## GLORYS to Atlantis Oceanographic Data Translation

### Part 1: GLORYS oceanic climate change projections using delta method downscaling

**Purpose:** Use a “delta method” to produce high-resolution oceanographic projections for multiple physical and biogeochemical variables, by translating coarse Earth System Model (ESM) outputs onto a higher resolution GLobal Ocean ReanalYsis and Simulation (GLORYS) grid. The GLORYS model is an ocean reanalysis, and does not provide projections. Hence, the general idea is to calculate the difference between ESM projections and a historical long-term mean (i.e., an ESM climatology), then apply those differences (the “delta”s) to the historical GLORYS climatology. In other words, we apply the changes from the projected ESMs to the GLORYS climatology to produce high-resolution projections.

This method has five major steps:

1. Calculate an ESM climatology on the ESM grid for each variable
2. Calculate an ESM delta (future minus ESM climatology)
3. Calculate a GLORYS climatology on the GLORYS grid
4. Convert the ESM delta to the GLORYS grid
5. Add the converted deltas to a GLORYS climatology

R scripts are in the `scripts` directory of atlantisserver05. The R scripts use command line calls to access the Climate Data Operators (https://code.mpimet.mpg.de/projects/cdo/)

**Variables**

Currently, there are 8 variables that are common to the 3 ESMs (IPSL, GFDL, and Hadley) and GLORYS. See `GLORYS\_and\_ESMs\_information.xlsx` in this directory:

* Biogeochemical variables (annual resolution in the ESMs, monthly resolution in the GLORYS data): Dissolved silicate (si), dissolved oxygen (o2), nitrate (no3)
* Physical variables (monthly resolution the ESMs and GLORYS): Sea water potential temperature (thetao), sea water salinity (so), eastward sea water velocity (uo), northward sea water velocity (vo), sea surface height above the geoid (zos)

It is important that variable units be checked for consistency between models. For example, values for no3, o2, and si are in mol/m3 in the ESMs, and mmol/m3 in GLORYS, so a conversion needs to be made.

**Historical reference period**

The “historical” time period is 1976-2005 for the ESMs and 1993-2018 for GLORYS. However to align timelines for the two oceanographic products, we append the projected years 2006 to 2018 to each historial ESM run, then choose the same years as GLORYS (1993-2018) for the calculation of both the ESM and GLORYS climatologies. Therefore, the climatology used for calculation of delta values is 1993-2018, and then the delta-method projections are made both forwards and backwards, resulting in a timeseries from 1976-2100—125 years of projected data.

By legacy convention, our California Current Atlantis model starts in 2013.

**1. ESM climatology**

Script: calc\_esm\_deltas.R

Append the years 2006:2018 to the 1976-2005 baseline for each ESM. Climatology is then calculated on the years 1993-2018, to match with the historical GLORYS period.

For each variable, calculate the long-term mean across the entire dataset, for each grid point (latitude/longitude/depth). Calculate a monthly mean for monthly-modelled variables or single mean for annually-modelled variables.

**2. ESM deltas**

Script: calc\_esm\_deltas.R

Calculated in the same command as step 1. Subtract the ESM climatology from each projected future year to produce a timeseries of delta values for each variable.**Option: for chlorophyll, calculate a proportional rather than an additive delta.** Divide each projected year by the climatology to obtain a “proportional delta”.

**3. GLORYS climatology**

Script: calc\_glorys\_climatology.R

Calculated in the same manner as in Step 1, for the entire GLORYS timeseries. Result is a monthly climatology for each GLORYS variable.

**4. Grid conversion**

Script: warp\_esm\_deltas\_glorys\_grid.R

Warp ESM deltas onto the GLORYS grid structure, to match the GLORYS lat/lon/depth slices. Each GCM grid is slightly different, but all are warped to the GLORYS grid using the same method. Distance-weighted interpolation from the four nearest neighbors is used for the horizontal grid conversion (CDO operator –remapbil), and linear interpolation for depth layers (CDO operator –intlevel).

**5. Calculate final output**

Script: add\_esm\_deltas\_glorys\_climatology.R

Add the grid-warped ESM deltas for each variable from Step 4 to the GLORYS climatology from Step 3 to create final converted datasets (for chlorophyll, we multiply). The final outputs for each variable are split into separate netCDF files for each year to prevent enormously unwieldy files.

Currently, all final outputs from Step 5, as well as warped ESM deltas from Step 4, are in the ‘delta\_method’ folder. Files are named as:

“[ESM]\_glorys\_[variable]\_[year].nc”

Examples:

“IPSL-CM5A-MR\_glorys\_zos\_2035.nc” contains the final output for sea surface height, based on the IPSL GCM, for the year 2035. The file has one variable (`zos\_glor`) with grid dimensions latitude, longitude, and 12 time steps (1 for each month of 2035).

“GFDL-ESM2M\_glorys\_thetao\_2095.nc” contains the final output for sea water temperature, based on the GFDL GCM, for the year 2095. The file has one variable (`thetao`) with grid dimensions latitude, longitude, and 12 time steps (1 for each month of 2095).

### Part 2: Translation onto Atlantis polygons

**Purpose:** After part 1, we have downscaled, projected oceanographic data for many variables and multiple different ESMs. The second challenge is making these data useful and readable for the Atlantis model. This means that we need to match the GLORYS geometry (3-dimensional grid raster/ncdf) to the Atlantis 3-dimensional box structure (i.e., many differently-sized polygons with multiple depth layers). These files form the basis of the Atlantis-required “forcing.prm” file, which governs the baseline oceanographic drivers of the ecosystem model.

One of the biggest challenges in this translation is translating point-estimate currents (*u* and *v* in the oceanography) into volumetric flows between Atlantis boxes and among different depth layers within single Atlantis boxes. The details of Atlantis geometry and general process for the incorporation of oceanographic data into Atlantis forcing files are provided in the Atlantis User Manual (available from [here](https://research.csiro.au/atlantis/home/about-atlantis/)), but here we describe the logic and calculations made specifically for the GLORYS to Atlantis translation for the California Current.

Beginning with the outputs of Part 1 (individual years of NetCDF output for state variables like temperature and salinity, and for fluxes *u* and *v*), we take the following general steps to produce NetCDF input forcing files for Atlantis:

1. Create matching keys that we can use to spatially join GLORYS grid points to Atlantis polygons and “faces” (interfaces between adjoining polygons), in three dimensions
2. Interpolate delta method GLORYS oceanographic outputs by depth to match the midpoints of Atlantis polygon depth layers.
3. Extract GLORYS from NetCDF and join data to the Atlantis geometry using the keys produced in Step 1. For state variables (temperature, salinity), calculate mean values by Atlantis box, depth layer, and time step.
4. For translation of fluxes, join *u* and *v* to Atlantis polygon faces (see details below). For each face and depth layer in each timestep, calculate the magnitude and direction of the net flux of water directly across the face—i.e., the flux orthogonal to the angle of the face.
5. Optionally, apply a correction for hyperdiffusion (see details below). Then, re-organize all fluxes by their “source” and “destination” Atlantis boxes/depths, and summarize such that we are left with a record, in each time step, of all exchanges of water between individual Atlantis boxes and depths.
6. Pack the above—mean values by time step, box, and depth for state variables, mean fluxes by time, source box, depth, and destination for fluxes—into NetCDF output that can be read directly by Atlantis as forcing files.

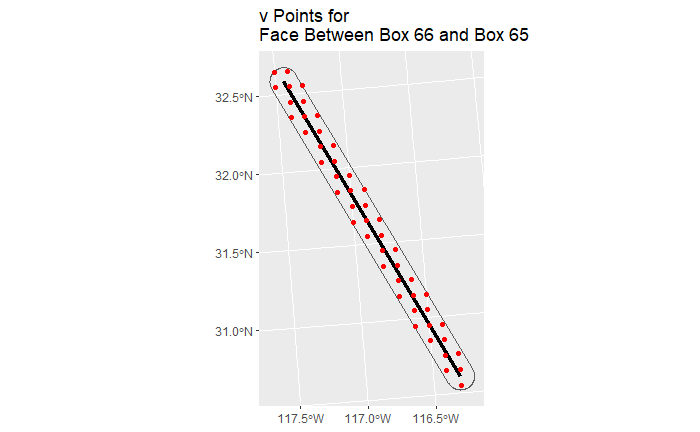
As in part 1, the following provides more detail on these general translation steps, with associated code scripts. The overall processing script is called `run\_GLORYS\_Atlantis\_translation.R`, but most of the analytical data work and translation functions are in two other scripts, ‘GLORYS\_Atlantis\_preprocessing.R’ and, ‘GLORYS\_to\_Atlantis\_fxns.R’.

**1. Matching Keys**

Script: GLORYS\_Atlantis\_preprocessing.R

We matched (GLORYS) points to (Atlantis) polygons using spatial joins. Because of the resolution of the GLORYS grid, it is possible that small polygons would not contain any points. To mitigate this issue, we relaxed the spatial join such that any point falling within a polygon *or within 10km* *of the polygon* would be matched.

For matching points to polygon faces, we used a buffering approach. All points that fell within a 10km buffer of each face were considered “matched” to that face (see Figure). We also keep track of the angle of each face, mostly in order to correctly calculate hydrodynamics in steps 4 and 5.



**2. Depth interpolation**

Script: GLORYS\_to\_Atlantis\_fxns.R

Using linear interpolation between depth layers, we calculated the values of all variables at the vertical midpoints of California Current Atlantis depth layers, at [-25,-75,-150,-375,-875, and-1800]m depths. This interpolation was performed mostly to make matching GLORYS points to Atlantis polygons faster and more efficient—this method removes the need to aggregate and average values by depth layer later in the process.

**3. Extract and join**

Script: GLORYS\_Atlantis\_preprocessing.R

The combination of this depth interpolation and the 3-dimensional grid matching from Step 1 meant that after this step, we had GLORYS data that mapped directly onto the Atlantis geometry. For state variables, we extracted GLORYS data from their NetCDFs and joined them to Atlantis, then calculated mean values of each variable for each polygon and depth layer.

**4. Calculate fluxes**

Script: GLORYS\_Atlantis\_preprocessing.R

Horizontal fluxes *u* and *v* were extracted and averaged from NetCDF in the same manner as for the state variables, but were joined to Atlantis polygon faces, not the polygons themselves. Then, the *u* and *v* data were joined, and a resultant flux vector (direction and magnitude) was calculated for each grid point and time step. Then, using the angle of that flux relative to the angle of the face itself, we calculated the magnitude of the flux orthogonal to each face—that is, the flow directly across each face—and kept track of its direction (i.e., which polygon the flow is originating from and which polygon it is entering).

**5. Hyperdiffusion correction**

Script: calc\_hyperdiffusion\_corrections.R

In Atlantis models, we usually have to correct for ‘hyperdiffusion’, i.e., the phenomenon that water fluxes cannot possibly instantly diffuse across entire, large Atlantis boxes as that water enters the box across a face. Therefore, we need to functionally ‘downweight’ the measures of flux to account for hyperdiffusion. To do this, we measure the distance from the midpoint of each face to the far side of the adjoining 'left' and 'right' boxes, at an angle which is orthogonal to the face. Basically, how far does water have entering through a face have to travel to reach the far side of the box? We then can use these correction factors, specific to each face, an apply them to the calculated fluxes from Step 4.

For example, consider the Atlantis box below, with a focal face in bold. At an angle orthogonal to the face, the red and blue rays measure the distance from the face to the opposite side of the adjoining box. These distances (the length of the red and blue lines), calculated for all faces, comprise our hyperdiffusion correction factors.

