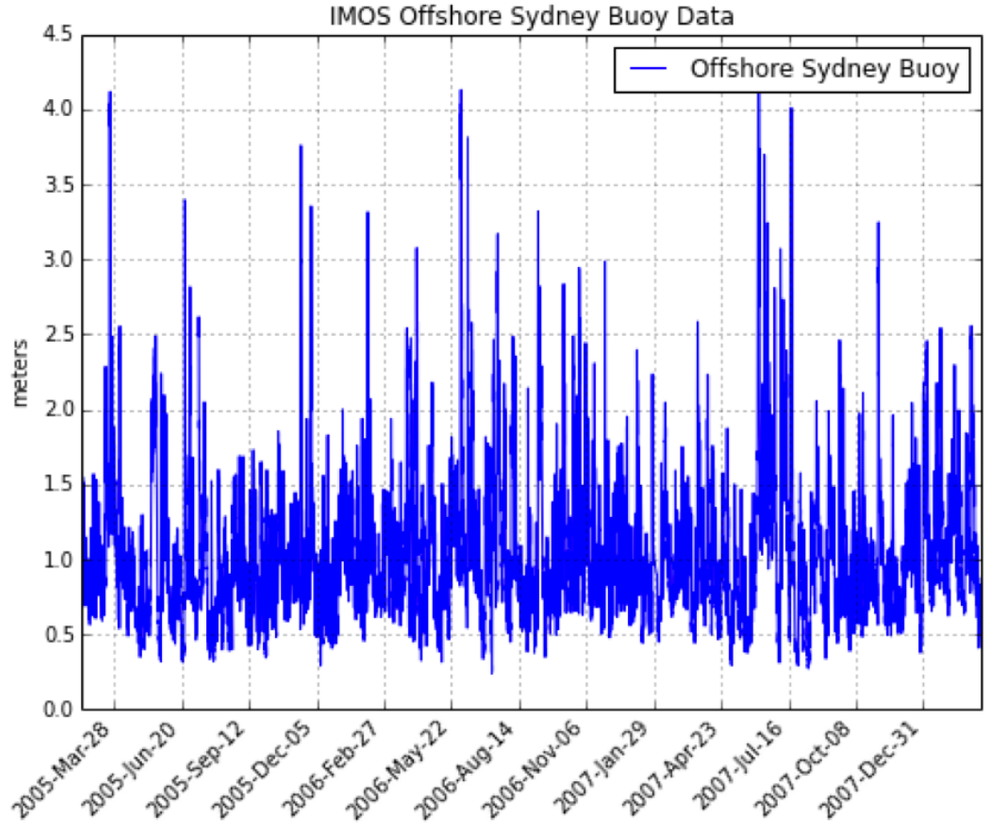
Ocean scientists routinely perform data 'crunching' to understand a particular system and need to access and query extensive lists of data.

Understanding how these data are stored, where they come from and how to quickly extract particular informations from them is what these series of notebooks are all about!

IMOS Ocean Radar Data Query



Let start by making a map of ocean radar data from IMOS. IMOS is the Australian Integrated Marine Observing System.



The Australian Coastal Ocean Radar Network (ACORN) has deployed over the last 10 years a series of HF radar in specific sites all around Australia. These ocean radar provide unprecedented time-resolved maps of surface currents over mesoscale areas (typically up to 150km x 150km) of coastal ocean. The ACORN system provides a basis for applied research into wave modeling and offers test sites for hydrodynamic modeling.

In this notebook, we will look at a dataset from the Turquoise Coast (close to Cervantes WA). You will query the data using THREDDS protocol, extract useful variables from the netcdf data format and perform a simple calculation of the magni-

tude of the current in the region at the time of acquisition and plot the result on a map.

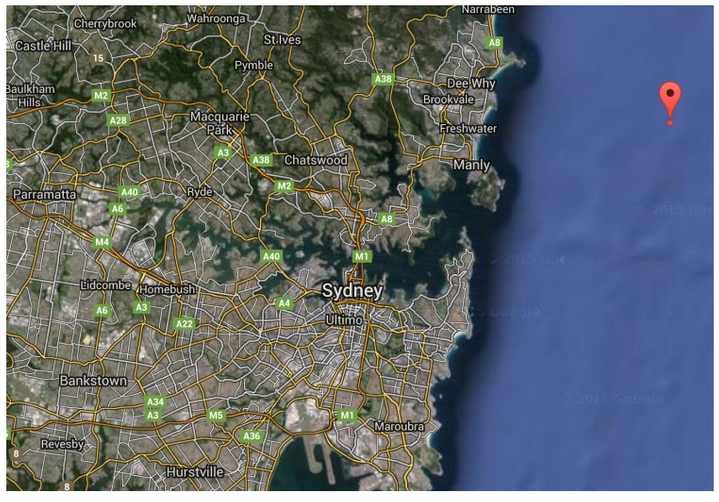
Q1. Follow the notebook. Knowing the zonal and meridional current components compute in a python cell the magnitude of the current. Save your map in your lab report.

Buoy Data Query

In this example, we will first use wave data collected by the NSW Public Works' Manly Hydraulics Laboratory (MHL) for the Office of Environment and Heritage. These data provide essential input to design, construction and performance monitoring of coastal zone projects undertaken by the NSW Government.

Uses of the wave data include:

- design, repair and performance monitoring of coastal structures
- assessment of coastal processes such as beach erosion and sediment transport
- baseline data for monitoring the impact of climate change



Location of the offshore Sydney buoy

- operational assistance for coastal construction projects
- monitoring severe weather conditions
- oceanographic research.

The NSW Wave Climate Program utilises a network of seven Waverider stations along the NSW coast. The buoys are located off Byron Bay, Coffs Harbour, Crowdy Head, Sydney, Port Kembla, Batemans Bay and Eden. To provide deepwater wave data, the buoys are typically moored in water depths between 60 and 100 metres, between 6 and 12 kilometres from the shoreline.

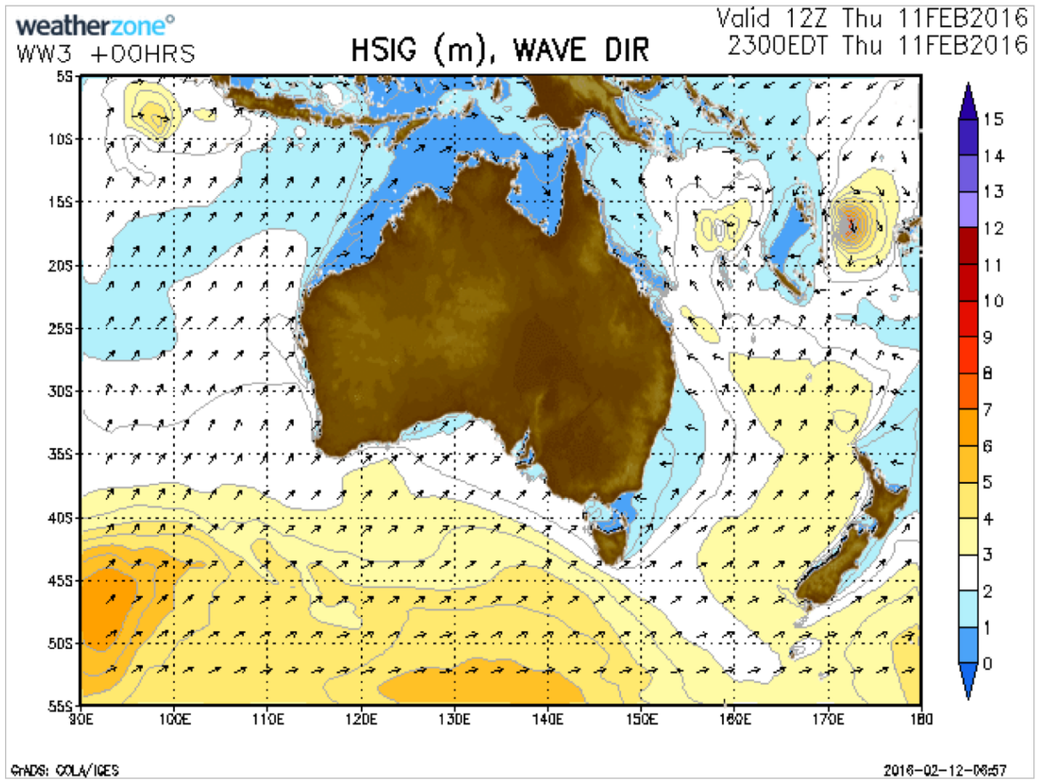
We will load the historical records for the buoy located Offshore Sydney. Like for our previous example the formatted data is in **netCDF**. We will first plot the wave mean height recorded at the buoy for the full temporal extend of the dataset.

Q2. Within the dataset what are the meaning of the following variables: **VAVH**, **VDIR**, **VAVT**.

Q3. Plot the root mean square wave height for the period ranging from February 2006 to February 2007.

Historical Model Data Query

There are several numerical model which are routinely used to forecast wave regimes. Here we will use one of the most established ones called **WaveWatch-III** (WW-3). In this model, physics includes amongst other things:



- wave field refraction,
- nonlinear resonant interactions.

Wind data is provided from the GDAS data assimilation system. Up to 2008, the model was limited to regions outside the surf zone where the waves are not strongly impacted by shallow depths. Since then the model has been extended to incorporate the effects of currents and is now used in shore applications.

The Australian Marine Weather website from the Bureau of Meteorology releases, everyday, forecasts maps from **WW-3 model**.

Follow the notebook to extract the predictions of significant wave height for the month of January 1998 in Maroubra Beach.

Q4. Using the same dataset plot the predictions of the **WW-3** model for *Manly beach* for both the significant wave height and the mean period.

Visualise in NbViewer

Follow these links to see the IPython notebooks that we are going to use for this part of the lab.

- LAB Data Query IPython material 1: nbviewer link
- LAB Data Query IPython material 2: nbviewer link

Instructions

In this 2nd part, we will build upon what you have learned in the previous notebooks.

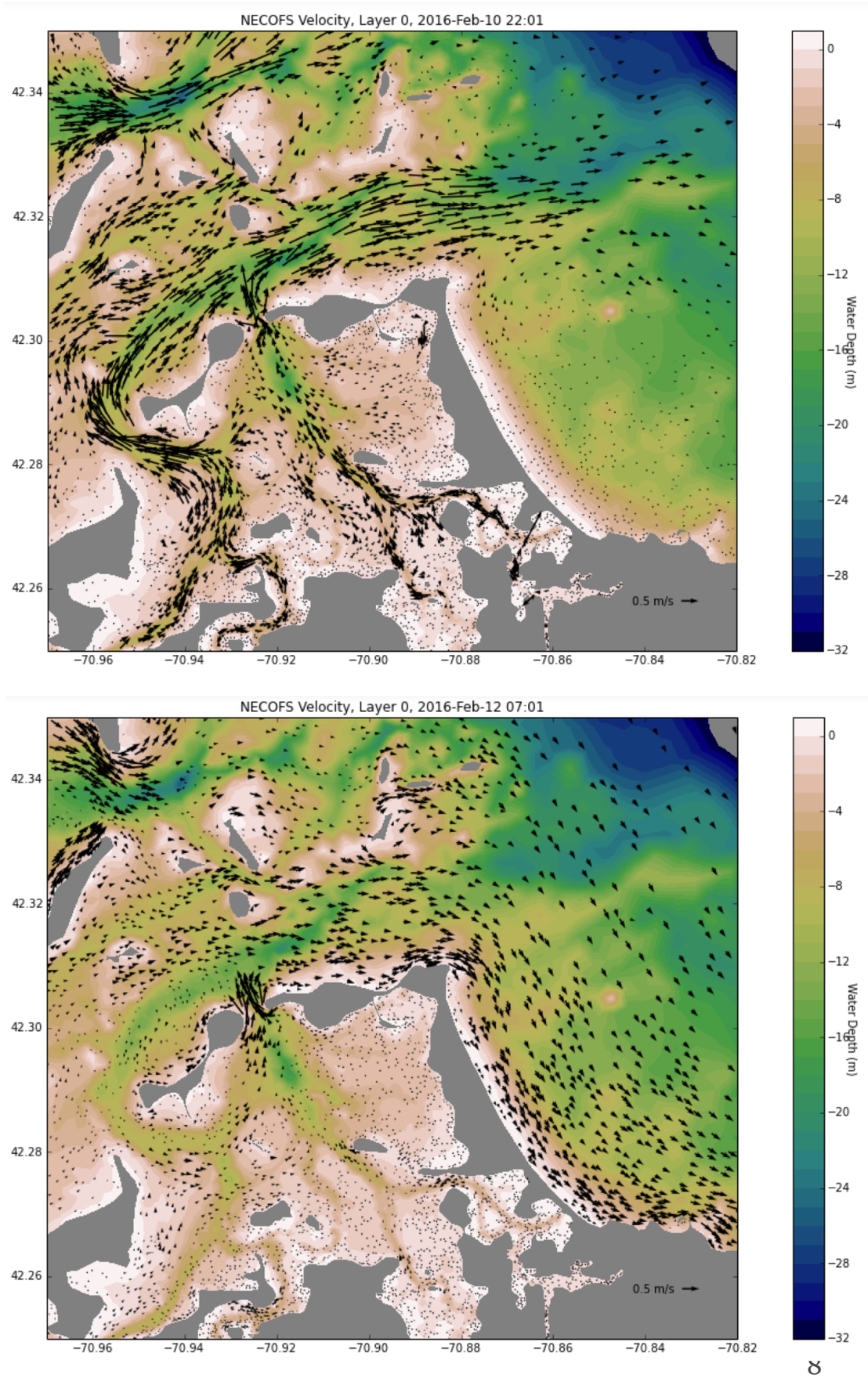
Again through the notes and the notebooks, you will see questions. You will need to provide solution for each of them.

Ocean Forecast Model Query

We have seen how to use global models (WW-3) to access historical ocean data. Now we will work with a forecast model called **FVCOM** (Finite-Volume, primitive equation Community Ocean Model) developed by the University of Massachusetts-Dartmouth in collaboration with the Woods Hole Oceanographic Institution. The model is well suited for simulating the circulation and ecosystem dynamics from global to estuarine scales, particularly for regions characterized by irregular complex coastlines, islands, inlets, creeks, and inter-tidal zones.

The model strengths not only rely on the physical processes that are simulated but also in its capacity to be coupled to other codes (meteorological, sediment transport, inundation models...).

The data that we are going to work with comes from the Northeast Coastal Ocean Forecast System (**NECOFS**) which is a US



integrated atmosphere-ocean model forecast system designed for the northeast US coastal region covering a computational domain from the south of Long-Island Sound to the north of the Nova Scotian Shelf.

The system includes:

1) the mesoscale meteorological model **WRF** (Weather Research and Forecasting model);

2) the unstructured grid Finite-Volume Community Ocean Model and Surface WAVE model with configuration for the Gulf of Maine/Georges Bank /New England Shelf (**FVCOM-GOM** and **FVCOM-SWAVE**);

3) the Massachusetts Coastal Waters domain FVCOM (Mass-Coastal FVCOM) and

4) fully wave-current coupled coastal inundation model systems for Scituate Harbor and Boston Harbor, MA, and Hamption-Seabrook Estuary, NH.

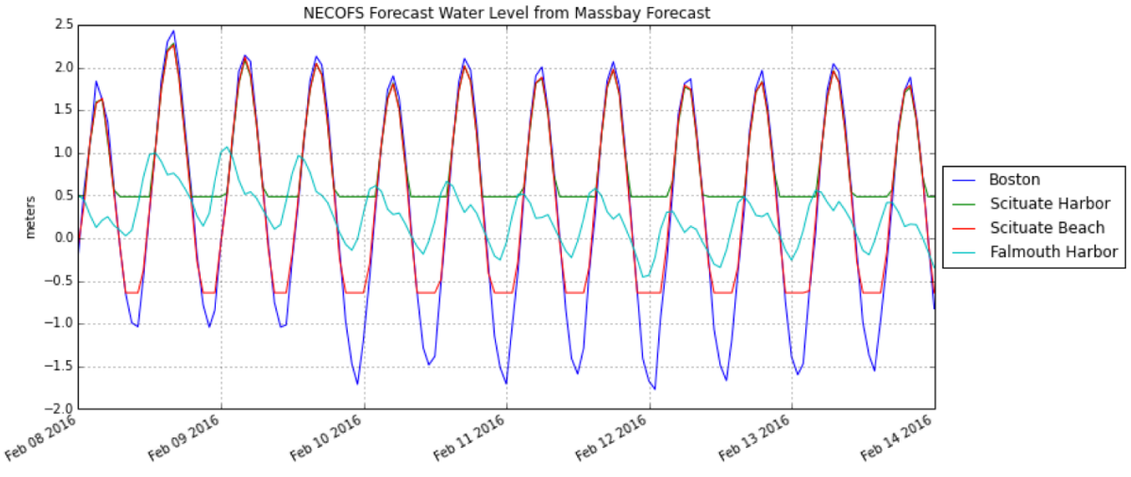
NECOFS has been validated by hindcast experiments from 1978 to present. This model is capable of reproducing accurately both tidal and subtidal motions in the Gulf of Maine/ Georges Bank/New England Shelf regions.

Using the notebook we will use hourly products from the NECOFS Web Map Server (WMS). NECOFS provides three days of forecast fields of surface winds, air pressure, air temperature, air humidity, sea surface heat flux, evaporation minus precipitation, sea level, water temperature, salinity, currents, wave heights, wave directions and wave frequencies as well as wintertime icing rates and storm-induced inundation areas. Just that :-) ....

Using the loaded dataset you will first have a look at the distribution of water depth and currents in the bay for the current time (US time actually).

Q5. Then have a look at the low and high tide time for today and plot two maps showing the forecast from FVCOM for these 2 times.

We will now look at the water levels forecasted for several stations in the Massachusetts Bay. The idea is that it is possible to refine the model by looking at buoy data from example which could be recorded on these stations.



Q6. You have on the next page a list of stations. The first four are the only one provided in the notebook file. Provide a plot showing the information for water level forecast for the entire set of stations.

	Lat	Lon
Station		
Boston	42.368186	-71.047984
Scituate Harbor	42.199447	-70.720090
Scituate Beach	42.209973	-70.724523
Falmouth Harbor	41.541575	-70.608020
Marion	41.689008	-70.746576
Marshfield	42.108480	-70.648691
Provincetown	42.042745	-70.171180
Sandwich	41.767990	-70.466219
Hampton Bay	42.900103	-70.818510
Gloucester	42.610253	-70.660570

Mapping Ocean Salinity

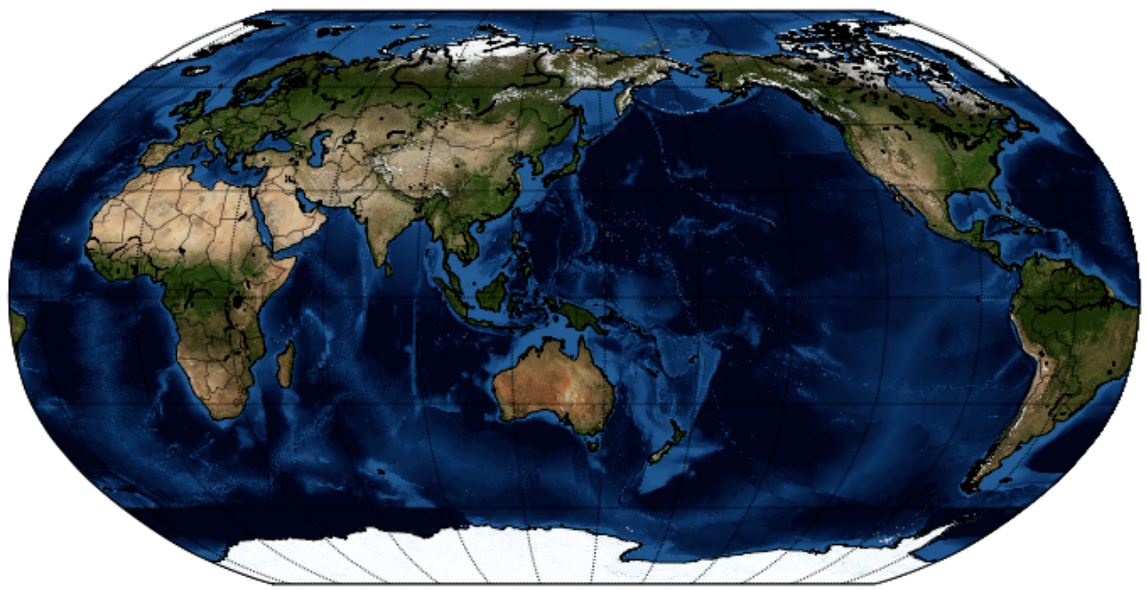
We will now use the Basemap library to quickly plot global topographic and salinity maps.

Making a simple map

Let's start out by making a simple map of the world. If you run the Ipython code, you should get a nice map of the globe, with good clean coastlines.

Let’s add some more detail to this map, starting with country borders and the latitude and longitude lines.

Before plotting data on this globe, we will adjust the perspective. Change the latitude and longitude parameters in the original Basemap definition to 0 and -100. You should see your



map centered along the equator. Then we will center it on Australia.

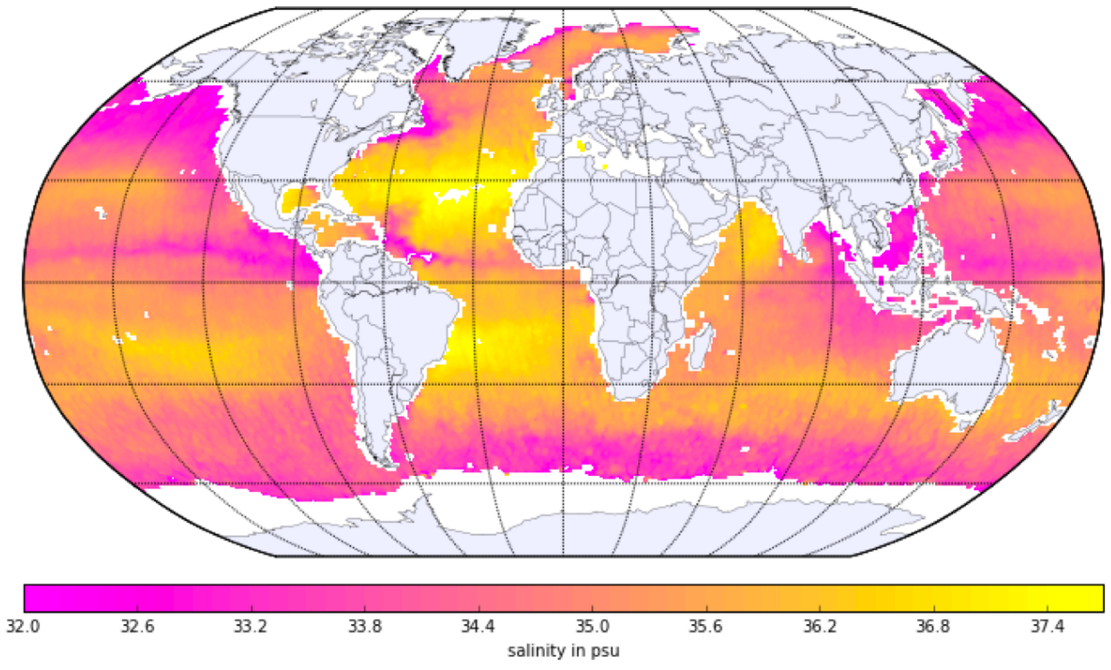
Now let's change the kind of map we're producing. Change the projection type to '*robin*'.

For a review of common map projections follow this link and this one as well projections.

Plotting global salinity map

I have provided you with a OpenDAP address to a global salinity map.

Q7. Provide the most detail references you can find about this dataset by search for it on the web.



Then follow the notebook to plot the salinity map using the '*robin*' projection system.

Zooming in

You will now look at the salinity distribution around Australia. Before that, you will need to find a way of zooming into a region using Basemap. This is important to know because there are many data sets specific to one region of the world, which would get lost when plotted on a map of the whole world. Some projections can not be zoomed in at all, so if things are not working well, make sure to look at the documentation.

The projection that you will use his the mercator project and you will need to specify the bounding box of your map.

Q8. Using the previous global dataset, provide a map of the salinity distribution around Australia.

Visualise in NbViewer

Follow these links to see the IPython notebooks that we are going to use for this part of the lab.

- LAB Data Query IPython material 3: nbviewer link
- LAB Data Query IPython material 4: nbviewer link

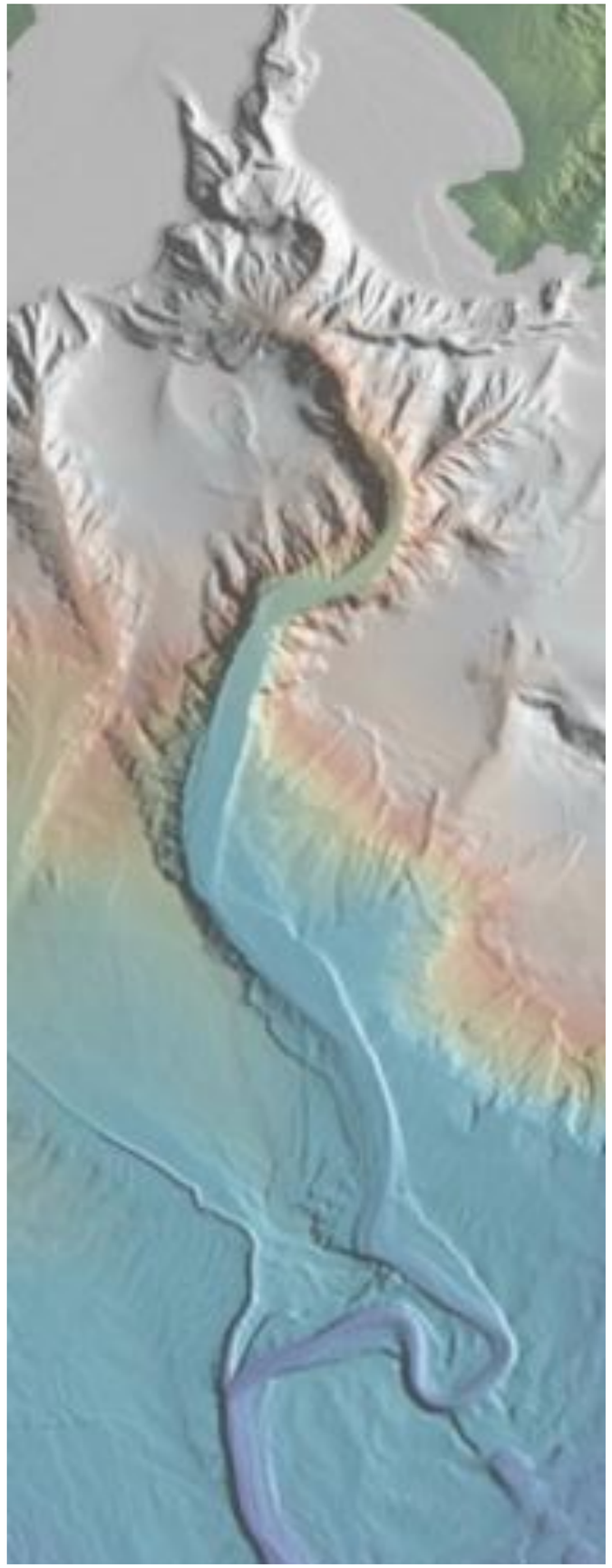
Instructions

Using IPython we will quantitatively estimate the relationship between submarine channel sinuosity, valley slope & latitude. There are 5 questions labeled Q, some with multiple parts. Provide solutions for all of them.

A note on submarine sedimentary systems

Submarine fans are accumulations of sediment deposited at the termini of land-to-deep-sea sediment-routing systems. At a given moment in time, sediment-routing systems comprise sediment source areas dominated by denudation, a zone of sediment transfer, and a terminal region of deposition, such as a submarine fan . The sediment-transfer zone between terrestrial source area and deep-sea depositional sink can include submarine canyon-channel systems. Submarine fans receive sediment through canyon-channel systems and are the largest detrital accumulations on Earth. Fan sediment includes the deposits of sediment gravity flows and other submarine mass movements. Perhaps the most widely recognized sediment-gravity-flow deposit is called a turbidite.

Because submarine fans and related turbidite systems are located at the termini of continent-draining sediment-routing systems, their deposits contain a wealth of proxy information pertaining to past climate and perturbations to their up-depositional-dip sedimentary systems.



For an overview of submarine sedimentary systems it is recommended to read the following article from Covault (2011) link.

Context

In 2011, Peakall et al. (see ref below) proposed that submarine channel sinuosity was in part induced by Coriolis force. They based their findings on a quantitative comparison of slope versus latitude for different submarine channel systems.

In 2013, Sylvester et al. (see ref below) took a closer look at the data provided by Peakall et al. and highlighted the importance of the nature of slope and sediment supply as the major factors controlling the development of submarine channel sinuosity.

Using their dataset and IPython, you will make your own analyses of the dataset.

References:

Peakall, J., Kane, I. A., Masson, D. G., Keevil, G., McCaffrey, W., & Corney, R. (2011). Global (latitudinal) variation in submarine channel sinuosity. Geology, 40(1), 11–14. [doi:10.1130/G32295.1]

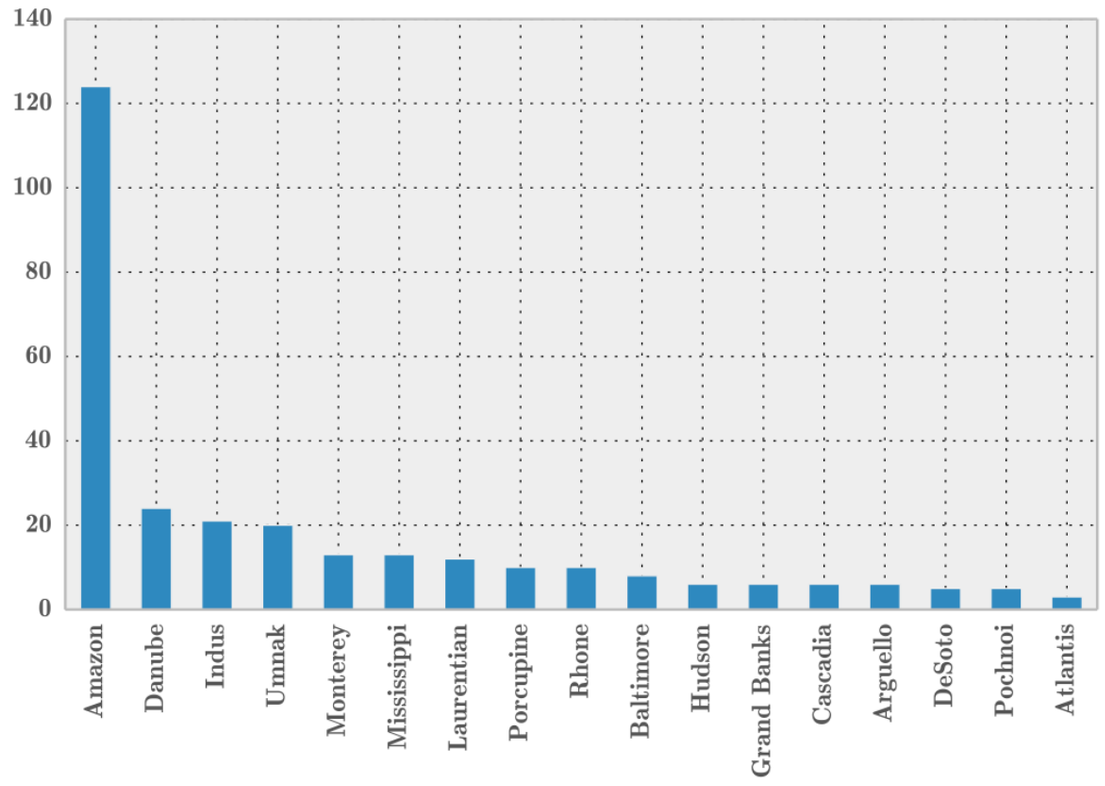
Sylvester, Z., Pirmez, C., Cantelli, A., & Jobe, Z. R. (2013). Global (latitudinal) variation in submarine channel sinuosity: COMMENT. Geology, 41(5), e287–e287. [doi:10.1130/G33548C.1]

Note:

This lab is based on the notebook and associated dataset from Sylvester, Z. (blog).

Dataset:

The majority of the data comes from the book by Clark and Pickering (1996), which also forms the basis of the analysis by Peakall et al. (2011). Data for the Amazon Channel comes from Pirmez and Flood (1995); and Popescu et al. (2001) (centerline digitized from map in paper and half-wavelength sinuosities calculated from centerline). See full references in papers.



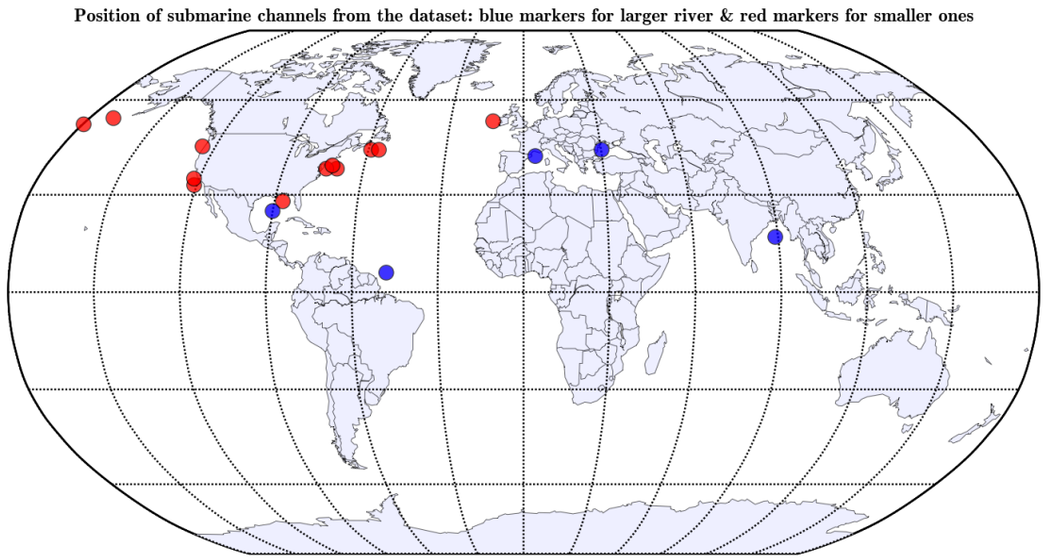
Quantitative analyses with IPython

Q1. What is the Coriolis force, and how it varies with latitude?

Open the ipython notebook *SubChannelSinuosity.ipynb* in a new web page. To work with the dataset we will use Pandas. **Pandas** is an easy-to-use data structures and data analysis tools for Python.

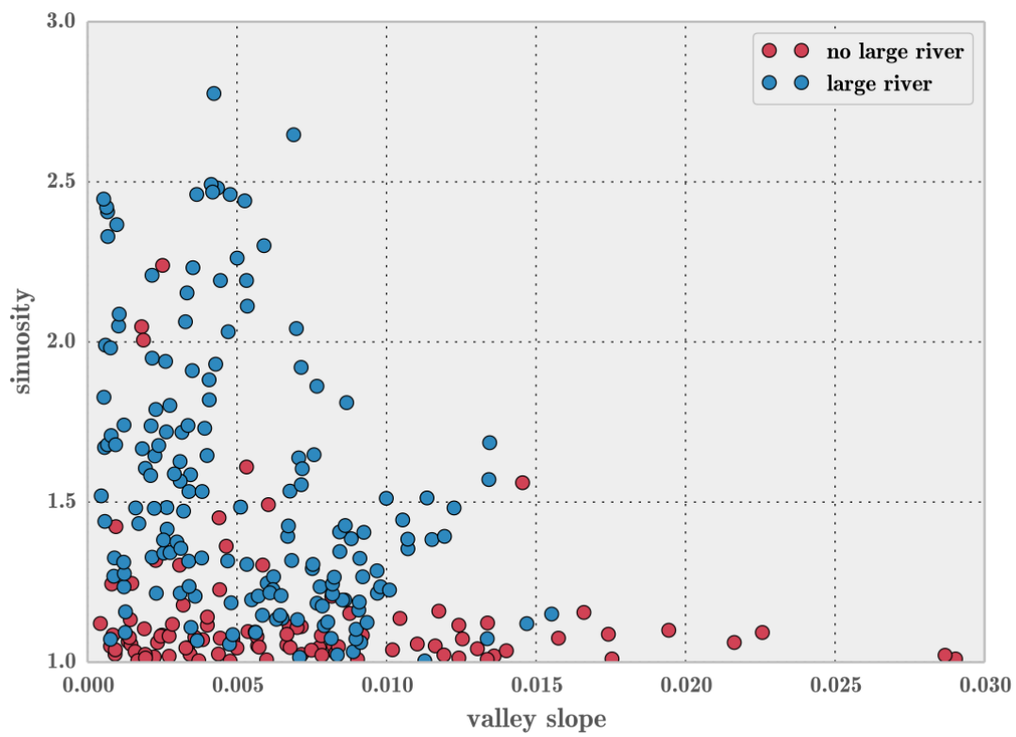
We first read the channel sinuosity dataset (csv file) and we plot the number of measure for each submarine channel system.

Q2. Explain why there are several measures for each submarine system?



We then load the second dataset containing the global coordinates of each of these systems (*canyon_loc.csv*).

From the combined dataset we will use **Basemap** library (remember the first part of this lab?) to visualise the global distribution of the dataset.



Q3. In the notebook the proposed script has been modified to plot only the large river systems. Find a way of plotting both types and to get the same map as the one above. Explain.

Sinuosity vs valley slope/latitude

From the dataset, we will first try to find if there are any variability in channel sinuosity with valley slope.

Q4. From the graph above, do you find any trend in the data-set that could help to understand the variability in sinuosity?

Q5. Using the script provided for the sinuosity versus valley slope write a new one to look at sinuosity versus latitude.

a. Do you find any trend in this new chart?

b. According to Peakall et al. (2011) what process could explain it?

Global river discharge dataset

In Sylvester et al. comment (2013), it is argued that another explanation could be that no significant sediment volumes reach deeper waters at higher latitudes due to the lack of large rivers that would provide the source.

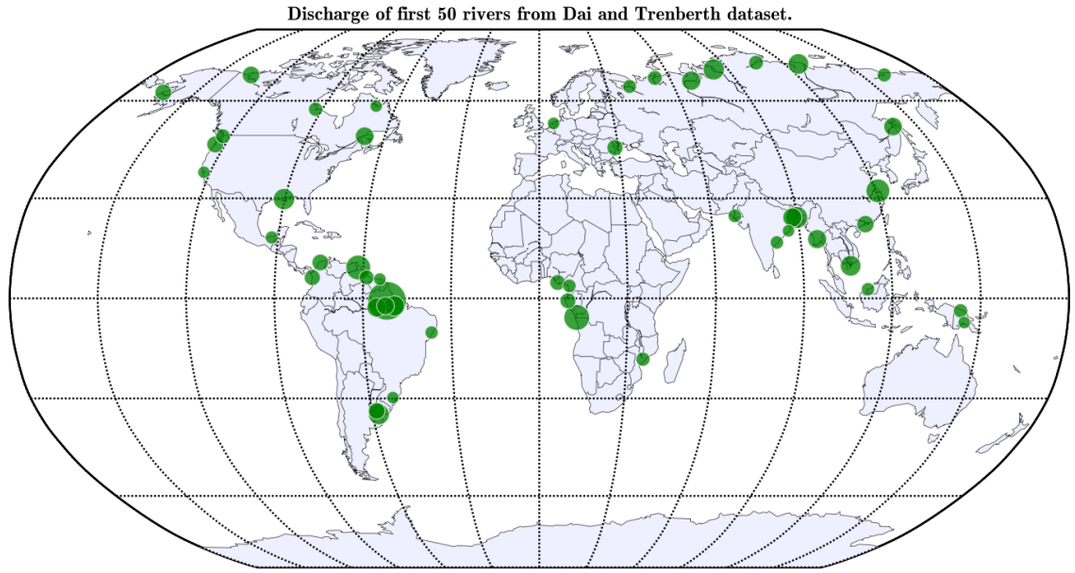
Using the dataset from Dai and Trenberth Global River Flow and Continental Discharge Database we will plot a world map of the discharge of largest rivers in the world using Basemap.

Once the dataset has been loaded into IPython we will work with only 4 columns:

- **lonm**: longitude
- **latm**: latitude
- **m2s_ra**: river mouth to station ratio of flow rate times 10,000
- **Vol**: station annual flow rate (km3/yr)

In order to get the river discharge we need to perform a simple manipulation of the dataset. The discharge D is given by:

$$D = \sqrt{Vol \frac{m2sra}{10000}}$$



Q6. Knowing that the squared root $\sqrt{}$ in IPython is given my **np.sqrt()** write the previous equation in the IPython cell. Copy/paste your python equation in your report.

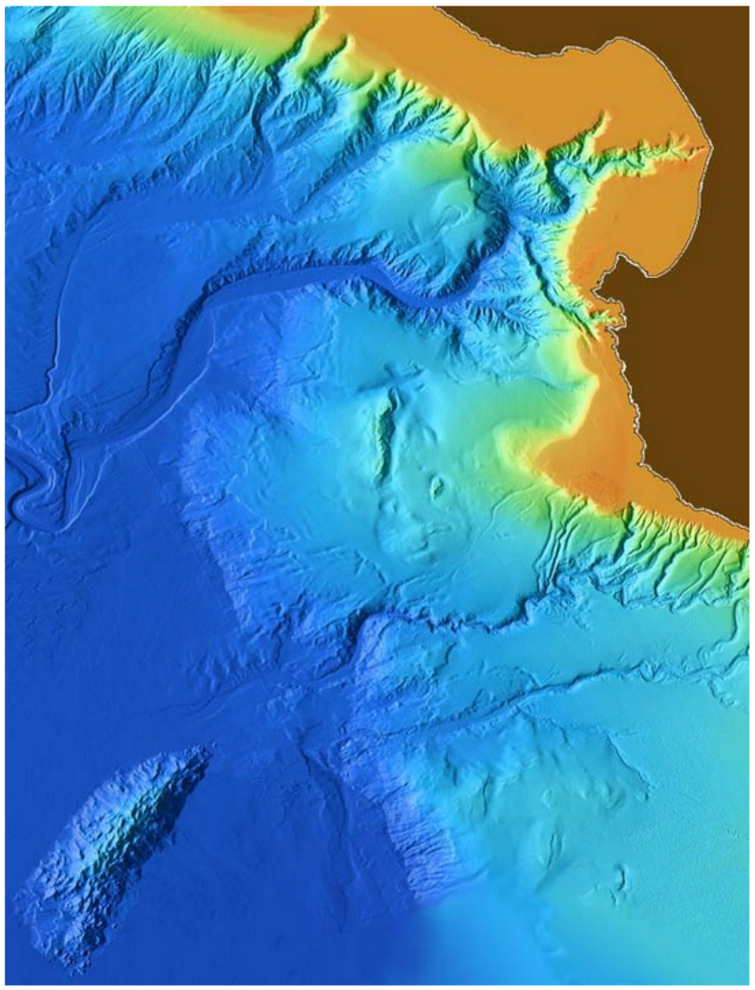
We then map the 50 biggest rivers discharge using Basemap.

Q7. In regards to Sylvester et al. comment (2013) and this last plot, can you see a relationship between river discharge and sinuosity variability?

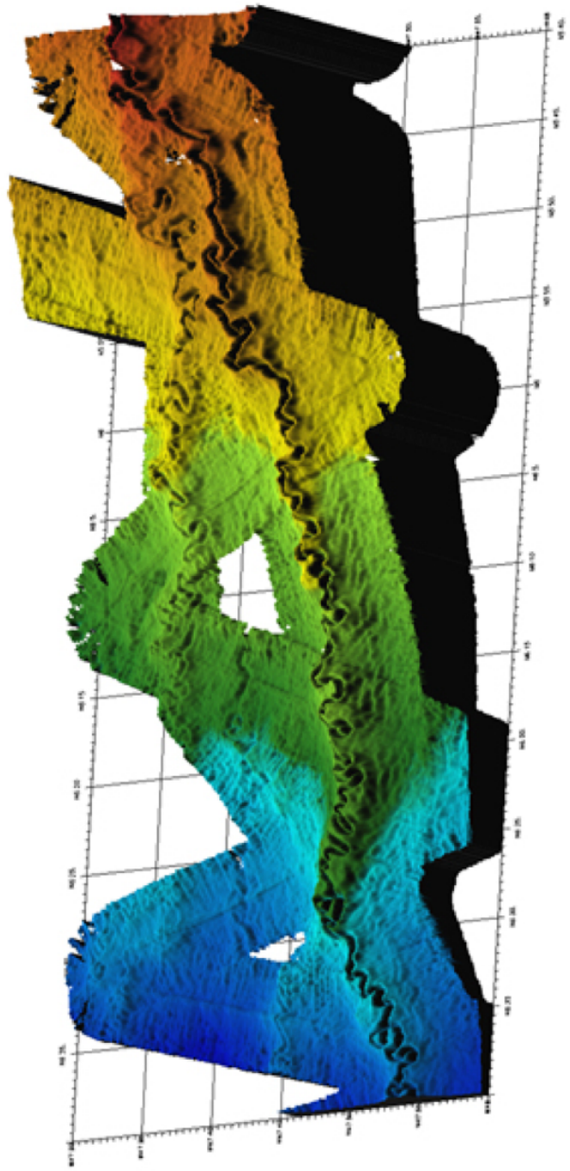
Interpreting: submarine channel morphologies

Q8. Based on what you have learn in this notebook and from both Peakall et al. (2011) and Sylvester et al. (2013), you might be able to have an initial guess on the most likely place where you might find these 2 types of channel systems. Explain.

Submarine channel system 1



Submarine channel system 2



Visualise in NbViewer

LAB Geological Processes 1 IPython material : nbviewer link

Coastal evolution model

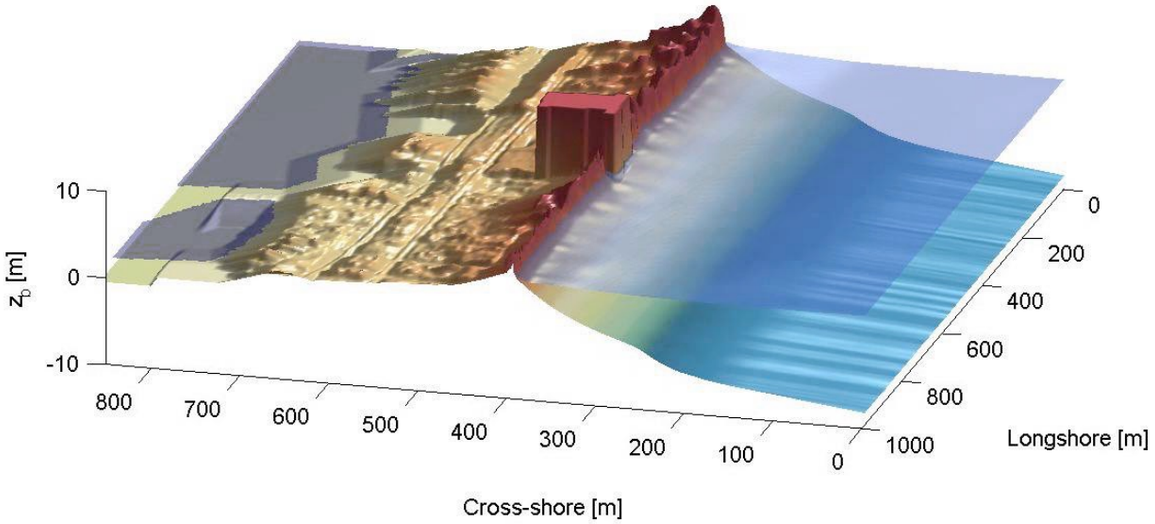
Instructions

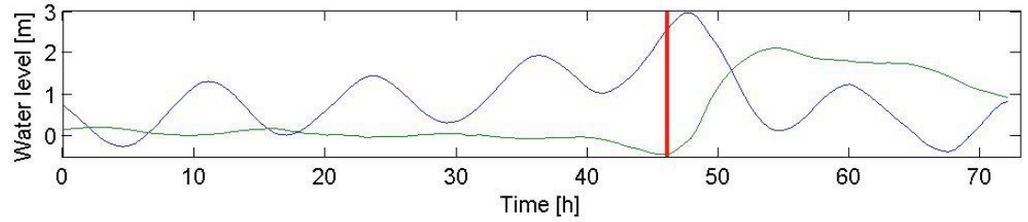
Using XBeach model we will explore the evolution of beach profiles subject to waves. We will design our own profiles and run several XBeach models. For this section, you will need to use IPython notebook to answer questions. Provide solutions.

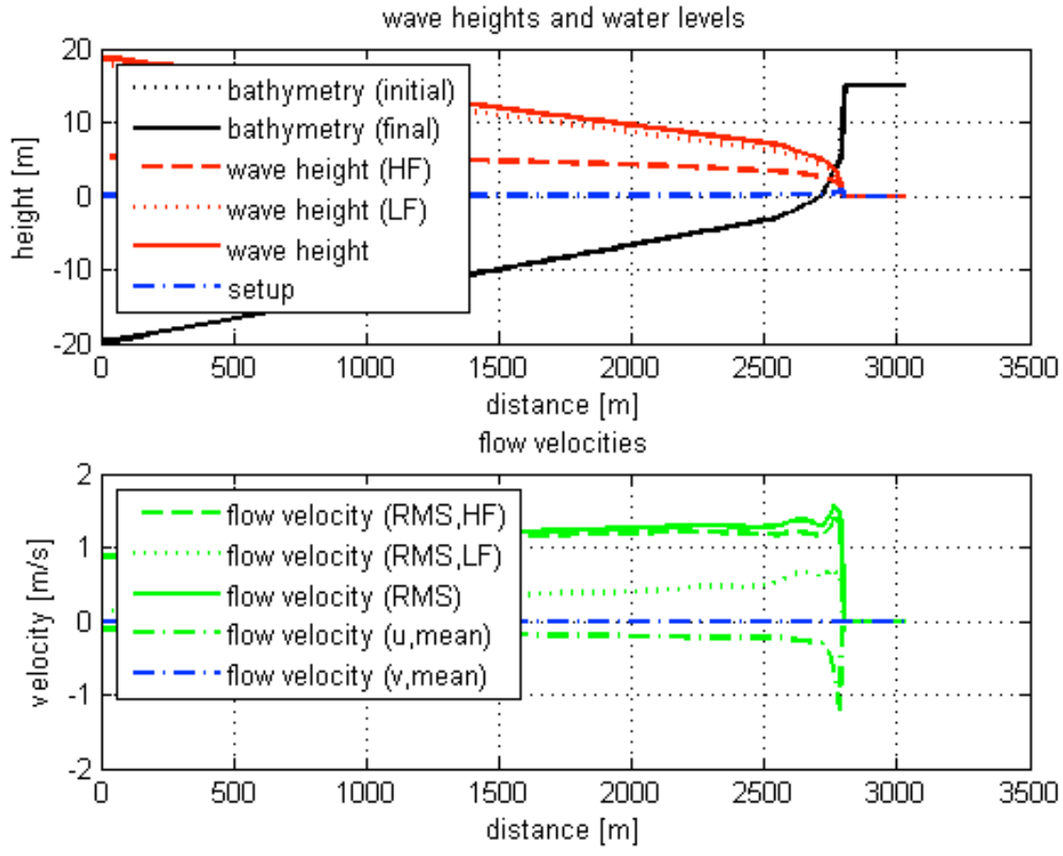
XBeach model

XBeach is an open-source numerical model which is originally developed to simulate hydrodynamic and morphodynamic processes and impacts on sandy coasts with a domain size of kilometers and on the time scale of storms.

The model includes the hydrodynamic processes of short wave transformation (refraction, shoaling and breaking), long wave (infragravity wave) transformation (generation, propagation and dissipation), wave-induced setup and unsteady currents, as well as overwash and inundation. The morphodynamic processes include bed load and suspended sediment transport, dune face avalanching, bed update and breaching. Effects of vegetation and of hard structures have been included. The model has been validated with a series of analytical, laboratory and field test cases using a standard set of parameter settings.



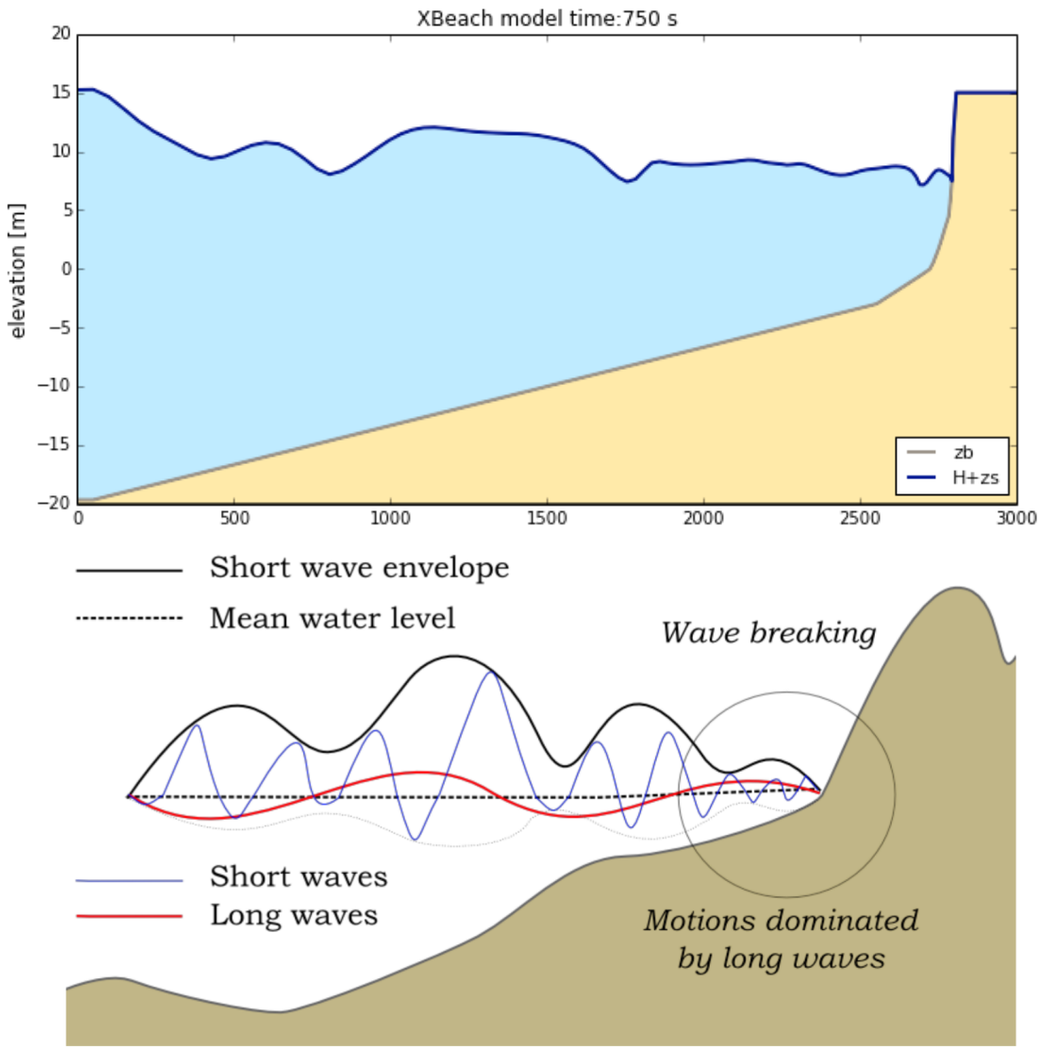




Beyond sandy coasts, the model has been applied to coral fringing and atoll reefs, in cooperation with and with funding by the University of Western Australia, the USGS and the Asian Development Bank.

It is recommended to look at the XBeach manual for an overview of the code capabilities as well as some understanding of the different parameters required to run the model.

Default model



The first model that we will run is a 1D beach profile experiencing over a duration of approximately 30 mins a wave regime define using a Jonswap spectra. The profile is divided in 2 sections: a smooth increase over the first 2500 m followed by a steeper slope when approaching the shoreline.

The underlying bed is supposed to be composed of only one type of sediment (sand) that will be subject to erosion and depsition due to wave erosion. The model handles both suspended and bed loads transport.

Running the model

To run this model you will need to execute XBeach in the directory which contains the model settings. This is the directory where the *params.txt* file is located which is called *default*.

From IPython server main directory page, open a **Terminal** by using the *New* tab on the top right hand side. It will open a Unix window that we will use to run the model.

Once open we need first to navigate to the default directory. This is done using the **cd** command:

$> cd /workspace/CoastalEvol/default

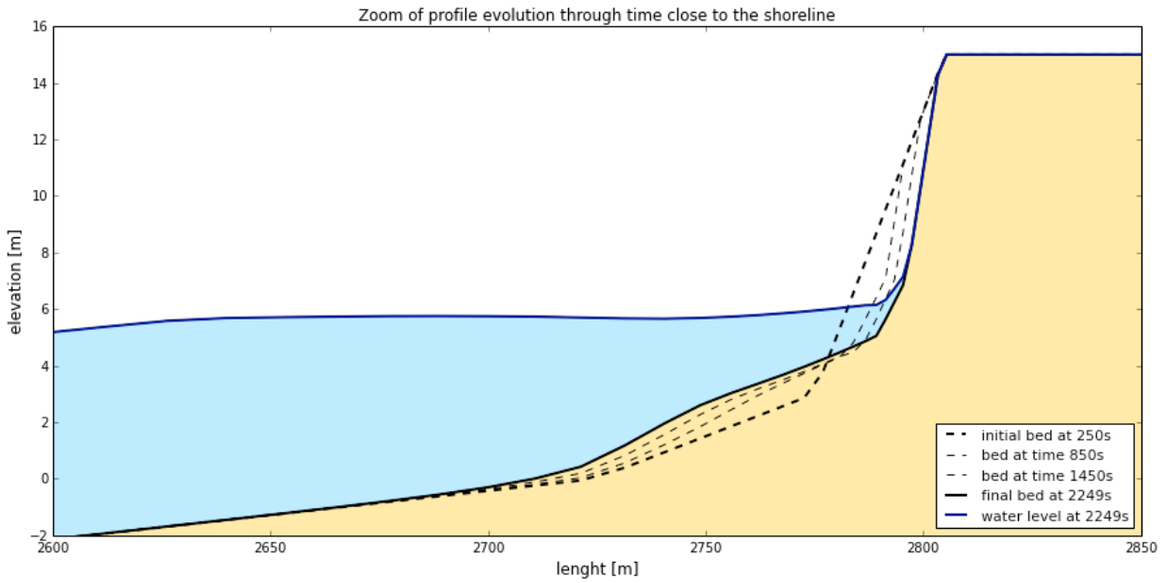
Press enter. You are now in the good directory to run XBeach simply type the following command

$> xbeach

You should see a bunch of informative lines on the window. If everything goes fine you should get your outputs in less than minute. XBeach is creating a netCDF file with all the information calculated during the run.

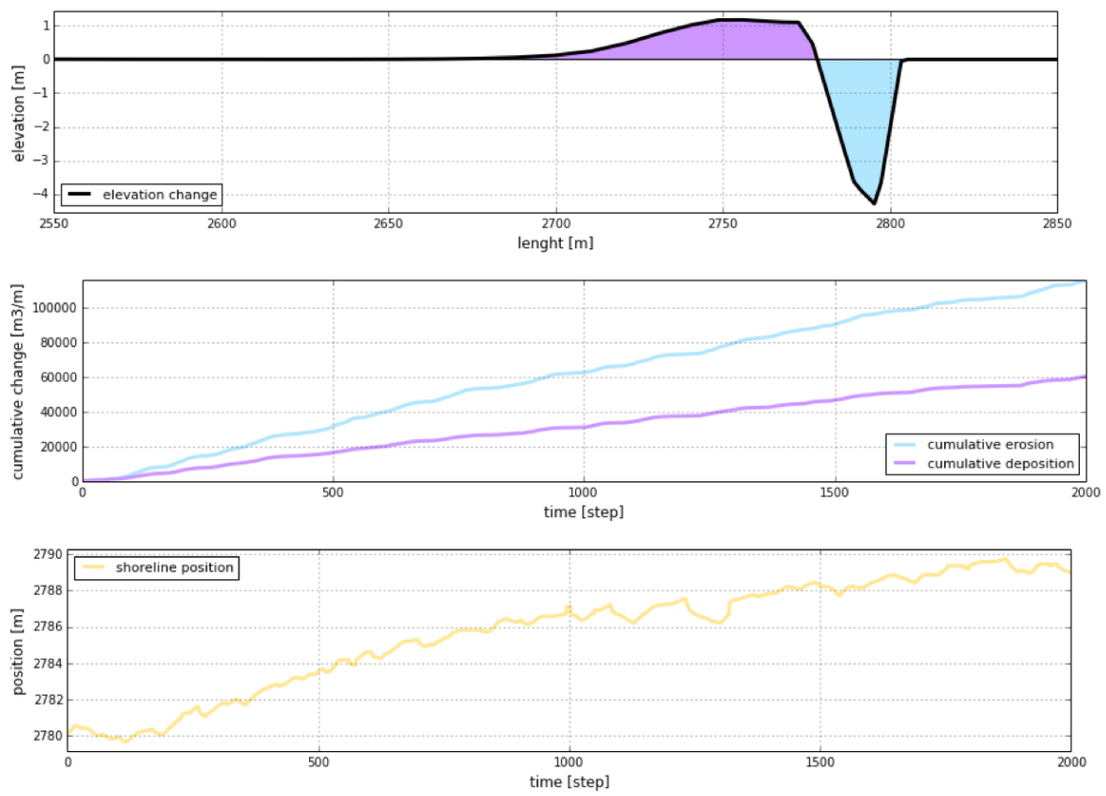
From the main directory page in the IPython server, navigate to the directory and check that the xboutput.nc file has been created.

Explore model outputs



Analysis methods for the following aspects of dune and beach morphology are given in the *BeachProfile* notebook:

- Profiles
- Hydrodynamics
- Sediment transports
- Morphodynamics



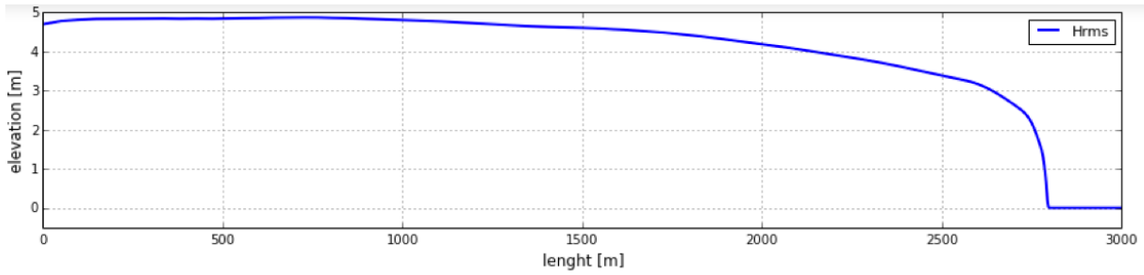
The collection of commands and screenshots provide an overview of what is possible.

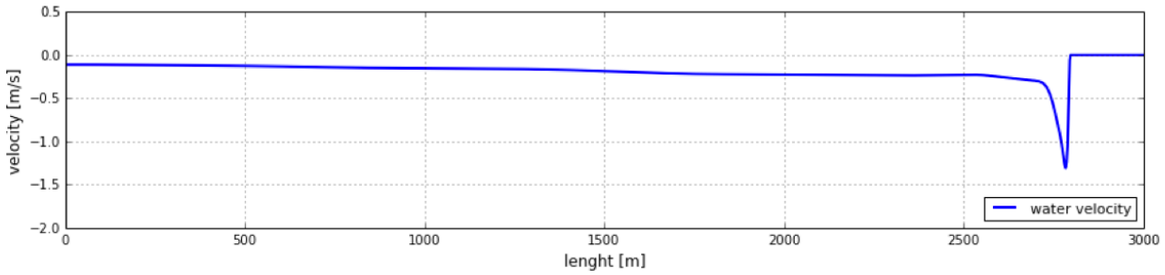
An example on how to setup your own profile is given in the *CreateProfile* notebook and can be used to define 2 sets of different profiles (1 smooth and another perturbed).

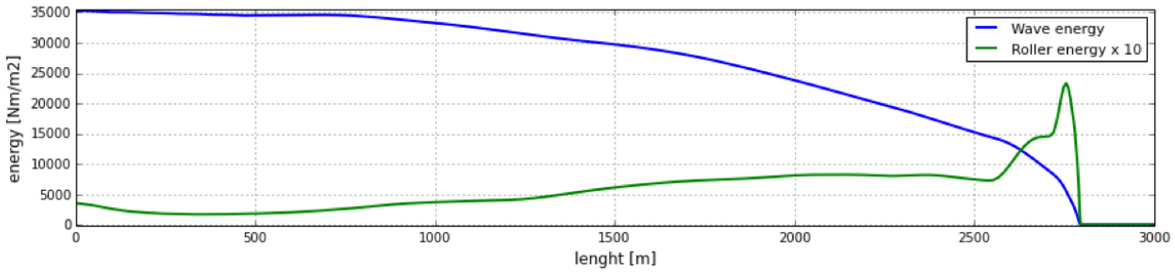
These examples are obviously illustrative of what can be done but of course, you are encouraged to write your own profile and create new analysis scripts!

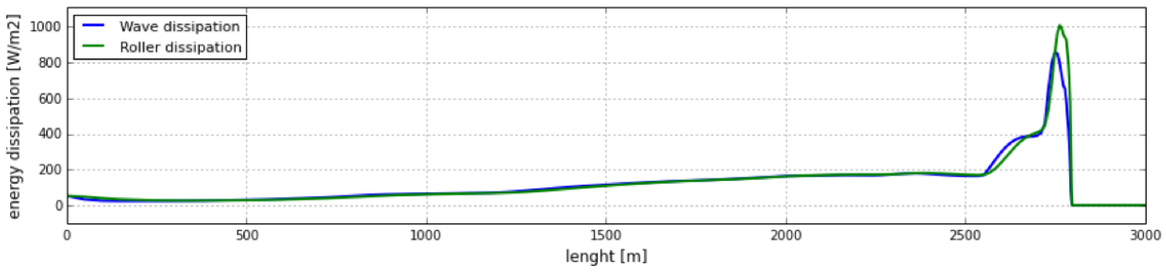
Q1. Using profiles 1 & 2 or 3 & 4 from the *CreateProfile* notebook or the ones that you have created, you will perform a comparative analysis of the hydrodynamics, sediment trans-

port and morphodynamics between each cases following what you did in the default example.









Visualise in NbViewer

Follow this link to see the IPython notebooks that you are going to use for this lab.

LAB IPython material 1: nbviewer link

LAB IPython material 2: nbviewer link

Instructions

We will explore the sensitivities of crenulate bay shorelines to wave climates using a vector-based one-line model called COVE. As for the other sections, there are several questions (Q.) in the following pages, we will use some IPython notebook to answer these questions. Provide solutions for all of them.

Shoreline one-line model

One-line models are coastal evolution models in which the coast is represented by a single line.

One-line coastal models make a number of simplifying assumptions in order to conceptualize the coast.

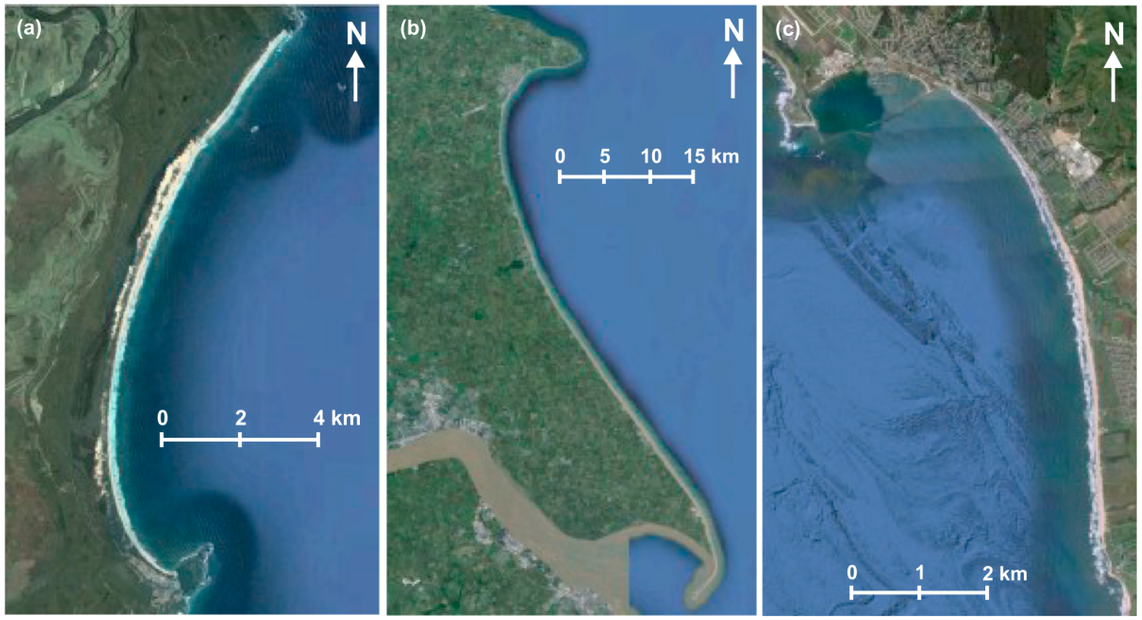
- First, short-term variations due to storms or rip currents, which tend to act in the cross-shore direction, are considered as temporary perturbations to the long-term trend of coastal change, causing fluctuations in shoreline position. As such the beach profile is assumed to maintain a constant time-averaged form, implying that depth contours are shore-parallel.
- Alongshore sediment transport occurs primarily in the surf zone, and it is assumed that cross-shore sediment transport

acts to maintain the equilibrium shoreface as it advances or retreats.

- Finally, it is assumed that alongshore sediment transport is driven by the delivery of energy to the surf zone, parameterized by the height and angle of incidence of breaking waves. Waves that impinge obliquely on the coastline will cause downdrift transport, and a variety of formulations are available to describe the relationship between wave conditions and the magnitude of alongshore sediment flux

One-line modeling of the evolution of sandy and "soft sediment" coastlines has proved an excellent exploratory tool to examine the dynamics of coastline evolution at mesoscales

Crenulated bay



Examples of crenulated bay shapes at different scales. (a) Hathead Bay, Eastern Australia. (b) Flamborough Head and the Holderness Coastline, East Yorkshire, UK. (c) Half Moon Bay, California, USA.

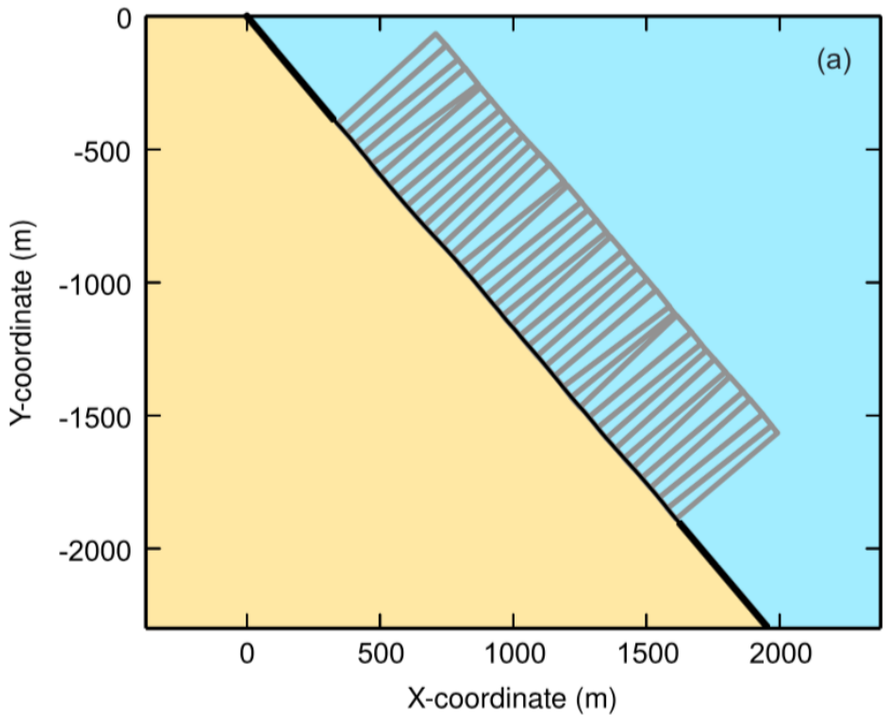
The curved planform morphology of embayed beaches can be observed at various length-scales at coastlines, from a few hundred meters to several kilometers. These log spiral-shaped, crenulate-shaped, hook-shaped, or zeta-shaped bays occur in the lee of headlands or man-made coastal structures where erosion and/or littoral drift is inhibited in the face of a dominant direction of wave incidence. A highly concave portion of shoreline forms on the downdrift side of the headland where the coastline is shadowed from the dominant wave direction and subject to waves that diffract around the headland.

Embayed beaches tend toward an equilibrium form under a prevailing wave climate. The planform morphology will adjust until gradients in alongshore sediment flux are minimized (net alongshore sediment flux is constant). Alongshore sediment flux will be negligible on an equilibrium coastline when there are no external sediment inputs. Subsequent changes in planform morphology may occur such as beach rotation, driven by changes in wave climate characteristics that alter alongshore sediment transport.

COVE model

The COVE model, that we will use in this notebook, conserves mass while still allowing high planform-curvature shorelines to be modeled. We will use the model to study the evolution of embayed beaches proximal to fixed headlands in order to explore the influence of a variable wave climate on predicted bay morphology and stability.

Initial conditions



Coastal cell geometry for a model instance at the beginning of a model run starting with a straight coastline with superimposed low amplitude noise in the coastline position

Boundary conditions

Initial model coastlines are straight and 2 km in length with an initial node spacing of 50 m.

The bounding ends of the coastline are fixed to represent headlands or fixed structures such as sea walls.

No sediment transport is permitted into the model domain across these boundaries, but sediment is permitted to escape out of the model domain by alongshore transport.

Wave climates

We will use for our tests an idealized wave climate. The offshore wave directions θ_0 are drawn from Gaussian functions defined by mean θ_{mean} and standard deviation θ_{std}. Offshore wave height H_0 and period T are also described using a narrow Gaussian function with H_{mean} and standard deviation H_{std}, and T_{mean} and T_{std}.

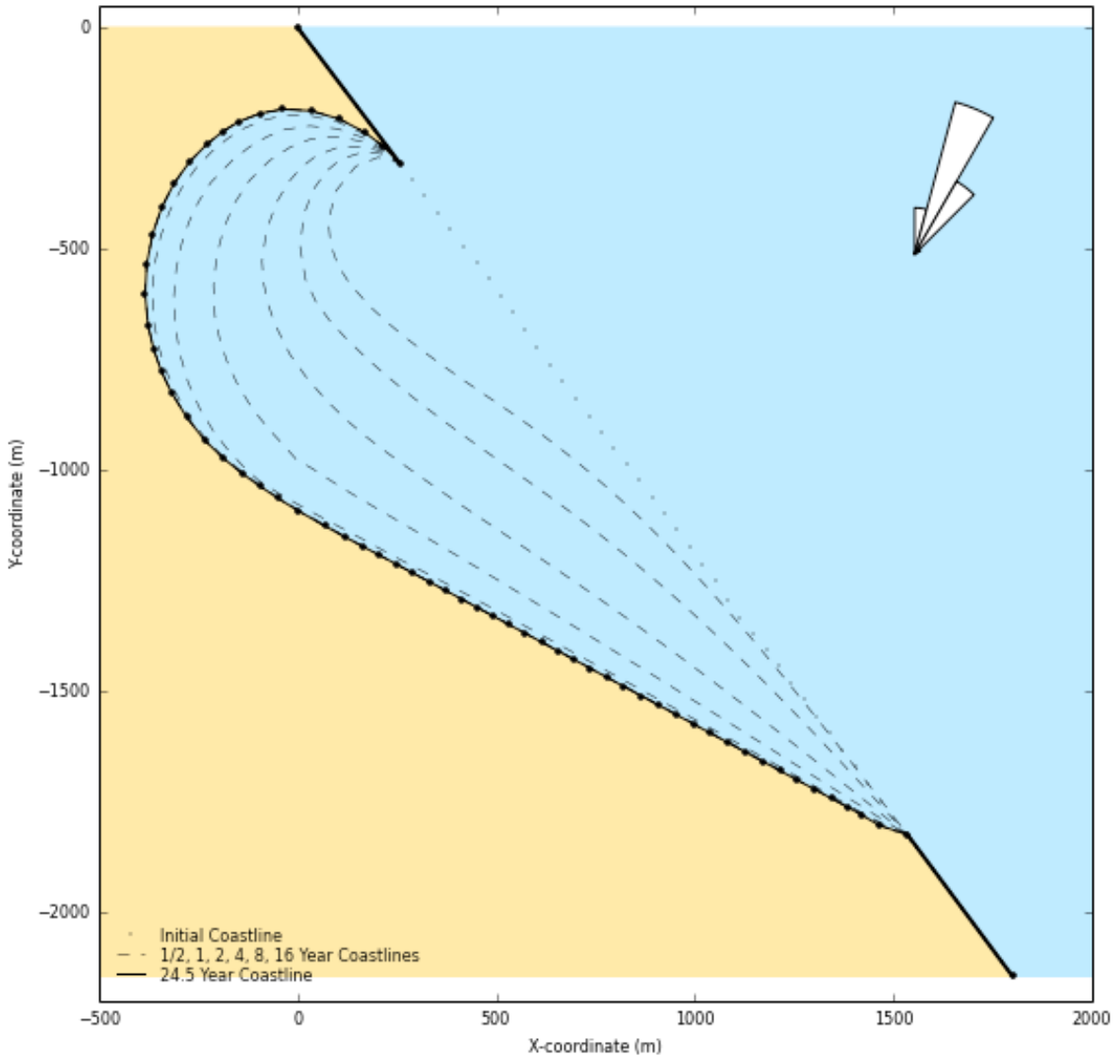
Running the COVE model

Open the BaySensitivity IPython notebook in the Jupyter server and start by running your first model using the following values for each parameter:

- $\theta_{mean} = 25^o$, $\theta_{std} = 10^o$
- $H_{mean} = 1m$, $H_{std} = 0.1m$
- $T_{mean} = 6s$, $T_{std} = 1s$

The model has been set to run over 20 years to ensure profile equilibrium. Using the notebook you will create a similar graph to the one on the next page of this lab notes.

Exploring shorelines sensitivity to wave climates



Taking this first model as an example you will now explore how the wave direction/spreading is influencing the development of this bay through time.

For the next simulations the following parameters are kept constant:

- $H_{mean} = 1m, H_{std} = 0.1m$
- $T_{mean} = 6s, T_{std} = 1s$

Q2. Run a series of model with $\theta_{mean} = 30^o$ and variation of spreading angles θ_{std} using successively a value of $10^o, 25^o$ and 40^o. For each of these angles plot the shoreline evolution over the 20 years of the simulation.

Q3. What is the effect of increasing the spread of wave directions for the embayment at the shadowed end of the bay. What is the reason for that?

Q4. Run a series of model with $\theta_{std} = 25^o$ and variation of dominant wave directions θ_{mean} using successively a value of 10^o and 40^o. For each of these angles plot the shoreline evolution over the 20 years of the simulation.

Q5. Using the two new models plus the one from Q1 with $\theta_{std} = 25^o$, explain what is the effect of increasing the obliquity of wave approach angles? What is the reason for that?

Visualise in NbViewer

Follow this link to see the IPython notebook that we are going to use for this lab.

LAB IPython material : nbviewer link

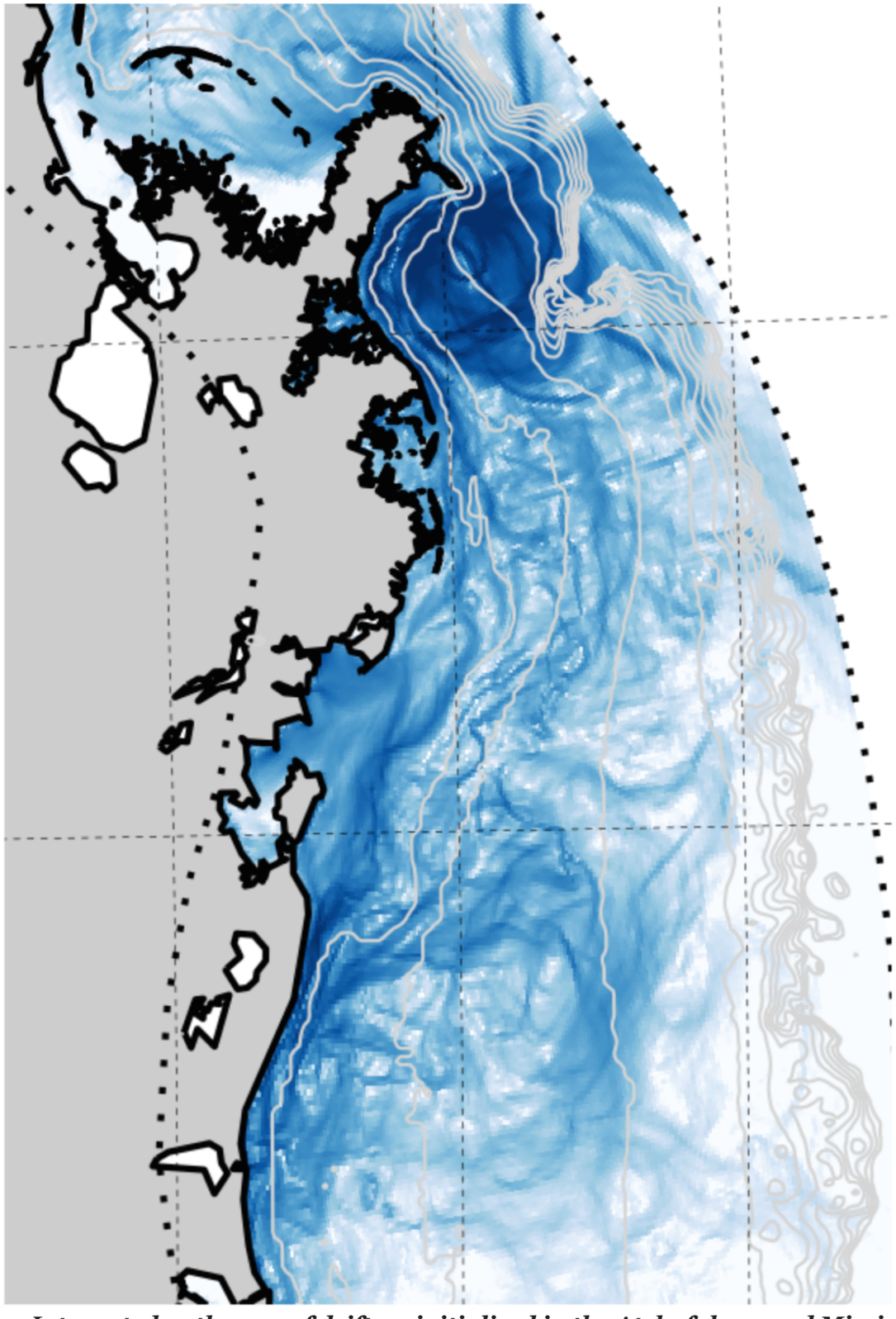
Instructions

Using a Lagrangian trajectory model (tracmass/tracPy) on the northern part of the Gulf of Mexico, you will simulate the tracks of several drifters at different locations and times. Two scenarios are proposed in this section. For each of them you will need to use IPython notebook to answer the questions. Provide solutions for all of them.

Drifters modeling

Drifters are used in oceanography and atmospherics in situ in order to demonstrate flow patterns created by individual fluid parcels. For example, in the ocean, drifters will often be released on the sea surface, and allowed to be passively transported with the flow, reporting their location via GPS at regular intervals. In this way, drifters are gathering data in a Lagrangian perspective. For example, one can analyse a set of surface drifters released in the northern Gulf of Mexico, using the tracks to better understand the dynamics of the underlying circulation fields.

Lagrangian trajectory modeling is a method of moving parcels through a fluid based on numerically modeled circulation fields. This approach enables analysis of many different drifter experiments for a much lower cost than is required to gather one relatively small set of drifters. Drifters are often used in on-going response work by the Office of Response and

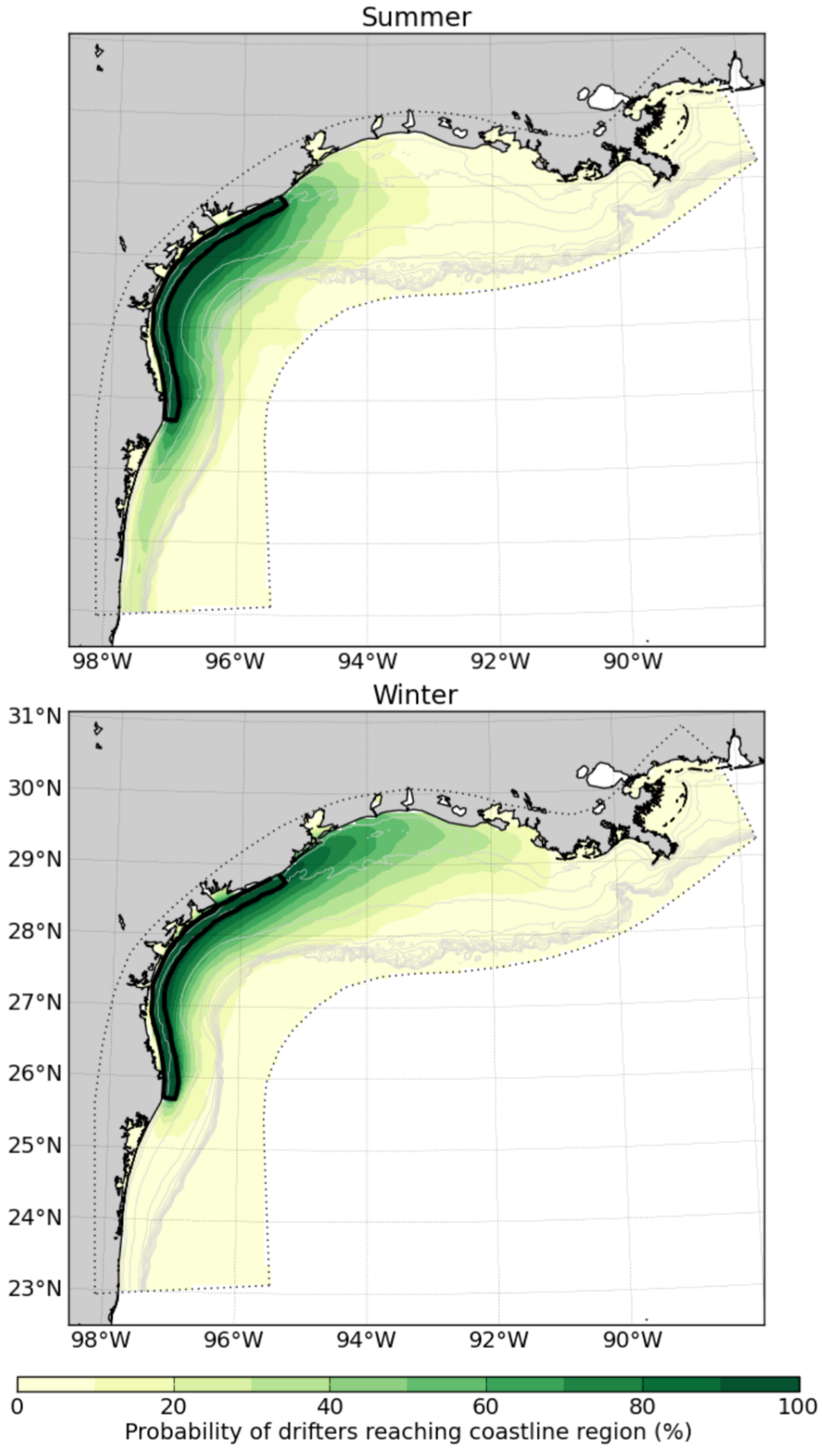


Integrated pathways of drifters initialized in the Atchafalaya and Mississippi river inputs to the numerical domain.

Restoration in the National Oceanic and Atmospheric Administration (NOAA). Using model output made available by various groups, responders apply their tool (General NOAA Oil Modeling Environment, GNOME) to simulate drifters and get best estimates of predicted oil transport.

Numerical drifters may be calculated online, while a circulation model is running, in order to use the highest resolution model-predicted velocity fields available in time (on the order of seconds to minutes). However, due to the high costs of the hydrodynamic computation, many repeated online simulations is not usually practical. In this case, Lagrangian trajectories can also be calculated offline, using the velocity fields at the stored temporal resolution (on the order of minutes to hours).

There are many sources of error in simulating offline Lagrangian trajectories. For example, the underlying circulation model must be capturing the dynamics to be investigated, and model output must be available often enough to represent the simulated flow conditions accurately. On top of that, the Lagrangian trajectory model must properly reproduce the transport pathways of the system. A given drifter's trajectory is calculated using velocity fields with a spatial resolution determined by the numerical model grid. To move the drifter, the velocity fields must be available at the drifter's location, which in general will not be co-located with all necessary velocity information. Many Lagrangian trajectory models use low- or high-order interpolation in space to extend the velocity information to the drifter location.



Tracmass is a Lagrangian trajectory model that runs natively on velocity fields that have been calculated on a staggered Arakawa C grid. Originally written about 2 decades ago, it has been used in several applications. The core algorithm for **Tracmass** is written in Fortran for speed, and has been wrapped in Python for increased usability. This code package together is called **TracPy**.

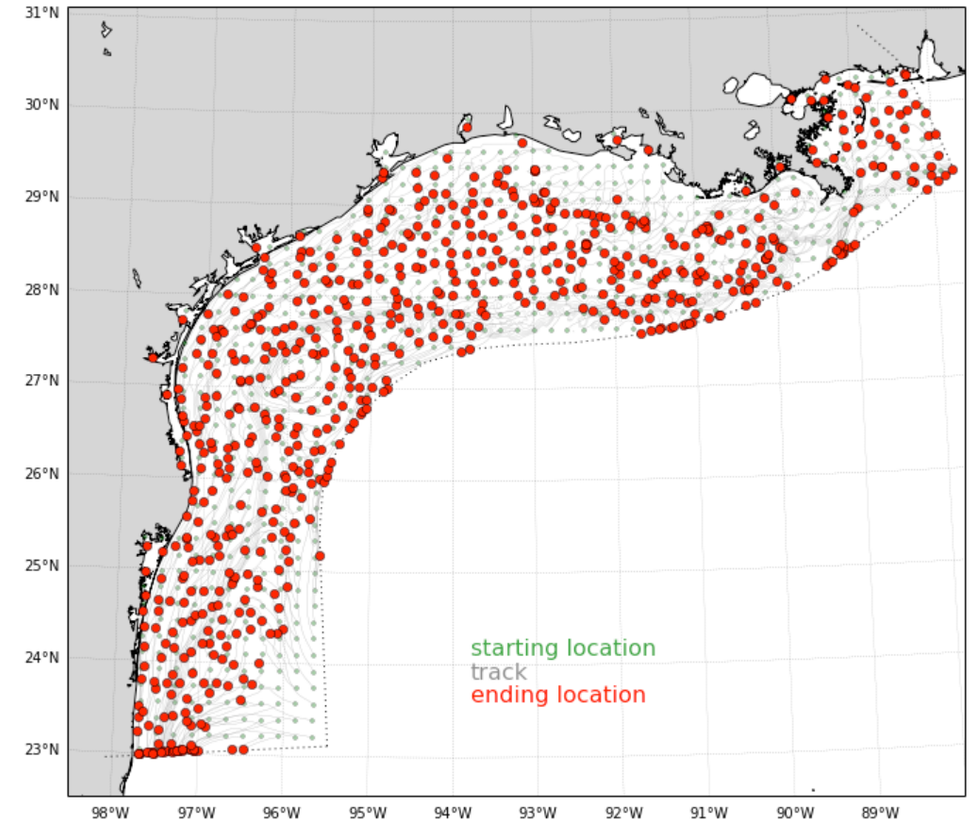
Using TracPy wrapper

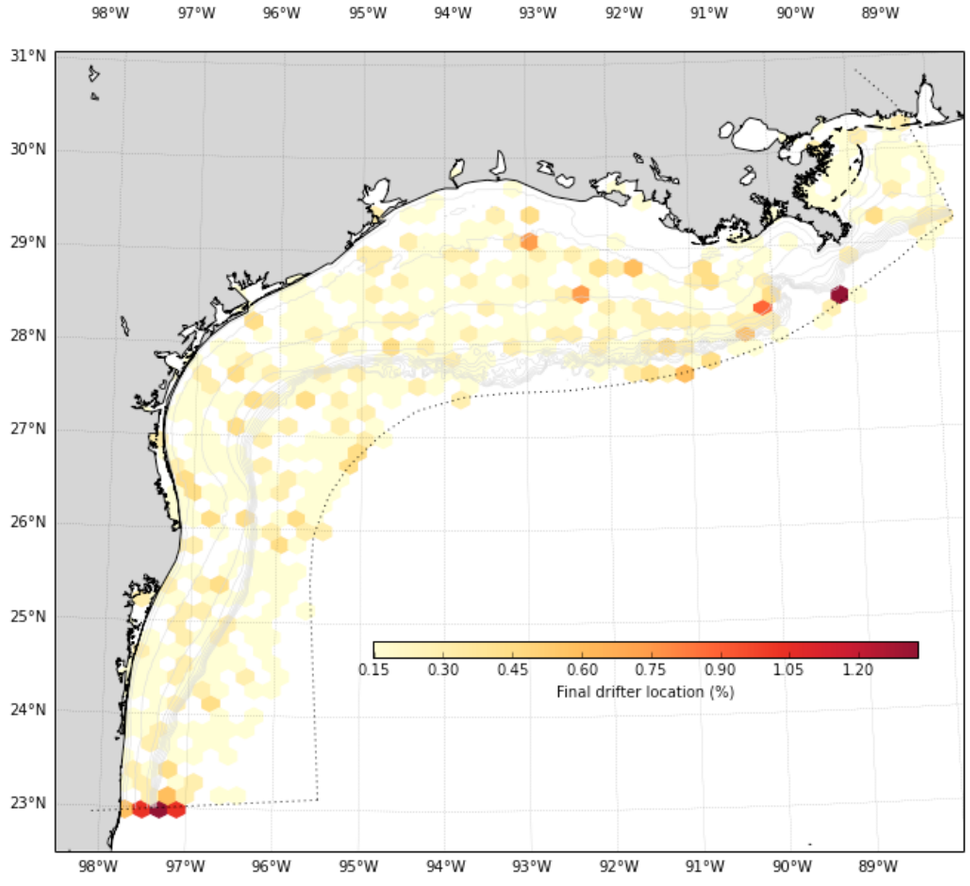
You will open the Drifter notebook in your local directory on the Jupyter server and start modeling the trajectory of some drifters located on the northern part of the Gulf of Mexico.

The underlying hydrodynamics that we will use are coming from a high resolution model of the Texas-Louisiana shelf for the years 2004-2012. **Tracmass** is able to read outputs from several well-used models, including **ROMS**, **MITgcm**, and **HyCOM**. Here we will get the results from **ROMS** (Regional Ocean Modeling System) model. The dataset is stored on a *thredds* server and is a netCDF file.

Following the IPython notebook you will see how the different parameters required to run the trajectory model need to be defined.

This first example consist in imposing a series of more than 2500 drifters on the numerical domain and looking at each individual track over a period of 3 days from November 25, 2009.





Tanker Ship Accidents

Suppose we are the 2nd of April 2009 and you are a coastal scientist working in the NOAA Oil Recovery Environment. You have been advised by the Texas Coastal Water Authority (CWA) of 3 tanker ships accidents in the bay happening at 0:00 AM, 10:00 AM and 6:00 PM on the 1st of April.

Texas CWA is asking your team for help in deciding which of these 3 accidents could have the most problematic impact on the coasts.

Based on the response strategy timeframe, your team has decided to use a 2 weeks window to analyse the potential trajectories for each of these oil spills. For each of the oil spills locations you have decided to put 100 drifters to take into account the spatial variability of the ocean dynamics in each place.

Here is the report from Texas CWA:

- *accident 1: happened at 0:00 AM, the ship was located in a area extending from (92W,28N) on the SW corner to (91.75W,28.5N) on the NE corner.*
- *accident 2: happened at 10:00 AM, the ship was located in a area extending from (97W,27N) on the SW corner to (96.75W,27.5N) on the NE corner.*
- *accident 3: happened at 6:00 PM, the ship was located in a area extending from (92W,28.5N) on the SW corner to (91.75W,29N) on the NE corner.*

Q6. Using a 24 hours time outputs for **TracPy** and keeping the other variables unchanged, provide for each of these accidents a plot of the oil spill trajectory in the next 15 days. Which of these accidents might be the more problematic?

Q7. If accident 2 occurred at the same time on the 1st of January 2009 instead, what would have been the drifters trajectories?

Sea Turtles Tracking

The 25th of July 2008, the Sea Turtle Stranding and Salvage Network (STSSN) has documented large numbers of sea turtles washed up on the north-central Gulf of Mexico.

The beaches are located in an area extending from (94.2W,29.5N) on the SW corner to (94.1W,29.8N) on the NE.

The majority of these turtles have been endangered juvenile green turtles of approximately 1 month old. At this age, these turtles usually follow main ocean currents.

As an expert in ocean dynamics within the STSSN, you have been asked to find their possible birth beaches for future marine life protection planning.

Q8. Using **TracPy** and 200 drifters (10 times 20 along the EW and NS directions respectively), track backward in time the migration of these turtles. Provide a plot of the computed trajectories. Which is the possible location where these turtles were born?

Visualise in NbViewer

Follow this link to see the IPython notebook that you are going to use for this lab.

LAB IPython material : nbviewer link