Structured and Unstructured Grid
Relocatable Ocean Platform
for Forecasting

SURF-NEMO User Guide

Version 1.00

Francesco Trotta, Nadia Pinardi, Luca Giacomelli

Table of contents

1. Welcome to the SURF-NEMO User Guide

Welcome to the user guide for the structured grid component of the **Structured and Unstructured Grid Relocatable Ocean Platform for Forecasting** (**SURF**), release version 1.00. This guide provides comprehensive information and instructions to help you make the most of the SURF-NEMO platform.

For details on updates and changes in this version, please refer to the release notes.

Note

PDF Documentation: The full user guide is also available for download in PDF format:

surf_nemo_1.00.pdf.

For the latest SURF-NEMO release, you can access and download it directly from the official website:

https://www.surf-platform.org

License Information

SURF is distributed as a free and open-source software package under the terms of the GNU General Public License (GPLv3).

2. Introduction

The **Structured and Unstructured grid Relocatable ocean platform for Forecasting (SURF)** is an open-source, on-demand ocean numerical modeling platform designed for setting up, running and analysing high-resolution nested ocean models in any region within a large-scale Ocean Forecasting System. Designed for flexibility and ease of use, SURF enables users, including those with minimal technical expertise, to easily set up downscaling experiments using a clear and well-documented JSON configuration file. SURF can be operated on commercially available personal computers or laptops, ensuring broad accessibility and flexibility.

SURF integrates two state-of-the-art ocean models, the structured grid model NEMO (Nucleus for European Modelling of the Ocean) and the unstructured grid model SHYFEM (Shallow Water Hydrodynamic Finite Element Model). This User Guide covers the structured grid component of SURF, known as **SURF-NEMO**. For information on the unstructured grid component, SURF-SHYFEM, please refer to the corresponding manual, available here .

This manual provides step-by-step instructions for running the platform, along with detailed explanations of the scripts and data structures, enabling users to modify or extend SURF according to their needs.

The package also includes case studies, complete with input datasets for bathymetry, coastlines, atmospheric , as well as coarser-resolution parent ocean data for initial and boundary conditions. Output datasets are also provided for validation and for checking the correct implementation.

For more information, please visit the SURF website at:

https://www.surf-platform.org

2.1 SURF-NEMO Relocatable ocean modelling platform

SURF-NEMO (Trotta et al. 2016, 2021) is a numerical platform designed for forecasting hydrodynamic and thermodynamic fields with high spatial and temporal resolution. It can be embedded into any region within a larger-scale ocean prediction system, which operates at a coarser resolution.

The platform utilizes a **one-way nesting** approach for downscaling, where coarser resolution parent model fields are interpolated onto the child grid, providing initial and lateral open boundary conditions for the fine-grid model. Additionally, it supports **multiple nesting capabilities**, allowing for consecutive nested models with progressively finer grid resolutions. Starting from a large-scale ocean model, it can achieve horizontal grid resolutions down to a few hundred meters. At each nesting level, the parent coarse-grid model supplies the initial and lateral boundary conditions for the nested SURF child components.

This relocatable ocean model system is intended to be a valuable tool for supporting various **Decision Support Systems (DSS)** that may require high-resolution ocean fields, such as oil spill monitoring, search and rescue operations, navigation routing, fisheries and tourism.

2.2 Virtual Machine Environment

SURF-NEMO is distributed as a **Virtual Machine (VM)** image created using **VirtualBox**. This VM contains all the necessary libraries and software required to set up, run, and analyze downscaling experiments. The source code includes the hydrodynamic

NEMO model, along with several pre- and post-processing tools, all installed within the VM environment. Detailed instructions for downloading and installing the package are provided in Chapter 3.1 of the Installation Guide.

A virtual machine is a software-based system that emulates a computer's operating system with virtual access to hardware resources such as CPU, RAM, networking, and storage. The operating system running inside the VM is referred to as the **guest**, and it operates within a window on your physical computer's operating system, known as the **host**.

The virtualization software used is the free and open-source **Oracle VM VirtualBox**, with the **Debian Linux** operating system installed inside the VM.

Virtual machines offer several advantages. They encapsulate an entire computing environment—operating system, applications, and data—within a single file. This makes setup easier compared to installing a full suite of software that must work together. A VM can be distributed as a pre-configured, ready-to-use system, simplifying both configuration and distribution. Additionally, VMs are versatile and can run on various hardware platforms.

2.3 Source Code

The **SURF-NEMO** source code is distributed as a compressed **tar.gz** archive, which includes the NEMO model code, a suite of pre- and post-processing tools, and a template user configuration file. Detailed instructions for downloading and installing the package are provided in Chapter 3.2 of the Installation Guide.

NEMO is an open-source ocean modeling framework, written in **Fortran 90** and optimized for parallel execution using **MPI**-based domain decomposition. Simulation outputs are stored in the **NetCDF** format, a widely-adopted standard for scientific data storage and exchange.

SURF-NEMO includes a suite of **preprocessing tools** for tasks such as mesh generation, dataset downloading, and remapping of input data, including ocean data for initial and lateral boundary conditions, as well as atmospheric forcing. It also offers **post-processing tools** to visualize and analyze simulation results.

The pre- and post-processing scripts are developed in **Julia**, **NCL**, **Python**, and **Fortran** programming languages. To efficiently handle NetCDF datasets, the platform utilizes **NetCDF Operators (NCO)** and **Climate Data Operators (CDO)**, both optimized for SURF to reduce computation time and ensure efficient memory usage. Currently, these pre- and post-processing tools operate in **serial mode** (i.e., executed on a single processor).

The structure of the SURF source code package is detailed in Appendix B.

3. Getting Started

This chapter provides a step-by-step guide to quickly get started with the **SURF-NEMO** platform. It includes instructions for downloading and installing the SURF Virtual Machine (VM) and all required SURF packages. Additionally, this guide will show you how to compile the source codes, run a case study experiment in the Gulf of Taranto, and analyze the results. Finally, you'll learn how to modify the user-configuration file for executing and analyzing new experiments.

The **template experiment** makes it easier to run simulations without requiring in-depth knowledge of the underlying scientific details. For most applications, only a limited number of default values need to be adjusted. However, for more advanced use cases, such as changing turbulence models or adjusting numerical schemes, users should consult the **NEMO** User Manual for a deeper understanding. There are also video tutorials available online here, designed for beginners to walk through the basic features of the SURF platform step by step.

3.1 Download and Install SURF Virtual Machine (VM)

The SURF platform is distributed as a Virtual Machine (VM) image, packaged within a ZIP archive. The naming scheme for the releases follows the format `surf_vm_VERSION.zip`, where `VERSION` represents the version number (e.g., `surf_vm_1.00.zip` for the current version). The instructions below explain how to download, install and configure the SURF VM in Oracle VirtualBox.

3.1.1 Installing Oracle VirtualBox

- Go to the VirtualBox download page and download the **VirtualBox base package** (version >=6) corresponding to your operating system (Windows, Mac, or Linux).

Figure 3.1. Downloads VirtualBox base package.

- Follow the installation instructions to install VirtualBox on your system.

- In addition to the base package, download the **VirtualBox Extension Pack**, which adds functionality like USB device support and remote desktop support. Double-click the downloaded extension pack and follow the installation steps. Please install the same version extension pack as your installed version of the VirtualBox base package.

VM_install_VirtualBoxExtentionPacks.png

Figure 3.2. Downloads VirtualBox Extension Pack.

3.1.2 Downloading and Installing SURF VM

- Visit the SURF platform website and download the current version of the SURF Virtual Machine (e.g., `surf_vm_1.00.zip`). The default installation path for VirtualBox VMs is typically `/Users/USERNAME/VirtualBox VMs/` on macOS, with similar paths for other operating systems. To download the SURF-VM directly from the terminal, use the following commands:

```
cd /Users/USERNAME/VirtualBox VMs;
wget https://www.surf-platform.org/repository/surf_vm/surf_vm_1.00/surf_vm_1.00.zip
```

In the virtual machines is installed the **Debian GNU/Linux 8.7 (jessie)** operating system and all the necessary libraries required. The Guest Additions have been also installed to optimize the guest operating system for better performance and usability.

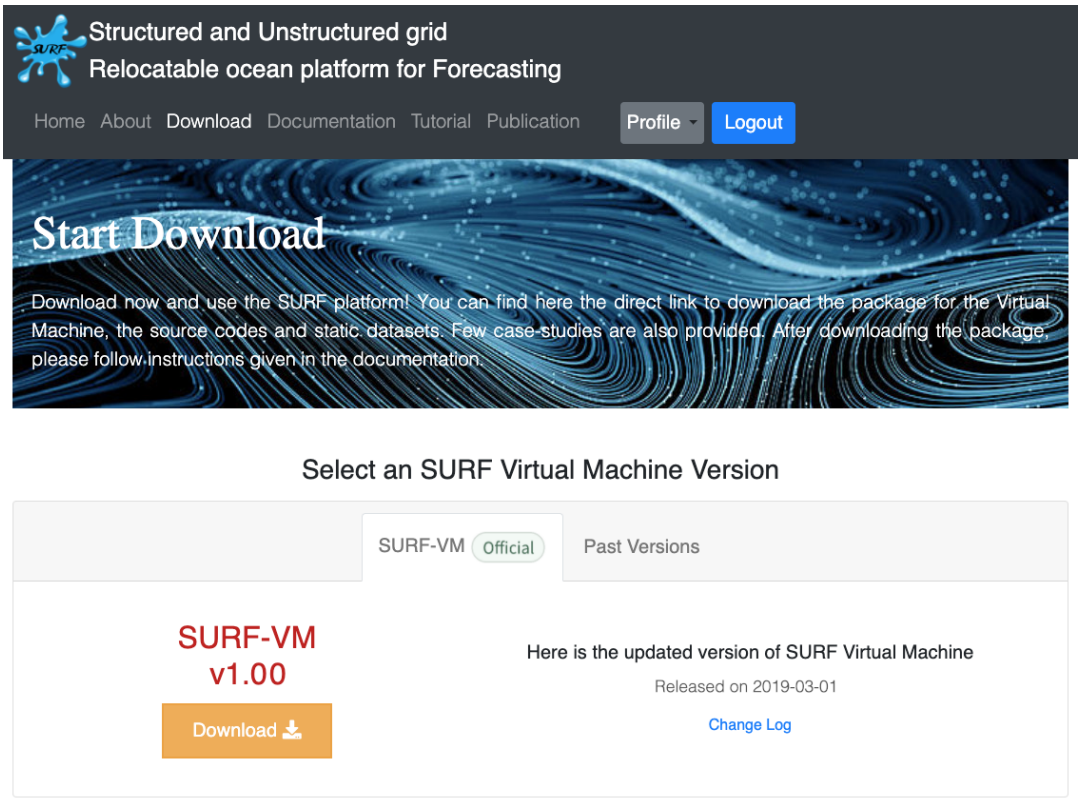


Figure 3.3. Downloads SURF Virtual Machine.

- Unzip the archive file into the VirtualBox directory:

```
unzip surf_vm_1.00.zip
```

- Open VirtualBox, navigate to **Machine > Add**, and select the `surf.vbox` file to add the SURF VM to the list of Virtual Machine. This file is an XML file that contains settings of the Machine.

Figure 3.4. Add SURF-VM in VirtualBox.

- Start the VM from the VirtualBox Manager by selecting it and pressing **Start** button at the top of the window. The VM Login should look like the figure 3.5 .

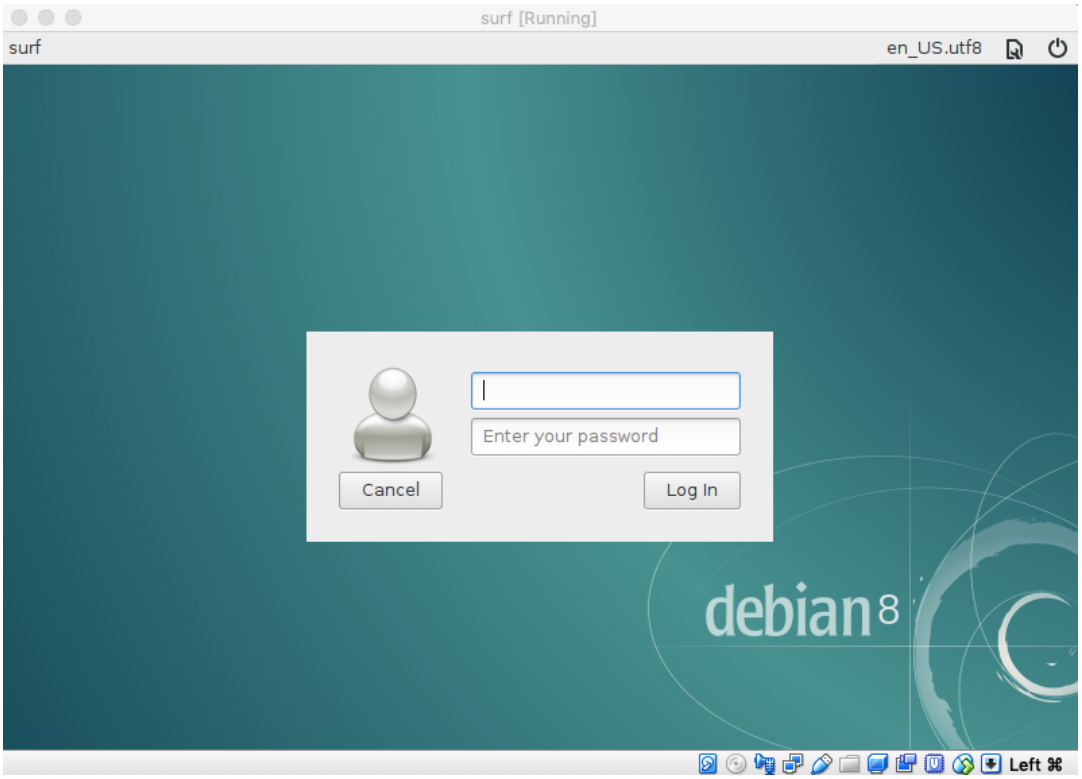


Figure 3.5. Start SURF-VM.

- Log in using the following credentials:
  - **Username:** surf
  - **Password:** surf2019

3.1.3 Disk Partitions mounted on the SURF Virtual Machine

The SURF VM package includes two VirtualBox Disk Image (VDI) files:

- `surf.vdi` : Contains the Debian GNU/Linux 8.7 (jessie) operating system.
- `surf_scratch.vdi` : Contains source code files, sample datasets, and experiments.

It is divided into two main partitions:

- the disk `/dev/sda` "mounted" as filesystems to the root directory `/`
- the disk `/dev/sdb` "mounted" in the directory `/scratch`.

From the guest operating system you can see the list of partitions by typing the following command:

```
sudo fdisk -l
```

3.1.4 **Shared Folders Between Host and Guest Systems**

You can mount other physical hard disks with VirtualBox (see the VirtualBox Manual for details). VirtualBox has the ability to **mount a shared folder between host and guest** in order to access files of your host system from within the guest system. There are a few steps involved:

- Shut down the VM, go to **Settings** > **Shared Folders** in VirtualBox.
- Click the Plus button and select the folder you want to share. Check the **Auto-mount** option and click OK.
- Restart the VM, and the shared folder will appear in `/media` with the prefix "sf_".

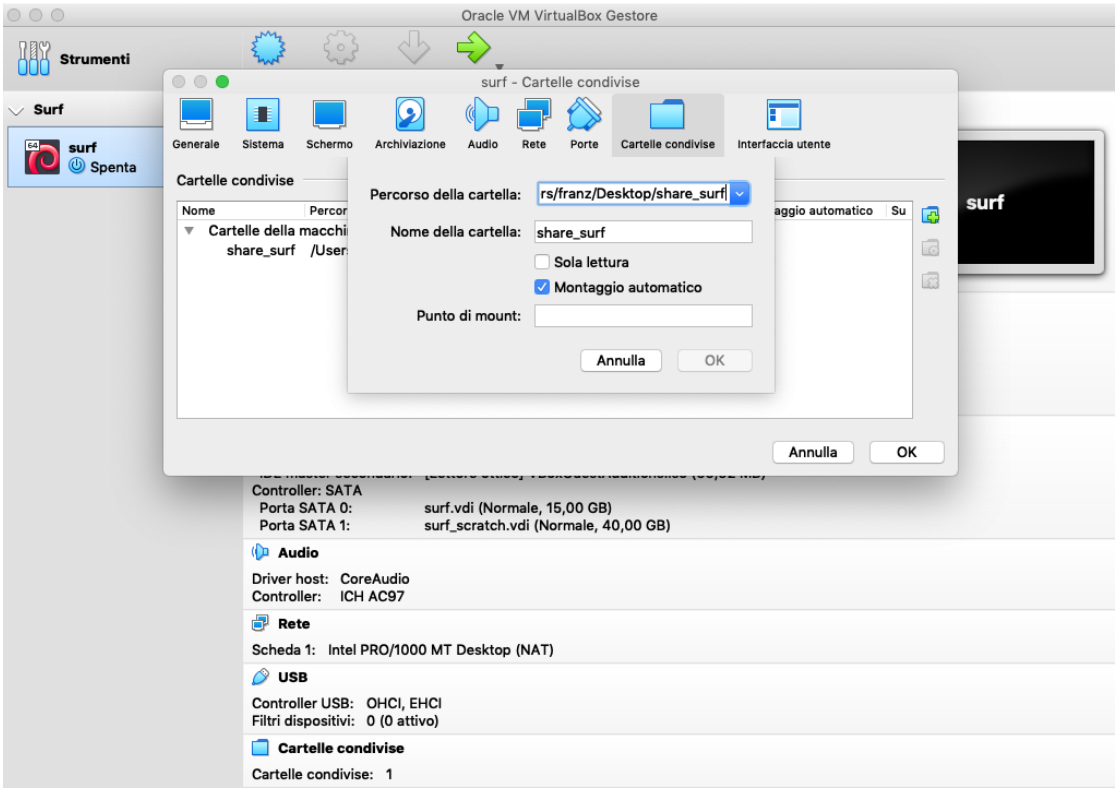


Figure 3.6. Mount shared folders.

3.1.5 **Configuring the SURF Virtual Machine**

By default, the VM surf is configurated as in table Table 3.1 . You can keep all defaults parameters or if it is not adequate for your application you can change settings. To change the configuration you need to shut down the virtual OS before you can edit settings.

- Select the surf VM in the VirtualBox Manager, right-click it and choose *Setting*.
- increase/decrease the number of cores based on your performance desires.
- increase/decrease the number of GB of RAM allocated to your VM according to the size of your computational domain.
- increase/decrease the video memory and scale factor of your screen

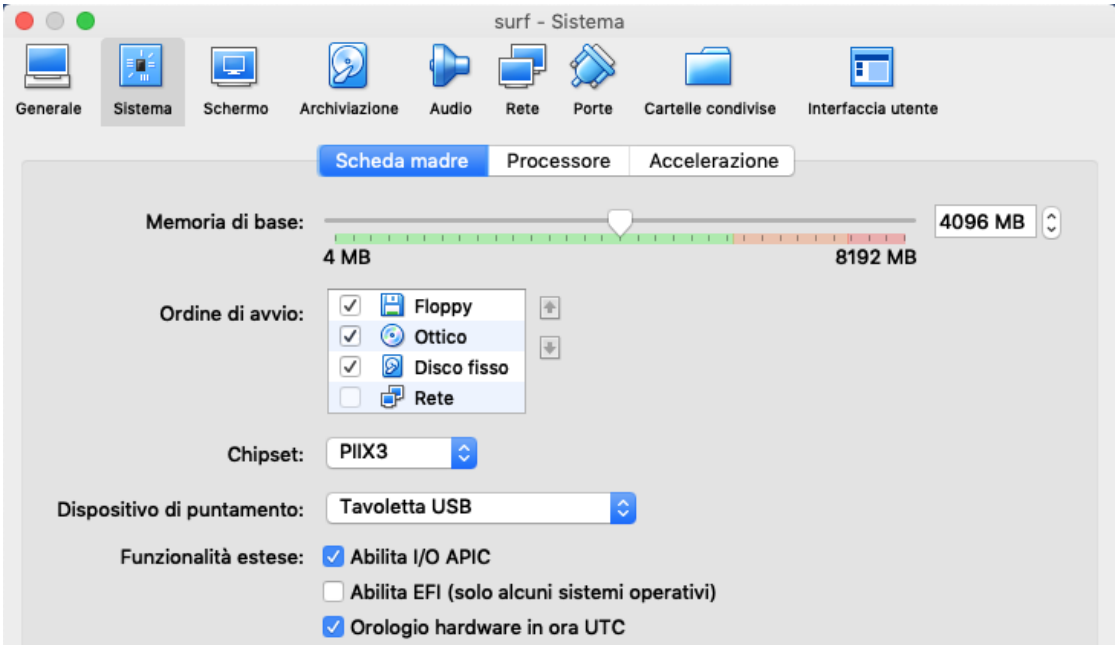


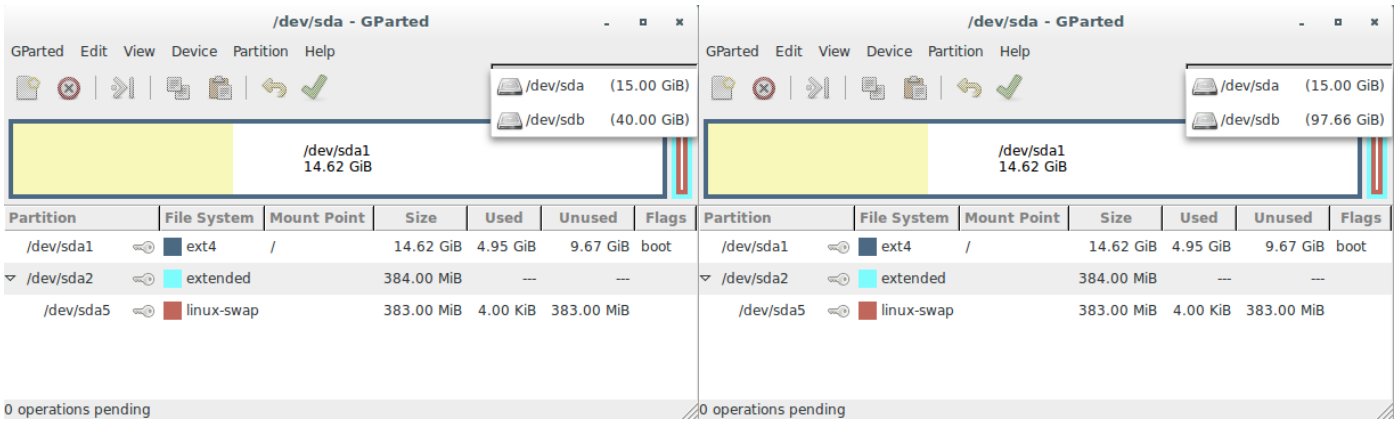
Figure 3.7. Change VM configurations.

If you want to add more storage space to a VM you can also expande the virtual hard disk. There are a few steps involved:

- With the VM Power off, open a terminal and move to the location of the surf_scratch.vdi file that you want to resize,
- At the terminal prompt, type the command:

```
VBoxManage modifyhd surf_scratch.vdi --resize SIZE_MB
```

- Restart the SURF VM and open the GParted application from the Application Menu
- Select the /dev/sdb partition (an unlocated drive space is now available). Resize to the unalocated area



(A) Before. *(B) After.*

Figure 3.8. Enlarge the virtual disk.

Parameter	Description	Values
Name	Name given to the VM	surf
Guest OS	Operating system running on this VM	Debian Linux
Memory	Amount of memory available to this VM	2 GB
Cores	Number of CPU cores being used by this VM	2
Disk Capacity	Total disk capacity available to this VM	40 GB
Network Adapters	Number of network adapters available to this VM	1
IP Address	IP address assigned to the VM	x

Table 3.1: Summary of Virtual Machine fields and configurations.

3.2 Download and Install SURF packages

The Virtual Machine you have installed does not contain the SURF packages (source codes and static datasets) and you need to download and install them. The scratch directory `/scratch` follows the directory structure as shown in Figure B.1. The SURF packages are packaged and distributed as a GZIP Compressed Tar Archive file (with a .tar.gz extension). The naming scheme for the releases follows the format `packageName_VERSION.tar.gz`, where `VERSION` is a number (e.g. *surf_nemo_1.00.tar.gz* for the current version of the surf_nemo package).

The instructions below explain how to install the packages in the SURF VM.

- Once logged in the VM surf, download the current version of the SURF-NEMO (surf_nemo_1.00.tar.gz) and SURF-DATASETS (surf_datasets_1.00.tar.gz) packages directly from the SURF platform website and save it in the directory `/scratch/surf/surf_install/releases/` (for simplicity, we abbreviate this location as `$SURF_RELEASES`). To download the SURF-NEMO and SURF-DATASETS packages, execute the following commands::

```
cd /scratch/surf/surf_install/releases
wget https://www.surf-platform.org/repository/surf_nemo/surf_nemo_1.00/surf_nemo_1.00.tar.gz
wget https://www.surf-platform.org/repository/surf_datasets/surf_datasets_1.00/
surf_datasets_1.00.tar.gz
```

- Go to the directory `$SURF_RELEASES` and run the installation bash script `install.sh` followed by the package name. Use the following commands to install both the SURF-NEMO and SURF-DATASETS packages:

```
cd /scratch/surf/surf_install/releases
install.sh surf_nemo_1.00.tar.gz
install.sh surf_datasets_1.00.tar.gz
```

  The installation process will extract the archive in the directory `/scratch/surf/surf_nemo/` and `/scratch/surf/surf_datasets/`, respectively, and will create a symbolic link `current` in this directory that points to the extracted folder (for simplicity, we abbreviate this location as `$SURF_NEMO`, `$SURF_DATASETS`, respectively).

For a detailed description of the directory structure and contents of each package refer to the Appendix B.

3.2.1 **Compiling the source code**

Once the installation of the SURF-NEMO package is complete, you need to compile the source codes in order to create the executable files needed to perform specific tasks. The executable files should not be recreated unless you need to modify the source code.

The compilation process is handled using the Unix/Linux `make` utility and the following tools:

• **Fortran 90 Compiler** – used for compiling the core numerical models.

• **C Preprocessor (`cpp`)** – processes C-style macros.

• **MPI Library** – required for running simulations in parallel mode.

• **netCDF Library** – allows reading and writing of data in portable netCDF format.

All of these tools are already installed and configured on the SURF platform, so no additional setup is required.

To compile the source codes go to the directory `/scratch/surf/surf_nemo/current/scripts/` and run the compilation bash script `compile.sh` followed by the package name (or by the word 'all' to compile all the packages):

```
cd /scratch/surf/surf_nemo/current/scripts;
./compile_codes.sh all
```

Compilation could take a few minutes and it will create the executable files for each program present in the SURF-NEMO package.

3.3 Running the Gulf of Taranto Case Study

In this case study, the SURF platform is used to implemente an high-resolution model in the Gulf of Taranto, located in the northern Ionian Sea. The simulation starts on 5 October 2014 at 00:00 and run until 7 October 2014 at 24:00. This section provides step-by-step instructions on how to set up, run, and analyze the downscaling experiment.

3.3.1 **Setting Up the Experiment**

1. Download the case study datasets (`gulfTaranto_20141005.tar.gz`) from the (SURF website). To download the SURF-VM, you can use the following commands:

```
cd /scratch/surf/indata_offline;
wget https://www.surf-platform.org/repository/surf_nemo/surf_nemo_1.00/case_studies/
gulfTaranto_20141005/gulfTaranto_20141005_indata.tar.gz
```

2. Extract them to the directory `/scratch/surf/indata_offline/` using the following command:

```
tar -zxvf gulfTaranto_20141005_indata.tar.gz
```

Note If you want to change the local repository path to some other location of your choice make sure to change the path in the user configuration file.

3. Create a new folder named gulfTaranto_20141005 in the `/scratch/from_GUI/` directory. This folder will serve as the Experiment ID, uniquely identifying the experiment:

```
cd /scratch/from_GUI;
mkdir gulfTaranto_20141005
```

4. Copy the configuration file setParFree.json from `/scratch/surf/surf_nemo/current/` into the newly created directory `/scratch/from_GUI/gulfTaranto_20141005/`. This file contains the configuration settings specific to this case study:

```
cp /scratch/surf/surf_nemo/current/setParFree.json /scratch/from_GUI/gulfTaranto_20141005/
```

3.3.2 **Running the Experiment**

To execute the experiment, navigate to the `scripts` directory and run the experiment using the `run_exp.jl` Julia script. Be sure to pass the experiment ID (in this case, `gulfTaranto_20141005`) as an argument:

```
cd /scratch/surf/surf_nemo/current/scripts;
julia run_exp.jl gulfTaranto_20141005
```

This command will generate a new folder named `gulfTaranto_20141005` in the `/scratch/surf/experiments/` directory. The newly created directory will follow a structure similar to the one shown in Figure B.1 (refer to the Appendix B for more details)

User can activate/deactivate specific tasks by setting logical parameters in the `set_lrun` section of the `setParFree.json` configuration file. Set each task to `True` or `False` depending on whether you want it enabled or disabled:

- `lrun_childMeshmask`: Enable the CHILD-MESHMASK GENERATION task.
- `lrun_regridPreAtm`: Enable the ATMOSPHERIC-DATA-REGRIDDING task.
- `lrun_regridPreOceIC`: Enable the OCEAN-IC-DATA-REGRIDDING task.
- `lrun_regridPreOceBC`: Enable the OCEAN-BC-DATA-REGRIDDING task.
- `lrun_regridPreWeights`: Set to `True` to compute the WEIGHT-FILEs for remapping in the regridding phase, or `False` to copy existing ones.
- `lrun_ocean`: Enable the execution of the NEMO model.

Here's an example of how the `set_lrun` section might look in `setParFree.json`:

```
{
  "id": "A001",
  "title": "set_lrun",
  "items": [
    { "name": "lrun_childMeshmask", "value": "True" },
    { "name": "lrun_regridPreAtm", "value": "True" },
    { "name": "lrun_regridPreOceIC", "value": "True" },
    { "name": "lrun_regridPreOceBC", "value": "True" },
    { "name": "lrun_regridPreWeights", "value": "True" },
    { "name": "lrun_ocean", "value": "True" }
  ]
}
```

3.3.3 **Post-processing the results**

The SURF platform includes a comprehensive suite of open-source tools designed for data analysis and visualization. Users can explore and visualize their data using the pre-installed **NcView** software, which provides interactive graphical capabilities. software, which provides interactive graphical capabilities. For more advanced data processing and visualization, the platform also provides a suite of NCL (NCAR Command Language) and Python scripts, giving users the flexibility to perform complex analyses and create customized visual outputs.

Visualizing the results with Ncview

NcView is a user-friendly tool designed for the quick visualization of netCDF data files. While its functionalities are limited, it is perfect for an initial exploration of your data. Typically you would use ncview to get a quick and easy, push-button look at your netCDF files. Users can create simple data movies, view data along different dimensions, inspect actual values, modify color maps, invert data, and more.

To launch **NcView**, use the following command, specifying the dataset you want to visualize:

```
ncview SURF_1h_20141006_20141006_grid_T.nc
```

Below is an example of the **NcView** user interface:

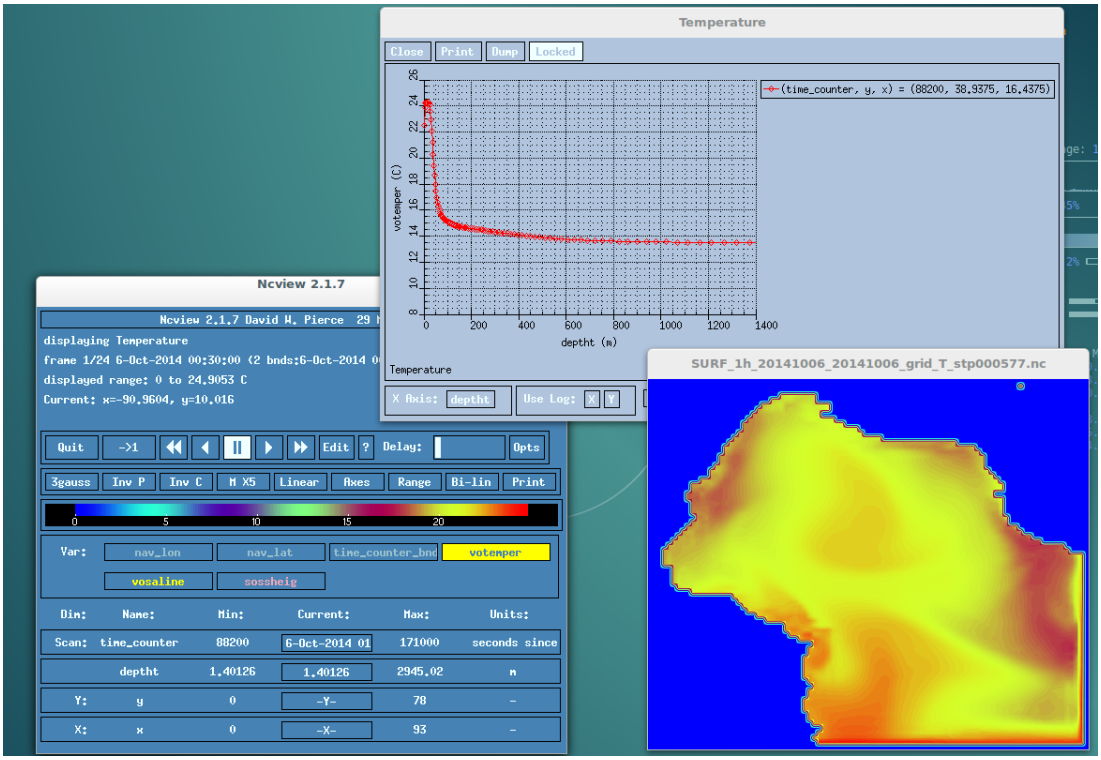


Figure 3.10. Screenshot of using NcView.

Analyzing and Visualizing results using NCAR graphic packages

The **NCAR Graphics** libraries offer advanced capabilities for the visualization of scientific data. **NCL** (NCAR Command Language) is an open-source, interpreted language designed to facilitate the analysis and visualization of geoscientific data. SURF-NEMO includes an extensive suite of **NCL** functions for post-processing tasks, such as:

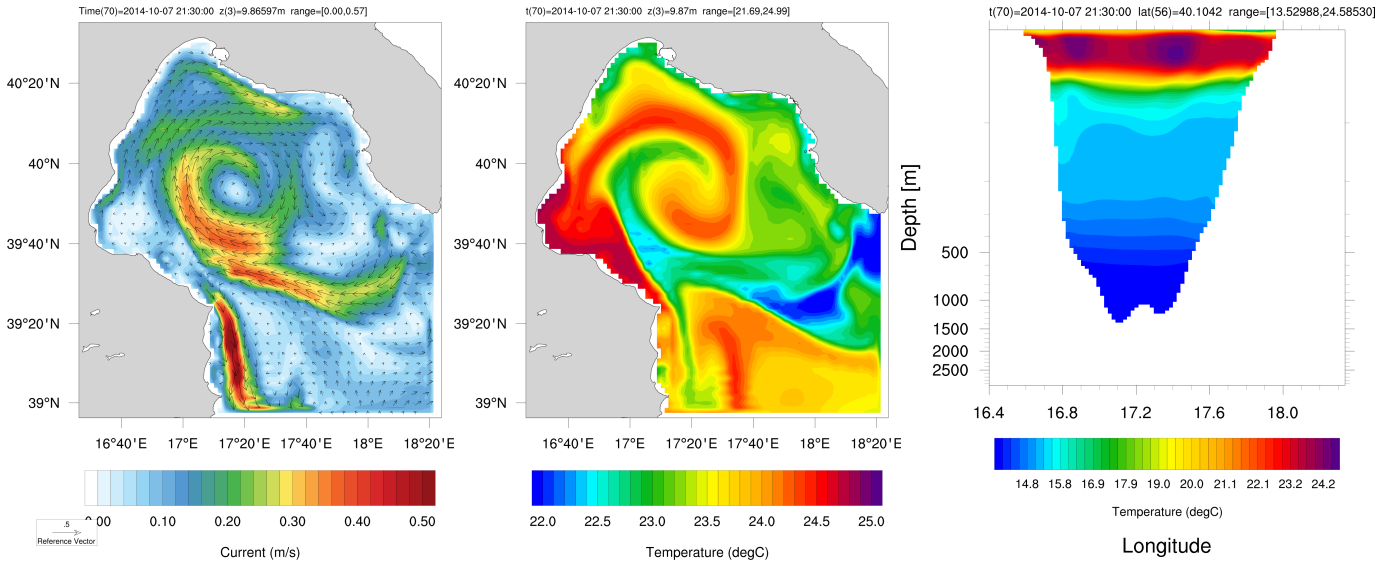
• Visualizing input and output datasets

• Comparing child and parent model fields

• Comparing simulation results with in-situ or satellite datasets

• Converting datasets into various formats

Example visualizations generated using **NCAR Graphics**:



(A) Surface current. *(B) Surface temperature.* *(C) Cross-section of temperature.*

Figure 3.11. Example figure generated using NCAR graphic packages.

To post-process the results of an existing experiment, you need to execute the Julia script `run_postProc.jl` followed by the experiment ID. Example for the case study experiment type the following command:

```
cd /scratch/surf/experiments/gulfTaranto_20141005/code/ocean/scripts/;
julia run_postproc.jl gulfTaranto_20141005
```

Post-processing tasks can be activated or deactivated by setting logical parameters in the **set_lrun_post** and **set_visual_lplot** sections of the configuration file *setParFree.json*. These parameters control various post-processing tasks, such as plotting specific domains, input fields, and comparison plots.

Set each task to `True` or `False` depending on whether you want it enabled or disabled:

- `lrun_visDom` to enable the plotting of the user-defined domains.
- `lrun_visIndata` to enable the plotting of the Indata Bat, Atm, OceIC, OceBC fields.
- `lrun_visExtrapdata` to enable the plotting of the Extrapdata Atm, OceIC, OceBC fields.
- `lrun_visRegriddata` to enable the execution of the OCEAN-IC-DATA-REGRIDDING task.
- `lrun_visOutdata` to enable the execution of the OCEAN-BC-DATA-REGRIDDING task.
- `lrun_chlVSpar` if you want to compute (=True) or just copy (=False) the WEIGHT-FILEs for REMAPPING in the Regridding phase.
- `lrun_surfVSctd` enables the execution of the NEMO code.
- `lrun_surfVSsat` enables the execution of the NEMO code.
- `lrun_surfVSmooring` enables the execution of the NEMO code.
- `lrun_surfVSferrybox` enables the execution of the NEMO code.

Here's an example of how the `set_lrun_post` section might look in `setParFree.json`:

```
{
  "id": "B000",
  "title": "set_lrun_post",
  "items": [
    { "name": "lrun_visDom", "value": "True" },
    { "name": "lrun_visIndata", "value": "True" },
    { "name": "lrun_visExtrapdata", "value": "True" },
    { "name": "lrun_visRegriddata", "value": "True" },
    { "name": "lrun_visOutdata", "value": "True" },
    { "name": "lrun_chlVSpar", "value": "True" },
    { "name": "lrun_surfVSctd", "value": "True" },
    { "name": "lrun_surfVSsat", "value": "True" },
    { "name": "lrun_surfVSmooring", "value": "True" },
    { "name": "lrun_surfVSferrybox", "value": "True" }
  ]
}
```

User can also configure which fields are plotted by modifying the `set_visual_lplot` section in the `setParFree.json` file:

- `lplotMesh` : Enable plotting of the Child MeshMask fields.
- `lplotBat` : Enable plotting of the Bathymetry fields.
- `lplotAtm` : Enable plotting of the Atmospheric fields.
- `lplotOceIC` : Enable plotting of the Ocean Initial Condition fields.
- `lplotOceBC` : Enable plotting of the Open Boundary Condition Ocean fields.
- `lplotOceBCbdy` : Enable plotting of the Open Boundary Condition Ocean fields.
- `lplotOceOut` : Enable plotting of the Output Ocean fields.

Here's an example of the `set_visual_lplot` section in `setParFree.json`:

```
{
   "id": "B001",
   "title": "set_visual_lplot",
   "items": [
      { "name": "lplotMesh", "value": "True" },
      { "name": "lplotBat", "value": "True" },
      { "name": "lplotAtm", "value": "True" },
      { "name": "lplotOceIC", "value": "True" },
      { "name": "lplotOceBC", "value": "True" },
      { "name": "lplotOceBCbdy", "value": "True" },
      { "name": "lplotOceOut", "value": "True" }
   ]
}
```

3.4 Make a new experiments

Let's assume you want create a new experiment to study the circulation of the Sermilik fjord in Greenland from 1 February 2017 at 00:00 to 7 February 2017 at 24:00. You can follow these steps:

- First, select an appropriate name for your experiment ID. For instance, you could name it something like `greenlandFjord_20170201`
- Navigate to the from_GUI directory, and create a new folder for this experiment.

```
cd /scratch/from_GUI/; mkdir greenlandFjord_20170201
```

- For this experiment, you will need a base configuration file to customize. Copy the template configuration file `/scratch/surf/surf_nemo/current/setParFree.json` in the directory `/scratch/from_GUI/greenlandFjord_20170201`

```
cp /scratch/surf/surf_nemo/current/setParFree.json ./greenlandFjord_20170201/
```

- Modify the user configuration file `setParFree.json` as needed for the Sermilik fjord experiment. You should adjust configuration parameters like domain, time period, and specific physical parameters.

```
param1 = xxx
param2 = xxx
param3 = xxx
param4 = xxx
```

- Once your configuration file is ready, it’s time to run the experiment. Navigate to the scripts directory where the simulation script `run_exp.jl` is located, and execute the simulation using Julia, passing the experiment ID `greenlandFjord_20170201` as an argument.

```
cd /scratch/surf/surf_nemo/current/scripts/;
julia run_exp.jl greenlandFjord_20170201
```

- Once the simulation has successfully completed, the next step is to analyze and visualize the results. Run the post-processing Julia script `run_postproc.jl`, followed by the experiment ID `greenlandFjord_20170201`:

```
cd /scratch/surf/surf_nemo/current/scripts/;
julia run_postproc.jl greenlandFjord_20170201
```

In principle you can simply use the template model and modify it to your needs, and not be too much concerned with the input files they create. But our advice is never to use the template model as black boxes. It is therefore important to understand how the codes work, which options they have and how their input files are structured.

3.5 Multiple downscaling experiments

SURF-NEMO package includes multiple nesting capability (i.e. consecutive nested models can be implemented with increasing grid resolutions). Let's assume you want to increase the spatial resolution of an existing experiment, such as the template experiment `gulfTaranto_20141005`, downscaling it to a finer resolution of 800 meters. Below are the steps to achieve this.

- Go to the existing experiment directory

```
cd /scratch/surf/experiments/gulfTaranto_20141005/
```

- Modify the user configuration file `setParFree.json` as needed for the multiple nesting experiment. You should adjust configuration parameters like domain, time period, and specific physical parameters.

```
param1 = xxx
param2 = xxx
param3 = xxx
param4 = xxx
```

- Once the configuration file is ready, from the directory `/scratch/surf/experiments/gulfTaranto_20141005/code/ocean/scripts/`, execute the Julia script `run_exp.jl` followed by the experiment ID `gulfTaranto_20141005`

```
cd /scratch/surf/experiments/gulfTaranto_20141005/code/ocean/scripts/;
julia run_exp.jl gulfTaranto_20141005
```

- After the simulation has completed, you can analyze and visualize the results. Run the Julia script `run_postproc.jl`, followed by the experiment ID `gulfTaranto_20141005`:

```
cd /scratch/surf/experiments/gulfTaranto_20141005/code/ocean/scripts/;
julia run_postproc.jl gulfTaranto_20141005
```

4. SURF-NEMO Workflow Overview

4.1 Workflow Overview

The schematic work-flow diagram in Fig. 4.1 outlines the key stages involved in the SURF-NEMO numerical platform. In the latest version, the workflow consists of the following steps:

1. **Model Configuration:**

   In this step, the user specifies the values of the input parameters for the specific dowscaling experiment, including preprocessing (such as data sources, interpolation methods, etc.), simulation (such as time steps, output frequency, subgrid scale parameterizations, etc.), and postprocessing settings (visualization and data analysis).

2. **Preprocessing** (Automated):

   This automated phase handles the downloading and preparation of the input datasets required for the simulation, such as bathymetry, coastline, atmospheric, and ocean data.

   - **Child Meshmask Generation:** Generates the Meshmask for the nested model ensuring an accurate representation of the region of interest,
   - **Input Data Regridding:** Remaps the input datasets like bottom topography, atmospheric , and initial and lateral open boundary conditions on the child model grid.

3. **Simulation** (Automated):

   In this phase, the NEMO ocean model is executed using the configured parameters. This automated step generates high-resolution fields that provide detailed descriptions of ocean dynamics in the selected region.

4. **Post-Processing:** (Automated):

   After the simulation is complete, post-processing is performed to visualize and analyze the model outputs. Various options for post-processing are available:

   - **Visualize Input, Regridded, and Output Datasets:** Explore the model's behavior at each stage by visualizing the input, regridded, and final model output datasets.
   - **Parent/Child Model Comparisons:** Analyzing differences between the parent (coarse-resolution) and child (high-resolution) model outputs.
   - **Validation with Observational Data:** Comparing the simulation results with in-situ measurements or satellite datasets to validate accuracy.
   - **Data Conversion:** Transforming the output datasets into different formats for further analysis, external use, or integration with other tools.

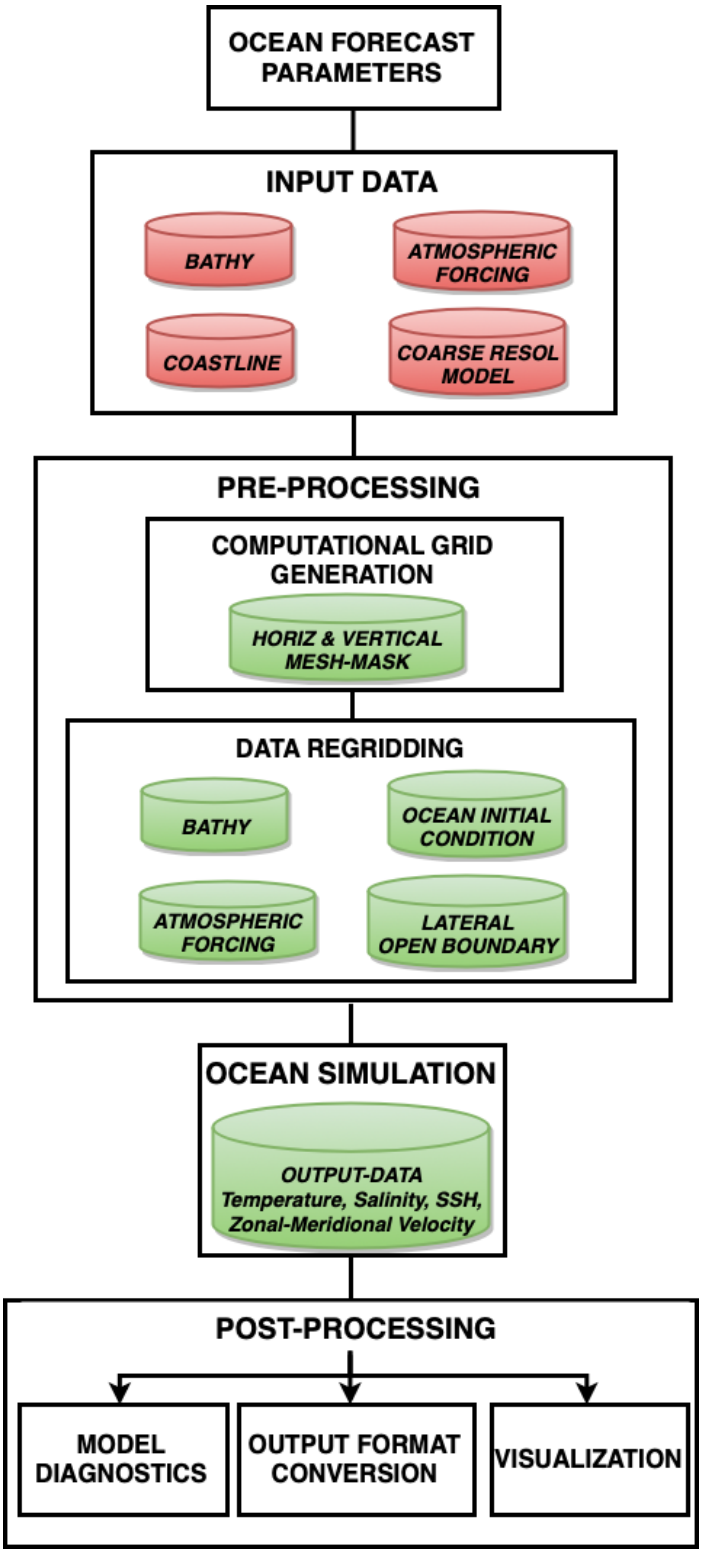


Figure 4.1. Work-flow of the Relocatable SURF-NEMO platform.

The graphical calling function flow, as shown in Figure 4.2, illustrates the complete sequence of steps executed by the program from start to finish. This step-by-step representation highlights the key stages the program passes through during its execution, ensuring users can understand the overall workflow. The process begins with child Meshmask generation, followed by input data remapping, progresses through the simulation phase, and concludes with visualization and data analysis.

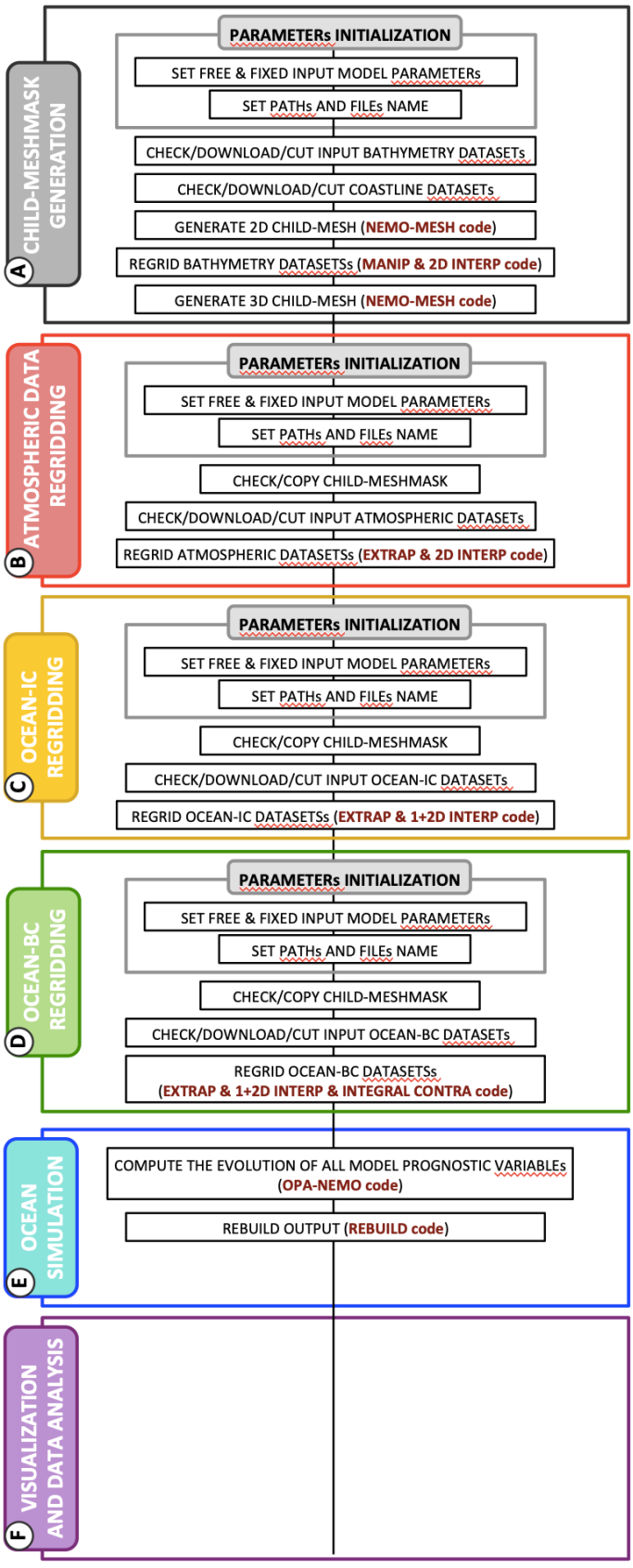


Figure 4.2. Graphical calling function flow of the Relocatable SURF-NEMO platform.

Many of the computational tasks in the workflow are interdependent, meaning they rely on each other to proceed. Figure 4.3 illustrates the dependency flow graph of these macro-tasks. Each node (labeled A through F) represents a distinct macro-task, and the solid edges between the nodes indicate data dependencies between them.

For example, node A serves as the starting point and branches out to nodes B, C, and D. Node E can only begin once all preceding tasks (B, C, and D) are completed. This flow ensures that tasks are executed in the proper order, maintaining the integrity of the overall process.

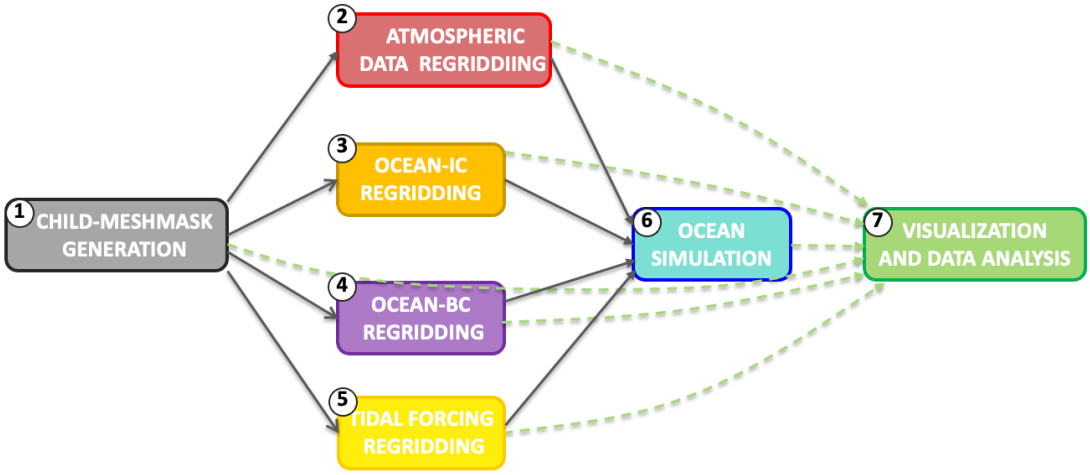


Figure 4.3. Dependency flow graph of macro-tasks.

4.2 Model Configuration

Each macro-task begins by initializing the input model parameters, which are loaded from the configuration files `setParFree.json` and `setParFixed.json`. These files contain user-free and fixed parameters, as detailed in Chapter 5, 6 and Appendix A.

During this phase, the procedures `read_inJsonFree` and `read_inJsonFixed` are executed to handle user-defined and fixed input parameters, respectively. These parameters are essential for running the NEMO model and associated pre- and post-processing tasks.

Additionally, the `set_pathData` and `set_fileData` procedures are called to establish the file paths and names required for the specific numerical simulation.

4.3 Child Meshmask Generation

After the model configuration phase, the child grid is generated. The NEMO model uses the Arakawa C grid for spatial discretization, where the state variables are defined on a staggered grid, as shown in Figure 4.4. In the C grid, scalar quantities such as temperature (T), salinity (S), pressure (p), and density (ρ) are defined at the center of each grid volume, while velocity components (zonal (u), meridional (v), and vertical (w)) are defined at the edges of the grid volumes, shifted by half a grid width in their respective directions.

During this phase, the following procedures are executed:

- Generation of the child 2D mesh.
- Interpolation of the source bathymetric dataset onto the generated child grid.
- Generation of the child 3D meshmask.

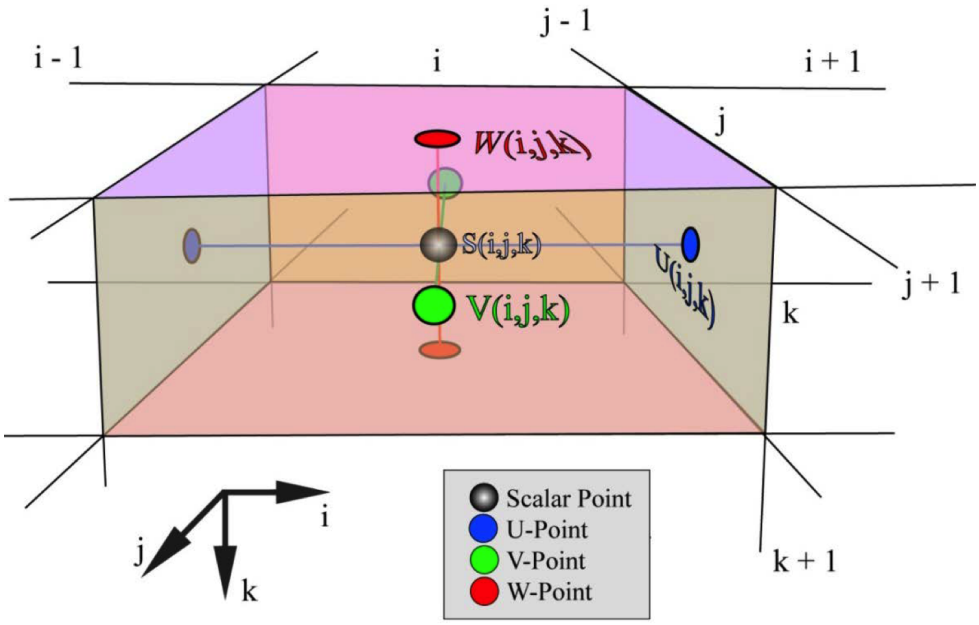


Figure 4.4. The staggered Arakawa C-grid used by NEMO ocean model.

4.3.1 **Horizontal grid**

The horizontal grid is generated using the NEMO code through a configuration called SURF-MESH. In this setup, SURF utilizes a rectangular (latitude-longitude) grid within a spherical coordinate system λ, φ, where λ represents longitude and φ represents latitude, both expressed in degrees.

To create the horizontal grid , the user specifies the number of grid points n_λ and n_φ in the zonal (longitude) and meridional (latitude) directions, respectively, along with the grid resolutions $\Delta\lambda$ and $\Delta\varphi$ (in degrees). Additionally, the grid's starting point $(\lambda, \varphi)_{1,1}$, which corresponds to the first row and column of the T grid, is defined.

The coordinates of the T points on the $\lambda\varphi$ plane are calculated as follows:

$$\lambda_{i,j} = \lambda_{11} + (i-1)\Delta\lambda \quad \text{with } i = 1.\ldots.n_\lambda \; \varphi_{i,j} = \varphi_{11} + (j-1)\Delta\varphi \quad \text{with } j = 1.\ldots.n_\varphi$$

In this grid system, the u and v points (velocity components) are offset by half a grid cell in the zonal and/or meridional directions, as illustrated in Figure 4.4.

4.3.2 **Bathymetry regridding**

During this phase, the bathymetric dataset is interpolated onto the child grid, which is required to generate the 3D meshmask. The following tasks are carried out:

- Accessing and downloading the source bathymetry and coastline datasets.
- Manipulating the bathymetry dataset.
- Spatial interpolation of the source bathymetry onto the child grid.

Access the bathymetry and coastline datasets

A procedure checks if the necessary input datasets are present in the experiment directory `$PATH_IDEXP/data/data00/indata/`. If any requested data is missing, the procedures `downlCoastlineInfile` and `downlBathyInfile` are executed to download the coastline and bathymetry datasets, respectively, from remote ftp or local data repositories as specified in the `setParFree.json` configuration file.

Manipulating and smoothing bathymetry

Before performing the spatial interpolation on the child grid, the source bathymetry dataset can be manipulated as specified in the `setParFree.json` configuration file. The available manipulation methods include:

- Adding a constant value to the surface elevation for the entire nested region (e.g., adjusting the water level for an inland body of water like the Caspian Sea).
- Setting maximum and minimum depth values (e.g., enforcing a minimum depth of 5 meters or a maximum depth of less than the actual depth).
- Defining land/sea grid points based on the input coastline.
- Setting maximum and minimum depths in specific sub-regions (e.g., to mask a particular area).
- Smoothing the bathymetry using the Shapiro filter, a high-order horizontal filter that efficiently removes small-scale noise without affecting the physical structures of a field. A Shapiro filter of a 2N order of accuracy is applied to a variable based on the expression:

$$\tilde{w_i} = F^{2N}(w_i) = \left[I + (-1)^{N-1} \frac{\delta^{2N}}{2^{2N}} \right](w_i) = w_i + (-1)^{N-1} \frac{\delta^{2N} w_i}{2^{2N}}$$

  where $\tilde{w_i}$ is the filtered value of variable w at point x_i, I is the identity operator and δ^{2N} is the even composition of the standard difference operator δ (Richtmyer, 1957). This filter is a discrete symmetric operator with a (2N + 1) point stencil. It acts as a low-pass filter that preserves the low-frequency content (i.e., largest wavelengths) and completely dissipates the high-frequency content (i.e., shortest wavelengths) from the original field.

Interpolation of bathymetric data

After manipulating the source bathymetry, spatial interpolation onto the child grid is performed using the Spherical Coordinate Remapping and Interpolation Package (SCRIP). Available interpolation methods are described in Section 4.4.4.

4.3.3 **Meshmask and Vertical grid**

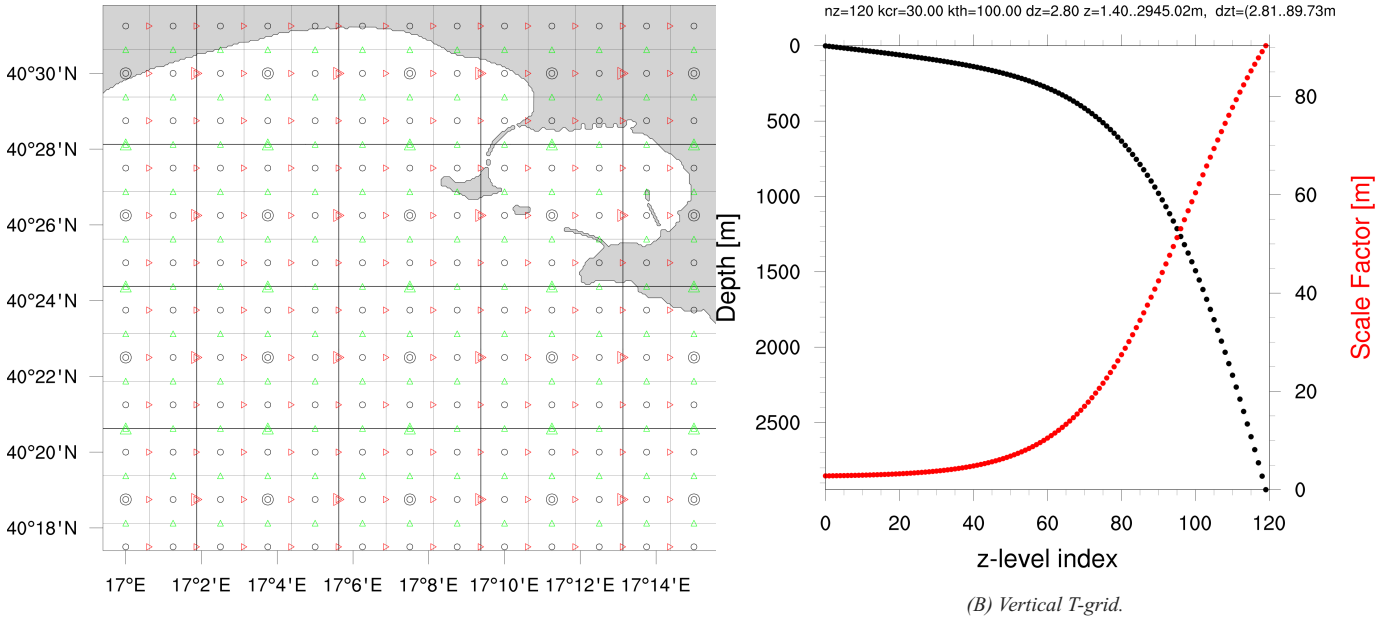
Once the final bathymetry is obtained, the 3D Meshmask can be generated using the MESH module of the NEMO code (within the SURF-MESH configuration). SURF uses a geopotential z-coordinate vertical grid with partial bottom cell representation of the bathymetry. Once the bathymetry $z = H(\lambda, \varphi)$ and the number of levels n_z are specified, the vertical location of the w- and t-levels (expressed in metres) is calculated using the following analytic expression:

$$z(k) = h_{sur} - h_0 k - h_1 log[cosh((k - h_{th})h_{cr})]$$

In this equation, the parameters h_{sur}, h_0, h_1, h_{th}, and h_{cr} need to be specified:

- h_{cr} represents the grid's stretching factor, controlling how the grid levels are distributed.
- h_{th} defines the approximate model level at which the maximum stretching occurs.

This expression enables stretched z-coordinate vertical levels to be defined, which are smoothly distributed along the water column, with appropriate thinning designed to better resolve the surface and intermediate layers. Through partial cell parameterization, the thickness of the bottom layer can vary as a function of the geographical location $(x, y)_{i,j}$ which allows a better representation of the real bathymetry.



(A) Horizontal grid.

(B) Vertical T-grid.

Figure 4.5. Example of horizontal (left) and vertical (right) numerical grid with, respectively, grid sizes of $\Delta\lambda$ and $\Delta\varphi$ in horizontal and Δz in vertical direction.

4.4 Input data Regridding

Regridding, also known as remapping, is the process of transforming data from a source grid to a destination grid while preserving the integrity of the original data. In this section, we explain the spatial extrapolation and interpolation procedures used in SURF to remap input fields onto the child grid.

During this phase, atmospheric , initial conditions, and lateral open boundary condition datasets are generated for the child grid. The steps involved in this process are:

- Accessing and downloading the necessary input datasets
- Rotating vector fields (if required)
- Extrapolating the input datasets
- Performing spatial interpolation of the source datasets onto the child grid
- Generating the lateral open boundary condition datasets

4.4.1 Access and download of the input datasets

A procedure is executed to check if the necessary input datasets are available in the experiment directory `$PATH_IDEXP/data/data00/indata/`. If any of the required data are missing, the procedures `downlAtmSrc`, `downlOceICSrc`, and `downlOceBCSrc` are automatically executed. These procedures download the atmospheric , initial conditions, and lateral boundary condition datasets, respectively, for the selected simulation period. The data are retrieved from remote or local repositories, as specified in the configuration file `setParFree.json`.

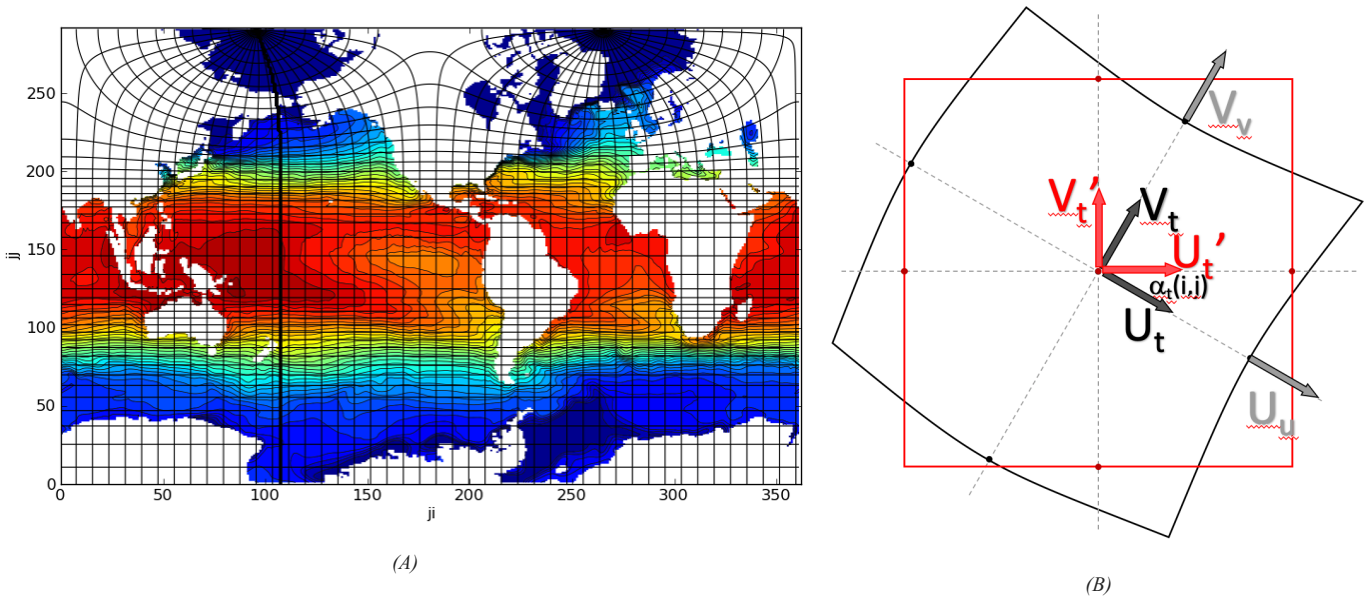
4.4.2 Rotation of horizontal velocity u, v

When the parent coarse resolution model is defined on a rotated or curvilinear grid (e.g., the global tripolar grid, Fig. 4.6(a)), an additional step is required to interpolate the horizontal velocity fields onto the child grid. In ocean models with a “distorted” grid, velocity vectors are aligned with the grid lines. In a staggered Arakawa C grid system, the components of the velocity field are defined at the cell edges (illustrated by the gray arrows in Fig. 4.6(b)).

A rotation of the velocity components in the latitudinal and longitudinal directions is necessary to convert the vectors from the local system (x, y) to a geographical system $(x^{'}, y^{'})$. This transformation ensures that `U` represents the zonal component (W-E direction) and `V` represents the meridional component (S-N direction) of the velocity vector. To achieve this, the vectors must be rotated according to the following equations:

$$U^{'}(x_t^{'}, y_t^{'}) = U(x_t, y_t) \cdot \cos(\alpha_t) - V(x_t, y_t) \cdot \sin(\alpha_t)$$
$$V^{'}(x_t^{'}, y_t^{'}) = U(x_t, y_t) \cdot \sin(\alpha_t) + V(x_t, y_t) \cdot \cos(\alpha_t)$$

For parent models that use rotated rectangular grids, the angle α_t remains nearly constant across the grid. However, for models with curvilinear tripolar grids (as seen in Fig. 4.6(a)), the angle α_t will vary within each grid cell.



(A) *(B)*

Figure 4.6. Example of curvilinear grid. Panel A shows an example of a tripolar grid. Panel B shows the horizontal velocity components defined on the source curvilinear grid (black arrows) and the destination rectilinear lat/lon grid (red arrows) after the rotation.

4.4.3 **Extrapolation method**

Before performing interpolation, the Sea-Over-Land (SOL) procedure in SURF is used to provide ocean field values in areas near the coastline where the parent model's solutions are not defined. This method iteratively extrapolates the input coarse-resolution ocean fields — such as salinity, temperature, sea surface height, and currents — onto land grid points. This process ensure that ocean fields at child-grid points near the coast can accurately be defined through interpolation.

In addition to ocean fields, the SOL procedure is also applied to various input atmospheric fields using the atmospheric land-sea mask. This prevents contamination from land-based atmospheric data, ensuring accurate representation at sea points near the land-sea boundaries.

The source code for the SOL procedure can be found in the directory `$PATH_SURFNEMO/utilities/extrapol/seaoverland`.

The resulting files are stored in the directory `$PATH_EXP/IDEXP/data/data00/extrapdata/`.

4.4.4 **Interpolation methods**

The extrapolation procedure described in the previous section provides the necessary input data for the interpolator. These procedures are based on the Spherical Coordinate Remapping and Interpolation Package (SCRIP) code. SCRIP is a software

package designed to compute weights for remapping and interpolating fields between grids in spherical coordinates. It is compatible with any grid on the surface of a sphere and currently supports the following five remapping options:

- **Conservative remapping:** First- and second-order conservative remapping, as described by Jones (1999, Monthly Weather Review, 127, 2204-2210).
- **Bilinear interpolation:** A slightly generalized version that uses a local bilinear approximation (applicable only to logically rectangular grids).
- **Bicubic interpolation:** Similar to bilinear interpolation but for higher-order accuracy (also only for logically rectangular grids).
- **Distance-weighted averaging:** The inverse-distance-weighted average of a user-specified number of nearest-neighbor values.
- **Particle remapping:** A conservative particle-based (Monte-Carlo-like) remapping scheme.

The source code for SCRIP can be found in the directory `$PATH_SURFNEMO/nemo/NEMOGCM/TOOLS/WEIGHTS`.

Regridding can be broken down into two stages. The first stage involves generating an interpolation weight matrix, which describes how points in the source grid contribute to points in the destination grid. In the second stage, values from the source grid are multiplied by this interpolation weight matrix to produce the corresponding values on the destination grid.

The SCRIP spatial interpolation procedure is applied to all the input fields required for running the NEMO code, including bathymetry, atmospheric , initial conditions, and lateral open boundary condition datasets. The resulting files are stored in the directory `$PATH_EXP/IDEXP/data/data00/regridata/`.

4.4.5 Lateral Open Boundary Condition

The lateral open boundary condition for the selected nested domain is implemented using the BDY module of NEMO. Two numerical algorithms are used to manage open boundary conditions, depending on the prognostic simulated variables:

- **Flather scheme** (Oddo and Pinardi, 2008): Used for **barotropic velocities** and **sea surface height**.
- **Flow relaxation scheme** (Engerdahl, 1995): Applied to **baroclinic velocities** and **active tracers**.

In SURF's default configuration, external data is provided along straight open boundary lines, with the Flow Relaxation area limited to a single internal grid point.

Since the parent coarse-resolution ocean model only provides the total velocity field, the interpolated total velocity field on the child grid is split into its **barotropic** and **baroclinic** components. To ensure that total transport across the open boundary is preserved after interpolation, an integral constraint method is applied. The process involves the following steps:

1. **Define the open boundary geometry**: This step involves defining the coordinates for the open boundary on the T, U, and V grids using `geometry_bdy` procedure. The resulting data arrays are written to the `coordinates.bdy.nc` file.
2. **Extract ocean fields at the open boundary**: In this step, ocean fields including active tracers, sea surface height, and barotropic and baroclinic velocities are extract at the open boundary T-U-V grid points using the `fields_bdy` procedure.

The generated files are stored in the directory `$PATH_EXP/IDEXP/data/data00/regridata/`.

4.4.6 **Integral Constraint at the open boundary**

The downscaling process in SURF is designed to that the volume transport across the open boundary (OB) of the child model matches that across the corresponding section of the parent model. At the eastern and western boundaries, the U-points (zonal velocities) are imposed using the following condition:

$$\int_{y_2}^{y_1} \int_{-H_{child}}^{\eta_{child}} U_{child} dz\, dy = \int_{y_2}^{y_1} \int_{-H_{parent}}^{\eta_{parent}} U_{parent} dz\, dy$$

Here, y_1, y_2 represent the limits of the open boundary section. η_{child}, H_{child} are the surface elevation and bathymetry at the boundary in the child model, while $\eta_{parent}, H_{parent}$ represent the surface elevation and bathymetry in the parent model. The terms U_{parent}, U_{child} denote the total zonal velocities in the parent and child models, respectively, normal to the western/eastern boundaries.

The corrected velocity component normal to the boundary, U_{child}, is calculated according to the method described in N. Pinardi et al. (2003):

$$U_{child}(x, y, z, t) = U_{interp} - U_{correction}$$

where U_{interp} is the U_{parent} interpolated on the child open boundary points and the velocity correction is given by

$$U_{correction} = \frac{M_{interp} - M_{parent}}{S}$$

In this equation:

- $M_{interp} = \int_{y_2}^{y_1} \int_{-H_{child}}^{\eta_{child}} U_{interp} dzdy$ is the volume transport of the interpolated zonal velocity across the open boundary in the child model.
- $M_{parent} = \int_{y_2}^{y_1} \int_{-H_{parent}}^{\eta_{parent}} U_{parent} dzdy$ is the volume transport of the parent model across the corresponding boundary section.
- $S = \int_{y_2}^{y_1} \int_{-H_{child}}^{\eta_{child}} dzdy$ is the area of the boundary section in the child model.

These conditions are similarly applied for the meridional velocity component (V-points) at the northern and southern boundaries. The integral constraint procedure ensures that the interpolation process preserves the net transport across the child model's lateral open boundary, preventing any artificial modifications to the total transport.

4.5 Simulation

Finally, the NEMO Fortran code is executed with the configured model parameters using `mpiexec`, allowing it to run across multiple processors for efficient parallel computation. During the execution, output files are continuously generated and updated at a fixed frequency. A logfile is also created to track the progress of the model run, providing step-by-step information on the current status.

Once the run is completed, all output files are stored in the experiment directory: `$PATH_IDEXP/data/data00/outdata/` (refer to Appendix B for more details).

4.6 Post-processing

After the simulation is complete, post-processing can be performed to visualize and analyze the model results. Several post-processing options are available:

- **Visualize Input, Regridded, and Output Datasets:** Explore the model's behavior at each stage by visualizing the input, regridded, and final model output datasets.
- **Parent/Child Model Comparisons:** Analyzing differences between the parent (coarse-resolution) and child (high-resolution) model outputs.
- **Validation with Observational Data:** Comparing the simulation results with in-situ measurements or satellite datasets to validate accuracy.
- **Data Conversion:** Transforming the output datasets into different formats for further analysis, external use, or integration with other tools.

5. Configuration Guide

5.1 User Configuration File - Preprocessing Sections

To execute the SURF-NEMO package, users need to configure various model parameters that define several aspects of the simulation, including the specific simulation region, simulation period, horizontal and vertical turbulence schemes, input datasets, interpolation methods, etc. These parameters are essential for the pre- and post-processing stages, as well as for generating the Fortran namelist required to execute the NEMO-OCE model.

The configuration process requires specifying the values of these parameters in the user configuration file, `setParFree.json`. This file is structured to organize all user-configurable input parameters into sections based on their respective functions. It has been designed to meet the needs of **intermediate-level users**, particularly PhD researchers, who are conducting downscaling experiments and validating specific aspects of the NEMO model.

Certain model parameters are predefined and fixed within the SURF source package. These fixed parameters are specified in the `setParFix.ncl` file. For more information about these fixed parameters, please refer to Appendix A.

In this chapter, we provide detailed explanations for each section of the user configuration file, including the acceptable parameter values, unit measurements, and the “reference value” used in the test case experiment (see Section 3.3).

5.1.1 Configuration file and JSON Object Structure

The user configuration file for the SURF-NEMO package is based on the JavaScript Object Notation (JSON) format, a simple, text-based method for storing and transmitting structured data. JSON is both "self-describing" and easy to read, making it an ideal choice for configuration files. It supports complex data types and structures and is widely used in web applications for its simplicity and flexibility.

JSON syntax is derived from JavaScript object notation and can represent data in two main forms: arrays and objects (associative arrays of name/value pairs, also referred to as properties). An **array** is a comma-separated list of values enclosed in square brackets `[ ]`, while an **object** is a collection of name/value pairs enclosed in curly brackets `{ }`. Each name/value pair consists of a field name (in double quotes) followed by a colon `:`, and then the field value. The value can be of various types, including numbers (integer or floating-point), strings (enclosed in double quotes), booleans (`true` or `false`), arrays, objects, or `null`.

In the SURF-NEMO package's configuration file, the top-level structure is the "sections" array, which contains a collection of objects. Each object represents a configuration section and includes three key properties:

- **Title** – A description of the section's contents.
- **ID** – A unique four-digit alphanumeric identifier (e.g., `id = A001`, `A002` for pre-processing sections, and `B001`, `B002` for post-processing sections).
- **Items** – An array of parameters specific to that section.

Each element of the "items" array is an object that consists of:

- **Name** – The name of the parameter.
- **Value** – The corresponding value for the parameter.
- **Data type** – The type of the parameter value (e.g., `int`, `float`, `double`, `bool`, `string`).
- **Description** – A brief explanation of what the parameter represents.

Figure 5.1 illustrates the structure of the JSON configuration file for the current version of the SURF-NEMO package. This organized format allows users to efficiently manage and customize their simulation parameters.

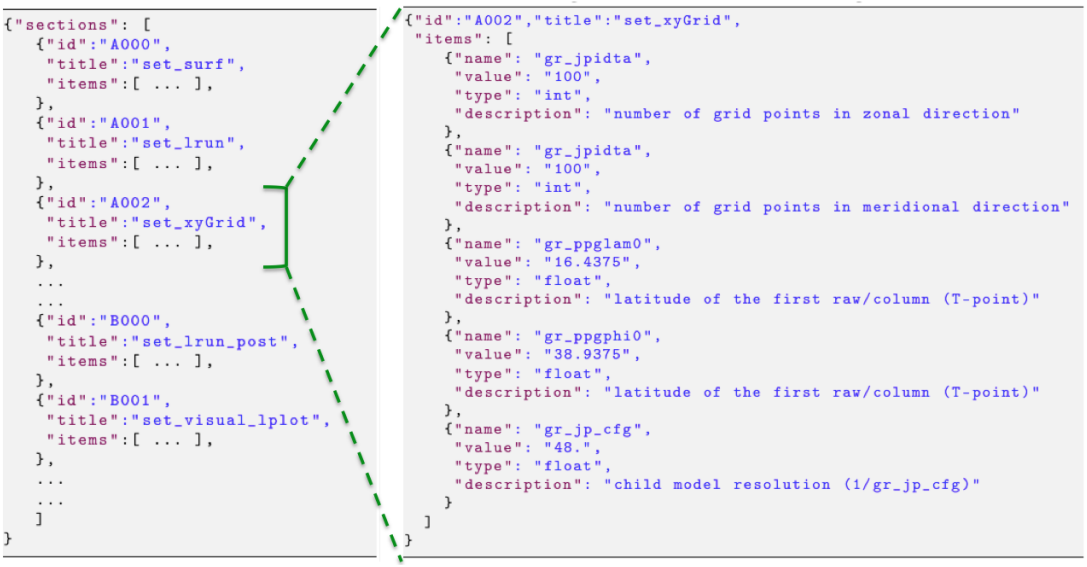


Figure 5.1. JSON representation for the SURF-NEMO user-configuration file.

5.1.2 General Configuration and Initialization

set_surf

This section contains the following two parameters:

Parameter	Value	Type	Description
`nnest`	1	int	number of nesting domain
`nameNestDomain`	gulfTaranto	string	name of the nest domain to simulate

set_lrun

This section includes logical parameters that control the activation or deactivation of specific tasks within the simulation. It determines which processes are enabled during the model's run.

Parameter	Value	Type	Description
`lrun_meshmask`	True	bool	enables the execution of the MESHMASK GENERATION phase
`lrun_regridPre`	True	bool	enables the execution of the DATA-REGRIDDING phase
`lrun_regridPreWeights`	True	bool	enables the computation/copy of WEIGHT-FILEs for input_fields REMAPPING (if lrun_regridPre=True)
`lrun_ocean`	True	bool	enables the execution of the NEMO code
`lrun_regridOutUV`	False	bool	enables the execution of the output-UV_fields REMAPPING (from UV GRID to T-GRID)
`lrun_regridOutUVWeghts`	False	bool	enables the computation/copy of WEIGHT-FILEs for output-UV_fields REMAPPING (if lrun_regridOutUV=True)
`lrun_shapFiltBat`	True	bool	enables the execution of the SHAPIRO Filter for Bathymetric datasets
`lrun_shapFiltOce`	False	bool	enables the execution of the SHAPIRO Filter for Ocean datasets

set_nemoCPPkey

This section contains the compilation keys used to compile the NEMO code.

Parameter	Value	Type	Description
`cppkey_zdf`	key_zdfric	string	name of CPP-keys for Vertical viscosity/diffusivity to check NEMO compilation: key_zdfric,key_zdftke (if =NOTUSED, cppkey not used)
`cppkey_spg`	key_dynspg_ts	string	name of CPP-keys for Surface Pressure Gradient to check NEMO compilation: key_dynspg_exp, key_dynspg_flt, key_dynspg_ts (if =NOTUSED, cppkey not used)
`cppkey_obc`	key_bdy	string	name of CPP-keys for Open Boundary Conditions to check NEMO compilation: key_bdy (if =NOTUSED, cppkey not used)
`cppkey_tide`	NOTUSED	string	name of CPP-keys for Tidal Potential forcing to check NEMO compilation: key_tide (if =NOTUSED, cppkey not used)
`cppkey_hpc`	key_mpp_mpi	string	name of CPP-keys for Massively Parallel Processing to check NEMO compilation: key_mpp_mpi (if =NOTUSED, cppkey not used)
`cppkey_dia`	NOTUSED	string	name of CPP-keys for Instantaneous output to check NEMO compilation: key_diainstant (if =NOTUSED, cppkey not used)
`cppkey_signfunc`	key_nosignedzero	string	name of CPP-keys to deal with intrinsic SIGN function (fortran compiler) to check NEMO compilation: key_nosignedzero (if =NOTUSED, cppkey not used)

5.1.3 **Spatial Grid Generation Settings**

set_xyGrid

This section defines the input parameters required for generating the horizontal grid of the model. The horizontal grid setup determines the spatial resolution and the domain layout of the simulation area.

Parameter	Value	Type	Description
`gr_xygridSpec`	0	int	parameters specification for the horizontal grid: (=0)z(nx,ny,x0,y0,dxy) (=1)z(x0,y0,x1,y1,dxy)
`gr_jpidta`	94	int	number of grid points in zonal direction to specify if xygridSpec=0 (if =NOTUSED, parameter not read)
`gr_jpjdta`	79	int	number of grid points in meridional direction to specify if xygridSpec=0 (if =NOTUSED, parameter not read)
`gr_ppglam0`	16.4375	float	longitude of the first raw and column T-point to specify if xygridSpec=0,1
`gr_ppglam1`	NOTUSED	float	longitude of the last raw and column T-point to specify if xygridSpec=1 (if =NOTUSED, parameter not read)
`gr_ppgphi0`	38.9375	float	latitude of the first raw and column T-point to specify if xygridSpec=0,1
`gr_ppgphi1`	NOTUSED	float	latitude of the last raw and column T-point to specify if xygridSpec=1 (if =NOTUSED, parameter not read)
`gr_jp_cfg`	48.	float	child model resolution (1/gr_jp_cfg) to specify if xygridSpec=0,1
`gr_jp_cfg_father`	16.	float	father model resolution (1/gr_jp_cfg_father)

set_zGrid

This section includes the input parameters used to generate the vertical grid. The vertical grid is essential for defining the stratification and depth levels of the model, impacting the vertical resolution of the simulation.

Parameter	Value	Type	Description
`gr_jpkdta`	120	int	number of vertical levels
`zgrid_spec`	1	int	parameters specification for the vertical grid: (=0)z(h0,h1,hsur,hcr,hth) (=1)z(dzmin,hmax,hcr,hth)
`gr_ppsur`	NOTUSED	double	parameter h_sur for the z-coord. transformation to specify if zgrid_spec=0 (if =NOTUSED, parameter not read)
`gr_ppa0`	NOTUSED	double	parameter h_0 for the z-coordinate transformation to specify if zgrid_spec=0 (if =NOTUSED, parameter not read)
`gr_ppa1`	NOTUSED	double	parameter h_1 for the z-coordinate transformation to specify if zgrid_spec=0 (if =NOTUSED, parameter not read)
`gr_ppkth`	100.0	double	parameter h_th which gives the approximate layer number above which stretching will be maximum (usually of order nz/2)
`gr_ppacr`	30.0	double	parameter h_cr which gives the grid stretching factor (the higher gr_ppacr, the smaller the stretching)
`gr_ppdzminw`	2.8	double	depth of the top (first) model layer depth of second 'w' level to specify if zgrid_spec=1 (if =NOTUSED, parameter not read)
`gr_pphmaxw`	2900.0	double	maximum depth of the ocean depth of the last 'w' level (set to 0.0 to be computed) to specify if zgrid_spec=1
`gr_dbletanh`	False	bool	enables the use of the double tanh function for vertical coordinates
`gr_ppa2`	NOTUSED	double	parameter h_2 to specify if gr_dbletanh=True (if =NOTUSED, parameter not read)
`gr_ppkth2`	NOTUSED	double	parameter h_th2 if gr_dbletanh=True (if =NOTUSED, parameter not read)
`gr_ppacr2`	NOTUSED	double	parameter h_cr2 if gr_dbletanh=True (if =NOTUSED, parameter not read)

5.1.4 **Simulation Timing Settings**

set_dateTime

This section specifies the parameters that define the simulation period and time discretization. It includes the start and end dates of the simulation, as well as the time steps for model calculations.

Parameter	Value	Type	Description
`start_date`	20141005	string	initial date of the simulation (the run starts at 00:00)
`ndays`	3	int	total number of simulation days
`ndays_spinup`	1	int	number of spin-up days
`dom_rdt`	150.	float	simulation 'baroclinic' time step (... 40,48,50,60,72,80,90,100,120,144...)
`runMan_write`	24	int	frequency of write in the output file expressed as the number of simulation time steps
`dom_btAuto`	False	bool	enables the automatic definition of baro-timestep to be just below a user-defined maximum Courant number dom_btCmax
`dom_btCmax`	0.8	float	maximum Courant number (allowed if dom_btAuto=True)
`dom_baro`	100	int	number of iterations of barotropic mode during dom_rdt (allowed if dom_btAuto=False)
`runMan_rstart`	True	bool	start from rest (False) or from a restart file (True)
`ndays_xsimu`	1.0	float	number of days per each restart simulation

5.1.5 **Boundary and Surface Condition Settings**

set_sbc

This section contains the parameters used to set up surface boundary conditions.

Parameter	Value	Type	Description
`sbc_iformulat`	0	int	surface boundary condition formulation to be used (=0)MFS bulk formulat, (=1)fluxform+ssRest,(=2)CORE formulation
`sbc_ltimeInterp`	True	bool	activate, or not, the time interpolation (=False) steplike shape forcing (=True) broken line shape forcing
`sbc_aprdyn`	False	bool	enables the inclusion of atmospheric pressure gradient in ocean and ice Eqs.
`sbc_sclapr`	1.	float	scaling factor to convert atmospheric pressure from hPa to Pa

set_obc

This section provides the parameters for defining lateral open boundary conditions.

Parameter	Value	Type	Description
obc_dyn2d	flather	string	algorithm of boundary condition for barotropic solution: flather
obc_dyn2d_dta	1	int	boundary data to use: (=0)Initial condition (=1)external data (=2)tidal forcing (=3)external data+tidal
obc_dyn3d	frs	string	algorithm of boundary condition for baroclinic velocities: frs, orlanski
obc_dyn3d_dta	1	int	boundary data to use: (=0)Initial condition (=1)external data
obc_tra	frs	string	algorithm of boundary condition for active tracers: frs, orlanski
obc_tra_dta	1	int	boundary data to use: (=0)Initial condition (=1)external data
obc_rimwidth	1	int	width of the FRS zone
obc_ltimeInterp	True	bool	activate, or not, the time interpolation (=False) steplike shape forcing (=True) broken line shape forcing
obc_lvelCorr	True	bool	activate the Integral Constraint method to preserve the total transport after the interpolation

set_tide

This section includes the parameters for configuring tidal components in the model. It allows for the inclusion of tidal effects either as the equilibrium tidal sea level and/or only at the lateral boundaries.

Parameter	Value	Type	Description
`tide_ltidePot`	False	bool	enables the use of tidal potential forcing
`tide_clname01`	K1	string	name of the 1-st tidal component (if =NOTUSED, component not used)
`tide_clname02`	O1	string	name of the 2-st tidal component (if =NOTUSED, component not used)
`tide_clname03`	P1	string	name of the 3-st tidal component (if =NOTUSED, component not used)
`tide_clname04`	Q1	string	name of the 4-st tidal component (if =NOTUSED, component not used)
`tide_clname05`	K2	string	name of the 5-st tidal component (if =NOTUSED, component not used)
`tide_clname06`	M2	string	name of the 6-st tidal component (if =NOTUSED, component not used)
`tide_clname07`	N2	string	name of the 7-st tidal component (if =NOTUSED, component not used)
`tide_clname08`	S2	string	name of the 8-st tidal component (if =NOTUSED, component not used)
`tide_clname09`	M4	string	name of the 9-st tidal component (if =NOTUSED, component not used)
`tide_clname10`	Mm	string	name of the 10-st tidal component (if =NOTUSED, component not used)
`tide_clname11`	Mf	string	name of the 11-st tidal component (if =NOTUSED, component not used)
`tide_clname12`	NOTUSED	string	name of the 12-st tidal component (if =NOTUSED, component not used)
`tide_clname13`	NOTUSED	string	name of the 13-st tidal component (if =NOTUSED, component not used)
`tide_clname14`	NOTUSED	string	name of the 14-st tidal component (if =NOTUSED, component not used)
`tide_clname15`	NOTUSED	string	name of the 15-st tidal component (if =NOTUSED, component not used)
`tide_clname16`	NOTUSED	string	name of the 16-st tidal component (if =NOTUSED, component not used)

5.1.6 Physical Parameterization Settings

set_eos

This section defines the parameters related to the equation of state of seawater.

Parameter	Value	Type	Description
`eos_type`	0	int	type of equation of state and Brunt-Vaisala frequency: (=-1)TEOS-10, (=0)EOS-80, (=1)S-EOS
`eos_useCT`	False	bool	enables the use of Conservative Temp. ==> surface CT converted in Pot. Temp. in sbcssm
`eos_a0`	0.1655	float	S-EOS coefficients: thermal expansion coefficient
`eos_b0`	0.76554	float	S-EOS coefficients: saline expansion coefficient
`eos_lambda1`	0.05952	float	S-EOS coefficients: cabbeling coefficient in T^2 (=0 for linear eos)
`eos_lambda2`	0.00074914	float	S-EOS coefficients: cabbeling coefficient in S^2 (=0 for linear eos)
`eos_mu1`	0.0001497	float	S-EOS coefficients: thermobaric coefficient in T (=0 for linear eos)
`eos_mu2`	1.109e-05	float	S-EOS coefficients: thermobaric coefficient in S (=0 for linear eos)
`eos_nu`	0.0024341	float	S-EOS coefficients: cabbeling coefficient in T*S (=0 for linear eos)

set_botFric

This section contains the parameters used to specify bottom friction settings

Parameter	Value	Type	Description
`botB_bfri2`	1.e-3	float	bottom drag coefficient (non-linear case)
`botB_bfeb2`	2.5e-3	float	bottom turbulent kinetic energy background (m²/s²)

set_xyturbTracers

This section includes the free input parameters for the parameterization of lateral subgrid-scale diffusion for tracers like temperature and salinity.

Parameter	Value	Type	Description
`tra_typeOperator`	1	int	type of the operator used (0)laplacian, (1)bilaplacian
`tra_eddycoeffSpec`	1	int	horizontal eddy coefficient specification (0)defined by coeff. tra_eddycoeff_child, (1)defined from coeff. tra_eddycoeff_father according to fat/child relation
`tra_eddycoeff_child`	-6.e8	float	horizontal eddy diffusivity (>0 (m^2/s) laplacian or <0 (m^4/s^2) bilaplacian) of the child model (if =NOTUSED, parameter not read)
`tra_eddycoeff_father`	-6.e8	float	horizontal eddy diffusivity (>0 (m^2/s) laplacian or <0 (m^4/s^2) bilaplacian) of the father model for fat/child coefficient relation (if =NOTUSED, parameter not read)
`tra_factor`	1.	float	factor used in fat/child coefficient relation (if laplacian: (a_child=factor*???), if bilaplacian: (a_child=factor*a_fat(Dx_child/Dx_fat)^4))

set_xyturbMomentum

This section includes the free input parameters that define the parameterization of lateral subgrid-scale viscosity for momentum.

Parameter	Value	Type	Description
`dyn_typeOperator`	1	int	type of the operator used (0)laplacian, (1)bilaplacian
`dyn_eddycoeffSpec`	1	int	horizontal eddy coefficient specification (0)defined by coeff. dyn_eddycoeff_child, (1)defined from coeff. dyn_eddycoeff_father according to fat/child relation
`dyn_eddycoeff_child`	-0.5e9	float	horizontal eddy viscosity (>0 (m^2/s) laplacian or <0 (m^4/s^2) bilaplacian) of the child model (if =NOTUSED, parameter not read)
`dyn_eddycoeff_father`	-1.e9	float	horizontal eddy viscosity (>0 (m^2/s) laplacian or <0 (m^4/s^2) bilaplacian) of the father model for fat/child coefficient relation (if =NOTUSED, parameter not read)
`dyn_factor`	1.	float	factor used in father/child coefficient relation (if laplacian: (a_child=factor*???), if bilaplacian: (a_child=factor*a_fat(Dx_child/Dx_fat)^4))

set_zturb

This section specifies the parameters for vertical turbulence, including eddy viscosity and diffusivity coefficients, which are essential for modeling vertical mixing and stratification in the water column.

Parameter	Value	Type	Description
zdyn_avm0	1.2e-5	float	vertical eddy viscosity [m²/s] (background Kz if not 'key_zdfcst')
zdyn_avt0	1.2e-6	float	vertical eddy diffusivity [m²/s] (background Kz if not 'key_zdfcst')
zdyn_avevd	10.	float	evd mixing coefficient [m²/s]
zdynric_avmri	1.e-2	float	maximum value of the vertical viscosity
zdynric_alp	5.	float	...
zdynric_ric	2	int	...
zdynric_ekmfc	0.7	float	...
zdynric_mldmin	1.0	float	...
zdynric_mldmax	1000.0	float	...
zdynric_wtmix	10.0	float	...
zdynric_wvmix	10.0	float	...
zdynric_mldw	False	bool	...
zdyntke_ediff	0.1	float	...
zdyntke_ediss	0.7	float	...
zdyntke_ebb	67.83	float	...
zdyntke_emin	1.e-6	float	...
zdyntke_emin0	1.e-4	float	...
zdyntke_mxl	2	int	...
zdyntke_pdl	1	int	...
zdyntke_lmxl0	True	bool	...
zdyntke_rmxl0	0.04	float	...
zdyntke_llc	True	bool	...
zdyntke_rlc	0.15	float	...
zdyntke_etau	0	int	...
zdyntke_efr	0.05	float	...
zdyntke_htau	1	int	...
zdyngls_emin	1.e-6	float	...
zdyngls_epsmin	1.e-12	float	...
zdyngls_length_lim	True	bool	...
zdyngls_clim_galp	0.53	float	...
zdyngls_sigpsi	True	bool	...

Parameter	Value	Type	Description
zdyngls_rcrban	100.	float	...
zdyngls_charn	70000.	float	...
zdyngls_hsro	0.02	float	...
zdyngls_frac_hs	1.3	float	...
zdyngls_z0_met	2	int	...
zdyngls_bc_surf	1	int	...
zdyngls_bc_bot	1	int	...
zdyngls_stab_func	2	int	...
zdyngls_clos	1	int	...
zdynkpp_kpprimix	True	bool	...
zdynkpp_difmiw	1.0e-04	float	...
zdynkpp_difsiw	0.1e-04	float	...
zdynkpp_riinfty	0.8	float	...
zdynkpp_difri	0.0050	float	...
zdynkpp_bvsqcon	-0.01e-07	float	...
zdynkpp_difcon	1.	float	...
zdynkpp_avb_k	0	int	...
zdynkpp_ave	1	int	...

5.1.7 **Dataset Download Configuration**

set_dataDownlCoast_urlName

This section defines the parameters used to construct the URL for accessing input coastline datasets. These datasets can be sourced from either a local directory or a remote repository.

Parameter	Value	Type	Description
`urlCoast_usr`	gofs	string	username to access the datasets from a remote FTP server
`urlCoast_pwd`	L##OOkkl!as	string	password to access the datasets from a remote FTP server
`urlCoast_urlbase`	file:///scratch/surf/surf_datasets/current/coastline/GSHHS_shp/(RESCOAST)	string	parametric URL name (i.e. ftp:/... or file:///...) for the coastline datasets. Parameters: (RESOL)
`urlCoast_resol`	h	string	name for spatial resolution used to replace the substring (RESCOAST) on the parametric URL name (if =NOTUSED, parameter not read)

set_dataDownlCoast_fileName

This section specifies the parameters required to generate the filenames of the input coastline datasets. These filenames help identify the relevant files in the storage location.

Parameter	Value	Type	Description
`fileCoast_lland`	True	bool	enables the use of the land coastline
`fileCoast_filebaseLand`	GSHHS_(RESCOAST)_L1.shp	string	files name for NOAA coastline datasets contains boundary between land and ocean (if fileCoast_lland=True). Parameters: (RESCOAST)
`fileCoast_llake`	False	bool	enables the use of the lake coastline
`fileCoast_filebaseLake`	GSHHS_(RESCOAST)_L2.shp	string	files name for NOAA coastline datasets contains boundary between lake and land (if fileCoast_llake=True). Parameters: (RESCOAST)
`fileCoast_lislandlake`	False	bool	enables the use of the isle-in-lake coastline
`fileCoast_filebaseIslandlake`	GSHHS_(RESCOAST)_L3.shp	string	files name for NOAA coastline datasets contains boundary between island-in-lake and lake (if fileCoast_lislandlake=True). Parameters: (RESCOAST)
`fileCoast_resol`	h	string	name for spatial resolution used to replace the substring (RESCOAST) on the parametric file name (if =NOTUSED, parameter not read)
`fileCoast_lcompression`	False	bool	enables if datasets you want to download are gzip compressed files (*.gz)
`fileCoast_lkeepSrcFull`	True	bool	enables if you want to keep the downloaded uncut datasets

set_dataDownlBat_urlName

This section defines the parameters used to construct the URL for accessing input bathymetry datasets. These datasets can be sourced from either a local directory or a remote repository.

Parameter	Value	Type	Description
`urlBat_usr`	gofs	string	username to access the datasets from a remote FTP server
`urlBat_pwd`	L##OOkkl!as	string	password to access the datasets from a remote FTP server
`urlBat_urlbase`	file:///scratch/surf/surf_datasets/current/bathymetry	string	parametric URL name (i.e. ftp:/... or file:///...) for the bathymetric datasets. Parameters: (RESOL)
`urlBat_resol`	h	string	name for spatial resolution used to replace the substring (RESOL) on the parametric URL name (if =NOTUSED, parameter not read)

set_dataDownlBat_fileName

This section specifies the parameters required to generate the filenames of the input bathymetry datasets. These filenames help identify the relevant files in the storage location.

Parameter	Value	Type	Description
`fileBat_filebase`	macroMED_bathyGEBCO.nc	string	parametric file name for the source bathymetric datasets. Parameters: (RESBAT)
`fileBat_resol`	NOTUSED	string	name for spatial resolution used to replace the substring (RESBAT) on the parametric file name (if =NOTUSED, parameter not read)
`fileBat_lcompression`	False	bool	enables if datasets you want to download are gzip compressed files (*.gz)
`fileBat_llonFlip`	False	bool	(=True) if longitude coordinate is in the 0 to 360 range, (=True) if longitude is in -180:+180 range
`fileBat_llonCycl`	True	bool	Logical variable=True if you use global data and longitude is cyclic
`fileBat_llatInv`	False	bool	enables if the dataset contains latitude decreasing through the pole
`fileBat_lkeepSrcFull`	True	bool	enables if you want to keep the downloaded uncut datasets

set_dataDownlBat_varName

This section outlines the parameters related to the variable names used within the input bathymetry datasets. These variable names are crucial for accurately extracting and utilizing the data within the files.

Parameter	Value	Type	Description
`srcDimBat_lon`	lon	string	name of the dimension for the longitude
`srcDimBat_lat`	lat	string	name of the dimension for the latitude
`srcCrdBat_lon`	lon	string	name of the variable for longitude coordinate
`srcCrdBat_lat`	lat	string	name of the variable for latitude coordinate
`srcVarBat_elev`	elevation	string	name of the variable for the Sea Floor Elevation

set_dataDownlAtmMesh_urlName

This section defines the parameters used to construct the URL for accessing atmospheric meshmask datasets. These datasets can be sourced from either a local directory or a remote repository.

Parameter	Value	Type	Description
`urlAtmMesh_usr`	gofs	string	username to access the input datasets before the spinup-time from a remote FTP server
`urlAtmMesh_pwd`	L##OOkkl!as	string	password to access the input datasets before the spinup-time from a remote FTP server
`urlAtmMesh_urlbase`	file:///scratch/surf/indata_offline/gulfTaranto_20141005/data/data00/ indata/atmosphere/srcFull	string	parametric URL name (i.e. ftp:/... or file:///...) for zonal air velocity datasets before the spinup-time. Parameters: (FIELD),YYYY(p)MM(p)DD(p)
`urlAtmMesh_velU`	v10m	string	name (for zonal air velocity) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_velV`	v10m	string	name (for meridional air velocity) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_mslp`	mslp	string	name (for mean sea-level pressure) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_cloudCov`	tcc	string	name (for total cloud cover) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_temp`	t2m	string	name (for air temperature) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_dpTemp`	d2m	string	name (for dewpoint temperature) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_prec`	precip	string	name (for total precipitation) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_tauU`	tauU	string	name (for zonal wind stress) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_tauV`	tauV	string	name (for meridional wind stress) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_qtot`	qtot	string	

Parameter	Value	Type	Description
			name (for total heat flux) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_qsr`	qsr	string	name (for solar Radiation Penetration) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_emp`	emp	string	name (for mass flux exchanged) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_tempS`	sst	string	name (for surface temperature) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_salS`	sss	string	name (for surface salinity) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_umid`	umid	string	name (for air humidity) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_radLW`	lwrd	string	name (for long wave radiation) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_radSW`	swrd	string	name (for short wave radiation) used to replace the substring (FIELD) on the parametric URL name
`urlAtmMesh_snow`	snow	string	name (for solid precipitation) used to replace the substring (FIELD) on the parametric URL name

set_dataDownlAtmMesh_fileName

This section specifies the parameters required to generate the filenames of the atmospheric meshmask datasets. These filenames help identify the relevant files in the storage location.

Parameter	Value	Type	Description
`fileAtmMesh_lcompression`	False	bool	enables if datasets you want to download are gzip compressed files (*.gz)
`fileAtmMesh_llonFlip`	False	bool	(=True) if longitude coord. is in the 0 to 360 range (=True) if longitude is in -180:+180 range
`fileAtmMesh_llonCycl`	True	bool	Logical variable=True if you use global data and longitude is cyclic
`fileAtmMesh_llatInv`	True	bool	enables if the dataset contains latitude decreasing through the pole
`fileAtmMesh_filebase_velU`	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the zonal air velocity datasets before the spinup-time (sbc_iformulat=0,2)
`fileAtmMesh_filebase_velV`	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the meridional air velocity datasets before the spinup-time (sbc_iformulat=0,2)
`fileAtmMesh_filebase_mslp`	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the Mean sea-level pressure datasets before the spinup-time (sbc_iformulat=0 or/and sbc_aprdyn)
`fileAtmMesh_filebase_cloudCov`	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the Total cloud cover datasets before the spinup-time (sbc_iformulat=0)
`fileAtmMesh_filebase_temp`	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the air temperature datasets before the spinup-time (sbc_iformulat=0,2)

Parameter	Value	Type	Description
`fileAtmMesh_filebase_dpTemp`	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the dewpoint temperature datasets before the spinup-time (sbc_iformulat=0)
`fileAtmMesh_filebase_prec`	YYYY(i)MM(i)DD(i)_YYYY(i+1)MM(i+1)DD(i+1)-ECMWF---AM025-MEDATL-bYYYY(i)MM(i)DD(i)_fc00-fv02.00_PREC.nc	string	parametric FILE name for the Total precipitation datasets before the spinup-time (sbc_iformulat=0,2)
`fileAtmMesh_filebase_tauU`	NOTUSED	string	parametric FILE name for the zonal Wind stress datasets before the spinup-time (sbc_iformulat=1)
`fileAtmMesh_filebase_tauV`	NOTUSED	string	parametric FILE name for the meridional Wind stress datasets before the spinup-time (sbc_iformulat=1)
`fileAtmMesh_filebase_qtot`	NOTUSED	string	parametric FILE name for the Total heat flux datasets before the spinup-time (sbc_iformulat=1)
`fileAtmMesh_filebase_qsr`	NOTUSED	string	parametric FILE name for the Solar Radiation Penetration datasets before the spinup-time (sbc_iformulat=1)
`fileAtmMesh_filebase_emp`	NOTUSED	string	parametric FILE name for the Mass flux exchanged datasets before the spinup-time (sbc_iformulat=1)
`fileAtmMesh_filebase_tempS`	NOTUSED	string	parametric FILE name for the Surface Temperature datasets before the spinup-time (sbc_iformulat=1)

Parameter	Value	Type	Description
`fileAtmMesh_filebase_salS`	NOTUSED	string	parametric FILE name for the Surface Salinity datasets before the spinup-time (sbc_iformulat=1)
`fileAtmMesh_filebase_umid`	NOTUSED	string	parametric FILE name for the Air Humidity datasets before the spinup-time (sbc_iformulat=1)
`fileAtmMesh_filebase_radLW`	NOTUSED	string	parametric FILE name for the Long Wave Radiation datasets before the spinup-time (sbc_iformulat=1)
`fileAtmMesh_filebase_radSW`	NOTUSED	string	parametric FILE name for the Short Wave Radiation datasets before the spinup-time (sbc_iformulat=1)
`fileAtmMesh_filebase_snow`	NOTUSED	string	parametric FILE name for the Solid precipitation datasets before the spinup-time (sbc_iformulat=1)
`fileAtmMesh_lkeepSrcFull`	True	bool	enables if you want to keep the downloaded uncut datasets
---			

set_dataDownlAtmMesh_varName

This section outlines the parameters related to the variable names used within the atmospheric meshmask datasets. These variable names are crucial for accurately extracting and utilizing the data within the files.

Parameter	Value	Type	Description
`srcDimAtmMesh_lon`	lon	string	name of the dimension for the longitude
`srcDimAtmMesh_lat`	lat	string	name of the dimension for the latitude
`srcDimAtmMesh_time`	time	string	name of the dimension for the time
`srcCrdAtmMesh_lon`	lon	string	name of the variable containing longitude coordinate
`srcCrdAtmMesh_lat`	lat	string	name of the variable containing latitude coordinate
`srcCrdAtmMesh_time`	time	string	name of the variable containing time coordinate
`srcVarAtmMesh_mask`	LSM	string	name of the variable containing the Land Sea Mask (if `sbc_iformulat=0,2`)
`srcVarAtmMesh_lont`	NOTUSED	string	name of the variable containing longitude coordinate (if `sbc_iformulat=1`)
`srcVarAtmMesh_lonu`	NOTUSED	string	name of the variable containing longitude coordinate (if `sbc_iformulat=1`)
`srcVarAtmMesh_lonv`	NOTUSED	string	name of the variable containing longitude coordinate (if `sbc_iformulat=1`)
`srcVarAtmMesh_latt`	NOTUSED	string	name of the variable containing latitude coordinate (if `sbc_iformulat=1`)
`srcVarAtmMesh_latu`	NOTUSED	string	name of the variable containing latitude coordinate (if `sbc_iformulat=1`)
`srcVarAtmMesh_latv`	NOTUSED	string	name of the variable containing latitude coordinate (if `sbc_iformulat=1`)
`srcVarAtmMesh_maskt`	NOTUSED	string	name of the variable containing the Land Sea Mask (if `sbc_iformulat=1`)
`srcVarAtmMesh_masku`	NOTUSED	string	name of the variable containing the Land Sea

Parameter	Value	Type	Description
			Mask (if `sbc_iformulat=1`)
`srcVarAtmMesh_maskv`	NOTUSED	string	name of the variable containing the Land Sea Mask (if `sbc_iformulat=1`)

set_dataDownlAtm_urlName_preSpinup

This section contains the parameters needed to build the URL for accessing pre-spinup atmospheric fields datasets. The datasets can be retrieved from either a local source or a remote repository.

Parameter	Value	Type	Description
urlAtmPre_usr	gofs	string	username to access the input datasets before the spinup-time from a remote ftp server
urlAtmPre_pwd	L##OOkkl!as	string	password to access the input datasets before the spinup-time from a remote ftp server
urlAtmPre_urlbase	file:///scratch/surf/indata_offline/gulfTaranto_20141005/data/data00/ indata/atmosphere/srcFull	string	parametric URL name (i.e. ftp:/... or file:///...) for atmospheric datasets before the spinup-time. Parameters: (FIELD),YYYY(p)MM(p)DD(p)
urlAtmPre_velU	v10m	string	name (for zonal air velocity) used to replace the substring (FIELD) on the parametric URL name
urlAtmPre_velV	v10m	string	name (for meridional air velocity) used to replace the substring (FIELD) on the parametric URL name
urlAtmPre_mslp	mslp	string	name (for mean sea-level pressure) used to replace the substring (FIELD) on the parametric URL name
urlAtmPre_cloudCov	tcc	string	name (for total cloud cover) used to replace the substring (FIELD) on the parametric URL name
urlAtmPre_temp	t2m	string	name (for air temperature) used to replace the substring (FIELD) on the parametric URL name
urlAtmPre_dpTemp	d2m	string	name (for dewpoint temperature) used to replace the substring (FIELD) on the parametric URL name
urlAtmPre_prec	precip	string	name (for total precipitation) used to replace the substring (FIELD) on the parametric URL name
urlAtmPre_tauU	tauU	string	name (for zonal wind stress) used to replace the substring (FIELD) on the parametric URL name
urlAtmPre_tauV	tauV	string	name (for meridional wind stress) used to replace the substring (FIELD) on the parametric URL name
urlAtmPre_qtot	qtot	string	name (for total heat flux) used to replace the substring

Parameter	Value	Type	Description
			(FIELD) on the parametric URL name
`urlAtmPre_qsr`	qsr	string	name (for solar Radiation Penetration) used to replace the substring (FIELD) on the parametric URL name
`urlAtmPre_emp`	emp	string	name (for mass flux exchanged) used to replace the substring (FIELD) on the parametric URL name
`urlAtmPre_tempS`	sst	string	name (for surface temperature) used to replace the substring (FIELD) on the parametric URL name
`urlAtmPre_salS`	sss	string	name (for surface salinity) used to replace the substring (FIELD) on the parametric URL name
`urlAtmPre_umid`	umid	string	name (for air humidity) used to replace the substring (FIELD) on the parametric URL name
`urlAtmPre_radLW`	lwrd	string	name (for long wave radiation) used to replace the substring (FIELD) on the parametric URL name
`urlAtmPre_radSW`	swrd	string	name (for short wave radiation) used to replace the substring (FIELD) on the parametric URL name
`urlAtmPre_snow`	snow	string	name (for solid precipitation) used to replace the substring (FIELD) on the parametric URL name

set_dataDownlAtm_fileName_preSpinup

This section provides the parameters required to create the filenames for the pre-spinup atmospheric fields datasets. These filenames are used to locate and manage the specific datasets within your data storage system.

Parameter	Value	Type	Description
fileAtmPre_lcompression	False	bool	enables if datasets you want to download are gzip compressed files (*.gz)
fileAtmPre_iProdDate	2	int	type of production (bulletin) date (=1):fixProdDate, (=2):varProdDate-DayofWeek
fileAtmPre_dateProdFixed	20170801	string	datasets production if iProdDate=1 is used in the URL/ files
fileAtmPre_dateProdDayofWeek	Wednesday	string	datasets production if iProdDate=2 is used in the URL/ files
fileAtmPre_filebase_velU	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the zonal air velocity datasets before the spinup-time (sbc_iformulat=0,2)
fileAtmPre_filebase_velV	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the meridional air velocity datasets before the spinup-time (sbc_iformulat=0,2)
fileAtmPre_filebase_mslp	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the Mean sea-level pressure datasets before the spinup-time (sbc_iformulat=0 or/and sbc_aprdyn)
fileAtmPre_filebase_cloudCov	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the Total cloud cover datasets before the spinup-time (sbc_iformulat=0)
fileAtmPre_filebase_temp	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the air temperature datasets before the

Parameter	Value	Type	Description
			spinup-time (sbc_iformulat=0,2)
fileAtmPre_filebase_dpTemp	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the dewpoint temperature datasets before the spinup-time (sbc_iformulat=0)
fileAtmPre_filebase_prec	YYYY(i)MM(i)DD(i)_YYYY(i+1)MM(i+1)DD(i+1)-ECMWF---AM025-MEDATL-bYYYY(i)MM(i)DD(i)_fc00-fv02.00_PREC.nc	string	parametric FILE name for the Total precipitation datasets before the spinup-time (sbc_iformulat=0,2)
fileAtmPre_filebase_tauU	NOTUSED	string	parametric FILE name for the zonal Wind stress datasets before the spinup-time (sbc_iformulat=1)
fileAtmPre_filebase_tauV	NOTUSED	string	parametric FILE name for the meridional Wind stress datasets before the spinup-time (sbc_iformulat=1)
fileAtmPre_filebase_qtot	NOTUSED	string	parametric FILE name for the Total heat flux datasets before the spinup-time (sbc_iformulat=1)
fileAtmPre_filebase_qsr	NOTUSED	string	parametric FILE name for the Solar Radiation Penetration datasets before the spinup-time (sbc_iformulat=1)
fileAtmPre_filebase_emp	NOTUSED	string	parametric FILE name for the Mass flux exchanged datasets before the spinup-time (sbc_iformulat=1)
fileAtmPre_filebase_tempS	NOTUSED	string	parametric FILE name for the Surface Temperature datasets before the

Parameter	Value	Type	Description
			spinup-time (sbc_iformulat=1)
`fileAtmPre_filebase_salS`	NOTUSED	string	parametric FILE name for the Surface Salinity datasets before the spinup-time (sbc_iformulat=1)
`fileAtmPre_filebase_umid`	NOTUSED	string	parametric FILE name for the Air Umidity datasets before the spinup-time (sbc_iformulat=1)
`fileAtmPre_filebase_radLW`	NOTUSED	string	parametric FILE name for the Long Wave Radiation datasets before the spinup-time (sbc_iformulat=1)
`fileAtmPre_filebase_radSW`	NOTUSED	string	parametric FILE name for the Short Wave Radiation datasets before the spinup-time (sbc_iformulat=1)
`fileAtmPre_filebase_snow`	NOTUSED	string	parametric FILE name for the Solid precipitation datasets before the spinup-time (sbc_iformulat=1)

set_dataDownlAtm_varName_preSpinup

This section details the parameters for specifying the field names of the pre-spinup atmospheric datasets. The field names identify the specific variables within the datasets, ensuring proper data extraction and processing.

Parameter	Value	Type	Description
srcDimAtmPre_lon	lon	string	name of the dimension for the longitude (sbc_iformulat=0,2)
srcDimAtmPre_lont	NOTUSED	string	name of the dimension for the longitude (sbc_iformulat=1)
srcDimAtmPre_lonu	NOTUSED	string	name of the dimension for the longitude (sbc_iformulat=1)
srcDimAtmPre_lonv	NOTUSED	string	name of the dimension for the longitude (sbc_iformulat=1)
srcDimAtmPre_lat	lat	string	name of the dimension for the latitude (sbc_iformulat=0,2)
srcDimAtmPre_latt	NOTUSED	string	name of the dimension for the latitude (sbc_iformulat=1)
srcDimAtmPre_latu	NOTUSED	string	name of the dimension for the latitude (sbc_iformulat=1)
srcDimAtmPre_latv	NOTUSED	string	name of the dimension for the latitude (sbc_iformulat=1)
srcDimAtmPre_time	time	string	name of the dimension for the time
srcCrdAtmPre_lon	lon	string	name of the variable containing longitude coordinate (sbc_iformulat=0,2)
srcCrdAtmPre_lont	NOTUSED	string	name of the variable containing longitude coordinate (sbc_iformulat=1)
srcCrdAtmPre_lonu	NOTUSED	string	name of the variable containing longitude coordinate (sbc_iformulat=1)
srcCrdAtmPre_lonv	NOTUSED	string	name of the variable containing longitude coordinate (sbc_iformulat=1)
srcCrdAtmPre_lat	lat	string	name of the variable containing latitude coordinate (sbc_iformulat=0,2)
srcCrdAtmPre_latt	NOTUSED	string	name of the variable containing latitude

Parameter	Value	Type	Description
			coordinate (sbc_iformulat=1)
srcCrdAtmPre_latu	NOTUSED	string	name of the variable containing latitude coordinate (sbc_iformulat=1)
srcCrdAtmPre_latv	NOTUSED	string	name of the variable containing latitude coordinate (sbc_iformulat=1)
srcCrdAtmPre_time	time	string	name of the variable containing time coordinate
srcVarAtmPre_velU	U10M	string	name of the variable for 10 metre zonal component of air velocity (sbc_iformulat=0,2)
srcVarAtmPre_velV	V10M	string	name of the variable for 10 metre meridional component of air velocity (sbc_iformulat=0,2)
srcVarAtmPre_mslp	MSL	string	name of the variable for Mean sea-level pressure (sbc_iformulat=0 or/and sbc_aprdyn)
srcVarAtmPre_cloudCov	TCC	string	name of the variable for Total cloud cover (sbc_iformulat=0)
srcVarAtmPre_temp	T2M	string	name of the variable for 2 metre temperature (sbc_iformulat=0,2)
srcVarAtmPre_dpTemp	D2M	string	name of the variable for 2 metre dewpoint temperature (sbc_iformulat=0)
srcVarAtmPre_prec	PREC	string	name of the variable for Total precipitation (sbc_iformulat=0,2)
srcVarAtmPre_tauU	tauU	string	name of the variable for tauU (sbc_iformulat=1)
srcVarAtmPre_tauV	tauV	string	name of the variable for tauV (sbc_iformulat=1)
srcVarAtmPre_qtot	qtot	string	name of the variable for qtot (sbc_iformulat=1)
srcVarAtmPre_qsr	qsr	string	name of the variable for qsr (sbc_iformulat=1)

Parameter	Value	Type	Description
srcVarAtmPre_emp	emp	string	name of the variable for emp (sbc_iformulat=1)
srcVarAtmPre_tempS	tempS	string	name of the variable for tempS (sbc_iformulat=1)
srcVarAtmPre_salS	salS	string	name of the variable for salS (sbc_iformulat=1)
srcVarAtmPre_umid	q2m	string	name of the variable for 2 metre humidity (sbc_iformulat=2)
srcVarAtmPre_radLW	lwrd	string	name of the variable for Long wave radiation (sbc_iformulat=2)
srcVarAtmPre_radSW	swrd	string	name of the variable for Short wave radiation (sbc_iformulat=2)
srcVarAtmPre_snow	snow	string	name of the variable for Solid precipitation (sbc_iformulat=2)

set_dataDownlAtm_urlName_postSpinup

This section contains the parameters needed to build the URL for accessing post-spinup atmospheric fields datasets. The datasets can be retrieved from either a local source or a remote repository.

Parameter	Value	Type	Description
urlAtmPost_usr	gofs	string	username to access the input datasets after the spinup-time from a remote ftp server
urlAtmPost_pwd	L##OOkkl!as	string	password to access the input datasets after the spinup-time from a remote ftp server
urlAtmPost_urlbase	file:///scratch/surf/indata_offline/gulfTaranto_20141005/data/data00/ indata/atmosphere/srcFull	string	parametric URL name for atmospheric datasets after the spinup-time. Parameters: (FIELD),YYYY(p)MM(p)DD(p)
urlAtmPost_velU	v10m	string	name (for zonal air velocity) used to replace the substring (FIELD) on the parametrinc url name
urlAtmPost_velV	v10m	string	name (for meridional air velocity) used to replace the substring (FIELD) on the parametrinc url name
urlAtmPost_mslp	mslp	string	name (for mean sea-level pressure) used to replace the substring (FIELD) on the parametrinc url name
urlAtmPost_cloudCov	tcc	string	name (for total cloud cover) used to replace the substring (FIELD) on the parametrinc url name
urlAtmPost_temp	t2m	string	name (for air temperature) used to replace the substring (FIELD) on the parametrinc url name
urlAtmPost_dpTemp	d2m	string	name (for dewpoint temperature) used to replace the substring (FIELD) on the parametrinc url name
urlAtmPost_prec	precip	string	name (for total precipitation) used to replace the substring (FIELD) on the parametrinc url name
urlAtmPost_tauU	tauU	string	name (for zonal wind stress) used to replace the substring (FIELD) on the parametrinc url name
urlAtmPost_tauV	tauV	string	name (for meridional wind stress) used to replace the substring (FIELD) on the parametrinc url name
urlAtmPost_qtot	qtot	string	name (for total heat flux) used to replace the substring

Parameter	Value	Type	Description
			(FIELD) on the parametrinc url name
`urlAtmPost_qsr`	qsr	string	name (for solar Radiation Penetration) used to replace the substring (FIELD) on the parametrinc url name
`urlAtmPost_emp`	emp	string	name (for mass flux exchanged) used to replace the substring (FIELD) on the parametrinc url name
`urlAtmPost_tempS`	sst	string	name (for surface temperature) used to replace the substring (FIELD) on the parametrinc url name
`urlAtmPost_salS`	sss	string	name (for surface salinity) used to replace the substring (FIELD) on the parametrinc url name
`urlAtmPost_umid`	umid	string	name (for air umidity) used to replace the substring (FIELD) on the parametrinc url name
`urlAtmPost_radLW`	lwrd	string	name (for long wave radiation) used to replace the substring (FIELD) on the parametrinc url name
`urlAtmPost_radSW`	swrd	string	name (for short wave radiation) used to replace the substring (FIELD) on the parametrinc url name
`urlAtmPost_snow`	snow	string	name (for solid precipitation) used to replace the substring (FIELD) on the parametrinc url name

set_dataDownlAtm_fileName_postSpinup

This section provides the parameters required to create the filenames for the post-spinup atmospheric fields datasets. These filenames are used to locate and manage the specific datasets within your data storage system.

Parameter	Value	Type	Description
fileAtmPost_lcompression	False	bool	enables if datasets you want to download are gzip compressed files (*.gz)
fileAtmPost_iProdDate	2	int	type of production (bulletin) date (=1):fixProdDate, (=2):varProdDate-DayofWeek
fileAtmPost_dateProdFixed	20170801	string	datasets production if iProdDate=1 is used in the URL/ files
fileAtmPost_dateProdDayofWeek	Wednesday	string	datasets production if iProdDate=2 is used in the URL/ files
fileAtmPost_filebase_velU	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the zonal air velocity datasets after the spinup-time
fileAtmPost_filebase_velV	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the meridional air velocity datasets after the spinup-time
fileAtmPost_filebase_mslp	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the Mean sea-level pressure datasets after the spinup-time
fileAtmPost_filebase_cloudCov	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the Total cloud cover datasets after the spinup-time
fileAtmPost_filebase_temp	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the air temperature datasets after the spinup-time
fileAtmPost_filebase_dpTemp	YYYY(i)MM(i)DD(i)-ECMWF---AM0125-MEDATL-bYYYY(i+1)MM(i+1)DD(i+1)_an-fv05.00.nc	string	parametric FILE name for the dewpoint temperature

Parameter	Value	Type	Description
			datasets after the spinup-time
`fileAtmPost_filebase_prec`	YYYY(i)MM(i)DD(i)_YYYY(i+1)MM(i+1)DD(i+1)-ECMWF---AM025-MEDATL-bYYYY(i)MM(i)DD(i)_fc00-fv02.00_PREC.nc	string	parametric FILE name for the Total precipitation datasets after the spinup-time
`fileAtmPost_filebase_tauU`	NOTUSED	string	parametric FILE name for the zonal Wind stress datasets after the spinup-time
`fileAtmPost_filebase_tauV`	NOTUSED	string	parametric FILE name for the meridional Wind stress datasets after the spinup-time
`fileAtmPost_filebase_qtot`	NOTUSED	string	parametric FILE name for the Total heat flux datasets after the spinup-time
`fileAtmPost_filebase_qsr`	NOTUSED	string	parametric FILE name for the Solar Radiation Penetration datasets after the spinup-time
`fileAtmPost_filebase_emp`	NOTUSED	string	parametric FILE name for the Mass flux exchanged datasets after the spinup-time
`fileAtmPost_filebase_tempS`	NOTUSED	string	parametric FILE name for the Surface Temperature datasets after the spinup-time
`fileAtmPost_filebase_salS`	NOTUSED	string	parametric FILE name for the Surface Salinity datasets after the spinup-time
`fileAtmPost_filebase_umid`	NOTUSED	string	parametric FILE name for the Air Umidity datasets after the spinup-time

Parameter	Value	Type	Description
`fileAtmPost_filebase_radLW`	NOTUSED	string	parametric FILE name for the Long Wave Radiation datasets after the spinup-time
`fileAtmPost_filebase_radSW`	NOTUSED	string	parametric FILE name for the Short Wave Radiation datasets after the spinup-time
`fileAtmPost_filebase_snow`	NOTUSED	string	parametric FILE name for the Solid precipitation datasets after the spinup-time

set_dataDownlAtm_varName_postSpinup

This section details the parameters for specifying the field names of the post-spinup atmospheric datasets. The field names identify the specific variables within the datasets, ensuring proper data extraction and processing.

Parameter	Value	Type	Description
srcDimAtmPost_lon	lon	string	name of the dimension for the longitude (sbc_iformulat=0,2)
srcDimAtmPost_lont	lon	string	name of the dimension for the longitude (sbc_iformulat=1)
srcDimAtmPost_lonu	NOTUSED	string	name of the dimension for the longitude (sbc_iformulat=1)
srcDimAtmPost_lonv	NOTUSED	string	name of the dimension for the longitude (sbc_iformulat=1)
srcDimAtmPost_lat	lat	string	name of the dimension for the latitude (sbc_iformulat=0,2)
srcDimAtmPost_latt	lat	string	name of the dimension for the latitude (sbc_iformulat=1)
srcDimAtmPost_latu	NOTUSED	string	name of the dimension for the latitude (sbc_iformulat=1)
srcDimAtmPost_latv	NOTUSED	string	name of the dimension for the latitude (sbc_iformulat=1)
srcDimAtmPost_time	time	string	name of the dimension for the time
srcCrdAtmPost_lon	lon	string	name of the variable containing longitude coordinate (sbc_iformulat=0,2)
srcCrdAtmPost_lont	lon	string	name of the variable containing longitude coordinate (sbc_iformulat=1)
srcCrdAtmPost_lonu	NOTUSED	string	name of the variable containing longitude coordinate (sbc_iformulat=1)
srcCrdAtmPost_lonv	NOTUSED	string	name of the variable containing longitude coordinate (sbc_iformulat=1)
srcCrdAtmPost_lat	lat	string	name of the variable containing latitude coordinate (sbc_iformulat=0,2)
srcCrdAtmPost_latt	lat	string	name of the variable containing latitude

Parameter	Value	Type	Description
			coordinate (sbc_iformulat=1)
srcCrdAtmPost_latu	NOTUSED	string	name of the variable containing latitude coordinate (sbc_iformulat=1)
srcCrdAtmPost_latv	NOTUSED	string	name of the variable containing latitude coordinate (sbc_iformulat=1)
srcCrdAtmPost_time	time	string	name of the variable containing time coordinate
srcVarAtmPost_velU	U10M	string	name of the variable for 10 metre zonal component of air velocity (sbc_iformulat=0,2)
srcVarAtmPost_velV	V10M	string	name of the variable for 10 metre meridional component of air velocity (sbc_iformulat=0,2)
srcVarAtmPost_mslp	MSL	string	name of the variable for Mean sea-level pressure (sbc_iformulat=0 or/ and sbc_aprdyn)
srcVarAtmPost_cloudCov	TCC	string	name of the variable for Total cloud cover (sbc_iformulat=0)
srcVarAtmPost_temp	T2M	string	name of the variable for 2 metre temperature (sbc_iformulat=0,2)
srcVarAtmPost_dpTemp	D2M	string	name of the variable for 2 metre dewpoint temperature (sbc_iformulat=0)
srcVarAtmPost_prec	PREC	string	name of the variable for Total precipitation (sbc_iformulat=0,2)
srcVarAtmPost_tauU	tauU	string	name of the variable for tauU (sbc_iformulat=1)
srcVarAtmPost_tauV	tauV	string	name of the variable for tauV (sbc_iformulat=1)
srcVarAtmPost_qtot	qtot	string	

Parameter	Value	Type	Description
			name of the variable for qtot (sbc_iformulat=1)
`srcVarAtmPost_qsr`	qsr	string	name of the variable for qsr (sbc_iformulat=1)
`srcVarAtmPost_emp`	emp	string	name of the variable for emp (sbc_iformulat=1)
`srcVarAtmPost_tempS`	tempS	string	name of the variable for tempS (sbc_iformulat=1)
`srcVarAtmPost_salS`	salS	string	name of the variable for salS (sbc_iformulat=1)
`srcVarAtmPost_umid`	q2m	string	name of the variable for 2 metre umidity (sbc_iformulat=2)
`srcVarAtmPost_radLW`	lwrd	string	name of the variable for Long wave radiation (sbc_iformulat=2)
`srcVarAtmPost_radSW`	swrd	string	name of the variable for Short wave radiation (sbc_iformulat=2)
`srcVarAtmPost_snow`	snow	string	name of the variable for Solid precipitation (sbc_iformulat=2)

set_dataDownIOceICMesh_urlName

This section defines the parameters used to construct the URL for accessing Ocean Initial Condition meshmask datasets. These datasets can be sourced from either a local directory or a remote repository.

Parameter	Value	Type	Description
urlOceICMesh_usr	gofs	string	username to access the datasets from a remote ftp server
urlOceICMesh_pwd	L##OOkkl!as	string	password to access the datasets from a remote ftp server
urlOceICMesh_urlbase	file:///scratch/surf/indata_offline/gulfTaranto_20141005/data/data00/ indata/ocean/oceanIC/srcFull	string	parametric URL (ftp:/ or file:///) name for ocean input IC-meshmask datasets. Parameters: (FIELD)
urlOceICMesh_lont	glamt	string	name (for longitude-Tgrid) used to replace the substring (FIELD) on the parametric URL name
urlOceICMesh_lonu	glamu	string	name (for longitude-Ugrid) used to replace the substring (FIELD) on the parametric URL name
urlOceICMesh_lonv	glamv	string	name (for longitude-Vgrid) used to replace the substring (FIELD) on the parametric URL name
urlOceICMesh_lonf	glamf	string	name (for longitude-Fgrid) used to replace the substring (FIELD) on the parametric URL name
urlOceICMesh_latt	gphit	string	name (for latitude-Tgrid) used to replace the substring (FIELD) on the parametric URL name
urlOceICMesh_latu	gphiu	string	

Parameter	Value	Type	Description
			name (for latitude-Ugrid) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_latv`	gphiv	string	name (for latitude-Vgrid) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_latf`	gphif	string	name (for latitude-Fgrid) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_deptht1d`	gdept_0	string	name (for depth-Tgrid) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_depthw1d`	gdepw_0	string	name (for depth-Wgrid) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_xscalfctt`	e1t	string	name (for scaleFactor-Tgrid x-direction) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_xscalfctu`	e1u	string	name (for scaleFactor-Ugrid x-direction) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_xscalfctv`	e1v	string	name (for scaleFactor-

Parameter	Value	Type	Description
			Vgrid x-direction) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_xscalfctf`	e1f	string	name (for scaleFactor-Fgrid x-direction) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_yscalfctt`	e2t	string	name (for scaleFactor-Tgrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_yscalfctu`	e2u	string	name (for scaleFactor-Ugrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_yscalfctv`	e2v	string	name (for scaleFactor-Vgrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_yscalfctf`	e2f	string	name (for scaleFactor-Fgrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_zscalfctt1d`	e3t_0	string	name (for scaleFactor-Tgrid z-direction) used to replace the substring (FIELD) on the parametric URL name

Parameter	Value	Type	Description
`urlOceICMesh_zscalfctw1d`	e3w_0	string	name (for scaleFactor-Wgrid z-direction) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_maskt`	tmask	string	name (for LandSea-mask-Tgrid) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_masku`	umask	string	name (for LandSea-mask-Ugrid) used to replace the substring (FIELD) on the parametric URL name
`urlOceICMesh_maskv`	vmask	string	name (for LandSea-mask-Vgrid) used to replace the substring (FIELD) on the parametric URL name

set_dataDownIOceICMesh_fileName

This section specifies the parameters required to generate the filenames of the Ocean Initial Condition meshmask datasets. These filenames help identify the relevant files in the storage location.

Parameter	Value	Type	Description
fileOceICMesh_filebase	meshmask_SYS4a3_IONIAN.nc	string	parametric FILE name for ocean source input data. Parameters: (FIELD)
fileOceICMesh_lcompression	False	bool	enables if datasets you want to download are gzip compressed files (*.gz)
fileOceICMesh_llonFlip	False	bool	(=True) if longitude coord. is in the 0 to 360 range (=True) if longitude is in -180:+180 range
fileOceICMesh_llonCycl	True	bool	Logical variable=True if you use global data and longitude is cyclic
fileOceICMesh_llatInv	False	bool	enables if the dataset contains latitude decreasing through the pole
fileOceICMesh_lont	coordinates	string	name (for longitude-Tgrid) used to replace the substring (FIELD) on the parametric URL name
fileOceICMesh_lonu	coordinates	string	name (for longitude-Ugrid) used to replace the substring (FIELD) on the parametric URL name
fileOceICMesh_lonv	coordinates	string	name (for longitude-Vgrid) used to replace the substring (FIELD) on the parametric URL name
fileOceICMesh_lonf	coordinates	string	name (for longitude-Fgrid) used to replace

Parameter	Value	Type	Description
			the substring (FIELD) on the parametric URL name
`fileOceICMesh_latt`	coordinates	string	name (for latitude-Tgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_latu`	coordinates	string	name (for latitude-Ugrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_latv`	coordinates	string	name (for latitude-Vgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_latf`	coordinates	string	name (for latitude-Fgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_depht1d`	meshmask_vertical	string	name (for depth-Tgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_depthw1d`	meshmask_vertical	string	name (for depth-Wgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_xscalfctt`	coordinates	string	name (for scaleFactor-Tgrid x-direction) used to replace the substring (FIELD) on the

Parameter	Value	Type	Description
			parametric URL name
`fileOceICMesh_xscalfctu`	coordinates	string	name (for scaleFactor-Ugrid x-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_xscalfctv`	coordinates	string	name (for scaleFactor-Vgrid x-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_xscalfctf`	coordinates	string	name (for scaleFactor-Fgrid x-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_yscalfctt`	coordinates	string	name (for scaleFactor-Tgrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_yscalfctu`	coordinates	string	name (for scaleFactor-Ugrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_yscalfctv`	coordinates	string	name (for scaleFactor-Vgrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_yscalfctf`	coordinates	string	name (for scaleFactor-Fgrid y-direction) used

Parameter	Value	Type	Description
			to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_zscalfctt1d`	meshmask_vertical	string	name (for scaleFactor-Tgrid z-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_zscalfctw1d`	meshmask_vertical	string	name (for scaleFactor-Wgrid z-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_maskt`	mesh_tmask	string	name (for LandSea-mask-Tgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_masku`	mesh_umask	string	name (for LandSea-mask-Ugrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_maskv`	mesh_vmask	string	name (for LandSea-mask-Vgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceICMesh_lkeepSrcFull`	True	bool	enables if you want keep the downloaded uncutted datasets

set_dataDownlOceICMesh_varName

This section outlines the parameters related to the variable names used within the Ocean Initial Condition meshmask datasets. These variable names are crucial for accurately extracting and utilizing the data within the files.

Parameter	Value	Type	Description
srcDimOceICMesh_lont	x	string	name of the dimension for the longitude
srcDimOceICMesh_lonu	x	string	name of the dimension for the longitude
srcDimOceICMesh_lonv	x	string	name of the dimension for the longitude
srcDimOceICMesh_lonw	x	string	name of the dimension for the longitude
srcDimOceICMesh_latt	y	string	name of the dimension for the latitude
srcDimOceICMesh_latu	y	string	name of the dimension for the latitude
srcDimOceICMesh_latv	y	string	name of the dimension for the latitude
srcDimOceICMesh_latw	y	string	name of the dimension for the latitude
srcDimOceICMesh_deptht	z	string	name of the dimension for the depth
srcDimOceICMesh_depthu	z	string	name of the dimension for the depth
srcDimOceICMesh_depthv	z	string	name of the dimension for the depth
srcDimOceICMesh_depthw	z	string	name of the dimension for the depth
srcDimOceICMesh_time	t	string	name of the dimension for the time
srcCrdOceICMesh_lont	nav_lon	string	name of the coordinate variable for the longitude
srcCrdOceICMesh_lonu	nav_lon	string	name of the coordinate variable for the longitude
srcCrdOceICMesh_lonv	nav_lon	string	

Parameter	Value	Type	Description
			name of the coordinate variable for the longitude
`srcCrdOceICMesh_latt`	nav_lat	string	name of the coordinate variable for the latitude
`srcCrdOceICMesh_latu`	nav_lat	string	name of the coordinate variable for the latitude
`srcCrdOceICMesh_latv`	nav_lat	string	name of the coordinate variable for the latitude
`srcCrdOceICMesh_deptht`	nav_lev	string	name of the coordinate variable for the depth
`srcCrdOceICMesh_depthw`	nav_lev	string	name of the coordinate variable for the depth
`srcCrdOceICMesh_time`	time_counter	string	name of the coordinate variable for the time
`srcVarOceICMesh_lont`	glamt	string	name of the variable containing longitude coordinate
`srcVarOceICMesh_lonu`	glamu	string	name of the variable containing longitude coordinate
`srcVarOceICMesh_lonv`	glamv	string	name of the variable containing longitude coordinate
`srcVarOceICMesh_lonf`	glamf	string	name of the variable containing longitude coordinate
`srcVarOceICMesh_latt`	gphit	string	name of the variable containing

Parameter	Value	Type	Description
			latitude coordinate
`srcVarOceICMesh_latu`	gphiu	string	name of the variable containing latitude coordinate
`srcVarOceICMesh_latv`	gphiv	string	name of the variable containing latitude coordinate
`srcVarOceICMesh_latf`	gphif	string	name of the variable containing latitude coordinate
`srcVarOceICMesh_depht1d`	gdept_0	string	name of the variable containing depth coordinate
`srcVarOceICMesh_depthw1d`	gdepw_0	string	name of the variable containing depth coordinate
`srcVarOceICMesh_xscalfctt`	e1t	string	name of the scale factors in zonal direction
`srcVarOceICMesh_xscalfctu`	e1u	string	name of the scale factors in zonal direction
`srcVarOceICMesh_xscalfctv`	e1v	string	name of the scale factors in zonal direction
`srcVarOceICMesh_xscalfctf`	e1f	string	name of the scale factors in zonal direction
`srcVarOceICMesh_yscalfctt`	e2t	string	name of the scale factors in meridional direction
`srcVarOceICMesh_yscalfctu`	e2u	string	name of the scale factors in meridional direction
`srcVarOceICMesh_yscalfctv`	e2v	string	name of the scale factors in meridional direction

Parameter	Value	Type	Description
`srcVarOceICMesh_yscalfctf`	e2f	string	name of the scale factors in meridional direction
`srcVarOceICMesh_zscalfctt1d`	e3t_0	string	name of the scale factors in vertical direction
`srcVarOceICMesh_zscalfctw1d`	e3w_0	string	name of the scale factors in vertical direction
`srcVarOceICMesh_maskt`	tmask	string	name of the land-sea-mask on T-points
`srcVarOceICMesh_masku`	umask	string	name of the land-sea-mask on U-points
`srcVarOceICMesh_maskv`	vmask	string	name of the land-sea-mask on V-points

set_dataDownIOceIC_urlName

This section contains the parameters needed to build the URL for accessing Ocean Initial Condition fields datasets. The datasets can be retrieved from either a local source or a remote repository.

Parameter	Value	Type	Description
`urlOceIC_usr`	gofs	string	Username to access the datasets from a remote FTP server
`urlOceIC_pwd`	L##OOkkl!as	string	Password to access the datasets from a remote FTP server
`urlOceIC_urlbase`	file:///scratch/surf/indata_offline/gulfTaranto_20141005/data/data00/ indata/ocean/oceanIC/srcFull	string	Parametric URL (ftp:/ or file:///) name for ocean input IC datasets. Parameters: (FIELD), YYYY(p)MM(p)DD(p), YYYY(i)MM(i)DD(i), YYYY(i-1)MM(i-1)DD(i-1), YYYY(i+1)MM(i+1)DD(i+1)
`urlOceIC_temp`	tem	string	Name (for Temperature) used to replace the substring (FIELD) on the parametric URL name
`urlOceIC_sal`	sal	string	Name (for Salinity) used to replace the substring (FIELD) on the parametric URL name
`urlOceIC_ssh`	ssh	string	Name (for Sea Surface Height) used to replace the substring (GRID) on the parametric URL name
`urlOceIC_velU`	cur	string	Name (for Zonal Current) used to replace the substring (GRID) on the parametric URL name
`urlOceIC_velV`	cur	string	Name (for Meridional Current) used to replace the substring (FIELD) on the parametric URL name
`urlOceIC_tempGrid`	gridT	string	Name (for Temperature) used to replace the substring (GRID) on the parametric URL name
`urlOceIC_salGrid`	gridT	string	Name (for Salinity) used to replace the substring (GRID) on the parametric URL name
`urlOceIC_sshGrid`	gridT	string	Name (for Sea Surface Height) used to replace the substring (GRID) on the parametric URL name
`urlOceIC_velUGrid`	gridU	string	Name (for Zonal Current) used to replace the substring (GRID) on the parametric URL name

Parameter	Value	Type	Description
urlOceIC_velVGrid	gridV	string	Name (for Meridional Current) used to replace the substring (GRID) on the parametric URL name

set_dataDownlOceIC_fileName

This section provides the parameters required to create the filenames for the pre-spinup Ocean Initial Condition fields datasets. These filenames are used to locate and manage the specific datasets within your data storage system.

Parameter	Value	Type	Description
fileOceIC_filebase	EXP1_EAS1_1d_YYYY(i)MM(i)DD(i)_ YYYY(i+1)MM(i+1)DD(i+1)_(GRID)_IONIAN.nc	string	Parametric FILE name for ocean (FIELD),YYYY(p)MM(p)DD(p),Y
fileOceIC_lcompression	False	bool	Enables if datasets you want to
fileOceIC_iProdDate	1	int	File name format to be downloa
fileOceIC_dateProdFixed	20170801	string	Datasets production if iProdDate
fileOceIC_dateProdDayofWeek	Wednesday	string	Datasets production if iProdDate
fileOceIC_temp	_votemper	string	Name (for Temperature) used to
fileOceIC_sal	_vosaline	string	Name (for Salinity) used to repla
fileOceIC_ssh	2D	string	Name (for Sea Surface Height)
fileOceIC_velU	_vozocrtx	string	Name (for Zonal Current) used t
fileOceIC_velV	_vomecrty	string	Name (for Meridional Current) u
fileOceIC_tempGrid	grid_T	string	Name (for Temperature) used to
fileOceIC_salGrid	grid_T	string	Name (for Salinity) used to repla
fileOceIC_sshGrid	grid_T	string	Name (for Sea Surface Height)
fileOceIC_velUGrid	grid_U	string	Name (for Zonal Current) used t
fileOceIC_velVGrid	grid_V	string	Name (for Meridional Current) u

set_dataDownIOceIC_varName

This section details the parameters for specifying the field names of the Ocean Initial Condition datasets. The field names identify the specific variables within the datasets, ensuring proper data extraction and processing.

Parameter	Value	Type	Description
srcDimOceIC_lont	x	string	Name of the dimension for the longitude
srcDimOceIC_lonu	x	string	Name of the dimension for the longitude
srcDimOceIC_lonv	x	string	Name of the dimension for the longitude
srcDimOceIC_lonw	x	string	Name of the dimension for the longitude
srcDimOceIC_latt	y	string	Name of the dimension for the latitude
srcDimOceIC_latu	y	string	Name of the dimension for the latitude
srcDimOceIC_latv	y	string	Name of the dimension for the latitude
srcDimOceIC_latw	y	string	Name of the dimension for the latitude
srcDimOceIC_deptht	deptht	string	Name of the dimension for the depth
srcDimOceIC_depthu	depthu	string	Name of the dimension for the depth
srcDimOceIC_depthv	depthv	string	Name of the dimension for the depth
srcDimOceIC_depthw	depthw	string	Name of the dimension for the depth
srcDimOceIC_time	t	string	Name of the dimension for the time
srcCrdOceIC_lont	nav_lon	string	Name of the coordinate variable for the longitude coordinate
srcCrdOceIC_lonu	nav_lon	string	Name of the coordinate variable for the longitude coordinate
srcCrdOceIC_lonv	nav_lon	string	Name of the coordinate variable for the longitude coordinate
srcCrdOceIC_lonw	nav_lon	string	Name of the coordinate variable for the longitude coordinate
srcCrdOceIC_latt	nav_lat	string	Name of the coordinate variable for the latitude coordinate
srcCrdOceIC_latu	nav_lat	string	Name of the coordinate variable for the latitude coordinate
srcCrdOceIC_latv	nav_lat	string	

Parameter	Value	Type	Description
			Name of the coordinate variable for the latitude coordinate
`srcCrdOceIC_latw`	nav_lat	string	Name of the coordinate variable for the latitude coordinate
`srcCrdOceIC_deptht`	deptht	string	Name of the coordinate variable for the depth coordinate
`srcCrdOceIC_depthu`	depthu	string	Name of the coordinate variable for the depth coordinate
`srcCrdOceIC_depthv`	depthv	string	Name of the coordinate variable for the depth coordinate
`srcCrdOceIC_depthw`	depthw	string	Name of the coordinate variable for the depth coordinate
`srcCrdOceIC_time`	time_counter	string	Name of the coordinate variable for the time coordinate
`srcVarOceIC_temp`	votemper	string	Name of the variable for the Temperature
`srcVarOceIC_sal`	vosaline	string	Name of the variable for the Salinity
`srcVarOceIC_ssh`	sossheig	string	Name of the variable for the Sea Surface Height
`srcVarOceIC_velU`	vozocrtx	string	Name of the variable for the Zonal Current
`srcVarOceIC_velV`	vomecrty	string	Name of the variable for the Merid. Current

set_dataDownlOceBCMesh_urlName

This section defines the parameters used to construct the URL for accessing Ocean Open Boundary Condition meshmask datasets. These datasets can be sourced from either a local directory or a remote repository.

Parameter	Value	Type	Description
`urlOceBCMesh_usr`	gofs	string	Username to access the datasets from a remote FTP server
`urlOceBCMesh_pwd`	L##OOkkl!as	string	Password to access the datasets from a remote FTP server
`urlOceBCMesh_urlbase`	file:///scratch/surf/indata_offline/gulfTaranto_20141005/data/data00/ indata/ocean/oceanBC/srcFull	string	Parametric URL (ftp:/ or file:///) name for ocean input BC-meshmask datasets
`urlOceBCMesh_lont`	glamt	string	Longitude-Tgrid used to replace the substring (FIELD) on the parametric URL name
`urlOceBCMesh_lonu`	glamu	string	Longitude-Ugrid used to replace the substring (FIELD) on the parametric URL name
`urlOceBCMesh_lonv`	glamv	string	Longitude-Vgrid used to replace the substring (FIELD) on the parametric URL name
`urlOceBCMesh_lonf`	glamf	string	Longitude-Fgrid used to replace the substring (FIELD) on the parametric URL name
`urlOceBCMesh_latt`	gphit	string	Latitude-Tgrid used to replace the substring (FIELD) on the parametric URL name
`urlOceBCMesh_latu`	gphiu	string	Latitude-Ugrid used to replace the substring (FIELD) on the parametric URL name

Parameter	Value	Type	Description
`urlOceBCMesh_latv`	gphiv	string	Latitude-Vgrid used to replace the substring (FIELD) on the parametric URL name
`urlOceBCMesh_latf`	gphif	string	Latitude-Fgrid used to replace the substring (FIELD) on the parametric URL name
`urlOceBCMesh_deptht1d`	gdept_0	string	Depth-Tgrid used to replace the substring (FIELD) on the parametric URL name
`urlOceBCMesh_depthw1d`	gdepw_0	string	Depth-Wgrid used to replace the substring (FIELD) on the parametric URL name
`urlOceBCMesh_xscalfctt`	e1t	string	Scale factor-Tgrid in the x-direction
`urlOceBCMesh_xscalfctu`	e1u	string	Scale factor-Ugrid in the x-direction
`urlOceBCMesh_xscalfctv`	e1v	string	Scale factor-Vgrid in the x-direction
`urlOceBCMesh_xscalfctf`	e1f	string	Scale factor-Fgrid in the x-direction
`urlOceBCMesh_yscalfctt`	e2t	string	Scale factor-Tgrid in the y-direction
`urlOceBCMesh_yscalfctu`	e2u	string	Scale factor-Ugrid in the y-direction
`urlOceBCMesh_yscalfctv`	e2v	string	Scale factor-Vgrid in the y-direction
`urlOceBCMesh_yscalfctf`	e2f	string	Scale factor-Fgrid in the y-direction
`urlOceBCMesh_zscalfctt1d`	e3t_0	string	

Parameter	Value	Type	Description
			Scale factor-Tgrid in the z-direction (1D)
`urlOceBCMesh_zscalfctw1d`	e3w_0	string	Scale factor-Wgrid in the z-direction (1D)
`urlOceBCMesh_zscalfctu`	e3u	string	Scale factor-Ugrid in the z-direction (3D)
`urlOceBCMesh_zscalfctv`	e3v	string	Scale factor-Vgrid in the z-direction (3D)
`urlOceBCMesh_maskt`	tmask	string	LandSea-mask for the T-grid
`urlOceBCMesh_masku`	umask	string	LandSea-mask for the U-grid
`urlOceBCMesh_maskv`	vmask	string	LandSea-mask for the V-grid

set_dataDownlOceBCMesh_fileName

This section specifies the parameters required to generate the filenames of the Ocean Open Boundary Condition meshmask datasets. These filenames help identify the relevant files in the storage location.

Parameter	Value	Type	Description
fileOceBCMesh_filebase	meshmask_SYS4a3_IONIAN.nc	string	Parametric FILE name for ocean source input data. Parameters: (FIELD)
fileOceBCMesh_lcompression	False	bool	Enables if datasets you want to download are gzip compressed files (*.gz)
fileOceBCMesh_llonFlip	False	bool	(=True) if longitude coord. is in the 0 to 360 range (=True) if longitude is in -180:+180 range
fileOceBCMesh_llonCycl	True	bool	Logical variable=True if you use global data and longitude is cyclic
fileOceBCMesh_llatInv	False	bool	Enables if the dataset contains latitude decreasing through the pole
fileOceBCMesh_lont	coordinates	string	Name (for longitude-Tgrid) used to replace the substring (FIELD) on the parametric URL name
fileOceBCMesh_lonu	coordinates	string	Name (for longitude-Ugrid) used to replace the substring (FIELD) on the parametric URL name
fileOceBCMesh_lonv	coordinates	string	Name (for longitude-Vgrid) used to replace the substring (FIELD) on the parametric URL name
fileOceBCMesh_lonf	coordinates	string	Name (for longitude-Fgrid) used to replace the substring (FIELD) on the parametric URL name
fileOceBCMesh_latt	coordinates	string	Name (for latitude-Tgrid) used to replace the substring (FIELD)

Parameter	Value	Type	Description
			on the parametric URL name
`fileOceBCMesh_latu`	coordinates	string	Name (for latitude-Ugrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_latv`	coordinates	string	Name (for latitude-Vgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_latf`	coordinates	string	Name (for latitude-Fgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_deptht1d`	meshmask_vertical	string	Name (for depth-Tgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_depthw1d`	meshmask_vertical	string	Name (for depth-Wgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_xscalfctt`	coordinates	string	Name (for scaleFactor-Tgrid x-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_xscalfctu`	coordinates	string	Name (for scaleFactor-Ugrid x-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_xscalfctv`	coordinates	string	Name (for scaleFactor-Vgrid x-direction) used to replace the substring (FIELD)

Parameter	Value	Type	Description
			on the parametric URL name
`fileOceBCMesh_xscalfctf`	coordinates	string	Name (for scaleFactor-Fgrid x-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_yscalfctt`	coordinates	string	Name (for scaleFactor-Tgrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_yscalfctu`	coordinates	string	Name (for scaleFactor-Ugrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_yscalfctv`	coordinates	string	Name (for scaleFactor-Vgrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_yscalfctf`	coordinates	string	Name (for scaleFactor-Fgrid y-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_zscalfctt1d`	meshmask_vertical	string	Name (for scaleFactor-Tgrid-1d z-direction) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_zscalfctw1d`	meshmask_vertical	string	Name (for scaleFactor-Wgrid-1d z-direction) used to replace the substring (FIELD) on the parametric URL name

Parameter	Value	Type	Description
`fileOceBCMesh_zscalfctu`	meshmask_e3u	string	Name (for scaleFactor-Ugrid-3d z-direction) used to replace the substring (FIELD) on the parametric URL name (needed if obc_lvelCorr=True)
`fileOceBCMesh_zscalfctv`	meshmask_e3v	string	Name (for scaleFactor-Vgrid-3d z-direction) used to replace the substring (FIELD) on the parametric URL name (needed if obc_lvelCorr=True)
`fileOceBCMesh_maskt`	mesh_tmask	string	Name (for LandSea-mask-Tgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_masku`	mesh_umask	string	Name (for LandSea-mask-Ugrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_maskv`	mesh_vmask	string	Name (for LandSea-mask-Vgrid) used to replace the substring (FIELD) on the parametric URL name
`fileOceBCMesh_lkeepSrcFull`	True	bool	Enables if you want to keep the downloaded uncutted datasets

set_dataDownlOceBCMesh_varName

This section outlines the parameters related to the variable names used within the Ocean Open Boundary Condition meshmask datasets. These variable names are crucial for accurately extracting and utilizing the data within the files.

Parameter	Value	Type	Description
srcDimOceBCMesh_lont	x	string	Name of the dimension for the longitude
srcDimOceBCMesh_lonu	x	string	Name of the dimension for the longitude
srcDimOceBCMesh_lonv	x	string	Name of the dimension for the longitude
srcDimOceBCMesh_lonw	x	string	Name of the dimension for the longitude
srcDimOceBCMesh_latt	y	string	Name of the dimension for the latitude
srcDimOceBCMesh_latu	y	string	Name of the dimension for the latitude
srcDimOceBCMesh_latv	y	string	Name of the dimension for the latitude
srcDimOceBCMesh_latw	y	string	Name of the dimension for the latitude
srcDimOceBCMesh_deptht	z	string	Name of the dimension for the depth
srcDimOceBCMesh_depthu	z	string	Name of the dimension for the depth
srcDimOceBCMesh_depthv	z	string	Name of the dimension for the depth
srcDimOceBCMesh_depthw	z	string	Name of the dimension for the depth
srcDimOceBCMesh_time	t	string	Name of the dimension for the time
srcCrdOceBCMesh_lont	nav_lon	string	Name of the coordinate variable for the longitude coordinate
srcCrdOceBCMesh_lonu	nav_lon	string	Name of the coordinate variable for the longitude coordinate
srcCrdOceBCMesh_lonv	nav_lon	string	

Parameter	Value	Type	Description
			Name of the coordinate variable for the longitude coordinate
`srcCrdOceBCMesh_latt`	nav_lat	string	Name of the coordinate variable for the latitude coordinate
`srcCrdOceBCMesh_latu`	nav_lat	string	Name of the coordinate variable for the latitude coordinate
`srcCrdOceBCMesh_latv`	nav_lat	string	Name of the coordinate variable for the latitude coordinate
`srcCrdOceBCMesh_deptht`	nav_lev	string	Name of the coordinate variable for the depth coordinate
`srcCrdOceBCMesh_depthw`	nav_lev	string	Name of the coordinate variable for the depth coordinate
`srcCrdOceBCMesh_time`	nav_lev	string	Name of the coordinate variable for the time coordinate
`srcVarOceBCMesh_lont`	glamt	string	Name of the variable containing longitude coordinate
`srcVarOceBCMesh_lonu`	glamu	string	Name of the variable containing longitude coordinate
`srcVarOceBCMesh_lonv`	glamv	string	Name of the variable containing longitude coordinate
`srcVarOceBCMesh_lonf`	glamf	string	Name of the variable containing longitude coordinate
`srcVarOceBCMesh_latt`	gphit	string	Name of the variable containing latitude coordinate
`srcVarOceBCMesh_latu`	gphiu	string	Name of the variable containing latitude coordinate

Parameter	Value	Type	Description
srcVarOceBCMesh_latv	gphiv	string	Name of the variable containing latitude coordinate
srcVarOceBCMesh_latf	gphif	string	Name of the variable containing latitude coordinate
srcVarOceBCMesh_depht1d	gdept_0	string	Name of the variable containing depth coordinate
srcVarOceBCMesh_depthw1d	gdepw_0	string	Name of the variable containing depth coordinate
srcVarOceBCMesh_xscalfctt	e1t	string	Name of the scale factors Tgrid in x-direction
srcVarOceBCMesh_xscalfctu	e1u	string	Name of the scale factors Ugrid in x-direction
srcVarOceBCMesh_xscalfctv	e1v	string	Name of the scale factors Vgrid in x-direction
srcVarOceBCMesh_xscalfctf	e1f	string	Name of the scale factors Fgrid in x-direction
srcVarOceBCMesh_yscalfctt	e2t	string	Name of the scale factors Tgrid in y-direction
srcVarOceBCMesh_yscalfctu	e2u	string	Name of the scale factors Ugrid in y-direction
srcVarOceBCMesh_yscalfctv	e2v	string	Name of the scale factors Vgrid in y-direction
srcVarOceBCMesh_yscalfctf	e2f	string	Name of the scale factors Fgrid in y-direction
srcVarOceBCMesh_zscalfctt1d	e3t_0	string	Name of the scale factors Tgrid-1d in z-direction
srcVarOceBCMesh_zscalfctw1d	e3w_0	string	Name of the scale factors Wgrid-1d in z-direction
srcVarOceBCMesh_zscalfctu	e3u	string	Name of the scale factors Ugrid-3d in z-direction (needed if obc_lvelCorr=True)
srcVarOceBCMesh_zscalfctv	e3v	string	

Parameter	Value	Type	Description
			Name of the scale factors Vgrid-3d in z-direction (needed if obc_lvelCorr=True)
`srcVarOceBCMesh_maskt`	tmask	string	Name of the land-sea-mask on T-points
`srcVarOceBCMesh_masku`	umask	string	Name of the land-sea-mask on U-points
`srcVarOceBCMesh_maskv`	vmask	string	Name of the land-sea-mask on V-points

set_dataDownlOceBC_urlName_preSpinup

This section contains the parameters needed to build the URL for accessing pre-spinup Ocean Open Boundary Condition fields datasets. The datasets can be retrieved from either a local source or a remote repository.

Parameter	Value	Type	Description
`urlOceBCPre_usr`	gofs	string	Username to access the datasets from a remote FTP server
`urlOceBCPre_pwd`	L##OOkkl!as	string	Password to access the datasets from a remote FTP server
`urlOceBCPre_urlbase`	file:///scratch/surf/indata_offline/gulfTaranto_20141005/data/data00/ indata/ocean/oceanBC/srcFull	string	Parametric URL name for ocean input BC datasets with various parameters
`urlOceBCPre_temp`	tem	string	Name for Temperature used to replace the substring (FIELD) in the parametric URL name
`urlOceBCPre_sal`	sal	string	Name for Salinity used to replace the substring (FIELD) in the parametric URL name
`urlOceBCPre_ssh`	ssh	string	Name for Sea Surface Height used to replace the substring (FIELD) in the parametric URL name
`urlOceBCPre_velU`	cur	string	Name for Zonal Current used to replace the substring (FIELD) in the parametric URL name
`urlOceBCPre_velV`	cur	string	Name for Meridional Current used to replace the substring (FIELD) in the parametric URL name
`urlOceBCPre_tempGrid`	gridT	string	Name for Temperature grid used to replace the substring (GRID) in the parametric URL name

Parameter	Value	Type	Description
`urlOceBCPre_salGrid`	gridT	string	Name for Salinity grid used to replace the substring (GRID) in the parametric URL name
`urlOceBCPre_sshGrid`	gridT	string	Name for Sea Surface Height grid used to replace the substring (GRID) in the parametric URL name
`urlOceBCPre_velUGrid`	gridU	string	Name for Zonal Current grid used to replace the substring (GRID) in the parametric URL name
`urlOceBCPre_velVGrid`	gridV	string	Name for Meridional Current grid used to replace the substring (GRID) in the parametric URL name

set_dataDownlOceBC_fileName_preSpinup

This section provides the parameters required to create the filenames for the pre-spinup Ocean Open Boundary Condition fields datasets. These filenames are used to locate and manage the specific datasets within your data storage system.

Parameter	Value	Type	Description
`fileOceBCPre_lcompression`	False	bool	Enables if datasets you want to download are gzip compressed files (*.gz)
`fileOceBCPre_iProdDate`	1	int	File Name format to be downloaded (=1):fixProdDate, (=2):varProdDate-DayofWeek
`fileOceBCPre_dateProdFixed`	20170801	string	Datasets production date if iProdDate=1 is used in the URL/ files
`fileOceBCPre_dateProdDayofWeek`	Wednesday	string	Datasets production day if iProdDate=2 is used in the URL/ files
`fileOceBCPre_filebase`	EXP1_EAS1_1d_YYYY(i)MM(i)DD(i)_ YYYY(i+1)MM(i+1)DD(i+1)_(GRID)_IONIAN.nc	string	Parametric FILE name for ocean source input data with various parameters
`fileOceBCPre_temp`	_votemper	string	Name for Temperature used to replace the substring (FIELD) in the parametric file name
`fileOceBCPre_sal`	_vosaline	string	Name for Salinity used to replace the substring (FIELD) in the parametric file name
`fileOceBCPre_ssh`	2D	string	Name for Sea Surface Height used to replace the substring (FIELD) in the parametric file name
`fileOceBCPre_velU`	_vozocrtx	string	Name for Zonal Current used to replace the substring (FIELD) in the parametric file name

Parameter	Value	Type	Description
`fileOceBCPre_velV`	_vomecrty	string	Name for Meridional Current used to replace the substring (FIELD) in the parametric file name
`fileOceBCPre_tempGrid`	grid_T	string	Name for Temperature grid used to replace the substring (GRID) in the parametric file name
`fileOceBCPre_salGrid`	grid_T	string	Name for Salinity grid used to replace the substring (GRID) in the parametric file name
`fileOceBCPre_sshGrid`	grid_T	string	Name for Sea Surface Height grid used to replace the substring (GRID) in the parametric file name
`fileOceBCPre_velUGrid`	grid_U	string	Name for Zonal Current grid used to replace the substring (GRID) in the parametric file name
`fileOceBCPre_velVGrid`	grid_V	string	Name for Meridional Current grid used to replace the substring (GRID) in the parametric file name

set_dataDownlOceBC_varName_preSpinup

This section details the parameters for specifying the field names of the pre-spinup Ocean Open Boundary Condition datasets. The field names identify the specific variables within the datasets, ensuring proper data extraction and processing.

Parameter	Value	Type	Description
srcDimOceBCPre_lont	x	string	Name of the dimension for the longitude
srcDimOceBCPre_lonu	x	string	Name of the dimension for the longitude
srcDimOceBCPre_lonv	x	string	Name of the dimension for the longitude
srcDimOceBCPre_lonw	x	string	Name of the dimension for the longitude
srcDimOceBCPre_latt	y	string	Name of the dimension for the latitude
srcDimOceBCPre_latu	y	string	Name of the dimension for the latitude
srcDimOceBCPre_latv	y	string	Name of the dimension for the latitude
srcDimOceBCPre_latw	y	string	Name of the dimension for the latitude
srcDimOceBCPre_deptht	deptht	string	Name of the dimension for the depth
srcDimOceBCPre_depthu	depthu	string	Name of the dimension for the depth
srcDimOceBCPre_depthv	depthv	string	Name of the dimension for the depth
srcDimOceBCPre_depthw	depthw	string	Name of the dimension for the depth
srcDimOceBCPre_time	time_counter	string	Name of the dimension for the time
srcCrdOceBCPre_lont	nav_lon	string	Name of the coordinate variable for the longitude
srcCrdOceBCPre_lonu	nav_lon	string	Name of the coordinate variable for the longitude
srcCrdOceBCPre_lonv	nav_lon	string	Name of the coordinate variable for the longitude
srcCrdOceBCPre_lonw	nav_lon	string	Name of the coordinate variable for the longitude
srcCrdOceBCPre_latt	nav_lat	string	Name of the coordinate variable for the latitude
srcCrdOceBCPre_latu	nav_lat	string	Name of the coordinate variable for the latitude
srcCrdOceBCPre_latv	nav_lat	string	Name of the coordinate variable for the latitude
srcCrdOceBCPre_latw	nav_lat	string	

Parameter	Value	Type	Description
			Name of the coordinate variable for the latitude
`srcCrdOceBCPre_deptht`	deptht	string	Name of the coordinate variable for the depth
`srcCrdOceBCPre_depthu`	depthu	string	Name of the coordinate variable for the depth
`srcCrdOceBCPre_depthv`	depthv	string	Name of the coordinate variable for the depth
`srcCrdOceBCPre_depthw`	depthw	string	Name of the coordinate variable for the depth
`srcCrdOceBCPre_time`	time_counter	string	Name of the coordinate variable for the time
`srcVarOceBCPre_temp`	votemper	string	Name of the variable for the Temperature
`srcVarOceBCPre_sal`	vosaline	string	Name of the variable for the Salinity
`srcVarOceBCPre_ssh`	sossheig	string	Name of the variable for the Sea Surface Height
`srcVarOceBCPre_velU`	vozocrtx	string	Name of the variable for the Zonal Current
`srcVarOceBCPre_velV`	vomecrty	string	Name of the variable for the Merid Current

set_dataDownlOceBC_urlName_postSpinup

This section contains the parameters needed to build the URL for accessing post-spinup Ocean Open Boundary Condition fields datasets. The datasets can be retrieved from either a local source or a remote repository.

Parameter	Value	Type	Description
`urlOceBCPost_usr`	gofs	string	Username to access the datasets from a remote FTP server
`urlOceBCPost_pwd`	L##OOkkl!as	string	Password to access the datasets from a remote FTP server
`urlOceBCPost_urlbase`	file:///scratch/surf/indata_offline/gulfTaranto_20141005/data/data00/ indata/ocean/oceanBC/srcFull	string	Parametric URL name for ocean input BC datasets with parameters for specific dates
`urlOceBCPost_temp`	tem	string	Name for Temperature used to replace the substring (FIELD) on the parametric URL name
`urlOceBCPost_sal`	sal	string	Name for Salinity used to replace the substring (FIELD) on the parametric URL name
`urlOceBCPost_ssh`	ssh	string	Name for Sea Surface Height used to replace the substring (FIELD) on the parametric URL name
`urlOceBCPost_velU`	cur	string	Name for Zonal Current used to replace the substring (FIELD) on the parametric URL name
`urlOceBCPost_velV`	cur	string	Name for Merid. Current used to replace the substring (FIELD) on the parametric URL name
`urlOceBCPost_tempGrid`	gridT	string	Name for Temperature used to replace the substring (GRID) on the parametric URL name

Parameter	Value	Type	Description
`urlOceBCPost_salGrid`	gridT	string	Name for Salinity used to replace the substring (GRID) on the parametric URL name
`urlOceBCPost_sshGrid`	gridT	string	Name for Sea Surface Height used to replace the substring (GRID) on the parametric URL name
`urlOceBCPost_velUGrid`	gridU	string	Name for Zonal Current used to replace the substring (GRID) on the parametric URL name
`urlOceBCPost_velVGrid`	gridV	string	Name for Merid. Current used to replace the substring (GRID) on the parametric URL name

set_dataDownlOceBC_fileName_postSpinup

This section provides the parameters required to create the filenames for the post-spinup Ocean Open Boundary Condition fields datasets. These filenames are used to locate and manage the specific datasets within your data storage system.

Parameter	Value	Type	Description
fileOceBCPost_lcompression	False	bool	Enables if datasets you want to download are gzip compressed files (*.gz)
fileOceBCPost_iProdDate	1	int	File Name format to be downloaded (=1):fixProdDate, (=2):varProdDate-DayofWeek
fileOceBCPost_dateProdFixed	20170801	string	Datasets production if iProdDate=1 is used in the URL/ files
fileOceBCPost_dateProdDayofWeek	Wednesday	string	Datasets production if iProdDate=2 is used in the URL/ files
fileOceBCPost_filebase	EXP1_EAS1_1d_YYYY(i)MM(i)DD(i)_ YYYY(i+1)MM(i+1)DD(i+1)_(GRID)_IONIAN.nc	string	Parametric FILE name for ocean source input data
fileOceBCPost_temp	_votemper	string	Name (for Temperature) used to replace the substring (FIELD) on the parametric file name
fileOceBCPost_sal	_vosaline	string	Name (for Salinity) used to replace the substring (FIELD) on the parametric file name
fileOceBCPost_ssh	2D	string	Name (for Sea Surface Height) used to replace the substring (FIELD) on the parametric file name
fileOceBCPost_velU	_vozocrtx	string	Name (for Merid. Current) used to replace the substring (FIELD) on the parametric file name
fileOceBCPost_velV	_vomecrty	string	Name (for Zonal Current) used to

Parameter	Value	Type	Description
			replace the substring (FIELD) on the parametric file name
`fileOceBCPost_tempGrid`	grid_T	string	Name (for Temperature) used to replace the substring (GRID) on the parametric file name
`fileOceBCPost_salGrid`	grid_T	string	Name (for Salinity) used to replace the substring (GRID) on the parametric file name
`fileOceBCPost_sshGrid`	grid_T	string	Name (for Sea Surface Height) used to replace the substring (GRID) on the parametric file name
`fileOceBCPost_velUGrid`	grid_U	string	Name (for Zonal Current) used to replace the substring (GRID) on the parametric file name
`fileOceBCPost_velVGrid`	grid_V	string	Name (for Merid. Current) used to replace the substring (GRID) on the parametric file name

set_dataDownlOceBC_varName_postSpinup

This section details the parameters for specifying the field names of the post-spinup Ocean Open Boundary Condition datasets. The field names identify the specific variables within the datasets, ensuring proper data extraction and processing.

Parameter	Value	Type	Description
srcDimOceBCPost_lont	x	string	Name of the dimension for the longitude
srcDimOceBCPost_lonu	x	string	Name of the dimension for the longitude
srcDimOceBCPost_lonv	x	string	Name of the dimension for the longitude
srcDimOceBCPost_lonw	x	string	Name of the dimension for the longitude
srcDimOceBCPost_latt	y	string	Name of the dimension for the latitude
srcDimOceBCPost_latu	y	string	Name of the dimension for the latitude
srcDimOceBCPost_latv	y	string	Name of the dimension for the latitude
srcDimOceBCPost_latw	y	string	Name of the dimension for the latitude
srcDimOceBCPost_deptht	deptht	string	Name of the dimension for the depth
srcDimOceBCPost_depthu	depthu	string	Name of the dimension for the depth
srcDimOceBCPost_depthv	depthv	string	Name of the dimension for the depth
srcDimOceBCPost_depthw	depthw	string	Name of the dimension for the depth
srcDimOceBCPost_time	time_counter	string	Name of the dimension for the time
srcCrdOceBCPost_lont	nav_lon	string	Name of the coordinate variable for the longitude
srcCrdOceBCPost_lonu	nav_lon	string	Name of the coordinate variable for the longitude
srcCrdOceBCPost_lonv	nav_lon	string	Name of the coordinate variable for the longitude

Parameter	Value	Type	Description
srcCrdOceBCPost_lonw	nav_lon	string	Name of the coordinate variable for the longitude
srcCrdOceBCPost_latt	nav_lat	string	Name of the coordinate variable for the latitude
srcCrdOceBCPost_latu	nav_lat	string	Name of the coordinate variable for the latitude
srcCrdOceBCPost_latv	nav_lat	string	Name of the coordinate variable for the latitude
srcCrdOceBCPost_latw	nav_lat	string	Name of the coordinate variable for the latitude
srcCrdOceBCPost_deptht	deptht	string	Name of the coordinate variable for the latitude
srcCrdOceBCPost_depthu	depthu	string	Name of the coordinate variable for the latitude
srcCrdOceBCPost_depthv	depthv	string	Name of the coordinate variable for the latitude
srcCrdOceBCPost_depthw	depthw	string	Name of the coordinate variable for the depth
srcCrdOceBCPost_time	time_counter	string	Name of the coordinate variable for the time
srcVarOceBCPost_temp	votemper	string	Name of the variable for the Temperature
srcVarOceBCPost_sal	vosaline	string	Name of the variable for the Salinity
srcVarOceBCPost_ssh	sossheig	string	Name of the variable for the Sea Surface Height
srcVarOceBCPost_velU	vozocrtx	string	Name of the variable for the Zonal Current
srcVarOceBCPost_velV	vomecrty	string	Name of the variable for the Merid Current

5.1.8 **Data Manipulation Settings**

set_manipolBat

This section defines the parameters for modifying and controlling bathymetry values within the simulation domain.

Parameter	Value	Type	Description
manipBat_seaLevel	0.0	float	Value to be added to bathymetry to modify the surface elevation (e.g., Caspian Sea 28 meters)
manipBat_minDepth	5.0	float	Minimum bathymetry value of the whole domain (at sea grid points)
manipBat_maxDepth	22000.0	float	Maximum bathymetry value of the whole domain (at sea grid points)
manipBat_lmoveDepth	True	bool	Enables the modification of bathymetry (to a given value >= 0) inside rectangular regions
manipBatMoveDepth_lonWest	18.0	float	Longitude of the western boundary of rectangular regions where you want to move the depth
manipBatMoveDepth_lonEast	18.375	float	Longitude of the eastern boundary of rectangular regions where you want to move the depth
manipBatMoveDepth_latSout	40.3	float	Latitude of the southern boundary of rectangular regions where you want to move the depth
manipBatMoveDepth_latNort	40.5625	float	Latitude of the northern boundary of rectangular regions where you want to move the depth
manipBatMoveDepth_depth	0.0	float	Depth of rectangular regions where you want to move the depth (if = 0.0,

Parameter	Value	Type	Description
			you add land points)

set_shapFiltBat

This section defines the parameters for applying a Shapiro filter to the bathymetry data.

Parameter	Value	Type	Description
`norder_shapFiltBat`	2	int	Order of the Shapiro Filter for Bathy data
`nloop_shapFiltBat`	4	int	Number of applications of the Shapiro Filter for Bathy data

set_shapFiltOce

This section defines the parameters for applying a Shapiro filter to the output ocean data.

Parameter	Value	Type	Description
`norder_shapFiltOce`	2	int	Order of the Shapiro Filter for outOce data
`nloop_shapFiltOce`	10	int	Number of applications of the Shapiro Filter for outOce data

5.1.9 Interpolation Method Settings

set_interpMethodBat

This section specifies the interpolation method to be used when processing bathymetric elevation data.

Parameter	Value	Type	Description
`interpMethodBat_elev`	bilin	string	Remapping option: bilin, bicub, distwgt

set_interpMethodAtm

This section defines the interpolation methods used for remapping the atmospheric data fields required by the simulation.

Parameter	Value	Type	Description
interpMethodAtm_velU	bilin	string	Remapping (sbc_iformulat=0,2) option: bilin, bicub, distwgt
interpMethodAtm_velV	bilin	string	Remapping (sbc_iformulat=0,2) option: bilin, bicub, distwgt
interpMethodAtm_temp	bilin	string	Remapping (sbc_iformulat=0,2) option: bilin, bicub, distwgt
interpMethodAtm_dpTemp	bilin	string	Remapping (sbc_iformulat=0) option: bilin, bicub, distwgt
interpMethodAtm_mslp	bilin	string	Remapping (sbc_iformulat=0 or/and sbc_aprdyn) option: bilin, bicub, distwgt
interpMethodAtm_cloudCover	bilin	string	Remapping (sbc_iformulat=0) option: bilin, bicub, distwgt
interpMethodAtm_prec	bilin	string	Remapping (sbc_iformulat=0,2) option: bilin, bicub, distwgt
interpMethodAtm_tauU	bilin	string	Remapping (sbc_iformulat=1) option: bilin, bicub, distwgt
interpMethodAtm_tauV	bilin	string	Remapping (sbc_iformulat=1) option: bilin, bicub, distwgt
interpMethodAtm_qtot	bilin	string	Remapping (sbc_iformulat=1) option: bilin, bicub, distwgt
interpMethodAtm_qsr	bilin	string	Remapping (sbc_iformulat=1) option: bilin, bicub, distwgt
interpMethodAtm_emp	bilin	string	Remapping (sbc_iformulat=1) option: bilin, bicub, distwgt

Parameter	Value	Type	Description
interpMethodAtm_tempS	bilin	string	Remapping (sbc_iformulat=1) option: bilin, bicub, distwgt
interpMethodAtm_salS	bilin	string	Remapping (sbc_iformulat=1) option: bilin, bicub, distwgt
interpMethodAtm_umid	bilin	string	Remapping (sbc_iformulat=2) option: bilin, bicub, distwgt
interpMethodAtm_radLW	bilin	string	Remapping (sbc_iformulat=2) option: bilin, bicub, distwgt
interpMethodAtm_radSW	bilin	string	Remapping (sbc_iformulat=2) option: bilin, bicub, distwgt
interpMethodAtm_snow	bilin	string	Remapping (sbc_iformulat=2) option: bilin, bicub, distwgt

set_interpMethodOceIC

This section defines the interpolation methods used for remapping the ocean initial condition data fields necessary for the simulation.

Parameter	Value	Type	Description
interpMethodOceIC_temp	bilin	string	Remapping option: bilin, bicub, distwgt
interpMethodOceIC_sal	bilin	string	Remapping option: bilin, bicub, distwgt
interpMethodOceIC_ssh	bilin	string	Remapping option: bilin, bicub, distwgt
interpMethodOceIC_velU	bilin	string	Remapping option: bilin, bicub, distwgt
interpMethodOceIC_velV	bilin	string	Remapping option: bilin, bicub, distwgt

set_interpMethodOceBC

This section defines the interpolation methods used for remapping the ocean boundary condition data fields required by the simulation.

Parameter	Value	Type	Description
`interpMethodOceBC_temp`	bilin	string	Remapping option: bilin, bicub, distwgt
`interpMethodOceBC_sal`	bilin	string	Remapping option: bilin, bicub, distwgt
`interpMethodOceBC_ssh`	bilin	string	Remapping option: bilin, bicub, distwgt
`interpMethodOceBC_velU`	bilin	string	Remapping option: bilin, bicub, distwgt
`interpMethodOceBC_velV`	bilin	string	Remapping option: bilin, bicub, distwgt

5.1.10 **Output Management Settings**

set_IvarOceOutT

This section defines the settings for controlling which oceanographic fields (on the T-grid) are written to the output files at the end of the simulation.

Parameter	Value	Type	Description
`lvarOceOutT_temp`	True	bool	Enables if you want to write Temperature (votemper) field in the output file
`lvarOceOutT_sal`	True	bool	Enables if you want to write Salinity (vosaline) field in the output file
`lvarOceOutT_tempS`	False	bool	Enables if you want to write Surface temperature (sosstsst) field in the output file
`lvarOceOutT_salS`	False	bool	Enables if you want to write Surface Salinity (sosaline) field in the output file
`lvarOceOutT_ssh`	True	bool	Enables if you want to write Sea Surface Height (sossheig) field in the output file
`lvarOceOutT_fluxWater_up`	False	bool	Enables if you want to write Net Upward Water Flux (sowaflup) field in the output file
`lvarOceOutT_fluxSalt_down`	False	bool	Enables if you want to write downward salt flux (sosfldow) field in the output file
`lvarOceOutT_fluxHeat_down`	False	bool	Enables if you want to write Net Downward Heat Flux (sohefldo) field in the output file
`lvarOceOutT_fluxHeatS_damp`	False	bool	Enables if you want to write Surface Heat Flux: Damping (sohefldp) field in the output file
`lvarOceOutT_fluxWaterS_damp`	False	bool	Enables if you want to write

Parameter	Value	Type	Description
			Surface Water Flux: Damping (sowafldp) field in the output file
`lvarOceOutT_fluxSaltS_damp`	False	bool	Enables if you want to write Surface salt flux: damping (sosafldp) field in the output file
`lvarOceOutT_runoffs`	False	bool	Enables if you want to write River runoffs (sorunoff) field in the output file
`lvarOceOutT_tempConcen`	False	bool	Enables if you want to write Concentration/ Dilution term on temperature (sosst_cd) field in the output file
`lvarOceOutT_salConcen`	False	bool	Enables if you want to write Concentration/ Dilution term on salinity (sosss_cd) field in the output file
`lvarOceOutT_radSW`	False	bool	Enables if you want to write Shortwave Radiation (soshfldo) field in the output file
`lvarOceOutT_depthTurb`	False	bool	Enables if you want to write Turbocline Depth (somixhgt) field in the output file
`lvarOceOutT_mld`	True	bool	Enables if you want to write Mixed Layer Depth 0.01 (somxl010) field in the output file
`lvarOceOutT_iceFrac`	False	bool	Enables if you want to write Ice fraction (soicecov) field in the output file
`lvarOceOutT_wind10`	False	bool	

Parameter	Value	Type	Description
			Enables if you want to write wind speed at 10m (sowindsp) field in the output file
`lvarOceOutT_bowlin`	False	bool	Enables if you want to write Bowl Index (sobowlin) field in the output file

set_lvarOceOutU

This section defines the settings for controlling which oceanographic fields (on the U-grid) are written to the output files at the end of the simulation.

Parameter	Value	Type	Description
`lvarOceOutU_velU`	True	bool	Enables if you want to write Zonal Current (vozocrtx) field in the output file
`lvarOceOutU_tauU`	True	bool	Enables if you want to write Zonal Wind Stress (sozotaux) field in the output file

set_lvarOceOutV

This section defines the settings for controlling which oceanographic fields (on the V-grid) are written to the output files at the end of the simulation.

Parameter	Value	Type	Description
`lvarOceOutV_velV`	True	bool	Enables if you want to write Meridional Current (vomecrty) field in the output file
`lvarOceOutV_tauV`	True	bool	Enables if you want to write Meridional Wind Stress (sometauy) field in the output file

set_lvarOceOutW

This section defines the settings for controlling which oceanographic fields (on the W-grid) are written to the output files at the end of the simulation.

Parameter	Value	Type	Description
`lvarOceOutW_velW`	False	bool	Enables if you want to write Vertical Velocity (vovecrtz) field in the output file
`lvarOceOutW_eddyDiffW`	False	bool	Enables if you want to write Vertical Eddy Diffusivity (votkeavt) field in the output file
`lvarOceOutW_eddyViscW`	False	bool	Enables if you want to write Vertical Eddy Viscosity (votkeavm) field in the output file

5.2 User Configuration File - Postprocessing Sections

In this chapter, we continue our exploration of the user configuration file `setParFree.json`, focusing specifically on section B, which manages the tools used for visualizing and analyzing model results. This section provides detailed options for the visualization of input and output datasets, comparison of child and parent model fields, and validation of simulation results with in situ or satellite observations. These tools are essential for ensuring the accuracy and reliability of the model results.

5.2.1 **Plot Selection Parameters**

set_lplot_post

This section defines the settings used to control the generation of plots during the post-processing phase of the simulation.

Parameter	Value	Type	Description
`lrun_visIndata`	True	bool	enable/disable the plotting of the Indata Bat, Atm, OceIC, OceBC fields
`lrun_visExtrapdata`	True	bool	enable/disable the plotting of the Extrapdata Atm, OceIC, OceBC fields
`lrun_visRegriddata`	True	bool	enable/disable the plotting of the Regriddata Bat, Atm, OceIC, OceBC, OceBCbdy fields
`lrun_visOutdata`	True	bool	enable/disable the plotting of the Outdata Ocean fields
`lrun_chlVSpar`	True	bool	enable/disable the plotting of the child VS. parent Ocean fields
`lrun_surfVSctd`	False	bool	enable/disable the plotting of the surf VS. ctd Ocean fields
`lrun_surfVSmooring`	False	bool	enable/disable the plotting of the surf VS. mooring Ocean fields
`lrun_surfVSferrybox`	False	bool	enable/disable the plotting of the surf VS. ferrybox Ocean fields
`lrun_surfVSsat`	False	bool	enable/disable the plotting of the surf VS. satellite Ocean fields
`lrun_regridOutUV`	False	bool	enables the execution of the output-UV_fields REMAPPING (from UV GRID to T-GRID)
`lrun_regridOutUVWeghts`	False	bool	enables the computation/copy of WHEGHT-FILEs for output-UV_fields REMAPPING (if lrun_regridOutUV=False)

set_visual_lplot

This section contains settings for visualizing various components of the model.

Parameter	Value	Type	Description
`lplotDom`	True	bool	enable/disable the plotting of the Nest Domains
`lplotBat`	True	bool	enable/disable the plotting of the Bathymetry fields
`lplotAtm`	True	bool	enable/disable the plotting of the Atmospheric fields
`lplotOceIC`	True	bool	enable/disable the plotting of the Initial Condition Ocean fields
`lplotOceBC`	True	bool	enable/disable the plotting of the Open Boundary Condition Ocean fields
`lplotOceBCbdy`	True	bool	enable/disable the plotting of the Open Boundary Condition Ocean fields

5.2.2 Bathymetry Visualization Settings

set_visual_lplotBat_mesh

This section defines the settings for visualizing the bathymetry mesh.

Parameter	Value	Type	Description
`lplotBat_mesh_n`	True	bool	Enable/disable the plotting of the Bathymetry Mesh
`lplotBat_mesh_nMultiP`	True	bool	Enable/disable the plotting of the Bathymetry Mesh

set_visual_lplotBat_mask

This section defines the settings for visualizing the bathymetry masks.

Parameter	Value	Type	Description
`lplotBat_mask_n`	True	bool	Enable/disable the plotting of the Bathymetry Mask
`lplotBat_mask_nMultiP`	True	bool	Enable/disable the plotting of the Bathymetry Mask

set_visual_lplotBat_maxDepth

This section defines the settings for visualizing the bathymetry.

Parameter	Value	Type	Description
`lplotBat_maxDepthxy_n`	True	bool	Enable/disable the plotting of the Bathymetry
`lplotBat_maxDepthxy_nMultiP`	True	bool	Enable/disable the plotting of the Bathymetry
`lplotBat_maxDepthx_n`	True	bool	Enable/disable the plotting of the Bathymetry
`lplotBat_maxDepthx_nMultiP`	True	bool	Enable/disable the plotting of the Bathymetry
`lplotBat_maxDepthy_n`	True	bool	Enable/disable the plotting of the Bathymetry
`lplotBat_maxDepthy_nMultiP`	True	bool	Enable/disable the plotting of the Bathymetry

5.2.3 Atmospheric Data Visualization Settings

set_visual_lplotAtm_mesh

This section contains settings for visualizing the atmospheric mesh.

Parameter	Value	Type	Description
`lplotAtm_mesh_n`	True	bool	Enable/disable the plotting of the Atmospheric Mesh
`lplotAtm_mesh_nMultiP`	True	bool	Enable/disable the plotting of the Atmospheric Mesh

set_visual_lplotAtm_mask

This section contains settings for visualizing the atmospheric masks.

Parameter	Value	Type	Description
`lplotAtm_mask_n`	True	bool	Enable/disable the plotting of the Atmospheric Mask
`lplotAtm_mask_nMultiP`	True	bool	Enable/disable the plotting of the Atmospheric Mask

set_visual_lplotAtm_velU

This section contains settings for visualizing the zonal wind (U component).

Parameter	Value	Type	Description
`lplotAtm_velU_nt`	True	bool	Enable/disable the plotting of the Zonal Wind
`lplotAtm_velU_tMultiP_n`	True	bool	Enable/disable the plotting of the Zonal Wind
`lplotAtm_velU_nMultiP_t`	True	bool	Enable/disable the plotting of the Zonal Wind

set_visual_lplotAtm_velV

This section contains settings for visualizing the meridional wind (V component).

Parameter	Value	Type	Description
`lplotAtm_velV_nt`	True	bool	Enable/disable the plotting of the Meridional Wind
`lplotAtm_velV_tMultiP_n`	True	bool	Enable/disable the plotting of the Meridional Wind
`lplotAtm_velV_nMultiP_t`	True	bool	Enable/disable the plotting of the Meridional Wind

set_visual_lplotAtm_vel

This section contains settings for visualizing the wind.

Parameter	Value	Type	Description
`lplotAtm_vel_nt`	True	bool	Enable/disable the plotting of the Wind
`lplotAtm_vel_tMultiP_n`	True	bool	Enable/disable the plotting of the Wind
`lplotAtm_vel_nMultiP_t`	True	bool	Enable/disable the plotting of the Wind

set_visual_lplotAtm_temp

This section contains settings for visualizing the air temperature.

Parameter	Value	Type	Description
`lplotAtm_temp_nt`	True	bool	Enable/disable the plotting of the Air Temperature
`lplotAtm_temp_tMultiP_n`	True	bool	Enable/disable the plotting of the Air Temperature
`lplotAtm_temp_nMultiP_t`	True	bool	Enable/disable the plotting of the Air Temperature

set_visual_lplotAtm_dpTemp

This section contains settings for visualizing the dew point temperature.

Parameter	Value	Type	Description
`lplotAtm_dpTemp_nt`	True	bool	Enable/disable the plotting of the Dew Point Temperature
`lplotAtm_dpTemp_tMultiP_n`	True	bool	Enable/disable the plotting of the Dew Point Temperature
`lplotAtm_dpTemp_nMultiP_t`	True	bool	Enable/disable the plotting of the Dew Point Temperature

set_visual_lplotAtm_mslp

This section contains settings for visualizing the sea-level pressure.

Parameter	Value	Type	Description
`lplotAtm_mslp_nt`	True	bool	Enable/disable the plotting of the Sea Level Pressure
`lplotAtm_mslp_tMultiP_n`	True	bool	Enable/disable the plotting of the Sea Level Pressure
`lplotAtm_mslp_nMultiP_t`	True	bool	Enable/disable the plotting of the Sea Level Pressure

set_visual_lplotAtm_cloudCov

This section contains settings for visualizing the cloud cover.

Parameter	Value	Type	Description
`lplotAtm_cloudCov_nt`	True	bool	Enable/disable the plotting of the Cloud Cover
`lplotAtm_cloudCov_tMultiP_n`	True	bool	Enable/disable the plotting of the Cloud Cover
`lplotAtm_cloudCov_nMultiP_t`	True	bool	Enable/disable the plotting of the Cloud Cover

set_visual_lplotAtm_prec

This section contains settings for visualizing the precipitation.

Parameter	Value	Type	Description
`lplotAtm_prec_nt`	True	bool	Enable/disable the plotting of the Dew Point Temperature
`lplotAtm_prec_tMultiP_n`	True	bool	Enable/disable the plotting of the Dew Point Temperature
`lplotAtm_prec_nMultiP_t`	True	bool	Enable/disable the plotting of the Dew Point Temperature

set_visual_lplotAtm_umid

This section contains settings for visualizing the air humidity.

Parameter	Value	Type	Description
`lplotAtm_umid_nt`	True	bool	Enable/disable the plotting of the Umidity
`lplotAtm_umid_tMultiP_n`	True	bool	Enable/disable the plotting of the Umidity
`lplotAtm_umid_nMultiP_t`	True	bool	Enable/disable the plotting of the Umidity

set_visual_lplotAtm_radLW

This section contains settings for visualizing the long wave radiative flux.

Parameter	Value	Type	Description
`lplotAtm_radLW_nt`	True	bool	Enable/disable the plotting of the Long Wave Radiative Flux
`lplotAtm_radLW_tMultiP_n`	True	bool	Enable/disable the plotting of the Long Wave Radiative Flux
`lplotAtm_radLW_nMultiP_t`	True	bool	Enable/disable the plotting of the Long Wave Radiative Flux

set_visual_lplotAtm_radSW

This section contains settings for visualizing the short wave radiative flux.

Parameter	Value	Type	Description
`lplotAtm_radSW_nt`	True	bool	Enable/disable the plotting of the Short Wave Radiative Flux
`lplotAtm_radSW_tMultiP_n`	True	bool	Enable/disable the plotting of the Short Wave Radiative Flux
`lplotAtm_radSW_nMultiP_t`	True	bool	Enable/disable the plotting of the Short Wave Radiative Flux

set_visual_lplotAtm_snow

This section contains settings for visualizing the solid precipitation.

Parameter	Value	Type	Description
`lplotAtm_snow_nt`	True	bool	Enable/disable the plotting of the Solid Precipitation
`lplotAtm_snow_tMultiP_n`	True	bool	Enable/disable the plotting of the Solid Precipitation
`lplotAtm_snow_nMultiP_t`	True	bool	Enable/disable the plotting of the Solid Precipitation

set_visual_lplotAtm_tauU

This section contains settings for visualizing the zonal wind stress.

Parameter	Value	Type	Description
`lplotAtm_tauU_nt`	True	bool	Enable/disable the plotting of the Zonal Wind stress
`lplotAtm_tauU_tMultiP_n`	True	bool	Enable/disable the plotting of the Zonal Wind stress
`lplotAtm_tauU_nMultiP_t`	True	bool	Enable/disable the plotting of the Zonal Wind stress

set_visual_lplotAtm_tauV

This section contains settings for visualizing the meridional wind stress.

Parameter	Value	Type	Description
`lplotAtm_tauV_nt`	True	bool	Enable/disable the plotting of the Meridional Wind stress
`lplotAtm_tauV_tMultiP_n`	True	bool	Enable/disable the plotting of the Meridional Wind stress
`lplotAtm_tauV_nMultiP_t`	True	bool	Enable/disable the plotting of the Meridional Wind stress

set_visual_lplotAtm_tau

This section contains settings for visualizing the wind stress.

Parameter	Value	Type	Description
`lplotAtm_tau_nt`	True	bool	Enable/disable the plotting of the Wind stress
`lplotAtm_tau_tMultiP_n`	True	bool	Enable/disable the plotting of the Wind stress
`lplotAtm_tau_nMultiP_t`	True	bool	Enable/disable the plotting of the Wind stress

set_visual_lplotAtm_qtot

This section contains settings for visualizing the total heat flux.

Parameter	Value	Type	Description
`lplotAtm_qtot_nt`	True	bool	Enable/disable the plotting of the Total Heat Flux
`lplotAtm_qtot_tMultiP_n`	True	bool	Enable/disable the plotting of the Total Heat Flux
`lplotAtm_qtot_nMultiP_t`	True	bool	Enable/disable the plotting of the Total Heat Flux

set_visual_lplotAtm_qsr

This section contains settings for visualizing the solar radiation penetration.

Parameter	Value	Type	Description
`lplotAtm_qsr_nt`	True	bool	Enable/disable the plotting of the Solar Radiation Penetration
`lplotAtm_qsr_tMultiP_n`	True	bool	Enable/disable the plotting of the Solar Radiation Penetration
`lplotAtm_qsr_nMultiP_t`	True	bool	Enable/disable the plotting of the Solar Radiation Penetration

set_visual_lplotAtm_emp

This section contains settings for visualizing the mass flux exchanged with the atmosphere.

Parameter	Value	Type	Description
`lplotAtm_emp_nt`	True	bool	Enable/disable the plotting of the Mass flux exchanged with the atmosphere
`lplotAtm_emp_tMultiP_n`	True	bool	Enable/disable the plotting of the Mass flux exchanged with the atmosphere
`lplotAtm_emp_nMultiP_t`	True	bool	Enable/disable the plotting of the Mass flux exchanged with the atmosphere

set_visual_lplotAtm_tempS

This section contains settings for visualizing the surface temperature.

Parameter	Value	Type	Description
`lplotAtm_tempS_nt`	True	bool	Enable/disable the plotting of the Surface Temperature
`lplotAtm_tempS_tMultiP_n`	True	bool	Enable/disable the plotting of the Surface Temperature
`lplotAtm_tempS_nMultiP_t`	True	bool	Enable/disable the plotting of the Surface Temperature

set_visual_lplotAtm_salS

This section contains settings for visualizing the surface salinity.

Parameter	Value	Type	Description
`lplotAtm_salS_nt`	True	bool	Enable/disable the plotting of the Surface Salinity
`lplotAtm_salS_tMultiP_n`	True	bool	Enable/disable the plotting of the Surface Salinity
`lplotAtm_salS_nMultiP_t`	True	bool	Enable/disable the plotting of the Surface Salinity

5.2.4 Ocean Data Visualization Options

set_visual_lplotOce_mesh

This section contains settings for visualizing the ocean mesh.

Parameter	Value	Type	Description
`lplotOce_meshTxy_n`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshTxy_nMultiP`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshUxy_n`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshUxy_nMultiP`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshVxy_n`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshVxy_nMultiP`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshTz_n`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshTz_nMultiP`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshWz_n`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshWz_nMultiP`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshTUVxy_n`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshTUVxy_nMultiP`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshTWz_n`	True	bool	Enable/disable the plotting of the Ocean Mesh
`lplotOce_meshTWz_nMultiP`	True	bool	Enable/disable the plotting of the Ocean Mesh

set_visual_lplotOce_mask

This section contains settings for visualizing the ocean mask.

Parameter	Value	Type	Description
`lplotOce_maskTxy_nz`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTxy_nMultiP_z`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTxz_ny`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTxz_nMultiP_y`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTyz_nx`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTyz_nMultiP_x`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTUVxy_n`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTUVxy_nMultiP`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTWxz_n`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTWxz_nMultiP`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTWyz_n`	True	bool	Enable/disable the plotting of the Ocean Mask
`lplotOce_maskTWyz_nMultiP`	True	bool	Enable/disable the plotting of the Ocean Mask

set_visual_lplotOce_temp

This section contains settings for visualizing the ocean temperature.

Parameter	Value	Type	Description
`lplotOce_tempxy_nzt`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempxy_zMultiP_nt`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempxy_tMultiP_nz`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempxy_nMultiP_zt`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempxz_nyt`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempxz_yMultiP_nt`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempxz_tMultiP_ny`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempxz_nMultiP_yt`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempyz_nxt`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempyz_xMultiP_nt`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempyz_tMultiP_nx`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempyz_nMultiP_xt`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempz_nxyt`	True	bool	Enable/disable the plotting of Temperature
`lplotOce_tempz_nMultiP_xyt`	True	bool	Enable/disable the plotting of Temperature

set_visual_lplotOce_sal

This section contains settings for visualizing the ocean salinity.

Parameter	Value	Type	Description
`lplotOce_salxy_nzt`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salxy_zMultiP_nt`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salxy_tMultiP_nz`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salxy_nMultiP_zt`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salxz_nyt`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salxz_yMultiP_nt`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salxz_tMultiP_ny`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salxz_nMultiP_yt`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salyz_nxt`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salyz_xMultiP_nt`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salyz_tMultiP_nx`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salyz_nMultiP_xt`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salz_nxyt`	True	bool	Enable/disable the plotting of Salinity
`lplotOce_salz_nMultiP_xyt`	True	bool	Enable/disable the plotting of Salinity

set_visual_lplotOce_dens

This section contains settings for visualizing the ocean density.

Parameter	Value	Type	Description
lplotOce_densxy_nzt	True	bool	Enable/disable the plotting of the Density
lplotOce_densxy_zMultiP_nt	True	bool	Enable/disable the plotting of the Density
lplotOce_densxy_tMultiP_nz	True	bool	Enable/disable the plotting of the Density
lplotOce_densxy_nMultiP_zt	True	bool	Enable/disable the plotting of the Density
lplotOce_densxz_nyt	True	bool	Enable/disable the plotting of the Density
lplotOce_densxz_yMultiP_nt	True	bool	Enable/disable the plotting of the Density
lplotOce_densxz_tMultiP_ny	True	bool	Enable/disable the plotting of the Density
lplotOce_densxz_nMultiP_yt	True	bool	Enable/disable the plotting of the Density
lplotOce_densyz_nxt	True	bool	Enable/disable the plotting of the Density
lplotOce_densyz_xMultiP_nt	True	bool	Enable/disable the plotting of the Density
lplotOce_densyz_tMultiP_nx	True	bool	Enable/disable the plotting of the Density
lplotOce_densyz_nMultiP_xt	True	bool	Enable/disable the plotting of the Density
lplotOce_densz_nxyt	True	bool	Enable/disable the plotting of the Density
lplotOce_densz_nMultiP_xyt	True	bool	Enable/disable the plotting of the Density

set_visual_lplotOce_velU

This section contains settings for visualizing the ocean zonal velocity (U-component).

Parameter	Value	Type	Description
lplotOce_velUxy_nzt	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUxy_zMultiP_nt	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUxy_tMultiP_nz	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUxy_nMultiP_zt	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUxz_nyt	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUxz_yMultiP_nt	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUxz_tMultiP_ny	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUxz_nMultiP_yt	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUyz_nxt	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUyz_xMultiP_nt	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUyz_tMultiP_nx	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUyz_nMultiP_xt	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUz_nxyt	True	bool	Enable/disable the plotting of the Zonal Velocity
lplotOce_velUz_nMultiP_xyt	True	bool	Enable/disable the plotting of the Zonal Velocity

set_visual_lplotOce_velV

This section contains settings for visualizing the ocean meridional velocity (V-component).

Parameter	Value	Type	Description
`lplotOce_velVxy_nzt`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVxy_zMultiP_nt`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVxy_tMultiP_nz`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVxy_nMultiP_zt`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVxz_nyt`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVxz_yMultiP_nt`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVxz_tMultiP_ny`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVxz_nMultiP_yt`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVyz_nxt`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVyz_xMultiP_nt`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVyz_tMultiP_nx`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVyz_nMultiP_xt`	True	bool	Enable/disable the plotting of the Meridional Velocity
`lplotOce_velVz_nxyt`	True	bool	Enable/disable the plotting of the

Parameter	Value	Type	Description
			Meridional Velocity
`lplotOce_velVz_nMultiP_xyt`	True	bool	Enable/disable the plotting of the Meridional Velocity

set_visual_lplotOce_velW

This section contains settings for visualizing the ocean vertical velocity (W-component).

Parameter	Value	Type	Description
`lplotOce_velWxy_nzt`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWxy_zMultiP_nt`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWxy_tMultiP_nz`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWxy_nMultiP_zt`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWxz_nyt`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWxz_yMultiP_nt`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWxz_tMultiP_ny`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWxz_nMultiP_yt`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWyz_nxt`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWyz_xMultiP_nt`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWyz_tMultiP_nx`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWyz_nMultiP_xt`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWz_nxyt`	True	bool	Enable/disable the plotting of the Vertical Velocity
`lplotOce_velWz_nMultiP_xyt`	True	bool	Enable/disable the plotting of the Vertical Velocity

set_visual_lplotOce_vel

This section contains settings for visualizing the ocean velocity.

Parameter	Value	Type	Description
`lplotOce_velxy_nzt`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velxy_zMultiP_nt`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velxy_tMultiP_nz`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velxy_nMultiP_zt`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velxz_nyt`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velxz_yMultiP_nt`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velxz_tMultiP_ny`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velxz_nMultiP_yt`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velyz_nyt`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velyz_xMultiP_nt`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velyz_tMultiP_nx`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velyz_nMultiP_xt`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velz_nxyt`	True	bool	Enable/disable the plotting of the Velocity
`lplotOce_velz_nMultiP_xyt`	True	bool	Enable/disable the plotting of the Velocity

set_visual_lplotOce_vort

This section contains settings for visualizing the ocean vorticity.

Parameter	Value	Type	Description
lplotOce_vortxy_nzt	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortxy_zMultiP_nt	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortxy_tMultiP_nz	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortxy_nMultiP_zt	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortxz_nyt	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortxz_yMultiP_nt	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortxz_tMultiP_ny	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortxz_nMultiP_yt	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortyz_nxt	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortyz_xMultiP_nt	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortyz_tMultiP_nx	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortyz_nMultiP_xt	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortz_nxyt	True	bool	Enable/disable the plotting of the Vorticity
lplotOce_vortz_nMultiP_xyt	True	bool	Enable/disable the plotting of the Vorticity

set_visual_lplotOce_tke

This section contains settings for visualizing the ocean Total Kinetic Energy (TKE).

Parameter	Value	Type	Description
`lplotOce_tkexy_nzt`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkexy_zMultiP_nt`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkexy_tMultiP_nz`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkexy_nMultiP_zt`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkexz_nyt`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkexz_yMultiP_nt`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkexz_tMultiP_ny`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkexz_nMultiP_yt`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkeyz_nxt`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkeyz_xMultiP_nt`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkeyz_tMultiP_nx`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkeyz_nMultiP_xt`	True	bool	Enable/disable the plotting of the Total Kinetic Energy
`lplotOce_tkez_nxyt`	True	bool	Enable/disable the plotting of the

Parameter	Value	Type	Description
			Total Kinetic Energy
lplotOce_tkez_nMultiP_xyt	True	bool	Enable/disable the plotting of the Total Kinetic Energy

set_visual_lplotOce_ssh

This section contains settings for visualizing the ocean Sea Surface Height (SSH).

Parameter	Value	Type	Description
lplotOce_ssh_nt	True	bool	Enable/disable the plotting of the Sea Surface Height
lplotOce_ssh_tMultiP_n	True	bool	Enable/disable the plotting of the Sea Surface Height
lplotOce_ssh_nMultiP_t	True	bool	Enable/disable the plotting of the Sea Surface Height

set_visual_plotOce_velWDens

This section contains settings for visualizing the ocean Vertical Velocity combined with Density.

Parameter	Value	Type	Description
`lplotOce_velWDensxy_nzt`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensxy_zMultiP_nt`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensxy_tMultiP_nz`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensxy_nMultiP_zt`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensxz_nyt`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensxz_yMultiP_nt`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensxz_tMultiP_ny`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensxz_nMultiP_yt`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensyz_nxt`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensyz_xMultiP_nt`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensyz_tMultiP_nx`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensyz_nMultiP_xt`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density
`lplotOce_velWDensz_nxyt`	True	bool	Enable/disable the plotting of the

Parameter	Value	Type	Description
			Vertical Velocity + Density
`lplotOce_velWDensz_nMultiP_xyt`	True	bool	Enable/disable the plotting of the Vertical Velocity + Density

set_visual_lplotOce_tempCur

This section contains settings for visualizing the ocean Temperature combined with Ocean Currents.

Parameter	Value	Type	Description
`lplotOce_tempCurxy_nzt`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCurxy_zMultiP_nt`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCurxy_tMultiP_nz`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCurxy_nMultiP_zt`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCurxz_nyt`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCurxz_yMultiP_nt`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCurxz_tMultiP_ny`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCurxz_nMultiP_yt`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCuryz_nxt`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCuryz_xMultiP_nt`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCuryz_tMultiP_nx`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCuryz_nMultiP_xt`	True	bool	Enable/disable the plotting of the Temperature + Current
`lplotOce_tempCurz_nxyt`	True	bool	Enable/disable the plotting of the

Parameter	Value	Type	Description
			Temperature + Current
`lplotOce_tempCurz_nMultiP_xyt`	True	bool	Enable/disable the plotting of the Temperature + Current

set_visual_lplotOce_chlVSpar

This section contains settings for visualizing the child VS parent fields.

Parameter	Value	Description
`lplot_chlVSpar`	True	Enable/disable the plotting of the child VS parent fields
`lplot_chlVSpar_grid`	True	Enable/disable the plotting of the mesh
`lplot_chlVSpar_temp`	True	Enable/disable the plotting of the Temperature
`lplot_chlVSpar_sal`	True	Enable/disable the plotting of the Salinity
`lplot_chlVSpar_ssh`	True	Enable/disable the plotting of the Sea Surface Height
`lplot_chlVSpar_velU`	True	Enable/disable the plotting of the Zonal Velocity
`lplot_chlVSpar_velV`	True	Enable/disable the plotting of the Meridional Velocity
`lplot_chlVSpar_vel`	True	Enable/disable the plotting of the Velocity
`lplot_chlVSpar_tke`	True	Enable/disable the plotting of the Total Kinetic Energy

set_visual_lplotOce_surfVSsat

This section contains settings for visualizing the comparison between surface data and satellite-derived Sea Surface Temperature (SST).

Parameter	Value	Description
`lplot_surfVSsat`	True	Enable/disable the plotting of the surf VS satellite SST
`lplot_surfVSsat_sst_grid`	True	Enable/disable the plotting of the SST mesh
`lplot_surfVSsat_sst_obs`	True	Enable/disable the plotting of the SST observed
`lplot_surfVSsat_sst_parent`	True	Enable/disable the plotting of the SST of parent model
`lplot_surfVSsat_sst_child`	True	Enable/disable the plotting of the SST of child model
`lplot_surfVSsat_sst_chlVSpar`	True	Enable/disable the plotting of the SST of child VS parent model
`lplot_surfVSsat_sst_RMSE`	True	Enable/disable the plotting of the SST-RMSE
`lplot_surfVSsat_sst_BIAS`	True	Enable/disable the plotting of the SST-BIAS

set_visual_plotOce_surfVSctd

This section contains settings for visualizing the comparison between surface data and CTD (Conductivity, Temperature, and Depth) profiles.

Parameter	Value	Description
`lplot_surfVSctd`	True	Enable/disable the plotting of the surf VS CTD fields
`lplot_surfVSctd_grid`	True	Enable/disable the plotting of the CTD mesh
`lplot_surfVSctd_temp_obs`	False	Enable/disable the plotting of the Temperature observed
`lplot_surfVSctd_temp_parent`	False	Enable/disable the plotting of the Temperature of parent model
`lplot_surfVSctd_temp_child`	False	Enable/disable the plotting of the Temperature of child model
`lplot_surfVSctd_temp_chlVSpar`	True	Enable/disable the plotting of the Temperature of child VS parent model
`lplot_surfVSctd_temp_RMSE_fstat`	True	Enable/disable the plotting of the Temperature-RMSE
`lplot_surfVSctd_temp_RMSE_flev`	True	Enable/disable the plotting of the Temperature-RMSE
`lplot_surfVSctd_temp_BIAS_fstat`	True	Enable/disable the plotting of the Temperature-BIAS
`lplot_surfVSctd_temp_BIAS_flev`	True	Enable/disable the plotting of the Temperature-BIAS
`lplot_surfVSctd_saline_obs`	False	Enable/disable the plotting of the Salinity observed
`lplot_surfVSctd_saline_parent`	False	Enable/disable the plotting of the Salinity of parent model
`lplot_surfVSctd_saline_child`	False	Enable/disable the plotting of the Salinity of child model
`lplot_surfVSctd_saline_chlVSpar`	True	Enable/disable the plotting of the Salinity of child VS parent model
`lplot_surfVSctd_saline_RMSE_fstat`	True	Enable/disable the plotting of the Salinity-RMSE
`lplot_surfVSctd_saline_RMSE_flev`	True	Enable/disable the plotting of the Salinity-RMSE
`lplot_surfVSctd_saline_BIAS_fstat`	True	Enable/disable the plotting of the Salinity-BIAS
`lplot_surfVSctd_saline_BIAS_flev`	True	Enable/disable the plotting of the Salinity-BIAS
`lplot_surfVSctd_TS_obs`	False	Enable/disable the plotting of the Temperature & Salinity observed
`lplot_surfVSctd_density_obs`	False	Logical variable for the Density observed

set_visual_lplotOce_surfVSferrybox

This section contains settings for visualizing the comparison between surface data and Ferrybox data.

Parameter	Value	Description
`lplot_surfVSferrybox`	True	Enable/disable the plotting of the surf VS ferrybox fields
`lplot_surfVSferrybox_grid`	True	Enable/disable the plotting of the ferrybox mesh
`lplot_surfVSferrybox_vel_obs`	False	Enable/disable the plotting of the Velocity observed
`lplot_surfVSferrybox_vel_parent`	False	Enable/disable the plotting of the Velocity of parent model
`lplot_surfVSferrybox_vel_child`	False	Enable/disable the plotting of the Velocity of child model
`lplot_surfVSferrybox_vel_chlVSpar`	True	Enable/disable the plotting of the Velocity of child VS parent model
`lplot_surfVSferrybox_vel_RMSE_fstat`	True	Enable/disable the plotting of the Velocity-RMSE
`lplot_surfVSferrybox_vel_RMSE_flev`	True	Enable/disable the plotting of the Velocity-RMSE
`lplot_surfVSferrybox_vel_BIAS_fstat`	True	Enable/disable the plotting of the Velocity-BIAS
`lplot_surfVSferrybox_vel_BIAS_flev`	True	Enable/disable the plotting of the Velocity-BIAS

set_visual_lplotOce_surfVSmooring

This section contains settings for visualizing the comparison between surface data and mooring data fields.

Parameter	Value	Description
`lplot_surfVSmooring`	True	Enable/disable the plotting of the surf VS mooring fields
`lplot_surfVSmooring_grid`	True	Enable/disable the plotting of the mooring mesh
`lplot_surfVSmooring_vel_obs`	False	Enable/disable the plotting of the Velocity observed
`lplot_surfVSmooring_vel_parent`	False	Enable/disable the plotting of the Velocity of parent model
`lplot_surfVSmooring_vel_child`	False	Enable/disable the plotting of the Velocity of child model
`lplot_surfVSmooring_vel_chlVSpar`	True	Enable/disable the plotting of the Velocity of child VS parent model
`lplot_surfVSmooring_vel_RMSE_fstat`	True	Enable/disable the plotting of the Velocity-RMSE
`lplot_surfVSmooring_vel_RMSE_flev`	True	Enable/disable the plotting of the Velocity-RMSE
`lplot_surfVSmooring_vel_BIAS_fstat`	True	Enable/disable the plotting of the Velocity-BIAS
`lplot_surfVSmooring_vel_BIAS_flev`	True	Enable/disable the plotting of the Velocity-BIAS

set_dataDownl_sstsat

This section of the JSON file contains settings for downloading and configuring Sea Surface Temperature (SST) satellite data.

Parameter	Value	Description
`iSSTSat`	L3	SST Satellite dataset to be used
`mis_gateway_url`	http://myocean.artov.isac.cnr.it/mis-gateway-servlet/Motu	Vertical grid parameters specification
`product_name`	SST_MED_SST_L3S_NRT_OBSERVATIONS_010_012	Vertical grid parameters specification
`datasetID`	SST_MED_SST_L3S_NRT_OBSERVATIONS_010_012_a	Vertical grid parameters specification
`serviceID`	http://purl.org/myocean/ontology/service/database#SST_MED_SST_L3S_NRT_OBSERVATIONS_010_012-TDS	...

set_dataDownl_ctd

This section includes settings for downloading and configuring CTD (Conductivity, Temperature, and Depth) data.

Parameter	Value	Description
`ldownl_ctd`	True	Download the CTD data
`urlbase_ctd`	ftp://ingvop.bo.ingv.it	URL for the resource available
`usr_ctd`	unibo	Username to access the data
`pwd_ctd`	9fqXLsirmd	Password to access the data
`filebase_ctd`	NO_201708_TS_MO_6201077.nc	File name
`iformatFile_ctd`	2	File format to be downloaded (0=txt, 1=netcdf)

5.2.5 General Visualization and Plot Configuration

set_visual_fileImg

This section contains settings related to the file format of the generated image.

Parameter	Value	Description
`fileImg_type`	png	Type of the image file to generate
`fileImg_wkWidth`	2400	Horizontal resolution (number of pixels) of the image file
`fileImg_wkHeight`	2400	Vertical resolution (number of pixels) of the image file

set_visual_lvis

This section contains parameters that control the visibility and display settings for various graphical elements in plots

Parameter	Value	Description
`lvis_title`	False	Enable/disable the visibility of a given string as the main title
`lvis_mplotDiff`	False	Enable/disable the visibility of ...
`lvis_leftString`	True	Enable/disable the visibility of a string above the plot's upper boundary and left
`lvis_rightString`	False	Enable/disable the visibility of a string above the plot's upper boundary and right
`lvis_axisLab`	True	Enable/disable the visibility of the X and Y axis titles
`lvis_tickmarks`	True	Enable/disable the visibility of the right, left, top, and bottom tick marks
`lvis_borders`	True	Enable/disable the visibility of the right, left, top, and bottom borders
`lvis_mapFillOn`	True	Enable/disable the visibility of the map area fill
`lvis_mapOutlineOn`	True	Enable/disable the visibility of the map area outlines
`lvis_vecRefAnnoOn`	True	Enable/disable the visibility of the reference vector annotation
`lvis_labbarOn`	True	Enable/disable the visibility of the LabelBar
`lvis_labbarLabelsOn`	True	Enable/disable the visibility of the LabelBar labels
`lvis_panelbarOn`	False	Enable/disable the visibility of the PanelBar
`lvis_infoContourOn`	False	Enable/disable the visibility of the info contour
`lvis_grid`	False	Enable/disable the visibility of the mesh grid
`lvis_myCoastline`	False	Enable/disable the visibility of the user coastline
`lvis_fillcurves`	False	Enables the filling of the area between two curves
`lvis_boxNest`	True	Enable/disable the visibility of the nest rectangular box

set_visual_graph

This section contains settings for graph and map visualization parameters.

Parameter	Value	Description
graph_projection	Mercator	Projection used for the map transformation (e.g., CylindricalEquidistant, Mercator, etc.)
graph_fillMode	AreaFill	How ContourPlot performs fill: (AreaFill), (RasterFill), (CellFill)
graph_minDistVec	0.015 0.018	Minimum distance (in NDC space) separating data locations of neighboring vectors
graph_thickArrowVec	2.0	Thickness of the line used to draw vector line arrows
graph_refMagnVecAtm	2.0	Reference magnitude used for the wind vector field
graph_refMagnVecOce	0.5	Reference magnitude used for the current vector field
graph_styleVec	CurlyVector	Style of glyph used to represent vector magnitude and direction (LineArrow, CurlyVector)
graph_iPlOrient	1	Panels orientation in multi-plots: (0) vertical, (1) horizontal
graph_nlevsBar	21	Number of levels for the color bar
graph_polymarkerSize	0.01	Size used to draw the marker in polymarker plots
graph_markLineMode	MarkLines	Draw curves using lines only (Lines), markers only (Markers), or both (MarkLines)
graph_lineThick	2.5	Thickness of the line used to draw the curves
graph_markerSize	0.01	Size used to draw the marker in xy-plots
graph_markers	16	Style of the markers in xy-plots
graph_boxThick	8.0	Thickness of the line used to draw the box
graph_boxColor	red	Color of the line used to draw the box

set_visual_colorsMap

This section specifies the color schemes used in plots.

Parameter	Value	Description
colorsMap_Rainbow0	BlAqGrYeOrReVi200	Color map for drawing a contour plot
colorsMap_Rainbow1	WhiteBlueGreenYellowRed	Color map for drawing a contour plot
colorsMap_EarthOcean	GMT_drywet	Color map for drawing a contour plot
colorsMap_BlueRed0	BlWhRe	Color map for drawing a contour plot
colorsMap_BlueRed1	cmp_b2r	Color map for drawing a contour plot
colorsMap_Line	black	Color used when drawing the corresponding curves
colorsMap_Mark	black	Color used when drawing the markers
colorsMap_LineFillPlus	PaleGreen	Fill color to use between two curves
colorsMap_LineFillMinus	lightcoral	Fill color to use between two curves

set_visual_labelNameMesh

This section contains settings for axis labels related to mesh coordinates used in visual representations.

Parameter	Value	Description
axisLabMesh_lon	Longitude	String to use as the Axis title for the longitude coordinate
axisLabMesh_lat	Latitude	String to use as the Axis title for the latitude coordinate
axisLabMesh_depth	Depth [m]	String to use as the Axis title for the vertical depth
axisLabMesh_thickness	Scale Factor [m]	String to use as the Axis title for the vertical thickness
axisLabMesh_vertIdx	z-level index	String to use as the Axis title for the index

set_visual_labelNameBat

This section contains settings for axis labels related to bathymetry used in visual representations.

Parameter	Value	Description
axisLabBat_bat	Bathymetry	String to use as the axis label for bathymetry

set_visual_labelNameAtm

This section contains settings for axis labels related to atmospheric data used in visual representations.

Parameter	Value	Description
`axisLabAtm_mslp`	Sea Level Pressure	String to use as the axis label for sea level pressure
`axisLabAtm_cloudCov`	Cloud Cover	String to use as the axis label for cloud cover
`axisLabAtm_temp`	Air Temperature	String to use as the axis label for air temperature
`axisLabAtm_dpTemp`	Dew Point Temperature	String to use as the axis label for dew point temperature
`axisLabAtm_vel`	Wind	String to use as the axis label for wind speed
`axisLabAtm_prec`	Precipitation	String to use as the axis label for precipitation
`axisLabAtm_tauU`	Wind stress along i-axis	String to use as the axis label for wind stress in the i-axis
`axisLabAtm_tauV`	Wind stress along j-axis	String to use as the axis label for wind stress in the j-axis
`axisLabAtm_qtot`	Total heat flux	String to use as the axis label for total heat flux
`axisLabAtm_qsr`	Solar Radiation Penetration	String to use as the axis label for solar radiation penetration
`axisLabAtm_emp`	Mass flux exchanged with the atmosphere	String to use as the axis label for mass flux exchange
`axisLabAtm_tempS`	Surface Temperature	String to use as the axis label for surface temperature
`axisLabAtm_salS`	Surface Salinity	String to use as the axis label for surface salinity
`axisLabAtm_velU`	Wind velocity along i-axis	String to use as the axis label for wind velocity in the i-axis
`axisLabAtm_velV`	Wind velocity along j-axis	String to use as the axis label for wind velocity in the j-axis
`axisLabAtm_vel`	Wind velocity	String to use as the axis label for wind velocity
`axisLabAtm_temp`	Air temperature at 2m	String to use as the axis label for air temperature at 2 meters
`axisLabAtm_umid`	Air humidity at 2m	String to use as the axis label for air humidity at 2 meters
`axisLabAtm_radLW`	Long wave radiative flux	String to use as the axis label for long wave radiative flux
`axisLabAtm_radSW`	Short wave radiative flux	String to use as the axis label for short wave radiative flux
`axisLabAtm_prec`	Total precipitation	String to use as the axis label for total precipitation
`axisLabAtm_snow`	Solid precipitation	String to use as the axis label for solid precipitation

set_visual_labelNameOce

This section contains settings for axis labels related to oceanographic data used in visual representations.

Parameter	Value	Description
`axisLabOce_temp`	Temperature	String to use as the axis label for temperature
`axisLabOce_sal`	Salinity	String to use as the axis label for salinity
`axisLabOce_dens`	Density	String to use as the axis label for density
`axisLabOce_velU`	Zonal Velocity	String to use as the axis label for zonal velocity
`axisLabOce_velV`	Meridional Velocity	String to use as the axis label for meridional velocity
`axisLabOce_velW`	Vertical Velocity	String to use as the axis label for vertical velocity
`axisLabOce_vel`	Current	String to use as the axis label for ocean current
`axisLabOce_vort`	Relative Vorticity	String to use as the axis label for relative vorticity
`axisLabOce_ssh`	Sea Surface Height	String to use as the axis label for sea surface height

set_visual_myWin

This section contains settings for defining the plot window with specific geographical and depth limits.

Parameter	Value	Description
`lmyWinsize`	False	Logical variable to enable using a user-defined section to plot
`minLonT`	17.0	Specific minimum longitude of the plot window
`maxLonT`	17.25	Specific maximum longitude of the plot window
`minLatT`	40.3	Specific minimum latitude of the plot window
`maxLatT`	40.52	Specific maximum latitude of the plot window
`minDepthT`	0.0	Specific minimum depth of the plot window
`maxDepthT`	400.0	Specific maximum depth of the plot window
`minThicknessT`	0.0	Specific minimum vertical resolution of the plot window
`maxThicknessT`	100.0	Specific maximum vertical resolution of the plot window
`lfixWinsize`	False	Logical variable for specific control
`inestFixWin`	1	Specific parameter for nested window control

5.2.6 Space-Time Sections Configuration

set_visual_mySect_datetime

This section defines the settings for visualizing specific time intervals or moments within the data.

Parameter	Value	Description
`myPlotSect_ldatetime`	True	Enables to visualize user-defined time sections (if False, initial time will be plotted)
`myPlotSect_itypeDatetime`	0	Type of the time specification at which the data are plotted: (=0) nt/dt, (=1) specific time
`myPlotSect_deltaDatetime`	2	Time interval used to define time sections if `itypeDatetime=0` (only divisible values: 1, 2, 3, 4, 6, 8, 12, 24)
`myPlotSect_datetime0`	2014-10-07_03:30:00	Specific date0 in the form YYYY-MM-DD_hhmmss if `itypeDatetime=1` (if =NOTUSED, parameter not read)
`myPlotSect_datetime1`	2014-10-07_05:30:00	Specific date1 in the form YYYY-MM-DD_hhmmss if `itypeDatetime=1` (if =NOTUSED, parameter not read)
`myPlotSect_datetime2`	2014-10-07_13:30:00	Specific date2 in the form YYYY-MM-DD_hhmmss if `itypeDatetime=1` (if =NOTUSED, parameter not read)
`myPlotSect_datetime3`	2014-10-07_21:30:00	Specific date3 in the form YYYY-MM-DD_hhmmss if `itypeDatetime=1` (if =NOTUSED, parameter not read)

set_visual_mySect_lev

This section defines the settings for visualizing specific levels or vertical intervals within the dataset.

Parameter	Value	Description
`myPlotSect_llev`	True	Enables visualization of user-defined level sections (if False, surface level will be plotted)
`myPlotSect_itypeLev`	1	Type of the level specification at which data are plotted: (0) nlev/dlev, (1) specific level
`myPlotSect_deltaLev`	2	Level interval used to define level sections if `itypeLev=0`
`myPlotSect_lev0`	9.8	Specific level0 at which data are plotted if `itypeLev=1` (if =NOTUSED, parameter not read)
`myPlotSect_lev1`	30.0	Specific level1 at which data are plotted if `itypeLev=1` (if =NOTUSED, parameter not read)
`myPlotSect_lev2`	NOTUSED	Specific level2 at which data are plotted if `itypeLev=1` (if =NOTUSED, parameter not read)
`myPlotSect_lev3`	NOTUSED	Specific level3 at which data are plotted if `itypeLev=1` (if =NOTUSED, parameter not read)

set_visual_mySect_lat

This section defines the settings for visualizing specific latitudes or latitude intervals within the dataset.

Parameter	Value	Description
`myPlotSect_llat`	True	Enables visualization of user-defined latitude sections (if False, mid latitude will be plotted)
`myPlotSect_itypeLat`	1	Type of the latitude specification at which data are plotted: (0) nlat/dlat, (1) specific latitude
`myPlotSect_deltaLat`	2	Latitude interval used to define latitude sections if `itypeLat=0`
`myPlotSect_lat0`	40.1	Specific latitude0 at which data are plotted if `itypeLat=1` (if =NOTUSED, parameter not read)
`myPlotSect_lat1`	40.2	Specific latitude1 at which data are plotted if `itypeLat=1` (if =NOTUSED, parameter not read)
`myPlotSect_lat2`	NOTUSED	Specific latitude2 at which data are plotted if `itypeLat=1` (if =NOTUSED, parameter not read)
`myPlotSect_lat3`	NOTUSED	Specific latitude3 at which data are plotted if `itypeLat=1` (if =NOTUSED, parameter not read)

set_visual_mySect_lon

This section defines the settings for visualizing specific longitudes or longitude intervals within the dataset.

Parameter	Value	Description
`myPlotSect_llon`	True	Enables visualization of user-defined longitude sections (if False, mid longitude will be plotted)
`myPlotSect_itypeLon`	1	Type of the longitude specification at which data are plotted: (0) nlon/dlon, (1) specific longitude
`myPlotSect_deltaLon`	2	Longitude interval used to define longitude sections if `itypeLon=0`
`myPlotSect_lon0`	17.4	Specific longitude0 at which data are plotted if `itypeLon=1` (if =NOTUSED, parameter not read)
`myPlotSect_lon1`	NOTUSED	Specific longitude1 at which data are plotted if `itypeLon=1` (if =NOTUSED, parameter not read)
`myPlotSect_lon2`	NOTUSED	Specific longitude2 at which data are plotted if `itypeLon=1` (if =NOTUSED, parameter not read)
`myPlotSect_lon3`	NOTUSED	Specific longitude3 at which data are plotted if `itypeLon=1` (if =NOTUSED, parameter not read)

set_visual_myPoints

This section defines the settings for visualizing specific geographic coordinates and depths within the dataset.

Parameter	Value	Description
`myPoints_lxyz`	True	Enables visualization of user-defined points
`myPoints_spec`	0	Type of points-coord specification at which the data are plotted: (0) lat, lon, depth index (1) lat, lon, depth coord
`myPoints_ilon`	2	Specific longitude-index at which the points are plotted if `spec=0`
`myPoints_ilat`	1	Specific latitude-index at which the points are plotted if `spec=0`
`myPoints_ilev`	2	Specific depth-index at which the points are plotted if `spec=0`
`myPoints_lon`	1.0	Specific longitude at which the points are plotted if `spec=1`
`myPoints_lat`	1.0	Specific latitude at which the points are plotted if `spec=1`
`myPoints_lev`	2.0	Specific depth at which the points are plotted if `spec=1`

5.2.7 ColorBar Configuration

set_visual_myColorBar

This section provides settings for customizing the color bar in visualizations

Parameter	Value	Description
`lmycolorBarBat`	True	Allows the user to set the range of values of the contour levels for the bathymetry data
`lmycolorBarAtm`	True	Allows the user to set the range of values of the contour levels for the atmospheric data
`lmycolorBarOce`	True	Allows the user to set the range of values of the contour levels for the ocean data

set_visual_myColorBarBat_maxDepth

This section specifies the color bar configuration for bathymetry data.

Parameter	Value	Description
`rangeBat_maxDepth_min`	0.0	Specifies the minimum contour level for the Bathymetry
`rangeBat_maxDepth_max`	1800.0	Specifies the maximum contour level for the Bathymetry
`rangeBat_maxDepth_nbar`	26	Specifies the number of intervals for the label bar
`rangeBat_maxDepthDiff_min`	-1.0	Specifies the minimum contour level for the Bathymetry difference
`rangeBat_maxDepthDiff_max`	1.0	Specifies the maximum contour level for the Bathymetry difference
`rangeBat_maxDepthDiff_nbar`	21	Specifies the number of intervals for the label bar

set_visual_myColorBarAtm_mslp

This section specifies the color bar configuration for sea level pressure data.

Parameter	Value	Description
`rangeAtm_mslp_min`	1011.0	Specifies the minimum contour level for the Sea Level Pressure
`rangeAtm_mslp_max`	1020.0	Specifies the maximum contour level for the Sea Level Pressure
`rangeAtm_mslp_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_mslpDiff_min`	-1.0	Specifies the minimum contour level for the Sea Level Pressure difference
`rangeAtm_mslpDiff_max`	1.0	Specifies the maximum contour level for the Sea Level Pressure difference
`rangeAtm_mslpDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_velU

This section specifies the color bar configuration for Zonal Wind data.

Parameter	Value	Description
`rangeAtm_velU_min`	-10.0	Specifies the minimum contour level for the Zonal Wind
`rangeAtm_velU_max`	10.0	Specifies the maximum contour level for the Zonal Wind
`rangeAtm_velU_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_velUDiff_min`	-1.0	Specifies the minimum contour level for the Zonal Wind difference
`rangeAtm_velUDiff_max`	1.0	Specifies the maximum contour level for the Zonal Wind difference
`rangeAtm_velUDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_velV

This section specifies the color bar configuration for Meridional Wind data.

Parameter	Value	Description
`rangeAtm_velV_min`	-10.0	Specifies the minimum contour level for the Meridional Wind
`rangeAtm_velV_max`	10.0	Specifies the maximum contour level for the Meridional Wind
`rangeAtm_velV_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_velVDiff_min`	-1.0	Specifies the minimum contour level for the Meridional Wind difference
`rangeAtm_velVDiff_max`	1.0	Specifies the maximum contour level for the Meridional Wind difference
`rangeAtm_velVDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_vel

This section specifies the color bar configuration for Wind data.

Parameter	Value	Description
`rangeAtm_vel_min`	-10.0	Specifies the minimum contour level for the Wind
`rangeAtm_vel_max`	10.0	Specifies the maximum contour level for the Wind
`rangeAtm_vel_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_velDiff_min`	-1.0	Specifies the minimum contour level for the Wind difference
`rangeAtm_velDiff_max`	1.0	Specifies the maximum contour level for the Wind difference
`rangeAtm_velDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_cloudCov

This section specifies the color bar configuration for Total Cloud Cover data.

Parameter	Value	Description
`rangeAtm_cloudCov_min`	0.0	Specifies the minimum contour level for the Cloud Cover
`rangeAtm_cloudCov_max`	100.0	Specifies the maximum contour level for the Cloud Cover
`rangeAtm_cloudCov_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_cloudCovDiff_min`	-1.0	Specifies the minimum contour level for the Cloud Cover difference
`rangeAtm_cloudCovDiff_max`	1.0	Specifies the maximum contour level for the Cloud Cover difference
`rangeAtm_cloudCovDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_temp

This section specifies the color bar configuration for Air Temperature data.

Parameter	Value	Description
`rangeAtm_temp_min`	285.0	Specifies the minimum contour level for the Temperature
`rangeAtm_temp_max`	296.0	Specifies the maximum contour level for the Temperature
`rangeAtm_temp_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_tempDiff_min`	-1.0	Specifies the minimum contour level for the Temperature difference
`rangeAtm_tempDiff_max`	1.0	Specifies the maximum contour level for the Temperature difference
`rangeAtm_tempDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_dpTemp

This section specifies the color bar configuration for Dew Point Temperature data.

Parameter	Value	Description
`rangeAtm_dpTemp_min`	285.0	Specifies the minimum contour level for the Dew Point Temperature
`rangeAtm_dpTemp_max`	290.0	Specifies the maximum contour level for the Dew Point Temperature
`rangeAtm_dpTemp_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_dpTempDiff_min`	-1.0	Specifies the minimum contour level for the Dew Point Temperature difference
`rangeAtm_dpTempDiff_max`	1.0	Specifies the maximum contour level for the Dew Point Temperature difference
`rangeAtm_dpTempDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_prec

This section specifies the color bar configuration for Total Precipitation data.

Parameter	Value	Description
`rangeAtm_prec_min`	1.0e-5	Specifies the minimum contour level for the Total Precipitation
`rangeAtm_prec_max`	5.0e-3	Specifies the maximum contour level for the Total Precipitation
`rangeAtm_prec_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_precDiff_min`	-1.0	Specifies the minimum contour level for the Total Precipitation difference
`rangeAtm_precDiff_max`	1.0	Specifies the maximum contour level for the Total Precipitation difference
`rangeAtm_precDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_snow

This section specifies the color bar configuration for Solid Precipitation data.

Parameter	Value	Description
`rangeAtm_snow_min`	1.0e-5	Specifies the minimum contour level for the Solid Precipitation
`rangeAtm_snow_max`	5.0e-3	Specifies the maximum contour level for the Solid Precipitation
`rangeAtm_snow_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_snowDiff_min`	-1.0	Specifies the minimum contour level for the Solid Precipitation difference
`rangeAtm_snowDiff_max`	1.0	Specifies the maximum contour level for the Solid Precipitation difference
`rangeAtm_snowDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_umid

This section specifies the color bar configuration for Humidity data.

Parameter	Value	Description
`rangeAtm_umid_min`	0.01	Specifies the minimum contour level for the Humidity
`rangeAtm_umid_max`	0.02	Specifies the maximum contour level for the Humidity
`rangeAtm_umid_nbar`	26	Specifies the number of intervals for the label bar
`rangeAtm_umidDiff_min`	-1.0	Specifies the minimum contour level for the Humidity difference
`rangeAtm_umidDiff_max`	1.0	Specifies the maximum contour level for the Humidity difference
`rangeAtm_umidDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_radLW

This section specifies the color bar configuration for Long Wave Radiative Flux data.

Parameter	Value	Description
`rangeAtm_radLW_min`	380.0	Specifies the minimum contour level for the Long Wave Radiative Flux
`rangeAtm_radLW_max`	440.0	Specifies the maximum contour level for the Long Wave Radiative Flux
`rangeAtm_radLW_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_radLWDiff_min`	-1.0	Specifies the minimum contour level for the Long Wave Radiative Flux difference
`rangeAtm_radLWDiff_max`	1.0	Specifies the maximum contour level for the Long Wave Radiative Flux difference
`rangeAtm_radLWDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_radSW

This section specifies the color bar configuration for Short Wave Radiative Flux data.

Parameter	Value	Description
`rangeAtm_radSW_min`	100.0	Specifies the minimum contour level for the Short Wave Radiative Flux
`rangeAtm_radSW_max`	300.0	Specifies the maximum contour level for the Short Wave Radiative Flux
`rangeAtm_radSW_nbar`	26	Specifies the number of intervals for the label bar
`rangeAtm_radSWDiff_min`	-1.0	Specifies the minimum contour level for the Short Wave Radiative Flux difference
`rangeAtm_radSWDiff_max`	1.0	Specifies the maximum contour level for the Short Wave Radiative Flux difference
`rangeAtm_radSWDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_tauU

This section specifies the color bar configuration for Zonal Wind Stress data.

Parameter	Value	Description
`rangeAtm_tauU_min`	-10.0	Specifies the minimum contour level for the Zonal Wind Stress
`rangeAtm_tauU_max`	10.0	Specifies the maximum contour level for the Zonal Wind Stress
`rangeAtm_tauUDiff_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_tauUDiff_min`	-1.0	Specifies the minimum contour level for the Zonal Wind Stress difference
`rangeAtm_tauUDiff_max`	1.0	Specifies the maximum contour level for the Zonal Wind Stress difference
`rangeAtm_tauUDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_tauV

This section specifies the color bar configuration for Meridional Wind Stress data.

Parameter	Value	Description
`rangeAtm_tauV_min`	-10.0	Specifies the minimum contour level for the Meridional Wind Stress
`rangeAtm_tauV_max`	10.0	Specifies the maximum contour level for the Meridional Wind Stress
`rangeAtm_tauVDiff_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_tauVDiff_min`	-1.0	Specifies the minimum contour level for the Meridional Wind Stress difference
`rangeAtm_tauVDiff_max`	1.0	Specifies the maximum contour level for the Meridional Wind Stress difference
`rangeAtm_tauVDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_tau

This section specifies the color bar configuration for Wind Stress data.

Parameter	Value	Description
`rangeAtm_tau_min`	-10.0	Specifies the minimum contour level for the Wind Stress
`rangeAtm_tau_max`	10.0	Specifies the maximum contour level for the Wind Stress
`rangeAtm_tauDiff_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_tauDiff_min`	-1.0	Specifies the minimum contour level for the Wind Stress difference
`rangeAtm_tauDiff_max`	1.0	Specifies the maximum contour level for the Wind Stress difference
`rangeAtm_tauDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_qtot

This section specifies the color bar configuration for Total Heat Flux data.

Parameter	Value	Description
`rangeAtm_qtot_min`	-10.0	Specifies the minimum contour level for the Total Heat Flux
`rangeAtm_qtot_max`	10.0	Specifies the maximum contour level for the Total Heat Flux
`rangeAtm_qtotDiff_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_qtotDiff_min`	-1.0	Specifies the minimum contour level for the Total Heat Flux difference
`rangeAtm_qtotDiff_max`	1.0	Specifies the maximum contour level for the Total Heat Flux difference
`rangeAtm_qtotDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_emp

This section specifies the color bar configuration for Mass flux exchanged data.

Parameter	Value	Description
`rangeAtm_emp_min`	-10.0	Specifies the minimum contour level for the Mass flux exchanged with the atmosphere
`rangeAtm_emp_max`	10.0	Specifies the maximum contour level for the Mass flux exchanged with the atmosphere
`rangeAtm_empDiff_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_empDiff_min`	-1.0	Specifies the minimum contour level for the Mass flux difference
`rangeAtm_empDiff_max`	1.0	Specifies the maximum contour level for the Mass flux difference
`rangeAtm_empDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_tempS

This section specifies the color bar configuration for Surface Temperature data.

Parameter	Value	Description
`rangeAtm_tempS_min`	-10.0	Specifies the minimum contour level for the Surface Temperature
`rangeAtm_tempS_max`	10.0	Specifies the maximum contour level for the Surface Temperature
`rangeAtm_tempSDiff_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_tempSDiff_min`	-1.0	Specifies the minimum contour level for the Surface Temperature difference
`rangeAtm_tempSDiff_max`	1.0	Specifies the maximum contour level for the Surface Temperature difference
`rangeAtm_tempSDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarAtm_salS

This section specifies the color bar configuration for Surface Salinity data.

Parameter	Value	Description
`rangeAtm_salS_min`	-10.0	Specifies the minimum contour level for the Surface Salinity
`rangeAtm_salS_max`	10.0	Specifies the maximum contour level for the Surface Salinity
`rangeAtm_salSDiff_nbar`	31	Specifies the number of intervals for the label bar
`rangeAtm_salSDiff_min`	-1.0	Specifies the minimum contour level for the Surface Salinity difference
`rangeAtm_salSDiff_max`	1.0	Specifies the maximum contour level for the Surface Salinity difference
`rangeAtm_salSDiff_nbar`	31	Specifies the number of intervals for the label bar for differences

set_visual_myColorBarOce_temp

This section specifies the color bar configuration for Temperature data.

Parameter	Value	Description
`rangeOce_temp_xy_min`	22.0	Specifies the minimum contour level for the horizontal section of the Temperature
`rangeOce_temp_xy_max`	25.0	Specifies the maximum contour level for the horizontal section of the Temperature
`rangeOce_temp_xy_nbar`	31	Specifies the intervals number of contour levels for the horizontal section of the Temperature
`rangeOce_tempDiff_xy_min`	-1.0	Specifies the minimum contour level for the horizontal section of the Temperature difference
`rangeOce_tempDiff_xy_max`	1.0	Specifies the maximum contour level for the horizontal section of the Temperature difference
`rangeOce_tempDiff_xy_nbar`	41	Specifies the intervals number of contour levels for the horizontal section of the Temperature difference
`rangeOce_temp_z_min`	14.0	Specifies the minimum contour level for the zonal section of the Temperature
`rangeOce_temp_z_max`	24.5	Specifies the maximum contour level for the zonal section of the Temperature
`rangeOce_temp_z_nbar`	41	Specifies the intervals number of contour levels for the zonal section of the Temperature
`rangeOce_tempDiff_z_min`	-1.0	Specifies the minimum contour level for the zonal section of the Temperature difference
`rangeOce_tempDiff_z_max`	1.0	Specifies the maximum contour level for the zonal section of the Temperature difference
`rangeOce_tempDiff_z_nbar`	41	Specifies the intervals number of contour levels for the zonal section of the Temperature difference

set_visual_myColorBarOce_sal

This section specifies the color bar configuration for Salinity data.

Parameter	Value	Description
`rangeOce_sal_xy_min`	37.2	Specifies the minimum contour level for the horizontal section of the Salinity
`rangeOce_sal_xy_max`	38.8	Specifies the maximum contour level for the horizontal section of the Salinity
`rangeOce_sal_xy_nbar`	41	Specifies the intervals number of contour levels for the horizontal section of the Salinity
`rangeOce_salDiff_xy_min`	-0.1	Specifies the minimum contour level for the horizontal section of the Salinity difference
`rangeOce_salDiff_xy_max`	0.1	Specifies the maximum contour level for the horizontal section of the Salinity difference
`rangeOce_salDiff_xy_nbar`	31	Specifies the intervals number of contour levels for the horizontal section of the Salinity difference
`rangeOce_sal_z_min`	37.2	Specifies the minimum contour level for the zonal section of the Salinity
`rangeOce_sal_z_max`	38.8	Specifies the maximum contour level for the zonal section of the Salinity
`rangeOce_sal_z_nbar`	41	Specifies the intervals number of contour levels for the zonal section of the Salinity
`rangeOce_salDiff_z_min`	-0.5	Specifies the minimum contour level for the zonal section of the Salinity difference
`rangeOce_salDiff_z_max`	0.5	Specifies the maximum contour level for the zonal section of the Salinity difference
`rangeOce_salDiff_z_nbar`	21	Specifies the intervals number of contour levels for the zonal section of the Salinity difference

set_visual_myColorBarOce_dens

This section specifies the color bar configuration for Density data.

Parameter	Value	Description
`rangeOce_dens_xy_min`	25.5	Specifies the minimum contour level for the horizontal section of the Density (10^3 kg/m³ has been subtracted)
`rangeOce_dens_xy_max`	26.8	Specifies the maximum contour level for the horizontal section of the Density (10^3 kg/m³ has been subtracted)
`rangeOce_dens_xy_nbar`	14	Specifies the intervals number of contour levels for the horizontal section of the Density
`rangeOce_densDiff_xy_min`	-1.0	Specifies the minimum contour level for the horizontal section of the Density difference
`rangeOce_densDiff_xy_max`	1.0	Specifies the maximum contour level for the horizontal section of the Density difference
`rangeOce_densDiff_xy_nbar`	31	Specifies the intervals number of contour levels for the horizontal section of the Density difference
`rangeOce_dens_z_min`	25.0	Specifies the minimum contour level for the zonal section of the Density
`rangeOce_dens_z_max`	29.0	Specifies the maximum contour level for the zonal section of the Density
`rangeOce_dens_z_nbar`	41	Specifies the intervals number of contour levels for the zonal section of the Density
`rangeOce_densDiff_z_min`	-1.0	Specifies the minimum contour level for the zonal section of the Density difference
`rangeOce_densDiff_z_max`	1.0	Specifies the maximum contour level for the zonal section of the Density difference
`rangeOce_densDiff_z_nbar`	31	Specifies the intervals number of contour levels for the zonal section of the Density difference

set_visual_myColorBarOce_velU

This section specifies the color bar configuration for Zonal Velocity data.

Parameter	Value	Description
`rangeOce_velU_xy_min`	-1.0	Specifies the minimum contour level for the horizontal section of the Zonal Velocity
`rangeOce_velU_xy_max`	1.0	Specifies the maximum contour level for the horizontal section of the Zonal Velocity
`rangeOce_velU_xy_nbar`	21	Specifies the intervals number of contour levels for the horizontal section of the Zonal Velocity
`rangeOce_velUDiff_xy_min`	-0.15	Specifies the minimum contour level for the horizontal section of the Zonal Velocity difference
`rangeOce_velUDiff_xy_max`	0.15	Specifies the maximum contour level for the horizontal section of the Zonal Velocity difference
`rangeOce_velUDiff_xy_nbar`	31	Specifies the intervals number of contour levels for the horizontal section of the Zonal Velocity difference
`rangeOce_velU_z_min`	-1.0	Specifies the minimum contour level for the zonal section of the Zonal Velocity
`rangeOce_velU_z_max`	1.0	Specifies the maximum contour level for the zonal section of the Zonal Velocity
`rangeOce_velU_z_nbar`	21	Specifies the intervals number of contour levels for the zonal section of the Zonal Velocity
`rangeOce_velUDiff_z_min`	-0.15	Specifies the minimum contour level for the zonal section of the Zonal Velocity difference
`rangeOce_velUDiff_z_max`	0.15	Specifies the maximum contour level for the zonal section of the Zonal Velocity difference
`rangeOce_velUDiff_z_nbar`	31	Specifies the intervals number of contour levels for the zonal section of the Zonal Velocity difference

set_visual_myColorBarOce_velV

This section specifies the color bar configuration for Meridional Velocity data.

Parameter	Value	Description
`rangeOce_velV_xy_min`	-1.0	Specifies the minimum contour level for the horizontal section of the Meridional Velocity
`rangeOce_velV_xy_max`	1.0	Specifies the maximum contour level for the horizontal section of the Meridional Velocity
`rangeOce_velV_xy_nbar`	21	Specifies the intervals number of contour levels for the horizontal section of the Meridional Velocity
`rangeOce_velVDiff_xy_min`	-0.15	Specifies the minimum contour level for the horizontal section of the Meridional Velocity difference
`rangeOce_velVDiff_xy_max`	0.15	Specifies the maximum contour level for the horizontal section of the Meridional Velocity difference
`rangeOce_velVDiff_xy_nbar`	31	Specifies the intervals number of contour levels for the horizontal section of the Meridional Velocity difference
`rangeOce_velV_z_min`	-1.0	Specifies the minimum contour level for the zonal section of the Meridional Velocity
`rangeOce_velV_z_max`	1.0	Specifies the maximum contour level for the zonal section of the Meridional Velocity
`rangeOce_velV_z_nbar`	21	Specifies the intervals number of contour levels for the zonal section of the Meridional Velocity
`rangeOce_velVDiff_z_min`	-0.15	Specifies the minimum contour level for the zonal section of the Meridional Velocity difference
`rangeOce_velVDiff_z_max`	0.15	Specifies the maximum contour level for the zonal section of the Meridional Velocity difference
`rangeOce_velVDiff_z_nbar`	31	Specifies the intervals number of contour levels for the zonal section of the Meridional Velocity difference

set_visual_myColorBarOce_velW

This section specifies the color bar configuration for Vertical Velocity data.

Parameter	Value	Description
`rangeOce_velW_xy_min`	-100.	Specifies the minimum contour level for the horizontal section of the Vertical Velocity
`rangeOce_velW_xy_max`	100.	Specifies the maximum contour level for the horizontal section of the Vertical Velocity
`rangeOce_velW_xy_nbar`	31	Specifies the intervals number of contour levels for the horizontal section of the Vertical Velocity
`rangeOce_velWDiff_xy_min`	-0.1	Specifies the minimum contour level for the horizontal section of the Vertical Velocity difference
`rangeOce_velWDiff_xy_max`	0.1	Specifies the maximum contour level for the horizontal section of the Vertical Velocity difference
`rangeOce_velWDiff_xy_nbar`	31	Specifies the intervals number of contour levels for the horizontal section of the Vertical Velocity difference
`rangeOce_velW_z_min`	-100.	Specifies the minimum contour level for the zonal section of the Vertical Velocity
`rangeOce_velW_z_max`	100.	Specifies the maximum contour level for the zonal section of the Vertical Velocity
`rangeOce_velW_z_nbar`	31	Specifies the intervals number of contour levels for the zonal section of the Vertical Velocity
`rangeOce_velWDiff_z_min`	-0.1	Specifies the minimum contour level for the zonal section of the Vertical Velocity difference
`rangeOce_velWDiff_z_max`	0.1	Specifies the maximum contour level for the zonal section of the Vertical Velocity difference
`rangeOce_velWDiff_z_nbar`	31	Specifies the intervals number of contour levels for the zonal section of the Vertical Velocity difference

set_visual_myColorBarOce_vel

This section specifies the color bar configuration for Current data.

Parameter	Value	Description
`rangeOce_vel_xy_min`	0.0	Specifies the minimum contour level for the horizontal section of the Current
`rangeOce_vel_xy_max`	0.5	Specifies the maximum contour level for the horizontal section of the Current
`rangeOce_vel_xy_nbar`	26	Specifies the intervals number of contour levels for the horizontal section of the Current
`rangeOce_velDiff_xy_min`	-0.15	Specifies the minimum contour level for the horizontal section of the Current difference
`rangeOce_velDiff_xy_max`	0.15	Specifies the maximum contour level for the horizontal section of the Current difference
`rangeOce_velDiff_xy_nbar`	31	Specifies the intervals number of contour levels for the horizontal section of the Current difference
`rangeOce_vel_z_min`	0.0	Specifies the minimum contour level for the zonal section of the Current
`rangeOce_vel_z_max`	1.0	Specifies the maximum contour level for the zonal section of the Current
`rangeOce_vel_z_nbar`	31	Specifies the intervals number of contour levels for the zonal section of the Current
`rangeOce_velDiff_z_min`	-0.15	Specifies the minimum contour level for the zonal section of the Current difference
`rangeOce_velDiff_z_max`	0.15	Specifies the maximum contour level for the zonal section of the Current difference
`rangeOce_velDiff_z_nbar`	31	Specifies the intervals number of contour levels for the zonal section of the Current difference

set_visual_myColorBarOce_vort

This section specifies the color bar configuration for Relative Vorticity data.

Parameter	Value	Description
`rangeOce_vort_xy_min`	-10.0	Specifies the minimum contour level for the horizontal section of the Relative Vorticity (for conversion, multiply by 10^5)
`rangeOce_vort_xy_max`	10.0	Specifies the maximum contour level for the horizontal section of the Relative Vorticity (for conversion, multiply by 10^5)
`rangeOce_vort_xy_nbar`	31	Specifies the number of intervals for the horizontal section of the Relative Vorticity (for conversion, multiply by 10^5)
`rangeOce_vortDiff_xy_min`	-0.15	Specifies the minimum contour level for the horizontal section of the Relative Vorticity difference (for conversion, multiply by 10^5)
`rangeOce_vortDiff_xy_max`	0.15	Specifies the maximum contour level for the horizontal section of the Relative Vorticity difference (for conversion, multiply by 10^5)
`rangeOce_vortDiff_xy_nbar`	31	Specifies the number of intervals for the horizontal section of the Relative Vorticity difference (for conversion, multiply by 10^5)
`rangeOce_vort_z_min`	-10.0	Specifies the minimum contour level for the zonal section of the Relative Vorticity (for conversion, multiply by 10^5)
`rangeOce_vort_z_max`	10.0	Specifies the maximum contour level for the zonal section of the Relative Vorticity (for conversion, multiply by 10^5)
`rangeOce_vort_z_nbar`	31	Specifies the number of intervals for the zonal section of the Relative Vorticity (for conversion, multiply by 10^5)
`rangeOce_vortDiff_z_min`	-0.15	Specifies the minimum contour level for the zonal section of the Relative Vorticity difference (for conversion, multiply by 10^5)
`rangeOce_vortDiff_z_max`	0.15	Specifies the maximum contour level for the zonal section of the Relative Vorticity difference (for conversion, multiply by 10^5)
`rangeOce_vortDiff_z_nbar`	31	Specifies the number of intervals for the zonal section of the Relative Vorticity difference (for conversion, multiply by 10^5)

set_visual_myColorBarOce_ssh

This section specifies the color bar configuration for Sea Surface Height data.

Parameter	Value	Description
`rangeOce_ssh_min`	-0.55	Specifies the minimum contour level for the Sea Surface Height
`rangeOce_ssh_max`	-0.35	Specifies the maximum contour level for the Sea Surface Height
`rangeOce_ssh_nbar`	41	Specifies the number of intervals for the Sea Surface Height
`rangeOce_sshDiff_min`	0.0	Specifies the minimum contour level for the Sea Surface Height difference
`rangeOce_sshDiff_max`	0.01	Specifies the maximum contour level for the Sea Surface Height difference
`rangeOce_sshDiff_nbar`	31	Specifies the number of intervals for the Sea Surface Height difference

6. Input/Output Management

6.1 Input Datasets for SURF

To execute the SURF-NEMO package, several input datasets are required. These include bathymetry datasets, which provide seafloor elevation; coastline datasets, which define the borders between land and sea; the initial condition dataset, containing initial values for model prognostic variables; and the boundary condition datasets, which contain variables necessary to impose boundary conditions for mass, momentum, and energy flows at the surface and lateral open boundaries of the domain.

Figure 6.1 illustrates the interfaces and external forcings acting on a typical computational domain.

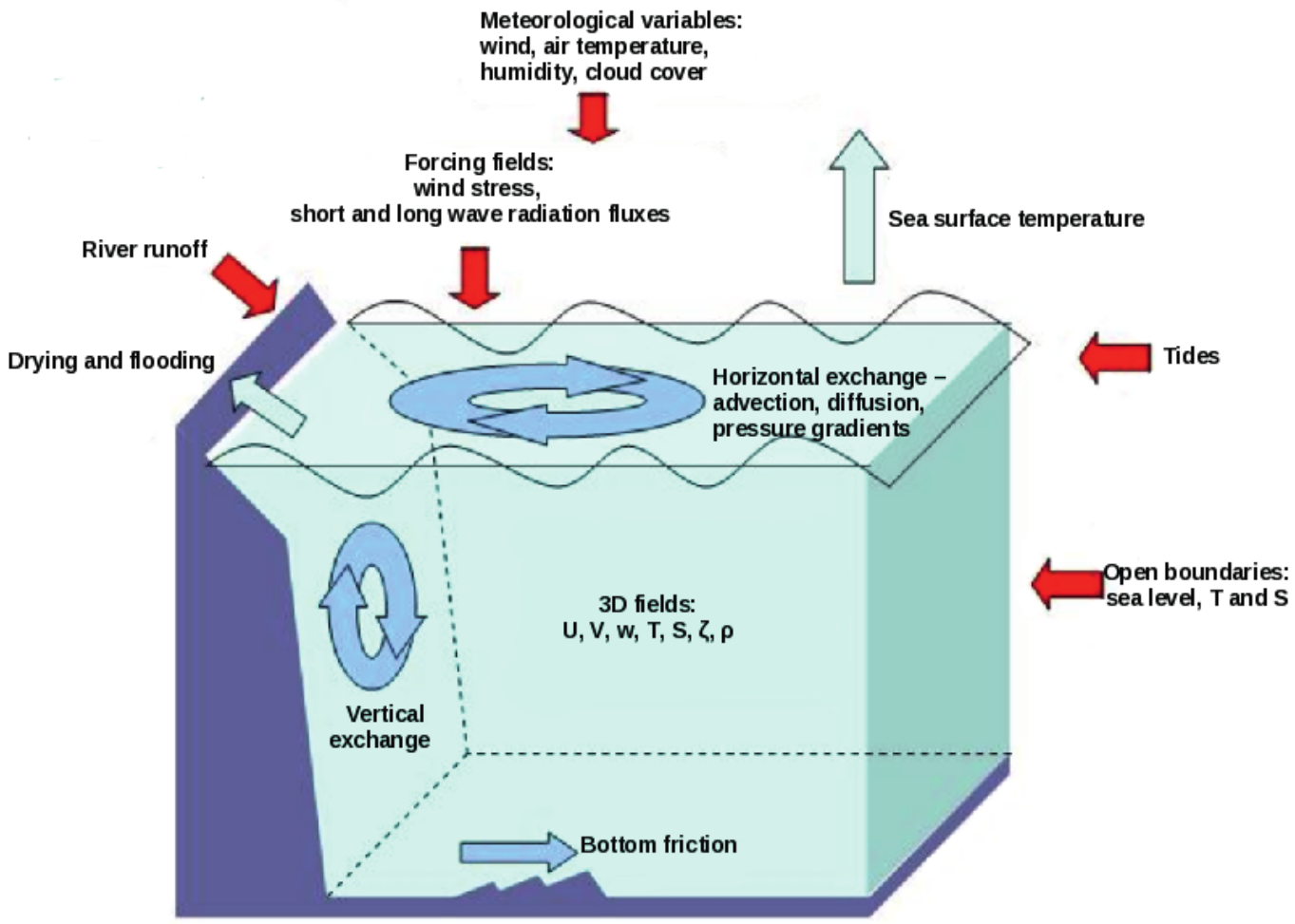


Figure 6.1. Schematized representation of the interface and external forcing acting on a typical computational domain.

6.1.1 Input Datasets

The input datasets for the model are provided in the classic NetCDF format for bathymetry, initial conditions, and lateral boundary conditions. NetCDF is a widely used file format in atmospheric and oceanic research which allows storage of different types of array based data, along with a short data description. The coastline datasets are instead provided in Shapefile format, a digital vector data format for geographic information system (GIS) software.

SURF allows also to use, if needed, two different model type of input data during the execution: analysis data for the spin-up period and forecast data afterward. Users can configure these parameters in the `setParFree.json` file, specifying path, filename, dimensions, variables, and other data characteristics.

Bathymetry Dataset

The bathymetry dataset contains the sea floor elevation and is required for generating the child meshmask file. Users need to configure the necessary parameters in the `set_dataDownlBat` section of the configuration file. The data are distributed on a curvilinear spherical grid (which may be regular or irregular) over a region containing the nested domain.

The bathymetry file contains an elevation variable (in meters) with a specific horizontal resolution. Elevation values are relative to a reference level and can either increase (positive values) or decrease (negative values) with increasing water depth. The coordinate variables (latitude and longitude) can be structured as one- or two-dimensional arrays.

An example CDL (Common Data Language) representation of a bathymetry file is shown in Listing 6.1.1.

```
netcdf bathymetry_filename {
dimensions:
   x = 300;
   y = 200;
variables: \\
   float lon(y,x);
      lon: units = "degrees_east";
   float lat(y,x);
      lat: units = "degrees_north";
   float elevation(y,x);
      elevation: units = "m";
}
```

Listing 6.1.1: Example of a netCDF file for bathymetry.

Users must specify the following logical parameters in the `set_dataDownlBat_fileName` section of the user configuration file:

- `fileBat_lcompression`: Indicates whether the file is compressed (.gzip) or not.
- `fileBat_llonFlip`: Specifies if the longitude coordinate is defined in the range [0:360] or [-180:+180].
- `fileBat_llatInv`: Indicates whether the dataset contains latitude values that decrease towards the poles.
- `fileBat_ldepthIncr`: Defines if the dataset contains seafloor elevation (positive values) that increases with increasing water depth.
- `fileBat_lkeepSrcFull`: Specifies whether the original downloaded file should be deleted after cropping it for the nested domain.

The available input bathymetry dataset included in the SURF package is the **General Bathymetric Chart of the Oceans (GEBCO) 2014**, a publicly accessible dataset that provides global bathymetric coverage with a resolution of 30 arc-seconds. More information can be found on the official GEBCO website here.

Coastline Dataset

The coastline dataset defines the borders between land and sea areas and is stored in shapefile format. It is required for generating the child meshmask. Users must configure the necessary parameters in the `set_dataDownlCoast` section of the configuration file.

The available coastline dataset within the SURF package is the **Global Self-consistent Hierarchical High-resolution Geography (GSHHG)** dataset, produced by the National Oceanic and Atmospheric Administration (NOAA). This dataset includes 20 shapefiles, each containing hierarchically arranged polygons that define shorelines. More information about GSHHG can be found on their official website here.

The GSHHG data are provided at five different resolutions, each stored as a separate shapefile:

• **f** (full): highest resolution with a resolution of xx m

• **h** (high): high resolution with a resolution of xx m

• **i** (intermediate): intermediate resolution with a resolution of xx m

• **l** (low): low resolution with a resolution of xx m

• **c** (coarse): coarsest resolution with a resolution of xx m

Each resolution level contains shorelines organized into four hierarchical categories:

• **L1**: Boundary between land and ocean

• **L2**: Boundary between lake and land

• **L3**: Boundary between islands within lakes and the lakes

• **L4**: Boundary between ponds within islands and the islands

Initial Condition Datasets

To initiate a model run, the initial values for the model's prognostic variables must be provided. These include temperature, salinity, sea surface height, and the zonal and meridional velocity components. Initial condition datasets are typically derived from coarse-grid model outputs. Users must configure the necessary parameters in the `set_dataDownlOceIC` section of the configuration file.

The data can be provided on a curvilinear spherical grid (which may be regular or irregular) using an unstaggered or staggered Arakawa-C grid arrangement within a region that contains the nested domain. The model assumes that all input ocean variables are defined on the same grid.

The coarse-resolution ocean files contain the following variables at a specified horizontal resolution:

• Potential Temperature [C]

• Salinity [PSU]

• Sea Surface Height [m]

• Zonal Velocity [ms^{-1}]

• Meridional Velocity [ms^{-1}]

An example CDL (Common Data Language) representation of an initial condition file is shown in Listing 6.1.2.

```
netcdf fields_filename {
dimensions:
   x = 40 ;
   y = 35 ;
   z = 72 ;
   time = UNLIMITED ; // (1 currently)
variables:
```

```
   float lont(y, x) ;
      lont:units = "degrees_east" ;
   float latt(y, x) ;
      latt:units = "degrees_north" ;
   float deptht(z) ;
      deptht:units = "m" ;
   double time(time) ;
      time_counter:units = "seconds since
                  1970-01-01 00:00:00" ;
   float temperature(time, z, y, x) ;
      temperature:units = "degC" ;
}
```

Listing 6.1.2: Example of a netCDF file for the Initial Condition temperature

To perform the extrapolation (SOL) of ocean fields (see Section 4.3 for more details), the parent land-sea mask file must be provided as an input dataset. The user needs to configure the required parameters in the `set_dataDownlOceICMesh` section of the configuration file.

This file contains information about the coarse-resolution ocean model grids, including the following variables:

• Longitude on TUVF grid points [$degree$]

• Latitude on TUVF grid points [$degree$]

• Depth on TUVF grid points [m]

• Land-sea mask on TUVF grid points [0-1]

• Scale factors on TUVF grid points [m]

An example CDL text representation of this file is shown in Listing 6.1.3.

```
netcdf meshmask_filename {
dimensions :
   x = 677;
   y = 253;
   z = 72;
   t = UNLIMITED; // (7 currently)
variables : \\
   float lon(y,x);
   float lat(y,x);
   float lev(z);
   double time(t);
   byte tmask(t,z,y,x);
   byte umask(t,z,y,x);
   byte vmask(t,z,y,x);
   byte fmask(t,z,y,x);
   float glamt(t,y,x);
   float glamu(t,y,x);
   float glamv(t,y,x);
   float glamf(t,y,x);
   float gphit(t,y,x);
   float gphiu(t,y,x);
   float gphiv(t,y,x);
   float gphif(t,y,x);
   double e1t(t,y,x);
   double e1u(t,y,x);
   double e1v(t,y,x);
   double e1f(t,y,x);
```

```
    double e2t(t,y,x);
    double e2u(t,y,x);
    double e2v(t,y,x);
    double e2f(t,y,x);
    double e3t(t,z,y,x);
    double e3u(t,z,y,x);
    double e3v(t,z,y,x);
    double e3w(t,z,y,x);
}
```

Listing 6.1.3: Example of a netCDF file for the Initial Condition meshmask.

Lateral Open Boundary Condition Datasets

In order to integrate the primitive equations, the NEMO ocean model needs to impose appropriate boundary conditions at the ocean-ocean interface (i.e., the sides of the domain not bounded by land). Lateral open boundary values for the model's prognostic variables must be specified for the entire simulation period. These include fields such as temperature, salinity, sea surface height, and velocity.

Users must configure the necessary parameters in the `set_dataDownlOceBC_preSpinup` and `set_dataDownlOceBC_postSpinup` sections of the configuration file. The data can be distributed on a curvilinear spherical grid (regular or irregular) with an unstaggered or staggered Arakawa-C grid arrangement covering the nested domain. The model assumes that all input ocean variables during the pre- and post-spinup periods are defined on the same grid.

The coarse-resolution ocean files contain the following variables at a given horizontal resolution and temporal frequency:

• Potential Temperature [C]

• Salinity [PSU]

• Sea Surface Height [m]

• Zonal Velocity [ms^{-1}]

• Meridional Velocity [ms^{-1}]

An example CDL (Common Data Language) representation of this file is shown in Listing 6.1.4.

```
netcdf fields_filename {
dimensions :
    x = 677;
    y = 253;
    z = 72;
    t = UNLIMITED; // (7 currently)
variables : \\
    float lont(x);
          lont: units = "degrees_east";
    float latt(y);
          latt: units = "degrees_north";
    float deptht(z);
          deptht: units = "m";
    double time(t);
          time: units = "seconds since
                   1970-01-01 00:00:00";
    float temperature(t,z,y,x);
          temperature: units = "degC";
}
```

Listing 6.1.4: Example of a netCDF file for Open boundary Condition temperature.

To perform the extrapolation (SOL) of ocean fields (see Section 4.3 for more details), the parent land-sea mask file must be provided as an input dataset. Users must configure the necessary parameters in the `set_dataDownlOceBCMesh` section of the configuration file.

This file contains the necessary information about the coarse-resolution ocean model grids and includes the following variables:

• Longitude on TUVF grid points [$degree$]

• Latitude on TUVF grid points [$degree$]

• Depth on TUVF grid points [m]

• Land-sea mask on TUVF grid points [0-1]

• Scale factor on TUVF grid points [m]

An example CDL text representation of this file is shown in Listing 6.1.5.

```
netcdf meshmask_filename {
dimensions :
   x = 677;
   y = 253;
   z = 72;
   t = UNLIMITED; // (7 currently)
variables : \\
   float lon(y,x);
   float lat(y,x);
   float lev(z);
   double time(t);
   byte tmask(t,z,y,x);
   byte umask(t,z,y,x);
   byte vmask(t,z,y,x);
   byte fmask(t,z,y,x);
   float glamt(t,y,x);
   float glamu(t,y,x);
   float glamv(t,y,x);
   float glamf(t,y,x);
   float gphit(t,y,x);
   float gphiu(t,y,x);
   float gphiv(t,y,x);
   float gphif(t,y,x);
   double e1t(t,y,x);
   double e1u(t,y,x);
   double e1v(t,y,x);
   double e1f(t,y,x);
   double e2t(t,y,x);
   double e2u(t,y,x);
   double e2v(t,y,x);
   double e2f(t,y,x);
   double e3t(t,z,y,x);
   double e3u(t,z,y,x);
   double e3v(t,z,y,x);
   double e3w(t,z,y,x);
}
```

Listing 6.1.5: Example of a netCDF file for the Initial Condition meshmask.

Atmospheric Forcing Datasets

To integrate the primitive equations, the NEMO ocean model needs to impose appropriate boundary conditions for the exchange of mass, momentum, and energy at the atmosphere-ocean interface. The following six fields must be provided for the integration domain:

1. Zonal components of surface ocean stress,
2. Meridional components of surface ocean stress,
3. Solar heat fluxes (Qsr),
4. Non-solar heat fluxes (Qns),
5. Water flux exchanged with the atmosphere (E-P), representing evaporation minus precipitation.

Additionally, an optional field can be provided:

1. Atmospheric pressure at the ocean surface (pa).

The NEMO model offers different methods for providing these fields, controlled by namelist variables (refer to the NEMO Manual). In the SURF platform, the choice of atmospheric forcing is determined by setting the `sbc_iformulat` parameter in the user configuration file:

- `sbc_iformulat=0` for the MFS bulk formulae,
- `sbc_iformulat=1` for the Flux formulation,
- `sbc_iformulat=2` for the CORE bulk formulae.

The atmospheric data are distributed on a regular unstaggered grid covering the nested domain. The model assumes that the input atmospheric variables for both the pre- and post-spinup periods are defined on the same mesh, though different meshes are allowed for different variables. The user must configure the required parameters in the `set_dataDownlAtm_preSpinup` and `set_dataDownlAtm_postSpinup` sections of the configuration file.

MFS BULK FORMULAE

The **MFS Bulk Formulae** are selected by setting `sbc_iformulat=0` in the user configuration file.

The atmospheric forcing files contain the following variables at a specific horizontal resolution and temporal frequency:

- 10 m Zonal Wind Component [ms^{-1}],
- 10 m Meridional Wind Component [ms^{-1}],
- 2 m Air Temperature [K],
- 2 m Dew Point Temperature [K],
- Mean Sea Level Pressure [Pa],
- Total Cloud Cover [%],
- Total Precipitation [m].

An example CDL text representation for an atmospheric forcing file with a 3-hour temporal frequency is shown in Listing 6.1.8.

```
netcdf atmFields_filename {
dimensions :
   lon = 245;
   lat = 73;
   time = UNLIMITED; // (8 currently)
variables : \\
   float lon(lon);
         lon: units = "degrees_east";
   float lat(lat);
         lat: units = "degrees_north";
   float time(time);
         time: units = "seconds since
                  1970-01-01 00:00:00";
   float T2M(time,lat,lon);
         T2M: units = "K";
}
```

Listing 6.1.8: Example of a netCDF file for the Atmospheric Forcing temperature.

CORE BULK FORMULAE

The **CORE Bulk Formulae** are selected by setting `sbc_iformulat=2` in the user configuration file.

The atmospheric forcing files contain the following variables at a specified horizontal resolution and temporal frequency:

• 10 m Zonal Wind Component [ms^{-1}],

• 10 m Meridional Wind Component [ms^{-1}],

• 2 m Air Temperature [K],

• 2 m Specific Humidity [%],

• Incoming Long-Wave Radiation [Wm^{-2}],

• Incoming Short-Wave Radiation [Wm^{-2}],

• Total Precipitation (Liquid + Solid) [$Kgm^{-2}s^{-1}$],

• Solid Precipitation [$Kgm^{-2}s^{-1}$].

An example CDL text representation for the atmospheric forcing file with a 3-hour temporal frequency is shown in Listing 6.1.9.

```
netcdf atmFields_filename {
dimensions :
   lon = 245;
   lat = 73;
   time = UNLIMITED; // (8 currently)
variables : \\
   float lon(lon);
         lon: units = "degrees_east";
   float lat(lat);
         lat: units = "degrees_north";
   float time(time);
         time: units = "seconds since
                  1970-01-01 00:00:00";
   float T2M(time,lat,lon);
```

```
          T2M: units = "K";
}
```

Listing 6.1.9: Example of a netCDF file for the Atmospheric Forcing temperature.

FLUX FORMULATION

The **Flux Formulation** is selected by setting `sbc_iformulat=1` in the user configuration file.

The atmospheric forcing files contain the following variables at a specific horizontal resolution and temporal frequency:

• Zonal Wind Stress [0 - 1],

• Meridional Wind Stress [0 - 1],

• Total Heat Flux [0 - 1],

• Solar Radiation Penetration [0 - 1],

• Mass Flux Exchanged [0 - 1],

• Surface Temperature [0 - 1],

• Surface Salinity [0 - 1].

An example CDL text representation for the atmospheric forcing file with a 3-hour temporal frequency is shown in Listing 6.1.10.

```
netcdf atmFields_filename {
dimensions :
   lon = 245;
   lat = 73;
   time = UNLIMITED; // (8 currently)
variables : \\
   float lon(lon);
         lon: units = "degrees_east";
   float lat(lat);
         lat: units = "degrees_north";
   float time(time);
         time: units = "seconds since
                  1970-01-01 00:00:00";
   float T2M(time,lat,lon);
         T2M: units = "K";
}
```

Listing 6.1.10: Example of a netCDF file for the Atmospheric Forcing temperature.

To perform the extrapolation (SOL) of atmospheric fields (see Section 4.3 for more details), the atmospheric meshmask file must be provided as an input dataset. Users need to configure the required parameters in the `set_dataDownlAtmMesh` section of the configuration file.

The atmospheric meshmask file contains the land-sea mask variable [0-1].

An example of CDL text representation of the atmospheric land-sea mask file is shown in Listing 6.1.11. The time dimension and coordinate variables can also be omitted if necessary.

```
netcdf atmFields_filename {
dimensions :
   lon = 245;
```

```
   lat = 73;
   time = UNLIMITED; // (1 currently)
variables : \\
   float lon(lon);
         lon: units = "degrees_east";
   float lat(lat);
         lat: units = "degrees_north";
   float time(time);
         time: units = "seconds since
                  1970-01-01 00:00:00";
   float LSM(time,lat,lon);
         LSM: units = "0-1";
}
```

Listing 6.1.11: Example of a netCDF file for the Atmospheric Forcing meshmask.

6.2 Output Model Datasets

The model generates several output files, including the meshmask file, restart file, and the output files for the T, U, V, and W grids, along with log files that record model execution details. These files are produced in the NetCDF (Network Common Data Form) format, which is widely used for storing and exchanging scientific data, particularly in climate and ocean modeling.

In the current version of SURF, the default I/O operations are managed by NEMO's diawri module, which handles file writing. The I/O server (for parallel data writing) is not yet included. The output files contain time-averaged values over specified intervals. The frequency of output writing is controlled by the nn_write parameter, set within the user configuration file (typically hourly means).

6.2.1 **Meshmask File**

The meshmask file (`mesh_mask.nc`) contains all the information of the child model grids, including:

• Longitude on T-U-V-F grid points [$degree$]

• Latitude on T-U-V-F grid points [$degree$]

• Depth on T-U-V-F grid points [m]

• Land-sea mask on T-U-V-F grid points [0-1]

• Scale factors on T-U-V-F grid points [m]

An example CDL (Common Data Language) representation of this file is shown in Listing 6.2.1.

```
netcdf mesh_mask {
dimensions:
    x = 94 ;
    y = 79 ;
    z = 120 ;
    t = UNLIMITED ; // (1 currently)
variables:
    float nav_lon(y, x) ;
    float nav_lat(y, x) ;
    float nav_lev(z) ;
    double time_counter(t) ;
    byte tmask(t, z, y, x) ;
    byte umask(t, z, y, x) ;
    byte vmask(t, z, y, x) ;
    byte fmask(t, z, y, x) ;
    byte tmaskutil(t, y, x) ;
    byte umaskutil(t, y, x) ;
    byte vmaskutil(t, y, x) ;
    byte fmaskutil(t, y, x) ;
    float glamt(t, y, x) ;
    float glamu(t, y, x) ;
    float glamv(t, y, x) ;
    float glamf(t, y, x) ;
    float gphit(t, y, x) ;
    float gphiu(t, y, x) ;
    float gphiv(t, y, x) ;
    float gphif(t, y, x) ;
    double e1t(t, y, x) ;
    double e1u(t, y, x) ;
    double e1v(t, y, x) ;
    double e1f(t, y, x) ;
```

```
    double e2t(t, y, x) ;
    double e2u(t, y, x) ;
    double e2v(t, y, x) ;
    double e2f(t, y, x) ;
    double ff(t, y, x) ;
    short mbathy(t, y, x) ;
    short misf(t, y, x) ;
    float isfdraft(t, y, x) ;
    double e3t_0(t, z, y, x) ;
    double e3u_0(t, z, y, x) ;
    double e3v_0(t, z, y, x) ;
    double e3w_0(t, z, y, x) ;
    float gdept_0(t, z, y, x) ;
    float gdepu(t, z, y, x) ;
    float gdepv(t, z, y, x) ;
    float gdepw_0(t, z, y, x) ;
    double gdept_1d(t, z) ;
    double gdepw_1d(t, z) ;
    double e3t_1d(t, z) ;
    double e3w_1d(t, z) ;

// global attributes:
        :file_name = "mesh_mask.nc" ;
}
```

Listing 6.2.1: CDL example for the meshmask datasets.

6.2.2 **T-Grid Ocean File**

The output T-Grid file (`SURF_1h_YYYYMMDD0_YYYYMMDD1_grid_T`) contains hourly fields defined on the Arakawa-T grid within the chid nested domain. This file includes:

• Temperature [C]

• Salinity [PSU]

• Sea Surface Temperature [C]

• Sea Surface Salinity [PSU]

• Sea Surface Height [m]

• Net Upward Water Flux [$Kg/m^2/s$]

• Concentration/Dilution Water Flux [$Kg/m^2/s$]

• Surface Salt Flux [$Kg/m^2/s$]

• Net Downward Heat Flux [W/m^2]

• Shortwave Radiation [W/m^2]

• Turbocline Depth [m]

• Mixed Layer Depth 0.01 [m]

• Ice Fraction [$0; 1$]

• Wind Speed at 10m [m/s]

• Surface Heat Flux: Damping [W/m^2]

• Surface Water Flux: Damping [$Kg/m^2/s$]

• Surface Salt Flux: Damping [$Kg/m^2/s$]

• Bowl Index [W point]

An example CDL representation of this file is shown in Listing 6.2.2.

```
netcdf SURF_1h_20141005_20141005_grid_T {
dimensions:
    x = 94 ;
    y = 79 ;
    deptht = 120 ;
    time_counter = UNLIMITED ; // (24 currently)
    bnds = 2 ;
variables:
    float nav_lon(y, x) ;
        nav_lon:standard_name = "longitude" ;
        nav_lon:long_name = "longitude" ;
        nav_lon:units = "degrees_east" ;
        nav_lon:_CoordinateAxisType = "Lon" ;
    float nav_lat(y, x) ;
        nav_lat:standard_name = "latitude" ;
        nav_lat:long_name = "latitude" ;
        nav_lat:units = "degrees_north" ;
        nav_lat:_CoordinateAxisType = "Lat" ;
    float deptht(deptht) ;
        deptht:long_name = "Vertical T levels" ;
```

```
        deptht:units = "m" ;
        deptht:positive = "down" ;
        deptht:axis = "Z" ;
        deptht:title = "deptht" ;
    double time_counter(time_counter) ;
        time_counter:standard_name = "time" ;
        time_counter:long_name = "Time axis" ;
        time_counter:bounds = "time_counter_bnds" ;
        time_counter:units = "seconds since 2014-10-05 00:00:00" ;
        time_counter:calendar = "standard" ;
        time_counter:axis = "T" ;
    double time_counter_bnds(time_counter, bnds) ;
    float votemper(time_counter, deptht, y, x) ;
        votemper:standard_name = "Temperature" ;
        votemper:long_name = "Temperature" ;
        votemper:units = "C" ;
        votemper:coordinates = "nav_lat nav_lon" ;
        votemper:_FillValue = 9.96921e+36f ;
        votemper:missing_value = 9.96921e+36f ;
        votemper:online_operation = "ave(x)" ;
        votemper:interval_operation = 150.f ;
        votemper:interval_write = 3600.f ;
    float vosaline(time_counter, deptht, y, x) ;
        vosaline:standard_name = "Salinity" ;
        vosaline:long_name = "Salinity" ;
        vosaline:units = "PSU" ;
        vosaline:coordinates = "nav_lat nav_lon" ;
        vosaline:_FillValue = 9.96921e+36f ;
        vosaline:missing_value = 9.96921e+36f ;
        vosaline:online_operation = "ave(x)" ;
        vosaline:interval_operation = 150.f ;
        vosaline:interval_write = 3600.f ;
    float sossheig(time_counter, y, x) ;
        sossheig:standard_name = "Sea Surface Height" ;
        sossheig:long_name = "Sea Surface Height" ;
        sossheig:units = "m" ;
        sossheig:coordinates = "nav_lat nav_lon" ;
        sossheig:_FillValue = 9.96921e+36f ;
        sossheig:missing_value = 9.96921e+36f ;
        sossheig:online_operation = "ave(x)" ;
        sossheig:interval_operation = 150.f ;
        sossheig:interval_write = 3600.f ;

// global attributes:
        :Conventions = "CF-1.1" ;
        :production = "An IPSL model" ;
        :file_name = "SURF_1h_20141005_20141005_grid_T.nc" ;
}
```

Listing 6.2.2: CDL example for the T-Grid output file.

6.2.3 **U-Grid Ocean File**

The output U-Grid file (`SURF_1h_YYYYMMDD0_YYYYMMDD1_grid_U`) contains hourly fields defined on the Arakawa-U grid within the chid nested domain. This file includes:

• Zonal Current [m/s]

• Wind Stress along the zonal axis [N/m^2]

An example CDL representation of this file is shown in Listing 6.2.3.

```
netcdf SURF_1h_20141005_20141005_grid_U {
dimensions:
    x = 94 ;
    y = 79 ;
    depthu = 120 ;
    time_counter = UNLIMITED ; // (24 currently)
    bnds = 2 ;
variables:
    float nav_lon(y, x) ;
        nav_lon:standard_name = "longitude" ;
        nav_lon:long_name = "longitude" ;
        nav_lon:units = "degrees_east" ;
        nav_lon:_CoordinateAxisType = "Lon" ;
    float nav_lat(y, x) ;
        nav_lat:standard_name = "latitude" ;
        nav_lat:long_name = "latitude" ;
        nav_lat:units = "degrees_north" ;
        nav_lat:_CoordinateAxisType = "Lat" ;
    float depthu(depthu) ;
        depthu:long_name = "Vertical U levels" ;
        depthu:units = "m" ;
        depthu:positive = "down" ;
        depthu:axis = "Z" ;
        depthu:title = "depthu" ;
    double time_counter(time_counter) ;
        time_counter:standard_name = "time" ;
        time_counter:long_name = "Time axis" ;
        time_counter:bounds = "time_counter_bnds" ;
        time_counter:units = "seconds since 2014-10-05 00:00:00" ;
        time_counter:calendar = "standard" ;
        time_counter:axis = "T" ;
    double time_counter_bnds(time_counter, bnds) ;
    float vozocrtx(time_counter, depthu, y, x) ;
        vozocrtx:standard_name = "Zonal Current" ;
        vozocrtx:long_name = "Zonal Current" ;
        vozocrtx:units = "m/s" ;
        vozocrtx:coordinates = "nav_lat nav_lon" ;
        vozocrtx:_FillValue = 9.96921e+36f ;
        vozocrtx:missing_value = 9.96921e+36f ;
        vozocrtx:online_operation = "ave(x)" ;
        vozocrtx:interval_operation = 150.f ;
        vozocrtx:interval_write = 3600.f ;

// global attributes:
        :Conventions = "CF-1.1" ;
        :production = "An IPSL model" ;
        :file_name = "SURF_1h_20141005_20141005_grid_U.nc" ;
}
```

Listing 6.2.3: CDL example for the U-Grid output file.

6.2.4 **V-Grid Ocean File**

The output V-Grid file (`SURF_1h_YYYYMMDD0_YYYYMMDD1_grid_V`) contains hourly fields defined on the Arakawa-V grid within the chid nested domain. This file includes:

• Meridional Current [m/s]

• Wind Stress along the meridional axis [N/m^2]

An example CDL representation of this file is shown in Listing 6.2.4.

```
netcdf SURF_1h_20141005_20141005_grid_V {
dimensions:
    x = 94 ;
    y = 79 ;
    depthv = 120 ;
    time_counter = UNLIMITED ; // (24 currently)
    bnds = 2 ;
variables:
    float nav_lon(y, x) ;
        nav_lon:standard_name = "longitude" ;
        nav_lon:long_name = "longitude" ;
        nav_lon:units = "degrees_east" ;
        nav_lon:_CoordinateAxisType = "Lon" ;
    float nav_lat(y, x) ;
        nav_lat:standard_name = "latitude" ;
        nav_lat:long_name = "latitude" ;
        nav_lat:units = "degrees_north" ;
        nav_lat:_CoordinateAxisType = "Lat" ;
    float depthv(depthv) ;
        depthv:long_name = "Vertical V levels" ;
        depthv:units = "m" ;
        depthv:positive = "down" ;
        depthv:axis = "Z" ;
        depthv:title = "depthv" ;
    double time_counter(time_counter) ;
        time_counter:standard_name = "time" ;
        time_counter:long_name = "Time axis" ;
        time_counter:bounds = "time_counter_bnds" ;
        time_counter:units = "seconds since 2014-10-05 00:00:00" ;
        time_counter:calendar = "standard" ;
        time_counter:axis = "T" ;
    double time_counter_bnds(time_counter, bnds) ;
    float vomecrty(time_counter, depthv, y, x) ;
        vomecrty:standard_name = "Meridional Current" ;
        vomecrty:long_name = "Meridional Current" ;
        vomecrty:units = "m/s" ;
        vomecrty:coordinates = "nav_lat nav_lon" ;
        vomecrty:_FillValue = 9.96921e+36f ;
        vomecrty:missing_value = 9.96921e+36f ;
        vomecrty:online_operation = "ave(x)" ;
        vomecrty:interval_operation = 150.f ;
        vomecrty:interval_write = 3600.f ;

// global attributes:
        :Conventions = "CF-1.1" ;
        :production = "An IPSL model" ;
        :file_name = "SURF_1h_20141005_20141005_grid_V.nc" ;
}
```

Listing 6.2.4: CDL example for the V-Grid output file.

6.2.5 **W-Grid Ocean File**

The output W-Grid file (`SURF_1h_YYYYMMDD0_YYYYMMDD1_grid_W`) contains hourly fields defined on the Arakawa-W grid within the chid nested domain. This file includes:

- Vertical velocity [m/s],
- Vertical Eddy Viscosity [m^2/s],
- Vertical Eddy Diffusivity [m^2/s].

An example CDL representation of this file is shown in Listing 6.2.5.

```
netcdf fields_filename {
netcdf SURF_1h_20141005_20141005_grid_W.nc {
dimensions:
   x = 94 ;
   y = 79 ;
   depthw = 120 ;
   time_counter = UNLIMITED ; // (24 currently)
   bnds = 2 ;
variables:
   float nav_lat(y, x);
      nav_lat:standard_name = "latitude";
      nav_lat:units = "degrees_north";
      nav_lat:long_name = "Latitude";
      nav_lat:nav_model = "Default grid";
   float nav_lon(y, x);
      nav_lon:standard_name = "longitude";
      nav_lon:units = "degrees_east";
      nav_lon:long_name = "Longitude";
      nav_lon:nav_model = "Default grid";
   float depthw(depthw);
      depthw:axis = "Z";
      depthw:standard_name = "model_level_number";
      depthw:units = "m";
      depthw:positive = "down";
      depthw:valid_min = 0.0f; // float
      depthw:valid_max = 2622.5078f; // float
      depthw:title = "depthw";
      depthw:long_name = "Vertical W levels";
   double time_counter(time_counter);
      time_counter:standard_name = "time" ;
      time_counter:long_name = "Time axis" ;
      time_counter:bounds = "time_counter_bnds" ;
      time_counter:units = "seconds since 2014-10-05 00:00:00" ;
      time_counter:calendar = "standard" ;
      time_counter:axis = "T" ;
   double time_counter_bnds(time_counter, tbnds);
   float votkeavm(time_counter, depthw, y, x);
      votkeavm:units = "m2/s";
      votkeavm:standard_name = "Vertical Eddy Viscosity";
      votkeavm:_FillValue = 9.96921E36f; // float
      votkeavm:long_name = "Vertical Eddy Viscosity";
      votkeavm:online_operation = "ave(x)";
      votkeavm:interval_operation = 150.0f; // float
      votkeavm:interval_write = 3600.0f; // float
```

```
      votkeavm:coordinates = "time_counter depthw nav_lat nav_lon";
   float votkeavt(time_counter, depthw, y, x);
      votkeavt:units = "m2/s";
      votkeavt:standard_name = "Vertical Eddy Diffusivity";
      votkeavt:_FillValue = 9.96921E36f; // float
      votkeavt:long_name = "Vertical Eddy Diffusivity";
      votkeavt:online_operation = "ave(x)";
      votkeavt:interval_operation = 150.0f; // float
      votkeavt:interval_write = 3600.0f; // float
      votkeavt:coordinates = "time_counter depthw nav_lat nav_lon";
   float vovecrtz(time_counter, depthw, y, x);
      vovecrtz:units = "m/s";
      vovecrtz:standard_name = "Vertical Velocity";
      vovecrtz:_FillValue = 9.96921E36f; // float
      vovecrtz:long_name = "Vertical Velocity";
      vovecrtz:online_operation = "ave(x)";
      vovecrtz:interval_operation = 150.0f; // float
      vovecrtz:interval_write = 3600.0f; // float
      vovecrtz:coordinates = "time_counter depthw nav_lat nav_lon";

  // global attributes:
         :Conventions = "CF-1.1";
         :production = "An IPSL model";
         :file_name = "SURF_1h_20141005_20141005_grid_W.nc";
}
```

Listing 6.2.5: CDL example for the W-Grid output file.

6.2.6 **Restart File**

The restart file (`SURF_restart_YYYYMMDD.nc`) contains the model state variables at two consecutive time step, allowing NEMO model to continue the simulation from that point onward. This file includes two consecutive time levels (referred to as 'before' and 'after') of the following variables:

• Temperature

• Salinity

• Velocity horizontal components

• Sea Surface Height

• ...

An example CDL representation of this file is shown in Listing 6.2.6

```
netcdf SURF_restart_20141005 {
dimensions:
    x = 94 ;
    y = 79 ;
    z = 120 ;
    t = UNLIMITED ; // (1 currently)
variables:
    float nav_lon(y, x) ;
    float nav_lat(y, x) ;
    float nav_lev(z) ;
    double time_counter(t) ;
    double kt ;
    double ndastp ;
```

```
    double adatrj ;
    double utau_b(t, y, x) ;
    double vtau_b(t, y, x) ;
    double qns_b(t, y, x) ;
    double emp_b(t, y, x) ;
    double sfx_b(t, y, x) ;
    double sbc_hc_b(t, y, x) ;
    double sbc_sc_b(t, y, x) ;
    double qsr_hc_b(t, z, y, x) ;
    double fraqsr_1lev(t, y, x) ;
    double rdt ;
    double rdttra1 ;
    double ub(t, z, y, x) ;
    double vb(t, z, y, x) ;
    double tb(t, z, y, x) ;
    double sb(t, z, y, x) ;
    double rotb(t, z, y, x) ;
    double hdivb(t, z, y, x) ;
    double sshb(t, y, x) ;
    double un(t, z, y, x) ;
    double vn(t, z, y, x) ;
    double tn(t, z, y, x) ;
    double sn(t, z, y, x) ;
    double rotn(t, z, y, x) ;
    double hdivn(t, z, y, x) ;
    double sshn(t, y, x) ;
    double rhop(t, z, y, x) ;

// global attributes:
        :file_name = "SURF_00000576_restart.nc" ;
}
```

Listing 6.2.6: CDL example for the restart datasets.

7. APPENDIX

7.1 Reference Configuration

7.1.1 **Reference Configuration for NEMO**

Within the SURF source package, several input model parameters for NEMO are fixed and predefined in the file `setParFix.ncl`. The fixed parameters include:

- **Tracer Advection Scheme:** The Monotonic Upstream Scheme for Conservation Laws (MUSCL) was used for tracer advection. This scheme provides high accuracy while maintaining monotonicity in the numerical solutions, preventing spurious oscillations.
- **Momentum Advection Scheme:** The Energy and Enstrophy Conservative (EEN) scheme was applied for momentum advection. This method, as described by Arakawa and Lamb (1981) and Barnier et al. (2006), conserves both energy and enstrophy, ensuring a physically realistic representation of momentum transport.
- **Lateral Close Boundary Conditions:** No-slip conditions were implemented on all closed lateral boundaries, ensuring that there is no unrealistic velocity at the boundaries of the computational domain.
- **Bottom Friction:** The bottom friction was parameterized using a quadratic function, which accurately simulates frictional forces at the ocean floor.

7.2 Scratch Partition and its directory structures

As mentioned in Chapter 6.1.1, the VM-SURF is divided into two partitions: the disk `/dev/sda` "mounted" at the root directory `/`, and the disk `/dev/sdb` "mounted" at `/scratch`. The scratch partition contains all the SURF packages and follows the directory structure illustrated in Figure B.1. The latest version of the release is structured as follows:

- The directory `surf_install/` contains the utilities necessary to manage all the operations of creation and installation of each package of the SURF platform.
- The directory `surf_datasets/` contains a list of static input datasets needed to run the SURF_NEMO package. With 'static' we mean here datasets which do not depend on the selected simulation period; i.e. bathymetry, coastline, parent meshmask, weight for remapping, meshmask and bathymetry remapped on the child grid.
- The directory `surf_nemo/` contains the sources code of the SURF-NEMO package.
- The directory `experiments/` contains all the experiments you have executed.

Below is a detailed description of the contents of these directories.

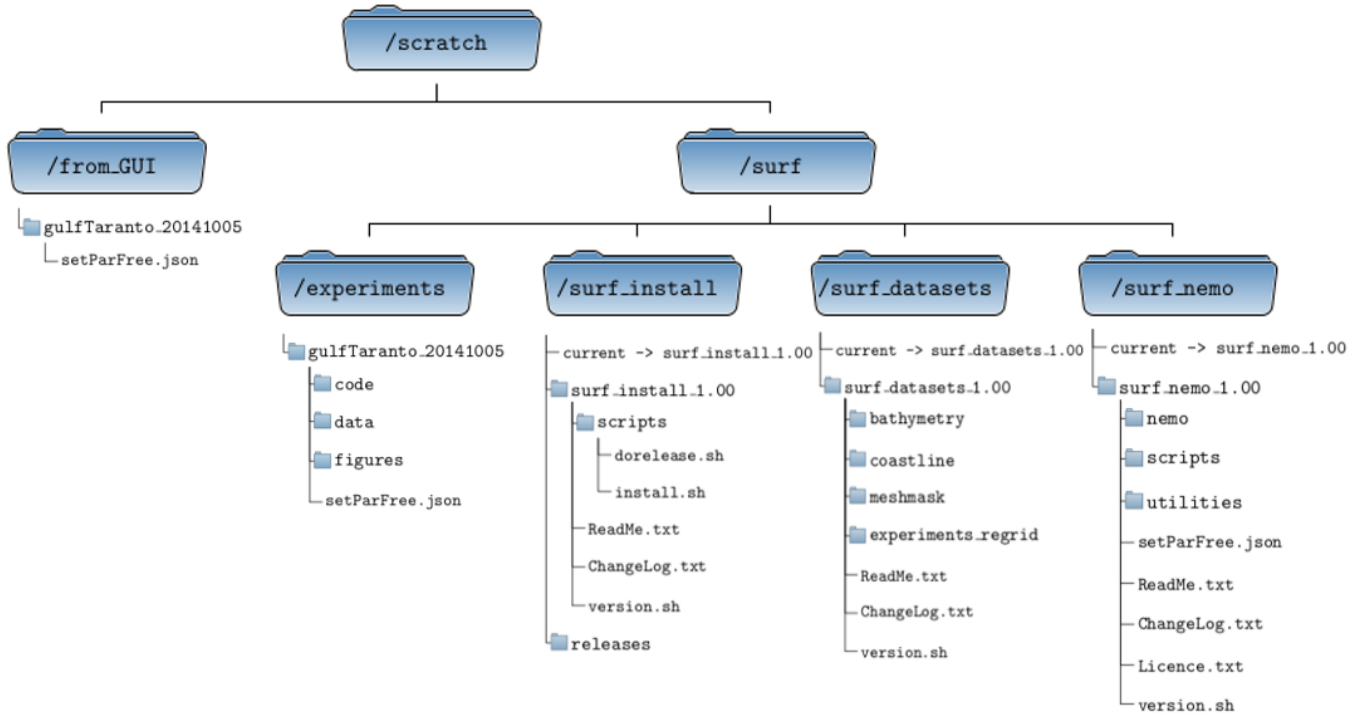


Fig. B.1 SURF package directories tree.

7.2.1 The surf_install Directory Structure

The **SURF-INSTALL** package is pre-installed in the SURF platform and is located in the directory `/scratch/surf/surf_install`. The folder `surf_install_1.00` has the following structure:

- The folder `scripts/` contains bash scripts to install (`install.sh`) and create (`dorelease.sh`) package releases.
- The `ChangeLog.txt` file documents all notable changes to the 'surf_install' package.
- The `ReadMe.txt` file provides an overview of the contents of the 'surf_install' package.
- The `vertion.sh` bash file displays the version number of the 'surf_install' package, shown in the upper-right corner of the VM desktop.

7.2.2 The surf_nemo Directory Structure

Once installed (see Section 6.2), the SURF-NEMO package is located in the directory `/scratch/surf/surf_nemo/`. The folder `surf_nemo_1.00` has the following structure:

- The folder `nemo/` contains the source code of the NEMO ocean model (v3.6).
- The folder `scripts/` holds pre- and post-processing scripts required to run the relocatable SURF model.
- The folder `utilities/` contains utility functions for specific pre-/post-processing tasks.
- The `setParFree.json` file is a template configuration file for case study experiments.
- The `ChangeLog.txt` file documents all notable changes to the 'surf_nemo' package.
- The `ReadMe.txt` file describes the contents of the 'surf_nemo' package.
- The `Licence.txt` file contains licensing information for the product.
- The `vertion.sh` bash file displays the version number of the 'surf_nemo' package, shown in the upper-right corner of the VM desktop.

7.2.3 The surf_datasets Directory Structure

Once installed (see Section 6.2), the SURF-DATASETS package is located in the directory `/scratch/surf/surf_datasets`. The folder `surf_datasets_1.00` is structured as follows:

- The `bathymetry/` folder contains the GEBCO Bathymetric datasets with 30 arc-second resolution.
- The `coastline/` folder includes GSHHG coastline datasets, provided by NOAA's National Geophysical Data Center (NGDC).
- The `meshmask/` folder stores meshmask files for the parent ocean model and atmospheric sources.
- The `experiments_regrid/` folder holds weight files for remapping ocean and atmospheric input data, along with meshmask and bathymetry files remapped onto the child grid. This is used when executing the SURF platform operationally.
- The `ChangeLog.txt` file documents all notable changes to the 'surf_datasets' package.
- The `ReadMe.txt` file describes the contents of the 'surf_datasets' package.
- The `vertion.sh` bash file shows the version number of the 'surf_datasets' package, displayed in the upper-right corner of the VM desktop.

7.2.4 **The experiments directory**

Once an experiment (e.g., `expID`) is executed, it is located in the directory `/scratch/surf/experiments/`. The folder `expID` has the following structure:

- A copy of the configuration file `setParFree.json` (from the directory `surf/from_GUI/expID/`).
- The `code/` folder contains a copy of the source code (from `surf/surf_nemo/current/`) used for the simulation.
- The `data/` folder stores all data used in the experiment:
  - Source input data (`data/indata/`),
  - Extrapolated data (`data/extrapoldata/`),
  - Regridded data (`data/regriddata/`),
  - Output data (`data/outdata/`).

  The input datasets are downloaded from local or web repositories for the selected simulation period.
- The `figure/` folder contains plots of:
  - Source input data (`figure/indata/`),
  - Extrapolated data (`figure/extrapoldata/`),
  - Regridded data (`figure/regriddata/`),
  - Output data (`figure/outdata/`),
  - Comparisons between child and parent (coarse resolution) data.

7.3 Linux Root Partition and the installed packages

As discussed in Chapter 6.1.1, the VM Surf is divided into two partitions: the disk `/dev/sda`, "mounted" at the root directory `/`, and the disk `/dev/sdb`, "mounted" at the `/scratch` directory. The root partition contains the Debian GNU/Linux 8.7 (jessie) operating system.

7.3.1 **Debian partition**

The operating system installed on this Virtual Machine is Debian, a free operating system (OS) that uses the Linux kernel. Debian comes with over 59,000 packages, precompiled software bundled for easy installation. Currently, Debian 8.7 (jessie) is installed in the VM. You can find the full list of available Debian packages here.

7.3.2 **Installed packages**

CDO - (v1.8.1)

The Climate Data Operator (CDO) is a collection of operators for standard processing of climate and forecast model data. It includes tools for statistical operations, arithmetic functions, data selection, subsampling, and spatial interpolation. CDO supports GRIB and NetCDF formats in one package.

curl - (v7.38.0)

curl is a free and open-source command-line tool used to transfer files and data from or to a server using protocols such as FTP, HTTP, HTTPS, SCP, SFTP, and SMB.

HDF5 - (v1.8.18)

The Hierarchical Data Format (HDF5) is a data model, library, and file format designed for storing and managing complex data. It supports an unlimited variety of datatypes, and is designed for flexible and efficient I/O and for high volume and complex data. HDF5 is portable and is extensible, allowing applications to evolve in their use of HDF5. The HDF5 Technology suite includes tools and applications for managing, manipulating, viewing, and analyzing data in the HDF5 format.

Julia - (v1.1.0)

Julia is a high-level, high-performance programming language, designed for numerical analysis and computational science. Julia is widely used for its speed and flexibility in handling scientific computing tasks.

MPICH2 - (v3.2)

MPICH, formerly known as MPICH2, is a freely available, high performance and widely portable implementation of the Message Passing Interface (MPI) standard. It efficiently supports different computation and communication platforms including commodity clusters, SMPs, massively parallel systems, and high-speed networks.

NCL - (v6.4.0)

The NCAR Command Language (NCL) is a free, interpreted language designed specifically for processing and visualizing scientific data.

Ncview - (v2.1.7)

Ncview is a visual browser for NetCDF files. It allows users to quickly view NetCDF data, create simple movies of the data, change color maps, and analyze various dimensions of the dataset.

NetCDF - (4.4.1.1)

The Network Common Data Form (NetCDF) is a set of software libraries and machine-independent data formats designed for the storage, access, and sharing of array-oriented scientific data. It is also a community standard for sharing scientific data.

Szip - (v2.1)

Szip is a compression library that provides lossless compression of scientific data, particularly in HDF5 datasets.

UDUNITS - (v2.2.24)

The UDUNITS package supports units of physical quantities, allowing arithmetic manipulation and conversion of units. It includes a command-line utility and a unit database in XML format.

zlib - (v1.2.11)

The zlib compression library provides in-memory compression and decompression functions, including integrity checks for uncompressed data.

7.4 Release Notes

This section contains documentation of all significant updates to the `surf_nemo` package, detailing new features, improvements, and changes made in each release.

7.4.1 Version 1.00 "What's New"

Release Date: 2019-03-01

This is the first release of surf_nemo package

7.5 Bibliography

- Engerdahl H (1995), *Use of the flow relaxation scheme in a three dimensional baroclinic ocean model with realistic topography* , Tellus 47A:365–382
- Oddo P, Pinardi N (2008), *Lateral open boundary conditions for nested limited area models: A scale selective approach* , Ocean Model 20:134–156
- Shapiro R (1970), *Smoothing, filtering, and boundary effects* , Rev Geophys Space Phys 8:359–387
- Shapiro R (1975), *Linear filtering,* , Math Comput 29:1094–1097
- Pinardi N. et al. (2003), *The Mediterranean ocean forecasting system: first phase of implementation (1998–2001)* , Annales Geophysicae, 21: 3-20
- Trotta F. et al. (2016), *A Structured and Unstructured Relocatable ocean plataform for Forecasting (SURF)* , Deep-Sea Research Part II
- Trotta Francesco, Pinardi Nadia, Fenu Elisa, Grandi Alessandro, Lyubartsev Vladyslav, *Multi-nest high-resolution model of submesoscale circulation features in the Gulf of Taranto* , OCEAN DYNAMICS, 2017, 67, pp. 1609 - 1625
- Trotta F. et al. (2021), *A Relocatable Ocean Modeling Platform for Downscaling to Shelf-Coastal Areas to Support Disaster Risk Reduction* , Front. Mar. Sci.,