MEDSLIK-II version 3.01 Virtual Machine manual

Augusto Sepp, CMCC Foundation, Italy

July 2021

1 Introduction

MEDSLIK-II v3.01 comes out after intensive testing of its sister version v2.1. No new developments were added to the original code. Three important changes were made though:

- useless and outdated code have been deleted,
- inaccuracies in the number of files required to run a simulation were fixed,
- unstable netCDF-ASCII format conversion for ocean fields was fixed,
- increased horizontal accuracy of ASCII ocean fields for very high resolution ocean fields.

2 MEDSLIK-II Virtual Machine Installation

Before downloading the MEDSLIK-II virtual machine (VM), install the most up-to-date Oracle Virtual Box and VM VirtualBox Extension Pack. Both installers can be found at *https://www.virtualbox.org/wiki/Downloads*. Once finished with the installation, proceed to downloading the Linux Mint MEDSLIK-II virtual machine (mdk2.ova), and adding it to your Virtual Box.

The MEDSLIK-II VM includes:

- MEDSLIK-II 3.01 installation found in */scratch/work/MEDSLIK_II_3.01/*
- Python Anaconda v2.7
- conda environment containing all packages required to pre and process MEDSLIK-II data. Activate the conda environment by opening the Terminal and typing: `conda activate mdk2`
- all python scripts necessary to prepare MEDSLIK-II input files and illustrate simulation results at */scratch/work/scripts*

3 How is the MEDSLIK-II VM configured?

All MEDSLIK-II-related files are found at */scratch/surf/*. The model installation is found in the *MEDSLIK_II_3.01/* folder and all Python support scripts are placed in *scripts* and divided into three folders: *scripts/pre_processing*, *scripts/post_processing*, and *scripts/templates*.

4 Testing your VM with the embedded case study

The VM includes a case study off the Yucatan Peninsula, Mexico. The hypothetical spill to be simulated has the following characteristics:

- *Spill date and time:* 13/04/2019 17:30 UTC
- *Spill location:* : 19.9891 N / -87.4167 E
- *Spill duration:* 30h
- *Oil API:* 28
- *Spilled volume:* 1000 tons
- *Simulation length:* 72h

All files required by MEDSLIK-II to simulate the scenario are found at */scratch/work/lab/yucatan_case* and are organized in subfolders:

- *oce_files* : contains MEDSLIK-compatible ocean fields (i.e., zonal and meridional velocities, and water temperature). Files are daily and contain 24 timesteps starting at 01:00 AM and ending at 00:00. There are three files for each day with one ocean variable each.
- *met_files* : contains MEDSLIK-compatible 10m wind fields (i.e., zonal and meridional velocities). Files are daily and contain 24 timesteps starting at 01:00 AM and ending at 00:00.
- *bnc_files* : contains MEDSLIK-compatible bathymetry map (i.e., *dtm.bath*) and coastline polygons (i.e., *dtm.map*). The bathymetry values are interpolated to the grid of input current fields.
- *xp_files* : contains a simulation setup file (i.e., *config1.txt*) and an *Extract_II.for* file adapted to the Yucatan experiment.
- *out_files* : folder used to contain simulation outputs and plots.

Place the required MEDSLIK-II inputs for simulating the Mexico case into the corresponding folders. Place the following commands in the VM Terminal:

```
# reduce typing
export gfolder='/scratch/work/lab/yucatan_case/'

export mfolder='/scratch/work/MEDSLIK_II_3.01/'

# link ocean fields
ln -sf $gfolder/oce_files/MDK*.nc $mfolder/METOCE_INP/PREPROC/OCE/

# link wind fields
ln -sf $gfolder/met_files/*.nc $mfolder/METOCE_INP/PREPROC/MET/

# link bath and coastline
ln -sf $gfolder/bnc_files/dtm.* $mfolder/DTM_INP/

# link MDK-II configuration file
ln -sf $gfolder/xp_files/config1.txt $mfolder/RUN/config1.txt

# link MDK-II Extract file adapted to Mexico case
ln -sf $gfolder/xp_files/Extract_II.for $mfolder/RUN/MODEL_SRC/Extract_II.for
```

MEDSLIK-II is almost ready to simulate the Yucatan case study. Go into the launching folder, compile the updated code and launch your test simulation:

```
cd /scratch/work/MEDSLIK_II_3.01/RUN

sh MODEL_SRC/compile.sh

./RUN.sh
```

MEDSLIK-II simulation results are placed at */scratch/work/MEDSLIK_II_3.01/OUT*. Move the output folder to the Yucatan case study folder:

```
mv $mfolder/OUT/MDK_SIM_ $gfolder/out_files
```

In order to visualize the simulation results, run the script *mdk_postprocessing.py*:

```
python $gfolder/scripts/post_processing/mdk_postprocessing.py
```

informing the experiment name (i.e., path containing MEDSLIK input files at `$gfolder/lab/`) as requested (see Figure 1). Post-processing is automatic and the output files (i.e., MEDSLIK-II raw outputs, surface and beached oil hourly concentration maps, and temporal evolution of the spill center of mass) are found at

`$gfolder/lab/yucatan_case/out_files/MDK_SIM_2019_04_13_yucatan_case`

```
(mdk2) mdk@mdk:/scratch/work/scripts/post_processing$ python mdk_postprocessing.py
type the name of the experiment youd like to postprocess (e.g.: "paria_case"): 'yucatan_case'
```

Figure 1: Script setup for the generation of post-processed outputs - Yucatan case

5 Setting up your own simulations

MEDSLIK-II can be used to simulate oil spills at any part of the ice-free ocean as long as appropriate inputs are supplied. Current, water temperature, 10m winds and bathymetry fields covering the area of interest as well as the local coastline polygons must be given in MEDSLIK-II-compatible format.

The following example shows how to generate the necessary input files to simulate a hypothetical spill event off the port of Rio Grande, Brazil with the following characteristics:

- *Spill date and time:* 15/07/2019 01:00
- *Spill location:* -31.8742 N, -51.7443 E
- *Spill duration:* 64h
- *Oil API:* 28
- *Spill Rate:* 1000 tons/h
- *Simulation length:* 72h

Five main steps must be followed by the user in order to launch an oil spill simulation: generate ocean fields, generate wind fields, generate bathymetry map, generate coastline polygon, and oil spill model setup. Detailed instructions on how to go through the listed steps are presented in the subsections below.

The simulated example relies on ocean fields supplied by the Marine Copernicus Service (CMEMS) and and wind fields supplied by the European Centre for Medium-Range Weather Forecasts (ECMWF). Certify yourself that you are already registered at the Marine Copernicus Service:

(https://resources.marine.copernicus.eu/?option=com_sla)

and at the European Centre for Medium-Range Weather Forecasts

(https://apps.ecmwf.int/registration/).

Add your new CMEMS username and password to *currents_download_MERCATOR_template.sh* (variables **CMEMS_USER** and **CMEMS_PASS** for username and password, respectively) found at */scratch/work/scripts/templates/*. Add your ECMWF Climate Data Store API key number to as indicated at *https://cds.climate.copernicus.eu/api-how-to*. The Climate Data Store API is already installed in the VM requiring solely the update of the user key number at */home/mdk2/.cdsapirc*.

5.1 Step 0: create experiment folder

For the sake of simplicity, the user is advised to create his/her experiment folder at */scratch/work/lab/* and organize input and output files into the preset folder names (see Section 4): *oce_files, met_files, bnc_files* and *out_files*.

5.2 Step 1: generate ocean fields

Ocean fields (i.e., E-W current, S-N current and sea surface temperature) must be provided into separate netCDF files:

- `File name:` MDK_ocean_yymmdd_U.nc, where *yy* correspond to the last two digits of the year, *mm* month and *dd* day. Files are daily containing hourly fields of W-E currents at four depths: 0m, 10m, 30m, and 120m.
- `File name:` MDK_ocean_yymmdd_V.nc, where *yy* correspond to the last two digits of the year, *mm* month and *dd* day. Files are daily containing hourly fields of S-N currents at four depths: 0m, 10m, 30m, and 120m.
- `File name:` MDK_ocean_yymmdd_T.nc, where *yy* correspond to the last two digits of the year, *mm* month and *dd* day. Files are daily containing hourly fields (starting at 01:00 AM UTC and ending at 00:00 included - 24 time steps) of water temperature at four depths: 0m, 10m, 30m, and 120m.

In order to run our example, a 72h long simulation starting on the 15/07/2019 01:00, we must generate daily files of the three fields (i.e., E-W current, S-N current and sea surface temperature) covering the spill area for the period between the 15th and the 18th of July, 2019. Such files must be structured as follows to be read by MEDSLIK-II:

- Temperature
  - `File name:` MDK_ocean_yymmdd_T.nc, where *yy* correspond to the last two digits of the year, *mm* month and *dd* day. Files are daily containing hourly fields.
  - `Dimensions:` *time_counter* for time, *y* for latitude and *x* for longitude, and *depth* for depth. Latitude starting from the southernmost grid point. Longitude starting from the westernmost point.
  - `Variables:` *nav_lat* and *nav_lon* for latitude and longitude fields dependent on the *(y)* and *(x)* dimensions, respectively (i.e., vectors). The variable *time_counter* contains the UTC time corresponding to each time step in hours since 1950-01-01 00:00. Four dimensional temperature fields are included in *votemper* and depend on *(time_counter, depth, y, x)*.
  - `Temporal resolution:` hourly starting from 01:00.
  - `Missing value:` 9999

  - `Units:` degrees Celsius

- E-W currents

  - `File name:` MDK_ocean_yymmdd_U.nc, where *yy* correspond to the last two digits of the year, *mm* month and *dd* day. Files are daily containing hourly fields.
  - `Dimensions:` *time_counter* time, *y* for latitude and *x* for longitude, and *depth* for depth. Latitude starting from the southernmost grid point. Longitude starting from the westernmost point.
  - `Variables:` *nav_lat* and *nav_lon* for latitude and longitude fields dependent on *(y,x)* dimensions, respectively (i.e., vectors). The variable *time_counter* contains the UTC time corresponding to each time step in hours since 1950-01-01 00:00. Four dimensional E-W current fields are included in *vozocrtx* which depends on *(time_counter, depth, y, x)*.
  - `Temporal resolution:` hourly starting from 01:00.
  - `Missing value:` 9999
  - `Units:` m/s

- S-N currents

  - `File name:` MDK_ocean_yymmdd_V.nc, where *yy* correspond to the last two digits of the year, *mm* month and *dd* day. Files are daily containing hourly fields.
  - `Dimensions:` *time_counter* time, *y* for latitude and *x* for longitude, and *depth* for depth. Latitude starting from the southernmost grid point. Longitude starting from the westernmost point.
  - `Variables:` *nav_lat* and *nav_lon* for latitude and longitude fields dependent on *(y,x)* dimensions, respectively (i.e., vectors). The variable *time_counter* contains the UTC time corresponding to each time step in hours since 1950-01-01 00:00. Four dimensional S-N current fields are included in *vomecrty* which depends on *(time_counter, depth, y, x)*.
  - `Temporal resolution:` hourly starting from 01:00.
  - `Missing value:` 9999
  - `Units:` m/s

The user can use any source of ocean fields, at any structured grid resolution, as long as MEDSLIK-II requirements are fully addressed by the customized files.

For the Rio Grande case, we will employ daily CMEMS Global Ocean Analysis and Forecast ocean fields at 1/12 degree resolution (dataset GLOBAL ANALYSIS FORECAST PHY 001 024). The script *download_n_transform_CMEMS.sh* found at *../scripts/pre_processing/* can be used to download, interpolate into hourly steps and transform CMEMS Global ocean fields into MEDSLIK-II format. The script is activated from the Terminal as follows (certify yourself that the conda environment *mdk2* is active):

```
conda activate mdk2

bash /scratch/work/scripts/pre_processing/download_n_transform_CMEMS_currents.sh
```

Setup the script by answering to the questions posed on your terminal. Check how the script has been setup for the Rio Grande case in Figure 2. You will find the ocean files in MEDSLIK-II compatible for at:
`../lab/rg_port/oce_files/`.

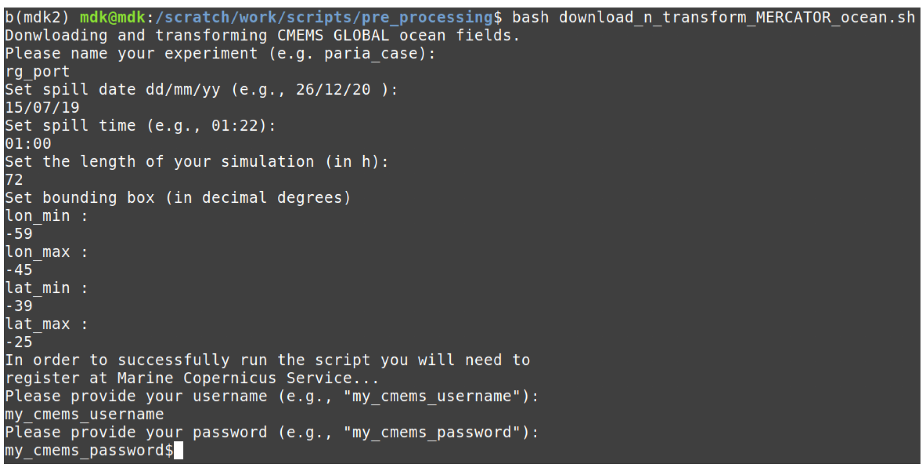


Figure 2: Script setup for the generation of current fields - Rio Grande case

5.3 Step 2: generate wind fields

Wind fields are given using a single file containing S-N and W-E winds at 10m height (in m/s). MEDSLIK-II ingests hourly resolution wind fields organized as follows:

- `File name:` YYYYmmdd.nc, where *YYYY* correspond to all the four digits of the year, *mm* month and *dd* day. Files are daily.
- `Dimensions:` *time* for time, *lat* for latitude and *lon* for longitude. Latitude starting from northernmost grid point. Longitude starting from westernmost point.
- `Variables:` *lat* and *lon* for latitude and longitude fields dependent on *(lat)* and *(lon)* dimensions, respectively. The variable *time* contains the UTC time corresponding to each time step in hours since 1950-01-01 00:00. The wind fields are included in *U10M* and *V10M* and depend on *(time,lat,lon)*.

- `Temporal resolution:` hourly starting from 01:00.

We will employ hourly wind fields from the ECMWF - ERA5 Reanalysis Dataset at 30km resolution. The user can use any source of wind fields, at any resolution, as long as MEDSLIK-II requirements are fully addressed by the customized files. The script *download_era.py* found at *../scripts/pre_processing/* can be used to download and transform ERA5 Reanalysis wind fields into MEDSLIK-II format. The script is activated as follows:

```
python /scratch/work/scripts/pre_processing/download_era5.py
```

Setup the script by answering to the questions posed on your terminal. Check how the script has been setup for the Rio Grande case in Figure 3.

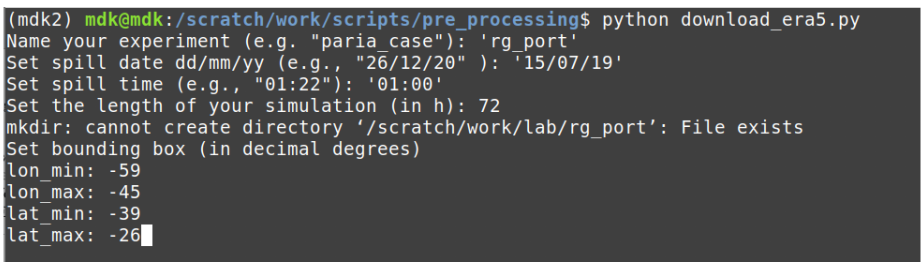


Figure 3: Script setup for the generation of wind fields - Rio Grande case

You will find the wind files in MEDSLIK-II compatible for at: `../lab/rg_port/met_files/`.

5.4 Step 3: generate bathymetry file

MEDSLIK-II allows users to generate their own bathymetry map for any part of the globe and at any resolution. For a given simulation, the bathymetry map should be given at the exact same grid of the adopted ocean fields. Users opting for using their own datasets should bear in mind that bathymetry input files (*dtm.bath*) must be given in the format described below:

- be sure to name it **dtm.bath**
- follow the file structure:
  - line 1: a good-enough description of your bathymetry file. For instance, ”GEBCO 30sec - derived bathymetry for the Patagonian shelf”
  - line 2: [minlong] [maxlong] [minlat] [maxlat] (space between variables with seven digits - 4 decimals - (in degrees))
  - line 3: [nx] [ny] (number of columns (E-W) and rows (S-N) in your source 2D bathymetry field)

– following lines: [depth value] (positive for values below the water line) with no decimals (4 digits). In case of land, put 9999. The bathymetry vector is based on the 2D bathymetry fields looping as follows:

```
counter = 1
for i in n_rows
     for j in n_columns
          bat1d(counter) = bat2d(i,j)
          counter = counter + 1
```

For the Rio Grande example, we will generate an interpolated bathymetry map based on the GEBCO database with original 30" resolution by running the dedicated script:

```
python /scratch/work/scripts/pre_processing/generate_bathymetry.py
```

Setup the script by answering to the questions posed on your terminal. Ocean fields generated at Section 5.2 will be accessed to extract geographic limits and computational grid. Certify yourself that the ocean fields have been generated correctly.

Check how the script has been setup for the Rio Grande case in Figure 4. You will find the bathymetry file in MEDSLIK-II compatible for at: `../lab/rg_port/bnc_files/`.

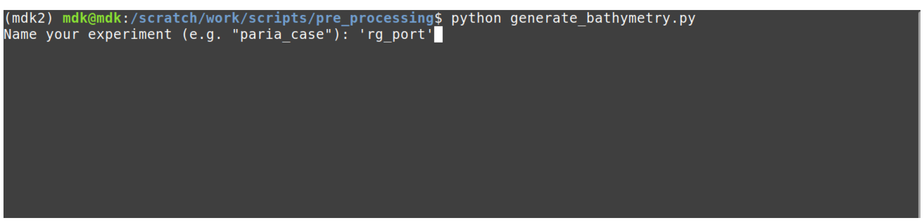


Figure 4: Script setup for the generation of interpolated bathymetry field - Rio Grande case

5.5 Step 4: generate GSHHS-based coastline

MEDSLIK-II currently employs the NOAA GSHHS coastline database to depict the coastline geometry. We will use the script *generate_coastline.py* to create a coastline file, *dtm.map*, for the Rio Grande example by running the script:

```
python /scratch/work/scripts/pre_processing/generate_coastline.py
```

Setup the script by answering to the questions posed on your terminal. Check how the script has been setup for the Rio Grande case in Figure 5. The coastline file will use the previously generated ocean files to extract the geographic boundaries of your study area.

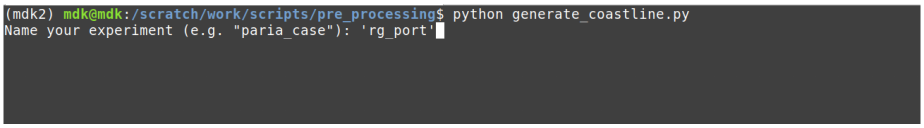


Figure 5: Script setup for the generation of coastline polygons - Rio Grande case

You will find the coastline file in MEDSLIK-II compatible for at:
`../lab/rg_port/bnc_files/`

Depending on the spatial scale of your study and on the number of simulations you are expected to perform, you may consider whether to use the full resolution file (more points and improved representation of the coastline and longer simulations) or high/intermediate (less points, poorer representation of the coastline but faster simulations) resolution.

Very high resolution applications are becoming more and more common in oceanography and, just like in the bathymetry case, the user may be interest in using his/her own coastline map in MEDSLIK-II. Again, it should not be that hard to build his/her own coastline file as long as the following guidelines are respected:

- be sure to name it **dtm.map**
- be sure your polygons never remain open.
- be sure it follows the file structure:
  - line 1: [number of coastline polygons, p, in your study area - 4 digits]
  - line 2: [number of points, n, forming the polygon p_1] [0]
  - line 3: $[p_{1_x}]$ $[p_{1_x}]$ - 8 digits with 5 decimals
  - line 4: $[p_{2_x}]$ $[p_{2_x}]$ - 8 digits with 5 decimals
  - line 3+n: $[p_{n_x}]$ $[p_{n_x}]$ - 8 digits with 5 decimals
  - line 3+n+1:[number of points, n, forming the polygon p_2] [0]
  - repeat procedure writing polygon points
  - this sequence should be repeated for all the polygons in your study area

5.6 Step 5: MEDSLIK-II model setup

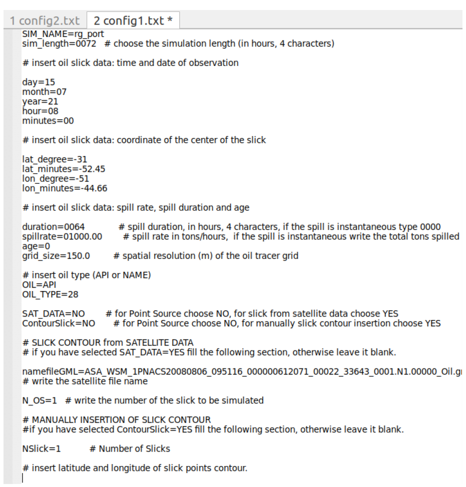
After generating the necessary input files, the user should proceed to setting up MEDSLIK-II and running the simulation. In this section, MEDSLIK-II will be configured to run the Rio Grande case. Start by cleaning up MEDSLIK-II input folders:

```
rm -f /scratch/work/MEDSLIK\_II\_3.01/METOCE\_INP/PREPROC/OCE/*.nc

rm -f /scratch/work/MEDSLIK\_II\_3.01/METOCE\_INP/PREPROC/MET/*.nc

rm -f /scratch/work/MEDSLIK\_II\_3.01/DTM\_INP/dtm.*
```

Proceed to adding the recently generated input files to their respective MEDSLIK-II folders:

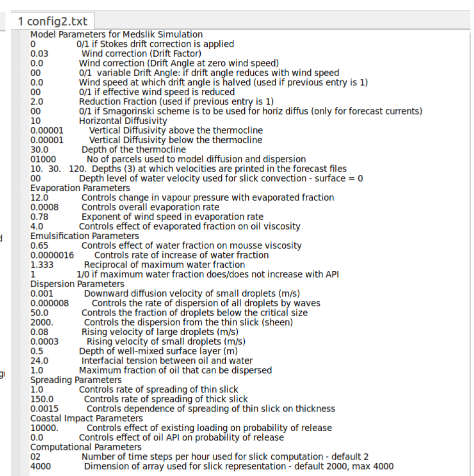
```
ln -sf /scratch/work/lab/rg_port/oce\_files/MDK*.nc
 /scratch/work/MEDSLIK\_II\_3.01/METOCE\_INP/PREPROC/OCE/

ln -sf /scratch/work/lab/rg_port/met\_files/2*.nc
 /scratch/work/MEDSLIK\_II\_3.01/METOCE\_INP/PREPROC/MET/

ln -sf /scratch/work/lab/rg_port/bnc\_files/dtm.bath
 /scratch/work/MEDSLIK\_II\_3.01/DTM\_INP/

ln -sf /scratch/work/lab/rg_port/bnc\_files/dtm.map
 /scratch/work/MEDSLIK\_II\_3.01/DTM\_INP/
```

Now that the necessary inputs can be accessed by MEDSLIK-II, the user should update the MEDSLIK-II configuration files (*config_1.txt* and *config_2.txt*) to the Rio Grande case. Check the example on Figure 6.



```
1 config2.txt   2 config1.txt *
SIM_NAME=rg_port
sim_length=0072   # choose the simulation length (in hours, 4 characters)

# insert oil slick data: time and date of observation

day=15
month=07
year=21
hour=08
minutes=00

# insert oil slick data: coordinate of the center of the slick

lat_degree=-31
lat_minutes=-52.45
lon_degree=-51
lon_minutes=-44.66

# insert oil slick data: spill rate, spill duration and age

duration=0064          # spill duration, in hours, 4 characters, if the spill is instantaneous type 0000
spillrate=01000.00       # spill rate in tons/hours,  if the spill is instantaneous write the total tons spilled
age=0
grid_size=150.0       # spatial resolution (m) of the oil tracer grid

# insert oil type (API or NAME)
OIL=API
OIL_TYPE=28

SAT_DATA=NO       # for Point Source choose NO, for slick from satellite data choose YES
ContourSlick=NO      # for Point Source choose NO, for manually slick contour insertion choose YES

# SLICK CONTOUR from SATELLITE DATA
# if you have selected SAT_DATA=YES fill the following section, otherwise leave it blank.

namefileGML=ASA_WSM_1PNACS20080806_095116_000000612071_00022_33643_0001.N1.00000_Oil.g
# write the satellite file name

N_OS=1   # write the number of the slick to be simulated

# MANUALLY INSERTION OF SLICK CONTOUR
#if you have selected ContourSlick=YES fill the following section, otherwise leave it blank.

NSlick=1         # Number of Slicks

# insert latitude and longitude of slick points contour.
```

(a) config1.txt



```
1 config2.txt
Model Parameters for Medslik Simulation
0             0/1 if Stokes drift correction is applied
0.03          Wind correction (Drift Factor)
0.0           Wind correction (Drift Angle at zero wind speed)
00            0/1  variable Drift Angle: if drift angle reduces with wind speed
0.0           Wind speed at which drift angle is halved (used if previous entry is 1)
00            0/1 if effective wind speed is reduced
2.0           Reduction Fraction (used if previous entry is 1)
00            0/1 if Smagorinski scheme is to be used for horiz diffus (only for forecast currents)
10            Horizontal Diffusivity
0.00001       Vertical Diffusivity above the thermocline
0.00001       Vertical Diffusivity below the thermocline
30.0          Depth of the thermocline
01000         No of parcels used to model diffusion and dispersion
10.  30.   120.  Depths (3) at which velocities are printed in the forecast files
00            Depth level of water velocity used for slick convection - surface = 0
Evaporation Parameters
12.0          Controls change in vapour pressure with evaporated fraction
0.0008        Controls overall evaporation rate
0.78          Exponent of wind speed in evaporation rate
4.0           Controls effect of evaporated fraction on oil viscosity
Emulsification Parameters
0.65          Controls effect of water fraction on mousse viscosity
0.0000016     Controls rate of increase of water fraction
1.333         Reciprocal of maximum water fraction
1             1/0 if maximum water fraction does/does not increase with API
Dispersion Parameters
0.001         Downward diffusion velocity of small droplets (m/s)
0.000008      Controls the rate of dispersion of all droplets by waves
50.0          Controls the fraction of droplets below the critical size
2000.         Controls the dispersion from the thin slick (sheen)
0.08          Rising velocity of large droplets (m/s)
0.0003        Rising velocity of small droplets (m/s)
0.5           Depth of well-mixed surface layer (m)
24.0          Interfacial tension between oil and water
1.0           Maximum fraction of oil that can be dispersed
Spreading Parameters
1.0           Controls rate of spreading of thin slick
150.0         Controls rate of spreading of thick slick
0.0015        Controls dependence of spreading of thin slick on thickness
Coastal Impact Parameters
10000.        Controls effect of existing loading on probability of release
0.0           Controls effect of oil API on probability of release
Computational Parameters
02            Number of time steps per hour used for slick computation - default 2
4000          Dimension of array used for slick representation - default 2000, max 4000
```

(b) config2.txt

Figure 6

Finally, the file */MEDSLIK_II_3.01/Extract_II.for* should be updated to load the ocean and wind fields generated for the Rio Grande simulation. The FOR-

TRAN functions responsible for loading wind and ocean fields found in *Extract_II.for* require the precise dimension of the two mesh grids and the user should update such parameters accordingly. Check on Figure 7 how to check the mesh size of Rio Grande metocean files.

```
(mdk2) mdk@mdk:/scratch/work/MEDSLIK_II_3.01/RUN$ ncdump -h ../METOCE_INP/PREPROC/OCE/MDK_ocean_210715_T.nc
netcdf MDK_ocean_210715_T {
dimensions:
        time_counter = UNLIMITED ; // (24 currently)
        y = 157 ;
        x = 169 ;
        depth = 4 ;
variables:
        double time_counter(time_counter) ;
                time_counter:units = "hours since 1950-01-01 00:00" ;
        double nav_lat(y) ;
                nav_lat:units = "degrees_north" ;
        double nav_lon(x) ;
                nav_lon:units = "degrees_east" ;
        double votemper(time_counter, depth, y, x) ;
                votemper:units = "deg C" ;
                votemper:missing_value = 9999. ;

// global attributes:
                :history = "MERCATOR Forecast - MyOcean -- Adapted to Medslik-II" ;
}
```

(a) nc file variables and dimensions : ocean fields

```
(mdk2) mdk@mdk:/scratch/work/MEDSLIK_II_3.01/RUN$ ncdump -h ../METOCE_INP/PREPROC/MET/20210715.nc
netcdf \20210715 {
dimensions:
        time = UNLIMITED ; // (24 currently)
        lat = 27 ;
        lon = 29 ;
variables:
        double time(time) ;
                time:units = "pythonic days" ;
        double lat(lat) ;
                lat:units = "degrees_north" ;
                lat:lat_min = -39. ;
        double lon(lon) ;
                lon:units = "degrees_east" ;
                lon:lon_min = -59. ;
        double U10M(time, lat, lon) ;
                U10M:units = "m*s-1" ;
                U10M:missing_value = 9999. ;
        double V10M(time, lat, lon) ;
                V10M:units = "m*s-1" ;
                V10M:missing_value = 9999. ;

// global attributes:
                :history = "GFS 10m winds -- Adapted to Medslik-II" ;
}
```

(b) nc file variables and dimensions : wind fields

Figure 7

Figure 8 shows how the functions **ExtractWIND** and **ExtractCURR** should be setup for the Rio Grande case. Search for the subroutines in *Extract_II.for* and replace the "imx" and "jmx" parameters with the appropriate values (Figure 7).

```
c***********************************************************************
      subroutine ExtractCURR(fc_dir,len_dir)

      parameter(ktmx=24, imx=169, jmx=157, kmx=4)
```

(a) ExtractCURR setup

```
c*******************************************************************
      subroutine ExtractWIND(fc_dir,len_dir)

      parameter(ktmx=24, imx=29, jmx=27)
```

(b) ExtractWIND setup

Figure 8: Setup of *Extract_II.for* - Rio Grande case

MEDSLIK-II is now set to run the Rio Grande case. Go to the RUN directory, compile the code and run the model:

```
cd /scratch/work/MEDSLIK\_II\_3.01/RUN/

sh MODEL\_SRC/compile.sh

./RUN
```

6 MEDSLIK-II output files and post processing

The MEDSLIK-II simulation outputs are found at */scratch/work/MEDSLIK_II_3.01/OUT*. The outputs of each simulations are placed in a folder (generically) named: *MDK_SIM_[YYYY]_[MM]_[DD]_[HHMM]_[xpname]*. Each folder contains:

- MEDSLIK simulation input files: *config1.txt, config2.txt, and medslik5.inp*
- *spill_properties.nc*: netCDF4 file containing information relative to each oil parcel properties at each time step
- *medslik.fte*: text file containing information relative to spill properties (e.g., evaporation, emulsification, etc) at each time step
- *MET*: folder containing the wind fields used in the oil spill simulation (ASCII-format)
- *OCE*: folder containing the current fields used in the oil spill simulation (ASCII-format)

The file *spill_properties* is organized as indicated below:

- `File name:` spill_properties.nc
- `Dimensions:` *parcel_id* refers to the parcel number and *time* refers to simulation time steps.
- `Variables:`
  - *latitude:* latitude of each oil parcel (degrees)
  - *longitude:* longitude of each oil parcel (degrees)
  - *evaporative_volume:* parcel volume relative to the evaporative component of the oil (cubic meters)
  - *non_evaporative_volume:* parcel volume relative to the non-evaporative component of the oil (cubic meters)
  - *water_fraction:* percentage of water in each parcel (%)
  - *particle_status:*
    - 0: not released

    - 1: released and spreading
    - 2: released and not spreading
    - 3: released and dispersed
    - 4: bottom
    - -n = parcel is beached. "n" corresponds to the total volume of oil seeped at the coastline.
    - 9: beyond boundary limits
  - *time:* time index
  - *total_fixed_oil:* total amount of oil beached and fixed at the coast (tonnes)
  - *viscosity_emulsion_1:* viscosity emulsion water-oil for mini-spill 1($Pa.s$)
  - *viscosity_oil_1:* viscosity oil for mini-spill 1($Pa.s$)
  - *density_emulsion_1:* density water-oil for mini-spill 1 (kg/m^3)
  - *water_fraction_1:* percentage of water in the emulsion for mini-spill 1 (%)
  - *viscosity_emulsion_2:* viscosity emulsion water-oil for mini-spill 2 ($Pa.s$)
  - *viscosity_oil_2:* viscosity oil for mini-spill 2 ($Pa.s$)
  - *density_emulsion_2:* density water-oil for mini-spill 2 (kg/m^3)
  - *water_fraction_2:* percentage of water in the emulsion for mini-spill 2 (%)
  - *volume_ratio:* volume ratio water/oil

- `Temporal resolution:` depends on the time step defined *a priori*.

The netCDF4 file *spill_properties.nc* is used to generate surface and beached oil concentration maps, and a time series of the spill center of mass geographic position. The user can generate the processed outputs using a single script activated by:

```
python /scratch/work/scripts/post_processing/mdk_postprocessing.py
```

By simply informing the name of your experiment, the script will be responsible for generating the processed outputs and placing them at the simulation folder (i.e., *../[xp_name]/out_files/MDK_SIM_[YYYY]_[MM]_[DD]_[HHMM]_[xpname]*). For the Rio Grande case, simply follow the example below:

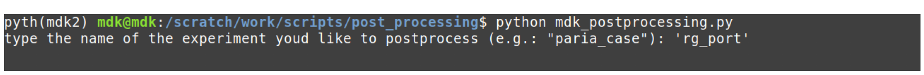


Figure 9: Script setup for the generation of post-processed outputs - Rio Grande case