Istituto Nazionale di Geofisica e Vulcanologia

Istituto Nazionale di Geofisca e Vulcanologia

- Sez. di Bologna -

MEDSLIK-II

lagrangian marine surface oil spill model

User Manual

D. Bruciaferri, and the MEDSLIK-II system team

- Version 1.02 -

September 2015

The development of the MEDSLIK-II model is supported by a formal agreement *Memorandum of Agreement for the Operation and Continued Development of MEDSLIK-II* signed by the following institutions:

- INGV: Istituto Nazionale di Geofisica e Vulcanologia
- CMCC: Centro Euro-Mediterraneo su Cambiamenti Climatici
- UNIBO-DIFA: Universita' di Bologna - DIpartimento di Fisica e Astronomia
- CNR-IAMC: Centro Nazionale delle Ricerche - Istituto per l'Ambiente Marino Costiero
- OC-UCY: Oceanography Centre at the University of Cyprus
- UNIBO-DICAM: Universita' di Bologna - Dipartimento di Ingegneria Civile, Chimica, Ambientale e dei Materiali

MEDSLIK-II Steering Committee members are:

Name	**Affiliation**	**e-mail**
Simona Simoncelli	INGV	simona.simoncelli@ingv.it
Giovanni Coppini	CMCC	giovanni.coppini@cmcc.it
Nadia Pinardi	UNIBO-DIFA	n.pinardi@sincem.unibo.it
Roberto Sorgente	CNR-IAMC	roberto.sorgente@cnr.it
George Zodiatis	OC-UCY	gzodiac@ucy.ac.cy
Renata Archetti	UNIBO-DICAM	renata.archetti@unibo.it

MEDSLIK-II system team members are:

Name	**Affiliation**	**e-mail**
Simona Simoncelli	INGV	simona.simoncelli@ingv.it
Michela DeDominicis	INGV*	michela.dedominicis@ingv.it
Diego Bruciaferri	INGV**	diego.bruciaferri@ingv.it
Andrea Cucco	CNR-IAMC	andrea.cucco@cnr.it
Costanza Tedesco	CNR-IAMC	costanza.tedesco@cnr.it
Svitlana Liubartseva	CMCC	svitlana.liubartseva@cmcc.it
Eric Jansen	CMCC	eric.jansen@cmcc.it
Xenia Panayidou	OC-UCY	xpanagid@ucy.ac.cy
Stavros Stylianou	OC-UCY	stylians@ucy.ac.cy
Hari Radhakrishnan	ORION	
Augusto Neves	UNIBO	augustosepp@gmail.com
Francesco Trotta	UNIBO	f.trotta@sincem.unibo.it

∗ Now at National Oceanography Centre, Liverpool

∗∗ Now at University of Plymouth, Plymouth

Contents

Chapter 1

Introduction

An oil spill is the release of any oily substance into the environment due to human activity, and is a form of pollution. The term is usually applied to marine oil spills, where oil is usually petroleum released into the ocean or coastal waters.

Fig. 1.1: The oil slick in the Gulf of Mexico is seen from a helicopter. (by Rick Loomis, Los Angeles Times, May 6, 2010)

Oil spills may be due to releases of crude oil from tankers, offshore platforms, drilling rigs and wells, as well as spills of refined petroleum products (such as gasoline, diesel) and their by-products or heavier fuels used by large ships

such as bunker fuel.

They endanger public health, imperil drinking waters, devastate natural resources, and disrupt the economy and when they occur human health and environmental quality are put at risk and every effort must be made to prevent or to clean them up promptly once they occur [5].

The ability to represent the transport and fate of an oil slick at the sea surface is a formidable task: by an accurate numerical representation of oil evolution and movement in seawater, the possibility to asses and reduce the oil-spill pollution risk can be greatly improved.

Many factors affect the motion and transformation of the slick. The most relevant of these are the meteorological and marine conditions at the air-sea interface (wind, waves and water temperatures); the chemical characteristics of the oil; its initial volume and release rates; and, finally, the marine currents at different space scales and timescales [4]. Therefore oil spill management strategies need to be developed together with the improvement of meteorological, ocean and wave forecasting models.

MEDSLIK-II is an oil spill model designed to predict the transport and weathering of an oil spill or to simulate the movement of a floating object and its code is a freely available community model which can be downloaded from *http://medslikii.bo.ingv.it*.

This Technical Report describes the properties and the features of the oil spill model MEDSLIK-II version 1.02 and it is organized as follows:

- In Chapter 1, a summary of the new features and properties of the version 1.02 is given.
- In Chapter 2, the governing equations solved by the model and the physical processes involved are briefly described.
- In Chapter 3, the model solution methodology and the numerical schemes and technics used in the model are presented.
- In Chapter 4, the software installation procedure and the description of the code structure are given.
- In Chapter 5, the Meteo-Oceanographic input data needed to force the model are presented.

- In Chapter 6, the steps needed to run both an oil spill-slicks simulation and a particle tracking Lagrangian simulation are described in detail.
- In Chapter 7, the three test cases performed to validate the model are described and the results are given.

1.1 The version 1.02

The new characteristics of the oil spill model MEDSLIK-II v1.02 can be divided into two groups: the **new physical-geometrical properties** and the **new model code features**.

The new physical-geometrical properties comprehend:

- The capability to simulate multiple independent oil spill sources or slicks. This means that each source/slick has its own oil properties (as spill rate and spill time, geometry, ecc.) and its own gravity center where the physical properties needed for the oil wheathering processes are calculated.
- The capability to be forced by Stokes' Drift velocities computed using a wind-wave model.
- The wind drift velocities are calculated using the total local wind fields and not the wind interpolated at the lonely gravity center.
- The possibility to use a 4th order Runge-Kutta numerical scheme to solve the advective-diffusive part of the active tracer equation.
- The possibility to switch off the transformation/reaction part of the model code to use MEDSLIK-II as a particle tracking Lagrangian model.

The new model code features are:

- A modular $Fortran90$ $ExtractII.F90$ subroutine : this subroutine is responsible for the pre-processing of the meteo-oceanographic data used to force MEDSLIK-II. With the new structure, it can be easily adapted to read any kind of netcdf input file.
- A more general linear time interpolation subroutine in the main code: now only 3 time interpolation subroutines exist, one for each kind of forcing. Each subroutine works with a specific kind of forcing at any time resolution.

- An external modular Fortran 90 package to create a netcdf output has been added.
- A visualization package based on python and PyNIO and PyNGL libraries (products of the Computational and Information System Laboratory at the National Centre for Atmospheric Reasearch, NCAR, Boulder, Colorado, USA) has been added.

Chapter 2

Governing Equations

This chapter summarizes the main aspects of the theory on which MEDSLIK-II is based. For a fully detailed description see [4], [3] and [2].

MEDSLIK-II is a Lagrangian marine surface oil spill model designed to simulate oil slick transport and transformation processes for realistic oceanic cases.

Oil discharged in the marine environment moves driven by the advection due to the large-scale flow field and by the dispersion caused by local turbulent flow components. While the oil moves, its concentration changes due to weathering processes.

2.1 The active tracer equation and the model state variables

The general equation for a tracer concentration, $C(\boldsymbol{x}, t)$, with units of mass over volume, mixed in the marine environment, is

$$\frac{\partial C}{\partial t} + \boldsymbol{U} \cdot \boldsymbol{\nabla} C = \boldsymbol{\nabla} \cdot (\boldsymbol{K} \cdot \boldsymbol{\nabla} C) + \sum_{j=1}^{M} r_j(\boldsymbol{x}, C(\boldsymbol{x}, t), t) \tag{2.1}$$

where $\frac{\partial}{\partial t}$ is the local time-rate-of-change operator, $\boldsymbol{U}$ is the three dimensional ocean current mean field, $\boldsymbol{K}$ is the diffusivity tensor which parameterizes the

turbulent effects, and $r_j(\boldsymbol{x}, C(\boldsymbol{x},t),t)$ are the M transformation rates that modify the tracer concentration by means of physical and chemical transformation processes.

Assuming that the slick is composed of oil constituent particles which move like water parcels and considering that the physical and chemical processes act on the entire slick rather than on the single particles properties, the active tracer equation 2.1 can be effectively split into two component equations:

$$\frac{\partial C_1}{\partial t} = \sum_{j=1}^{M} r_j(\boldsymbol{x}, C_1(\boldsymbol{x},t),t) \tag{2.2}$$

$$\frac{\partial C}{\partial t} = -\boldsymbol{U}\cdot\boldsymbol{\nabla}C_1 + \boldsymbol{\nabla}\cdot(\boldsymbol{K}\cdot\boldsymbol{\nabla}C_1) \tag{2.3}$$

where C_1 is the oil concentration solution solely due to the weathering processes, while the final time rate of change of C is given by the advection-diffusion acting on C_1.

Equation 2.2 describes the evolution in time of the oil slick concentration due to the transformation processes acting on the total oil slick volume: therefore, **oil slick state variables** have to be defined.

On the other hand, in order to solve equation 2.3 by using a Lagrangian particle formalism, it is needed to discretize the oil slick in particles with associated **particle state variables**, some of them deduced from the oil slick state variables.

Finally, four **structural state variables** C_S, C_D, C_C and C_B are defined, which describes, respectively, the oil concentration at the surface, in the subsurface, adsorbed on the coasts and sedimented at the bottom (see Fig. 2.1).

Table 2.1 lists all the *Structural*, *Slick* and *Particle* state variables defined in MEDSLIK-II.

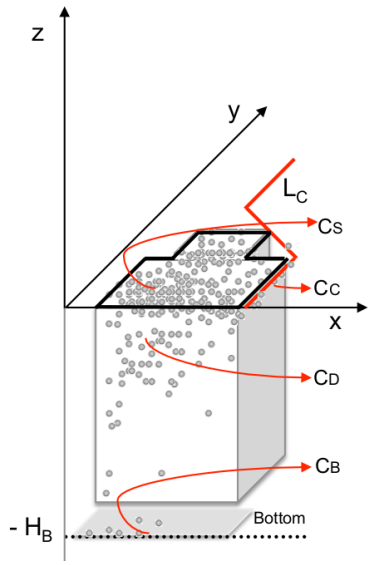


Fig. 2.1: Schematic view of the oil concentration classes (the grey spheres represent the oil particles and L_C the coast line) (modified from [4])

Table 2.1: Oil spill model state variables (modified from [4])

Variable name	Variable type	Variable Description	Dimensions
$C_S(\boldsymbol{x},t)$	Structural	Oil concentration at the surface	Kg m^{-2}
$C_D(\boldsymbol{x},t)$	Structural	Oil concentration dispersed	Kg m^{-2}
$C_C(\boldsymbol{x},t)$	Structural	Oil concentration on the coast	Kg m^{-2}
$C_B(\boldsymbol{x},t)$	Structural	Oil concentration on the bottom	Kg m^{-2}
$V_S(\boldsymbol{x},t)$	Slick	Oil slick surface volume	m^3
$V_D(\boldsymbol{x},t)$	Slick	Oil slick dispersed volume	m^3
$V^{TK}(\boldsymbol{x},t)$	Slick	Thick part of the surface oil slick volume	m^3
$V^{TN}(\boldsymbol{x},t)$	Slick	Thin part of the surface oil slick volume	m^3
$A^{TK}(\boldsymbol{x},t)$	Slick	Surface area of the thick part of the surface oil slick volume	m^2
$A^{TN}(\boldsymbol{x},t)$	Slick	Surface area of the thin part of the surface oil slick volume	m^2
$T^{TK}(\boldsymbol{x},t)$	Slick	Surface thickness of the thick part of the surface oil slick volume	m
$T^{TN}(\boldsymbol{x},t)$	Slick	Surface thickness of the thin part of the surface oil slick volume	m
$\boldsymbol{x}_k(t) = (x_k(t), y_k(t), z_k(t))$	Particle	Particle position	m
$v_{NE}(n_k,t)$	Particle	Non-evaporative surface oil volume particle attribute	m^3
$v_E(n_k,t)$	Particle	Evaporative surface oil volume particle attribute	m^3
$\sigma(n_k,t) = 0,1,2,<0$	Particle	Particle status index (on surface, dispersed, sedimented, on coast)	-

2.2 The Oil Weathering-Transformation Equation

In MEDSLIK-II, the concentration C_1 in the weathering equation 2.2 represents the oil concentration at the surface C_S and the oil concentration dispersed in the water column C_D.

Therefore, the oil weathering equation in MEDSLIK-II comprehends the following

$$\frac{dC_S}{dt} = \frac{\rho}{A}\frac{dV_S}{dt} \tag{2.4}$$

$$\frac{dC_D}{dt} = \frac{\rho}{A}\frac{dV_D}{dt} \tag{2.5}$$

where A is the unit area, V_S is the oil slick surface volume, V_D is the oil slick dispersed volume and ρ is the sea water density.

Regarding the surface oil which arrives close to the coasts and is adsorbed, the concentration of beached oil is given by

$$C_C(\boldsymbol{x}, t) = \frac{\rho}{L_C} V_C \tag{2.6}$$

where L_C is a coastline segment and V_C is the adsorbed oil volume calculated from the oil particle state variables.

In the present version of the model, the oil concentration on the bottom, C_B , is not computed, and it is simply represented by a number of oil particles that reach the bottom.

The weathering acting on the surface oil comprehends three main processes: **Evaporation**, which occurs for first and usually acts on the lighter fractions of the oil, **Dispersion** of the remaining oil fractions below the water surface and **Spreading**, which is responsible for the mechanical spreading of the spill over the water surface under the action of gravitational forces (see Fig. 2.2).

Equations 2.4 and 2.5 are solved by using the Mackay et al. (1980) transformation process algorithms, which are based on the assumption that the oil surface volume can be divided into a thin part, V^{TN} , and a thick part, V^{TK}.

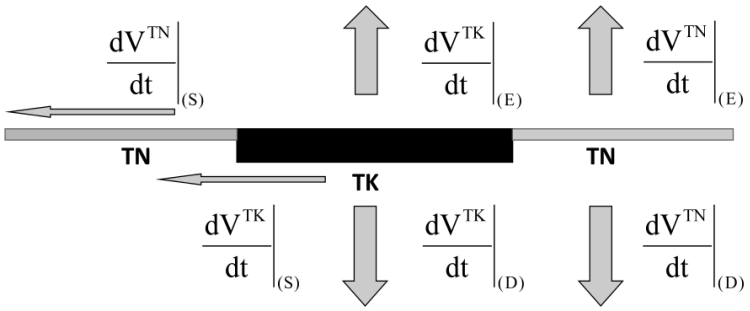


Fig. 2.2: Weathering processes using Mackay's approach. TK indicates the thick slick and TN the thin slick. V^{TK} and V^{TN} are the surface oil volumes of the thick and thin part of the slick and the suffixes indicate evaporation (E), dispersion (D) and spreading (S) (from[4])

Consequently, the weathering processes are considered separately for the thick slick and thin slick and the prognostic equations are

$$\frac{dV^{TK}}{dt} = \frac{dV^{TK}}{dt}\Big|_{(E)} + \frac{dV^{TK}}{dt}\Big|_{(D)} + \frac{dV^{TK}}{dt}\Big|_{(S)} \tag{2.7}$$

$$\frac{dV^{TN}}{dt} = \frac{dV^{TN}}{dt}\Big|_{(E)} + \frac{dV^{TN}}{dt}\Big|_{(D)} + \frac{dV^{TN}}{dt}\Big|_{(S)} \tag{2.8}$$

where the suffixes indicate evaporation (E), dispersion (D) and spreading (S).

All the oil slick state variables are defined only at the slick's central geographical position (Gravity Centre, hereinafter GC), which is updated after each advection-diffusion time step. In the case of multiple slicks, each slick is characterized by its own GC.

The oil spill centre position $\boldsymbol{x}_C^s = (x_C^s(t), y_C^s(t))$ is defined as

$$\begin{cases} x_C^s = \frac{\sum_{k=1}^{N_s} x_k^s(t)}{N_s} \\ \\ y_C^s = \frac{\sum_{k=1}^{N_s} y_k^s(t)}{N_s} \end{cases} \tag{2.9}$$

where s ($1 < s <$ number of slicks) is an integer which identifies the s^{th} slick, N_s is the number of oil constituent particles into which the s^{th} slick is broken and $(x_k^s(t), y_k^s(t))$ is the position at time t of the k^{th} particle of the s^{th} slick.

2.3 The Advection-diffusion Equation

Medslik-II before solves the weathering equation 2.2 and then uses the oil concentration solution C_1 to solve the advection-diffusion equation 2.3 .

In order to solve the latter with a Lagrangian particle formalism, the oil slick is before discretized in N_{TOT} particles with associated particle state variables and then the oil concentration is computed by assembling the particles together with their associated properties.

The horizontal current field used in the Lagrangian model is taken to be the sum of different components:

$$\begin{cases} \sigma = 0, \quad d\boldsymbol{x}_k(t) = [\boldsymbol{U}_C(x_k, y_k, 0, t) + \boldsymbol{U}_W(x_k, y_k, t) \\ \qquad\qquad\qquad + \boldsymbol{U}_S(x_k, y_k, t)]dt + \boldsymbol{U}_D(x_k, y_k, t)]dt + d\boldsymbol{x'}_k(t) \\ \\ \sigma = 1, \quad d\boldsymbol{x}_k(t) = \boldsymbol{U}_C(x_k, y_k, z_k, t)dt + d\boldsymbol{x'}_k(t) \end{cases} \tag{2.10}$$

where σ is the particle status index which identifies if a particle is on the sea surface ($\sigma = 1$) or dispersed into the water column ($\sigma = 0$), $\boldsymbol{U}_C$ is the wind, buoyancy, and pressure-driven large-scale current velocity field, $\boldsymbol{U}_W$ is the local wind velocity correction term (wind drift term), $\boldsymbol{U}_S$, called hereafter wave current term, is the velocity due to wave-induced currents or surface Stokes drift, $\boldsymbol{U}_D$ is the wind drag correction due to emergent part of the objects at the surface and $d\boldsymbol{x'}_k(t)$ is the displacement due to the turbulent diffusion.

In Medslik-II the turbulent diffusion is parameterized with a random walk scheme as follows:

$$d\boldsymbol{x}'_k(t) = \sqrt{2\boldsymbol{K}dt}\boldsymbol{Z} \tag{2.11}$$

where $\boldsymbol{K}$ is the turbulent diffusion diagonal tensor and $\boldsymbol{Z}$ is a vector of independent random numbers used to model the Brownian random walk processes. The three components of $\boldsymbol{K}$ are K_h, K_h, K_v. Usually, the isotropic K_h is set equal to 2 $m^2 s^{-1}$ while K_v is set to be 0.01 $m^2 s^{-1}$ in the mixed layer (assumed to be 30 m deep) and 0.0001 $m^2 s^{-1}$ below it.

Chapter 3

Numerical Approaches

This chapter describes the main numerical aspects which have to be considered in order to run Medslik-II. The main consequences of the *new physical-geometrical properties* on the numerical features of the model as well as the *4th order Runge-Kutta scheme* implemented in this new version of Medslik-II are presented and described.

For a fully detailed description of the numerical approaches used in Medslik-II see [4] and [3] .

This chapter is organized as follows:

- First, the model solution methodology is outlined (section 3.1) .
- Then, how MEDSLIK-II sets the initial conditions for particle and slick variables at the surface for single or multiple spills or slicks is discussed in section 3.2.
- Finally, in section 3.3 the Euler Forward scheme as well as the 4th order Runge-Kutta scheme used to solve the governing equations of the model are described.

3.1 Model solution methodology

In order to solve coherently the transformation and the advection-diffusion equations, an algorithm wich links the oil slick and the particle state variables is needed. The model *sequential solution method* is represented schematically in Fig. 3.1.

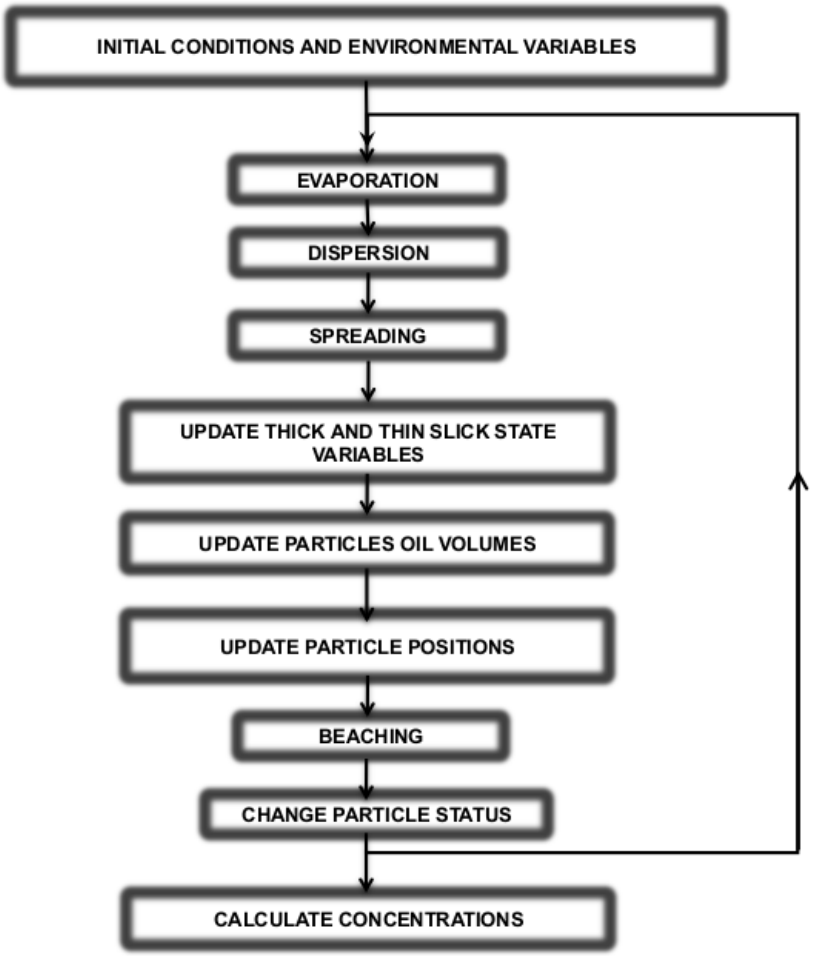


Fig. 3.1: MEDSLIK-II model solution procedure methodology (from [4])

3.2 Initial conditions

A surface oil release into the marine environment can be instantaneus or continuous: in the first case, all the oil is spilled instantly, while in the latter the leakage may last for several hours or even months. Furthermore, the simulated scenario could involve an oil slick detected by satellite or at sight. Finally, the simulated oil spills or detected slicks can be more than one.

Medslik-II version 1.02 is designed to simulate instantaneous or continuous single spills, instantaneous multiple spills and single or multiple detected slicks.

The approach used is the same adopted in the version 1.01 (see [4] for a detailed explanation), except for the fact that the structural, oil and particle state variables are defined for each spill/slick.

A **continuous oil spill release** is modeled dividing the total spilled oil into a number of sub-spills consisting of a given part of the oil released at the spill location during the prescribed computational time interval. As each sub-spill is moved away from the source, the total spill becomes a chain of sub-spills. In a continuous release, a given fraction of the total amount of particles used in the simulation is released with each sub-spill.

In the case of an **instantaneous release**, at the beginning of the simulation all the oil is released on the sea surface at the spill location and all the particles are released.

Satellite or at sight detected slicks are modeled as instantaneous spills except for the fact that the total amount of oil and particles of each slick are released inside the boundary line which identifies and describes the shape of the slicks. Therefore, in this case each oil spill source is not described by a single location, but by an ensamble of geographical points which identify a contour line.

The oil-spill data required to define a numerical oil-spill initial condition are:

- LOCATION : the geographical coordinates of the oil source location. In the case of a detected slick, the geographical coordinates of all the points which identify the oil slick polygonal contour line have to be provided.

- START TIME : it is the start time of the spill or the time of the slick detection.
- VOLUME : it is the total amount of oil released in a spill or the total oil of a slick.
- DURATION : the spill duration (for a detected slick it is zero).
- AGE : age of the oil spill from initial arrival in the sea. This information can be easily provided to MEDSLIK-II by satellite monitoring systems.

If the simulated scenario consists of multiple spills-slicks, all these informations have to specified for each spill-slick.

3.3 Numerical time integration schemes

Medslik-II solves its governing differential equations by using finite-difference numerical methods.

- The version 1.02 allows to choose an Euler forward scheme or a 4^{th} order Runge-Kutta scheme to solve the **Lagrangian horizontal particle motion** (equation 2.10).

  Euler Forward scheme

  The particle position at time step $t + \Delta t$ is calculated as follows:

  $$\boldsymbol{x}_k(t + \Delta t) = \boldsymbol{x}_k(t) + \boldsymbol{U}(\boldsymbol{x}_k, t)\Delta t + \Delta \boldsymbol{x'}_k(t) \tag{3.1}$$

  where $\boldsymbol{x}_k(t)$ represents the position of the k^{th} paricle at the current time step t, Δt is the discrete Lagrangian time step, $\boldsymbol{U}(\boldsymbol{x_k}, t)$ is the total Eulerian velocity field at the current time step at the particle position and $\Delta \boldsymbol{x'}_k(t)$ is the particle displacement due to turbulent motion.

  The total velocity field at the current time step at the particle position is computed performing a linear interpolation in time between successive input velocity fields, a bi-linear interpolation in horizontal space using the four external field grid points of the Eulerian model field (oceanographic or meteorological) nearest the particle position and a

linear vertical interpolation between the two field at the Eulerian model levels nearest the particle depth.

Normally, the Lagrangian time step Δt is taken to be 1800 s.

4^{th} order Runge-Kutta scheme

The particle position at time step $t + \Delta t$ is

$$\boldsymbol{x}_k(t + \Delta t) = \boldsymbol{x}_k(t) + \frac{1}{6}\Big(\boldsymbol{K}_1 + 2\boldsymbol{K}_2 + 2\boldsymbol{K}_3 + \boldsymbol{K}_4\Big)\Delta t + \Delta \boldsymbol{x'}_k(t) \quad (3.2)$$

where

$$\begin{aligned}
\boldsymbol{K}_1 &= \boldsymbol{U}(\boldsymbol{x}_k, t) \\
\boldsymbol{K}_2 &= \boldsymbol{U}(\boldsymbol{x}_k + \frac{1}{2}\boldsymbol{K}_1\Delta t, t + \frac{\Delta t}{2}) \\
\boldsymbol{K}_3 &= \boldsymbol{U}(\boldsymbol{x}_k + \frac{1}{2}\boldsymbol{K}_2\Delta t, t + \frac{\Delta t}{2}) \\
\boldsymbol{K}_4 &= \boldsymbol{U}(\boldsymbol{x}_k + \boldsymbol{K}_3\Delta t, t + \Delta t)
\end{aligned}$$

The 4^{th} order Runge-Kutta scheme needs the total velocity field at time steps t, $t + \Delta t/2$ and $t + \Delta t$ at the position $\boldsymbol{x}_k$, $\boldsymbol{x}_k + \frac{1}{2}\boldsymbol{K}_1\Delta t$, $\boldsymbol{x}_k + \frac{1}{2}\boldsymbol{K}_2\Delta t$ and $\boldsymbol{x}_k + \boldsymbol{K}_3\Delta t$. If the Runge-Kutta method is chosen, before a linear interpolation in time between successive input velocity fields is performed to compute the total velocity field at the necessary time steps. Then, a bi-linear horizontal spatial interpolation and a vertical linear interpolation are carried out to find the field values at the needed positions.

- On the other hand, **the weathering equations** 2.8 are solved with the Euler forward scheme, but with a different time step, the so-called weathering time step. The surface oil volumes of the thick and thin part of the slick are computed as follows:

$$V^{TK}(t + \delta t) = V^{TK}(t) + \frac{dV^{TK}}{dt}\delta t \quad (3.3)$$

$$V^{TN}(t + \delta t) = V^{TN}(t) + \frac{dV^{TN}}{dt}\delta t \quad (3.4)$$

where $\frac{dV^{TK}}{dt}$ and $\frac{dV^{TN}}{dt}$ are calculated using the empirical relationships which parameterize the weathering processes and the weathering time step δt is equal to $\frac{\Delta t}{30}$.

Chapter 4

Installing MEDSLIK-II

This chapter describes the technical details needed to properly install Medslik-II version 1.02. Furthermore, in section 4.5 a description of the code structure is given.

Medslik-II version 1.02 is written in FORTRAN-77/90, with in no machine-dependent elements, so that it can be installed without modifications on most platforms.

The architecture currently supported is Linux (tested on Ubuntu 10.04 LTS, Lubuntu 14.04, Debian 7 and Centos 5).

4.1 Downloading the code

MEDSLIK-II is a freely available community model and its code can be downloaded from the web page `http://medslikii.bo.ingv.it`, at the `Download` section.

After registration and be logged in, two links, named `Version_1.01` and `Version_1.02` respectively, are accessible. Each link leads to the homepage of the corresponding version of the model: here there are the links to download the code, to download the user manual and to download all the data needed to run the test cases.

After downloading the version 1.02 code, it is needed to move and extract the gzip compressed tar archive in the chosen installation path:

```
mv /${DOWNLOAD_FOLDER}/MEDSLIK_II_1.02.tar.gz /${INSTALLATION_FOLDER}
cd /${INSTALLATION_FOLDER}
tar -xvzf MEDSLIK_II_1.02.tar.gz
```

The variables `${DOWNLOAD_FOLDER}` and `${INSTALLATION_FOLDER}` have to be replaced by the user specific path of the `DOWNLOAD` and user chosen `INSTALLATION` folders, respectively. Because the installation process requires write, read and execute permissions, it is needed that the user has the necessary permits for the chosen installation folder.

4.2 Structure of the model

Extracting the gzip-tarball file creates the `/${INSTALLATION_FOLDER}/MEDSLIK_II_1.02` directory which is organized with the following structure:

Fig. 4.1: Tree list of the directories into which the model code is organized.

The description of the meaning and the function of each folder is given below:

`./DATA/` : Contains the directories for the NetCDF meteo-oceanographic model output data files needed to run a simulation

`./DATA/MET/` : Contains the NetCDF atmospheric model output data files needed for the simulation

`./DATA/OCE/` : Contains the NetCDF oceanographic model output data files needed for the simulation

`./DATA/WAV/` : Contains the NetCDF wave model output data files needed for the simulation

`./EXE/` : Contains the executable files created after having compiled the Fortran code and all the Python code needed to pre-process the data

`./OUTPUT/` : Contains the simulation output directories

`./PLOT/` : Contains the Python code needed to plot the simulation results

`./RUN/` : Contains the `data` and `INP_DATA` directories and all the shell scripts needed to configure and run a simulation

`./RUN/data/` : Contains bathymetry files for several regions of the Mediterranean Sea and other associated data files

`./RUN/INP_DATA/` : Contains the directories for the ASCII formatted meteo-oceanographic model output data files. These files are written after pre-processing of the original NetCDF files stored in `./DATA/*/`

`./RUN/INP_DATA/MET/` : Contains the ASCII formatted atmospheric model output data files

`./RUN/INP_DATA/OCE/` : Contains the ASCII formatted oceanographic model output data files

`./RUN/INP_DATA/WAV/` : Contains the ASCII formatted wave model output data files

`./RUN/template/` : Contains template files for the input and the parameter files.

`./SRC/` : Contains the model FORTRAN-77/90 source code

4.3 External libraries needed for computing and plotting

Nowdays, meteo-oceanographic model output data are usually provided as NetCDF datasets and therefore Medslik-II is designed to work with NetCDF files. Furthermore, in version 1.02 an external modular FORTRAN-90 package to create a Medslik-II NetCDF output file has been added.

Concerning the plotting procedure, a visualization package based on Python and PyNIO and PyNGL libraries has been added in this last version of the model. PyNGL and PyNIO libraries are products of the Computational and Information System Laboratory at the National Centre for Atmospheric Reasearch, NCAR, Boulder, Colorado, USA. PyNGL, as well as the associated modules PyNIO and NumPy, are all freely available.

NetCDF

NetCDF is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. NetCDF was developed and is maintained at Unidata. Unidata provides data and software tools for use in geoscience education and research. Unidata is part of the University Corporation for Atmospheric Research (UCAR) Community Programs (UCP). Unidata is funded primarily by the National Science Foundation.

The NetCDF homepage may be found at `http://www.unidata.ucar.edu/netcdf`. The NetCDF source-code is hosted at GitHub, and may be found directly at `http://github.com/Unidata/netcdf-c`.

In order to run Medslik-II version 1.02 it is needed to install NetCDF 4.3 or later. Documentation on how to install properly NetCDF4 FORTRAN libraries can be found at `http://www.unidata.ucar.edu/netcdf`.

PyNGL

PyNGL (`https://www.pyngl.ucar.edu/newusers.shtml`) is a Python language module used to visualize scientific data, with an emphasis on high quality 2D visualizations. A working knowledge of Python is assumed. The

NumPy extension module is used for array processing.

PyNGL provide Python interfaces to most of the graphics and file input/output functionality existing in the NCAR Command Language (NCL). A knowledge of NCL is not necessary because they are meant to be independent from NCL and used as a stand-alone suite of Python functions. In a few circumstances where the NCL documentation applies directly to PyNGL and there is no ambiguity, links are made to the NCL documentation from the PyNGL documentation.

Documentation on how to install properly PyNGL can be found at `https://www.pyngl.ucar.edu/Download/`.

PyNIO

PyNIO (`https://www.pyngl.ucar.edu/Nio.shtml`) is a Python package that allows read and/or write access to a variety of data formats using an interface modeled on NetCDF. PyNIO is composed of a C library which contains the same data I/O code used in NCL, a scripting language developed for analysis and visualization of geo-scientific data. In a few circumstances where the NCL documentation applies directly to PyNIO and there is no ambiguity, links are made to the NCL documentation from the PyNIO documentation.

Currently supported formats include NetCDF3 and NetCDF4 classic format (read/write), GRIB1 and GRIB2 (read-only), HDF4 (read/write for Scientific DataSets only), HDF-EOS2 and HDF-EOS5 (read-only for Grid and Swath data only), CCM history files (read-only), OGR-Open Geospatial Consortium's Simple Feature formats (ESRI Shapefile, MapInfo, GMT, Tiger) (read-only).

PyNIO interfaces with the Python environment using the multi-dimensional array module, NumPy. Documentation on how to install properly PyNIO can be found at `https://www.pyngl.ucar.edu/Download/` .

4.4 Compiling and linking the code

In order to easily compile the code, a bash script named `compile.sh` exists inside the directory `MEDSLIK_II_1.02/SRC/`.

A **GNU Fortran compiler** is needed.

To compile properly the model FORTRAN code, first it is needed to open the `compile.sh` file with any editor and change the bash variable `NETCDF_DIR` adding the exact absolute path of the NetCDF libraries installation directory.

Once you are sure that the file has execute permission, just run the build process with the following command: `./compile.sh`.

4.5 Structure of the code

The Medslik-II code comprehends FORTRAN-77/90 source code and Bash and Python scripts.

Bash scripts are located into the `RUN` directory and they are:

- `medslik_II.sh` : it is the main script and it controls the pre and post processing, the model run and plot procedures.
- `RUN_drifter_TESTCASE.sh`, `euler_RUN_drifter_TESTCASE.sh` and `runge_RUN_drifter_TESTCASE.sh` : they run the drifters test case (see 7).

Python scripts control the oil initial conditions processing and the output plotting procedures.

The oil initial condition processing python scripts are located inside the `EXE` directory (because Python source code is automatically runtime compiled into Python byte code by the CPython interpreter):

- `read_oil_data.py` : it reads the oil related variables into the `medslik_inputfile.txt` (`$OIL` and `$OIL_TYPE`) (see ??) and processes these informations in order to find the specific characteristics of the simulated oil.

- `ReadSatData.py` : it is able to read data for satellite detected oil spills and processes these informations in Medslik-II model format.

The python scripts which plot MEDSLIK-II `version_1.02` output results are located inside the `PLOT` directory. They are

- `MEDSLIKII_plot_slick.py`: it is a general scripts able to plot MEDSLIKII oil simulations output results stored in NetCDF output files.
- `MEDSLIKII_plot_oilfate.py`: it is able to plot MEDSLIKII oil simulations output results written in the `medslik.fte` file, which contains the trend over time of: the oil volume spilled, the percentage of oil evaporated, the percentage of oil on the surface, the percentage of oil dispersed, the percentage of oil on the coast, the oil-water emulsion viscosity, the oil viscosity, the oil density, the fraction of water contained in the oil-water emulsion, etc.
- `MEDSLIKII_plot_traj.py`: it is a general script able to plot MEDSLIKII Particle Lagrangian simulations output results.
- `plot_slick_ALGERIA.py`: it is an AlGERIAN test-case scripts, meaning that it is a customized version of the `MEDSLIKII_plot_slick.py` file for the ALGERIAN test case: it uses an appropriate palette for the oil concentration and it plots satellite observations at the right time-step (see 7 for a detailed description of this test case).
- `plot_slick_LEBANON.py`: it is a LEBANON test-case scripts, meaning that it is a customized version of the `MEDSLIKII_plot_slick.py` file for the LEBANON test case: it uses an appropriate palette for the oil concentration and it plots satellite observations at the right time-step (see 7 for a detailed description of this test case).
- `plot_slick_SRG.py`: it is a SERIOUS GAME test-case scripts, meaning that it is a customized version of the `MEDSLIKII_plot_slick.py` file for the SERIOUS GAME test case: it uses an appropriate palette for the oil concentration and it plots oil samples and drifter release locations at the right time-step (see 7 for a detailed description of this test case).
- `plot_traj_BEACON.py`: it is a SERIOUS GAME test-case scripts, meaning that it is a customized version of the `MEDSLIKII_plot_traj.py` file for the SERIOUS GAME test case: it plots BEACON real trajectory and results for all the three simulations performed in this test case (see 7 for a detailed description of this test case).

- `plot_traj_CODE.py`: it is a SERIOUS GAME test-case scripts, meaning that it is a customized version of the `MEDSLIKII_plot_traj.py` file for the SERIOUS GAME test case: it plots CODE real trajectory and results for all the three simulations performed in this test case (see 7 for a detailed description of this test case).

- `plot_traj_ISPHERE.py`: it is a SERIOUS GAME test-case scripts, meaning that it is a customized version of the `MEDSLIKII_plot_traj.py` file for the SERIOUS GAME test case: it plots I-SPHERE real trajectory and results for all the three simulations performed in this test case (see 7 for a detailed description of this test case).

- `plot_traj_SLDMB.py`: it is a SERIOUS GAME test-case scripts, meaning that it is a customized version of the `MEDSLIKII_plot_traj.py` file for the SERIOUS GAME test case: it plots SLDMB real trajectory and results for all the three simulations performed in this test case (see 7 for a detailed description of this test case).

FORTRAN-77/90 source code is located inside the `SRC` directory. It comprehends the following files:

- `Extract_II.F90` : This program reads meteo-oceanographic data from NetCDF model output files and writes the required data into Medslik-II ASCII formatted input files. It contains a main program and two modules, namely `env` and `utils`.

  `env` contains the declaration statments for all the variables used by the main program.

  `utils` contains all the subroutines used by the main program.

  Figure 4.2 represents the call graph for the `Extract_II.F90` program.

- `jday.f` : This program gets a day in the format `YYYYMMDD` as first argument and returns a date in the same format adding or subtracting a number of days given as a second argument.

- `lat_lon.F90` : This program computes the geographical limits of the area affected by the spill and writes the limits in the `RUN/medslik.tmp` file.

- `create_netcdf.F90` : This program reads MEDSLIK-II Ascii formatted outputs and creates, by the bilinear interpolation and linear interpo-

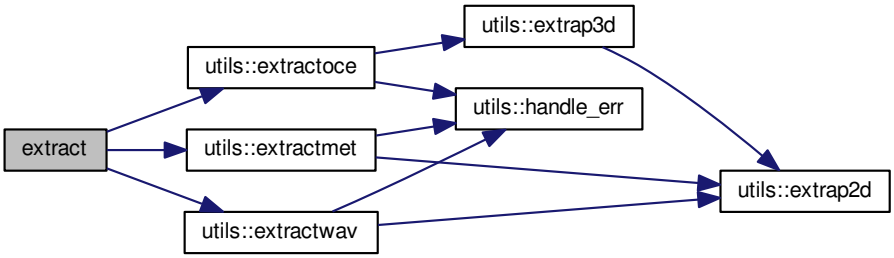


Fig. 4.2: Call graph for the Extract program.

lation methods, netcdf outputs of the Oil Concentration, the Ocean and the Wave-induced currents and the Wind velocities fields. Wind velocities are defined at the gravity centre point of each slick. On the other hand, ocean forcing fields are defined on their original grids and oil concentration is defined on the 150mx150m grid. This software uses the following external modules: `module_phymat.F90`, `module_interpolation.F90` and `module_netcdf.F90`.

Figure 4.3 represents the call graph for the `Extract_II.F90` program.

- `medslik_II.for` : This is the Medslik-II v1.02 program.

  The program `medslik_ii` consists of a `main` and several subroutines and functions. Figure 4.4 represents the call graph for the `medslik_II.for` program.

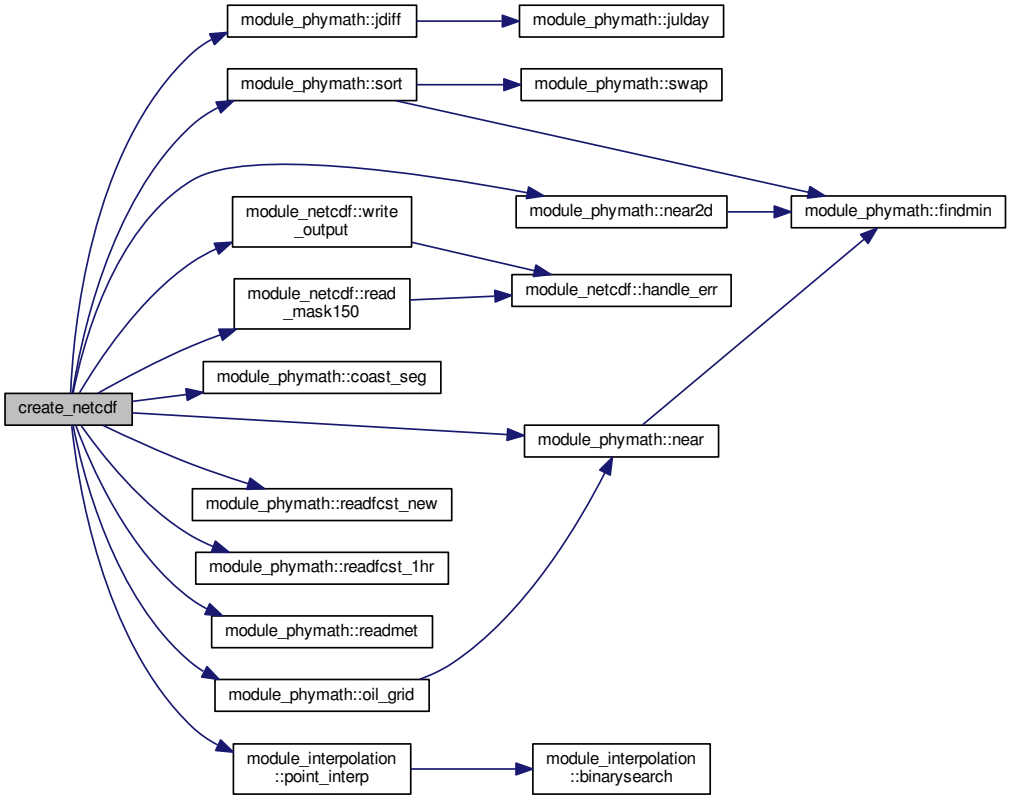


Fig. 4.3: Call graph for the create netcdf program.

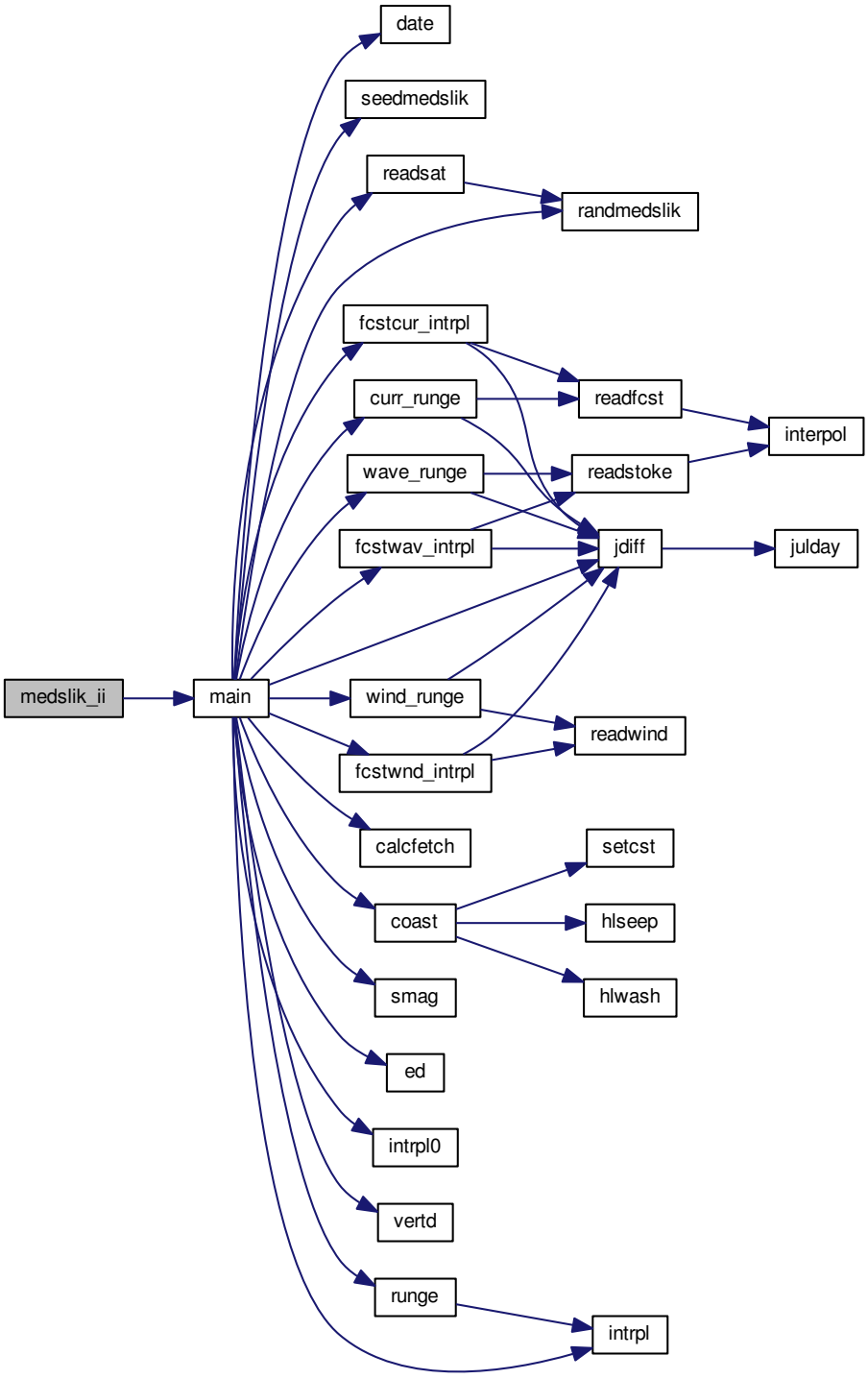


Fig. 4.4: Call graph for the medslikii program.

Chapter 5

Meteo-Oceanographic input forcings

Meteorological and oceanographic data are needed to force the model.

The meteo-oceanographic forecasts-analysis NetCDF data needed for a simulation have to be placed in one of the directories listed below:

- `MEDSLIK_II_1.02/DATA/MET` is for the atmospheric wind forcing data;
- `MEDSLIK_II_1.02/DATA/OCE` is for the oceanographic (i.e, current and sea surface temperature) forcing data;
- `MEDSLIK_II_1.02/DATA/WAV` is for the oceanographic wave-induced current forcing data;

Medslik-II 1.02 code is designed to use the following meteo-oceanographic model output fields.

5.1 Ocean Current and SST data

Copernicus Mediterranean Forecasting System products

These data can be daily or hourly average forecasts and analysis and are produced in NetCDF format for the whole Mediterranean Sea. Copernicus

Mediterranean Forecasting System data and the MFS system are fully described at `http://marine.copernicus.eu/` and `http://medforecast.bo.ingv.it/` respectively.

In order to run a simulation, two files named as follows are needed for each simulated *YYYYMMDD* date :

Daily mean data

- For forecast data

1) `YYYYMMDD_dm-INGV--RFVL-MFSs4e-MED-b${bulletin_date}_fc-fv05.00.nc`
2) `YYYYMMDD_dm-INGV--TEMP-MFSs4e-MED-b${bulletin_date}_fc-fv05.00.nc`

- For analysis data

1) `YYYYMMDD_dm-INGV--RFVL-MFSs4e-MED-b${bulletin_date}_an-fv05.00.nc`
2) `YYYYMMDD_dm-INGV--TEMP-MFSs4e-MED-b${bulletin_date}_an-fv05.00.nc`

Hourly mean data

- For forecast data

1) `YYYYMMDD_hm-INGV--RFVL-MFSs4e-MED-b${bulletin_date}_fc-fv05.00.nc`
2) `YYYYMMDD_hm-INGV--TEMP-MFSs4e-MED-b${bulletin_date}_fc-fv05.00.nc`

- For analysis data

1) `YYYYMMDD_hm-INGV--RFVL-MFSs4e-MED-b${bulletin_date}_an-fv05.00.nc`
2) `YYYYMMDD_hm-INGV--TEMP-MFSs4e-MED-b${bulletin_date}_an-fv05.00.nc`

where `${bulletin_date}` is the data production date (`YYYYMMDD`).

2008 Mediterranean Forecasting System hourly mean products

These data are hourly average forecasts produced in NetCDF format for the whole Mediterranean Sea. They have been created following the time convenction on which the MFS system was based in 2006-2008 and they have been added in the Medslik-II code in order to run the Algeria and Libano test cases with the same data used for the Medslik-II 1.01 test cases.

In order to run a simulation, three files named as follows are needed for

each simulated *yyyymmdd* date :

1) `MEDffEyyyymmdd_01_grid_T.nc`

2) `MEDffEyyyymmdd_01_grid_U.nc`

3) `MEDffEyyyymmdd_01_grid_V.nc`

where T, U and V mean, respectively, Temperature, Zonal ocean current velocity and Meridional ocean current velocity.

5.2 Atmospheric Wind data

European Centre for Medium-Range Weather Forecasts products

The ECMWF wind forecasts and analyses are available for the whole Mediterranean Sea with a horizontal resolution of 0.125 degrees since March 23^{th}. They are snapshots every 6 hours (at hours 00, 06, 12, 18). Between the 11^{th} of January 2009 and the 23^{th} of March 2015 the ECWMF wind forecasts and analyses products were available for the whole Mediterranean Sea with a horizontal resolution equal to 1/4 of degree while before the 11^{th} of January 2009 the ECWMF analysis were produced at a resolution of 0.5 degrees. The temporal resolution of the forecasts and analysis did not change during the different versions developed.

In order to run a simulation, one file named as follows is needed for each simulated *YYYYMMDD* date:

0.5 degrees horizontal resolution (for forecast or analysis data)

- `YYYYMMDD_6hi-INGV-ECMWF-AM050-MED.nc`

0.25 degrees horizontal resolution (for forecast or analysis data)

- `YYYYMMDD_6hi-INGV-ECMWF-AM025-MED.nc`

0.125 degrees horizontal resolution (for forecast or analysis data)

- `YYYYMMDD_6hi-INGV-ECMWF-AM0125-MED.nc`

5.3 Ocean Wave-induced current data

Copernicus Mediterranean Forecasting System hourly instantaneous products

These data are hourly instantaneuos forecasts produced in NetCDF format for the whole Mediterranean Sea. Copernicus Mediterranean Forecasting System data and the MFS system are fully described at `http://marine.copernicus.eu/` and `http://medforecast.bo.ingv.it/` respectively.

In order to run a simulation, one file named as follows is needed for each simulated *YYYYMMDD* date:

- `YYYYMMDD_hi-INGV--WAVE-MFSs4e-MED-b${bulletin_date}_fc-fv03.00.nc`

Chapter 6

Running MEDSLIK-II

This chapter describes the steps needed in order to run both an oil spill-slicks simulation and a particle tracking Lagrangian simulation.

All the files needed to specify the initial conditions and the physical parameters which will be used in a simulation as well all the bash scripts which run the model are located inside the directory `RUN`.

- `medslik_inputfile.txt` is the file where initial conditions have to be specified.
- `medslik5.par` is the file where the parameters to be used in the simulation are set.
- `medslik_II.sh` is the bash script which controls the model run.

6.1 Parameter file

`medslik5.par` is the file where the simulation parameters are set.

It is organized in the following sections: *Physical Parameters*, *Evaporation Parameters*, *Emulsification Parameters*, *Dispersion Parameters*, *Spreading Parameters*, *Coastal Impact Parameters* and *Computational Parameters*.

In the **Physical Parameters** section it is possible to control the physics which governs the model activating or not some physical processes, choosing between some simulation options and setting some coefficients values.

Here, it is possible to activate or not the wave-induced current term, the wind-drift term and the wind-drag term of the advection-diffusion equation 2.10.

The **wave-induced current term**, $\boldsymbol{U}_S$, can be simulated by using the Medslik-II Jonswap parameterization or with wave-induced current velocities from wave model outputs. The choice is made writing `01` or `02`, respectively, in the line corresponding to the `Stokes' Drift`. If `00` is written in the same line the wave-induced current term is not activated. If the Medslik-II Jonswap parameterization is chosen (01), then the 4^{th} order Runge-Kutta scheme can not be chosen.

The **wind-drift term**, $\boldsymbol{U}_W$, is activated by setting the `Drift Factor` and the `Drift Angle at zero wind speed` with values different from `0.00` and `0.0` respectively. The `Drift Factor` and the `Drift Angle at zero wind speed` represent the α and β coefficients of the parameterization formula for the wind-drift term (see [4] and [2] for a detailed description).

The **wind-drag term**, $\boldsymbol{U}_D$, is activated by setting the `Drift Factor` with values different from `0.00` and setting the `Drift Angle at zero wind speed` equal to `0.0`. In this case the `Drift Factor` represents the γ coefficient of the parameterization formula for the wind-drag term (see [2] for a detailed description).

The `Variable Drift Angle` flag allows the model to use a drift angle that decreases as the wind speed increases; if this flag is activated (flag set equal to 01), then a wind speed at which the drift angle is reduced by 50 % must then be entered in the line below. The default is that such a reduction is not made.

The `Reduction of Forecast Wind Speed` flag: when using forecast water circulation in a simulation, the forecast water velocities already include the effect of the wind forces on the water surface. It may be thus be considered appropriate in some cases to reduce the wind speed used in the drift formula by a fraction of the wind used in the forecasts. This can be done writing 01 in `effective wind speed` and entering in the line below the reduction fraction (between 0 and 1). The default is that such a reduction is not made.

`Smagorinsky Scheme`: Choosing 01 allows the horizontal diffusivity to be computed from the water currents using the Smagorinsky scheme. The default value is 0.

The default value for the `Horizontal Diffusivity` is 2.0 $m^2 s^{-1}$. A larger value will cause the slick to spread faster.

The model allows the use of two values of `Vertical Diffusivity`: a larger value in the top well-mixed layer and a smaller value below the mixed layer. The defaults are 0.01 and 0.0001 $m^2 s^{-1}$ respectively.

The ocean current fields are extracted from NetCDF files and are given in the the ASCII formatted files in `RUN/INP_DATA/OCE` directory at fixed `depths`. Whatever depths are used in the text files (they can only be changed by modifying the `Extract_II.F90` code) the same depths must be entered in the parameters file. In MEDSLIK-II version 1.02 the ocean current fields are given at the surface, 10 m, 30 m and 120 m.

`Selection of currents for convection of slick`: it is possible to choose between 4 options: 00 for surface currents, 01 for 10 m deep currents, 02 for 30 m deep currents and 03 for 120 m deep currents.

The default strategy is to use surface velocities for advection of the slick and to ignore the wind drift altogether (by setting the `Drift factor` equal to 0.0).

6.2 Input file

`medslik_inputfile.txt` is the file where the simulation initial conditions have to be declared. It has been changed from the version `1.01` in order to account for multiple spills/slicks, the 4^{th} order Runge-Kutta scheme and the particle tracking Lagrangian mode. Now, it comprehends the following bash variables:

- `SIM_NAME` : the name of the simulation. It will be a part of the output directory name.
- `RESTART` : the hour of the simulation restart file.
- `TRACKMODE` : type of the simulation: 0 = Oil fate and transport, 1 = Lagrangian trajectory.

- `OCEAN` : A string which specifies the ocean current forcing type (see the file `forcing_medslik.txt` for a list of the forcing available in the version 1.02).
- `WIND` : A string which specifies the wind forcing type (see the file `forcing_medslik.txt` for a list of the forcing available in the version 1.02).
- `WAVE` : A string which specifies the wave forcing type (see the file `forcing_medslik.txt` for a list of the forcing available in the version 1.02).
- `SIM_LENGTH` : it is a 4 digits integer which indicates the lenght of the simulation.
- `NUM_SCHEME` : it indicates the numerical scheme chosen: Euler = 0, 4^{th} order Runge-Kutta = 1

- `GRID_SIZE` : it indicates the spatial resolution (in meters) of the oil tracer grid.
- `OUTPUT_NAME` : it a 4 letters string which will be a part of the output files name.
- `STEP_OUTPUT` : it represents the time interval for printing results to output files.
- `NSLICK`: it indicates the total number of spills/slicks which will be simulated.
- `SAT_DATA`: it indicates if the simulation will envolve satellite detected slicks or not. It can be equal to "YES" or "NO". For point source/s it has to be always "NO".
- `NAMEFILE_GML` : if the previous variable is set as "YES", this variable indicates the absolute path where the `Xml` file is located.
- `CONTOURSLICK` : it indicates if the simulation will envolve at sight detected slicks, meaning that the user will manually provide, for each simulated slicks, the coordinates of all the corners which approximate the boundary line which identifies and describes the shape of the slicks. It can be equal to "YES" or "NO". For point source/s it has to be always "NO".
- `OIL` : it is a string which indicates how the oil type will be declared, by "NAME" or "API".

- `OIL_TYPE` : depending on the previous variable, it can be a real number indicating the API of the simulated oil or a string which identifies the name of the oil.
- `AGE` : age of the oil spill from initial arrival in the sea.

The next section of variables consists of a list of the **N** oil slicks/spills ordered from the nearest to the farthest in time to the time start of the simulation. For each N^{th} oil slicks/spills it is needed to specify:

- date of the N^{th} (the bold **n** represents an integer which specify the identity of the slick) slick observation/oil released:
  - day: S**n**DD: it is a 2 digit integer
  - month: S**n**MM: it is a 2 digit integer
  - year: S**n**YY: it is a 2 digit integer
- time of the N^{th} (**n**) slick observation/oil released:
  - hour: S**n**HR: it is a 2 digit integer
  - minutes: S**n**MN: it is a 2 digit integer
- spill duration of the N^{th} (**n**) slick/spill, in hours:
  - spill duration: S**n**durath: it is an integer. If the spill is instantaneous or it is a detected slick type 0.
- spill rate in tons/hours of the N^{th} (**n**) slick/slick: if the spill is instantaneous or it is an observed slick write the total tons spilled
  - spill rate: S**n**spllrt: it is an integer. If the spill is instantaneous or it is a detected slick write the total tons spilled
- Longitude and latitude (in decimal degree) of the N^{th} (**n**) slick/spill: For point sources spill, write the coordinates of the N^{th} point (k=1). For polygon detected slicks, write the coordinates for each K^{th} (**k**) corner of the N^{th} slick.
  - Longitude: S**n**lon[**k**]
  - Latitude: S**n**lat[**k**]

Here, an example for 2 quadrilateral detected slicks is given:

```
S1DD=17
S1MM=05
S1YY=14
```

```
S1HR=05
S1MN=38
S1durath=0
S1spllrt=34
S1lon[1]=10.0069
S1lat[1]=42.9656
S1lon[2]=10.0120
S1lat[2]=42.9656
S1lon[3]=10.0120
S1lat[3]=42.9706
S1lon[4]=10.0069
S1lat[4]=42.9706
S2DD=17
S2MM=05
S2YY=14
S2HR=05
S2MN=38
S2durath=0
S2spllrt=1
S2lon[1]=9.9831
S2lat[1]=42.9567
S2lon[2]=9.9881
S2lat[2]=42.9567
S2lon[3]=9.9881
S2lat[3]=42.9617
S2lon[4]=9.9831
S2lat[4]=42.9617
```

Here, an example for a continuous single point spill source is given:

```
S1DD=01
S1MM=11
S1YY=15
S1HR=15
S1MN=08
S1durath=24
S1spllrt=50
S1lon[1]=12.0669
S1lat[1]=43.9371
```

Medslik-II version 1.02 is designed to simulate **instantaneous** or **continuous single spills**, **instantaneous multiple spills** and **single** or **multiple**

detected slicks.

This means that for **multiple spills** S**n**durath has to be set always equal to 0 and that for **single** or **multiple detected slicks** S**n**DD, S**n**MM, S**n**YY, S**n**HR and S**n**MN have to be the same for each N^{th} slicks.

If the **Particle tracking Lagrangian simulations** flag is activated (`TRACKMODE=1`), then the variables `OIL`, `OIL_TYPE`, `AGE`, S**n**durath and S**n**spllrt can be set with any value you want since they are not used in the computation.

6.3 Running the model

After having properly edited the `medslik_inputfile.txt`, modified the `medslik5.par` files and placed the needed forcing files into the right directories, just type the following command to run the model:

```
cd RUN
./medslik_II.sh
```

While the simulations is running, several files needed by the model code are created in the `RUN` directory and they can be checked to understand what is the model state. Here it is a brief description of each run time created files:

- `medslik_pnts.inp` : if point spill sources are simulated, this file contains a list of the geographical coordinates of the spill sources.
- `medslik_plgs.inp`: if polygonal slicks are simulated, this file contains a list of the geographical coordinates of all the corners which describe the slicks shape.
- `files-oce`: it contains informations about the time resolution of the ocean current model outputs forcing fields.
- `files-met`: it contains informations about the time resolution of the wind model outputs forcing fields.
- `files-wav`: it contains informations about the time resolution of the wave-indeuced current model outputs forcing fields.
- `medslik5.inp`: this is the file which is directly read by the model: it contains all the input informations written in the `medslik_inputfile.txt`.

- `medslik.tmp`: it contains the geographical limits of the area affected by the spill.
- `Extract.log`: it is the log-file for the program `Extract_II.F90`.
- `medslik.log`: it is the log-file for the program `medslik_II.for`.
- `coast.map`: if `TRACKMODE=0`, this file contains the coordinates of all the coastal segments which are in the simulated area.

6.4 Output files

Medslik-II output files are located in `OUTPUT/MEDSLIKII_YYYY_MM_DD_hhmm_${SIM_NAME}`, where YYYY is the year, MM the month, DD the day, hh the hour, mm the minutes and `${SIM_NAME}` the name of the simulation.

The folder always contains:

- the input files `medslik_inputfile.txt`, `medslik5.inp`, `medslik.tmp`, `medslik_pnts.inp` and `medslik_plgs.inp`.
- the parameter file `medslik5.par`.
- the `OCE`, `MET` and `WAV` (if waves are used) directories which contains the ASCII formatted data used to force the model.

If an "**Oil fate and transport**" simulation is performed (`TRACKMODE=0`), then the output folder contains also:

- the `medslik.fte` file, which contains the trend over time of: the oil volume spilled, the percentage of oil evaporated, the percentage of oil on the surface, the percentage of oil dispersed, the percentage of oil on the coast, the oil-water emulsion viscosity, the oil viscosity, the oil density, the fraction of water contained in the oil-water emulsion, etc.
- `${OUTPUT_NAME}hhhh.srf` files: they contain the values of the oil concentration on the surface; hhhh indicates the number of hours after the start of the spill.
- `${OUTPUT_NAME}hhhh.dsp` files: they contain the values of the dispersed oil concentration;
- `${OUTPUT_NAME}hhhh.cst` files: they contain the values of the oil concentration on the coast;

- `${OUTPUT_NAME}.nc` file: it contains data about the Oil Concentration, the Ocean and the Wave-induced currents and the Wind velocities fields. Wind velocities are defined at the gravity centre point of each slick. Ocean forcing fields are defined on their original grids. Oil concentration is defined on the 150m×150m grid.

If a "**Lagrangian trajectory**" simulation is performed (`TRACKMODE=1`), then the output folder contains:

- `${OUTPUT_NAME}.trj`: it contains the position in time (trajectory) of the gravity centre of the particles used in the advection-diffusion simulation.

Chapter 7

Test cases

In order to test the correct operation of the MEDSLIK-II version 1.02 system, the meteo-oceanographic data and input files needed to run 3 test cases are provided. In this chapter, the procedure for each test case to download the data and to set-up the model is described.

At MEDSLIK-II web page (`http://medslikii.bo.ingv.it`), after registration and be logged in it is possible to access the `Version_1.02` homepage (just clicking the corresponding link). Here 5 links exist: the first is to **Download the model**, the second is to **Download the User Manual** and the latter three are to download the test cases data.

7.1 Algeria test case

This test case simulates 3 satellite detected oil slicks off the Algerian coast (August 6-10, 2008) certified by REMPEC. A detailed description of this experiment can be found in [3].

The data needed to run this test case can be downloaded through the `Algeria_test_case` link at the MEDSLIK-II `Version_1.02` homepage.

After the download is finished, it is needed to go to the `${DOWNLOAD_DIRECTORY}` and type the following commands:

```
cd ${DOWNLOAD_DIRECTORY}
tar -xvzf Algeria_test_case.tar.gz
cd Algeria_test_case
cp OCE/* ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/OCE/
cp MET/* ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/MET/
gunzip ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/*/*
```

This procedure will extract the `Algeria_test_case.tar.gz` archive and will copy the meteo-oceanographic data needed for the simulation into the right directories. The last command will extract the gzipped NetCDF data files.

After that, it is needed to copy the set-up files (`medslik_inputfile.txt` and `medslik5.par`) into the `RUN` directory:

```
cp SET-UP/medslik_inputfile.txt ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
cp SET-UP/medslik5.par ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
```

This will overwrite the `medslik_inputfile.txt` and `medslik5.par` files located in the `RUN` directory. A default template of these two files can be found in the `RUN/template` directory.

To run the model, just type

```
./medslik_II.sh
```

The output directory will be located at `${INSTALLATION_DIR}/MEDSLIK-II_v1.02/OUTPUT`.

Inside the `${DOWNLOAD_DIRECTORY}/Algeria_test_case/OUTPUT/` directory you can find the reference output and plot files which can be compared to the local system outputs to check the correct operation of the model.

The oil initial conditions, physical and numerical parameters and meteo-oceanographic forcings used in this test case simulation are listed in table 7.1.

Table 7.1: Oil initial conditions, physical-numerical parameters and meteo-oceanographic forcings used in Algeria test case

Start date	06-06-2008
Start hour	09:51
Simulation length	36 hours
Ocean Forcing	MFS hourly mean fields
Wind Forcing	ECMWF 1/2 degree resol.
Wave Forcing	NONE
Physical processes	MEDSLIK-II JONSWAP param.
Number of Slicks	3
API	22
TONS of 1^{st} slick	240 tons
TONS of 2^{nd} slick	225 tons
TONS of 3^{rd} slick	215 tons
NUM SCHEME	EULER

Figure 7.1 represent output maps after 25 hours of simulation performed with the Euler Forward scheme.

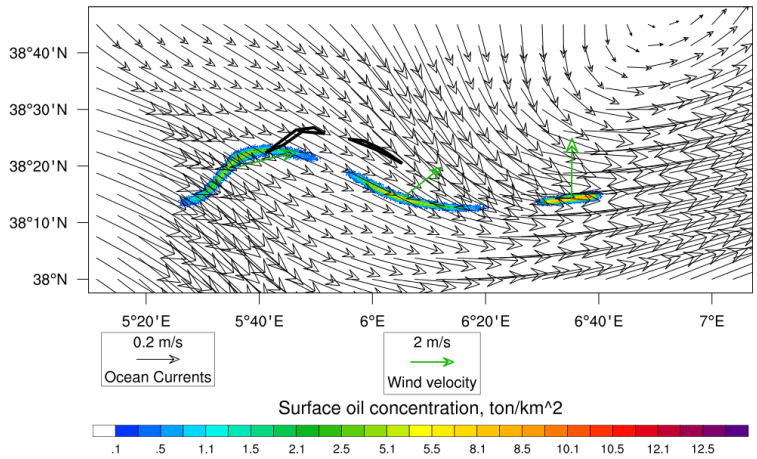


Fig. 7.1: Output maps after 25 hours of simulation performed with the Euler Forward scheme. It represents oil-slick concentration, current fields (black arrows), wind in the gravity centre of each slick (green arrows) and satellite observations (black slicks)

.

7.2 Lebanon test case

This test case simulates the Lebanon coastal spill caused by air raids on coastal power plants on July-August 2006. A detailed description of this experiment can be found in [1].

The data needed to run this test case can be downloaded through the `Lebanon_test_case` link at the MEDSLIK-II `Version_1.02` homepage.

The instructions to run this test case simulation are given in the following list:

```
cd ${DOWNLOAD_DIRECTORY}
tar -xvzf Lebanon_test_case.tar.gz
cd Lebanon_test_case
cp OCE/* ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/OCE/
cp MET/* ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/MET/
gunzip ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/*/*
cp SET-UP/medslik_inputfile_euler.txt ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/medslik_inputfile.txt
cp SET-UP/medslik5.par ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
cd ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
./medslik_II.sh
```

This set of command will run a simulation with the Euler Forward numerical scheme activated.

In order to run a simulation with the 4^{th} order Runge-Kutta numerical scheme activated it is needed just to type the following commands:

```
cp SET-UP/medslik_inputfile_runge.txt ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/medslik_inputfile.txt
cp SET-UP/medslik5.par ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
cd ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
./medslik_II.sh
```

The output directories will be:

`${INSTALLATION_DIR}/MEDSLIK-II_v1.02/OUTPUT/EULER`
`${INSTALLATION_DIR}/MEDSLIK-II_v1.02/OUTPUT/RUNGE`

Inside the `${DOWNLOAD_DIRECTORY}/Lebanon_test_case/OUTPUT/` directory you can find the reference output and plot files which can be compared to the local system outputs to check the correct operation of the model.

N.B.: Because of the simulated period is very long (660 hours), the computational times of the simulation and the NetCDF creation can be very long: it is due to the fact that the re-gridding algorithm has to process a big amount of grid points, given that the grid on which the oil concentration is calculated is a 150m $\times$ 150m grid.

The oil initial conditions, physical and numerical parameters and meteo-oceanographic forcings used in this test case simulation are listed in table 7.2.

Figures 7.2 and 7.3 represents output maps after 147 and 216 hours of

Table 7.2: Oil initial conditions, physical-numerical parameters and meteo-oceanographic forcings used in Lebanon test case

Start date	13-07-2006
Start hour	08:00
Simulation length	660 hours
Ocean Forcing	MFS hourly mean fields
Wind Forcing	ECMWF 1/2 degree resol.
Wave Forcing	NONE
Physical processes	WIND DRAG = 1.2 %
Number of Spills	1
API	20
Duration of the release	144 hours
Spill rate	130.35 tons/hours
NUM SCHEME	EULER and RUNGE-KUTTA

simulation, respectively, both performed with the Euler Forward and the 4^{th} order Runge-Kutta schemes, respectively (a) and (b) in each figure.

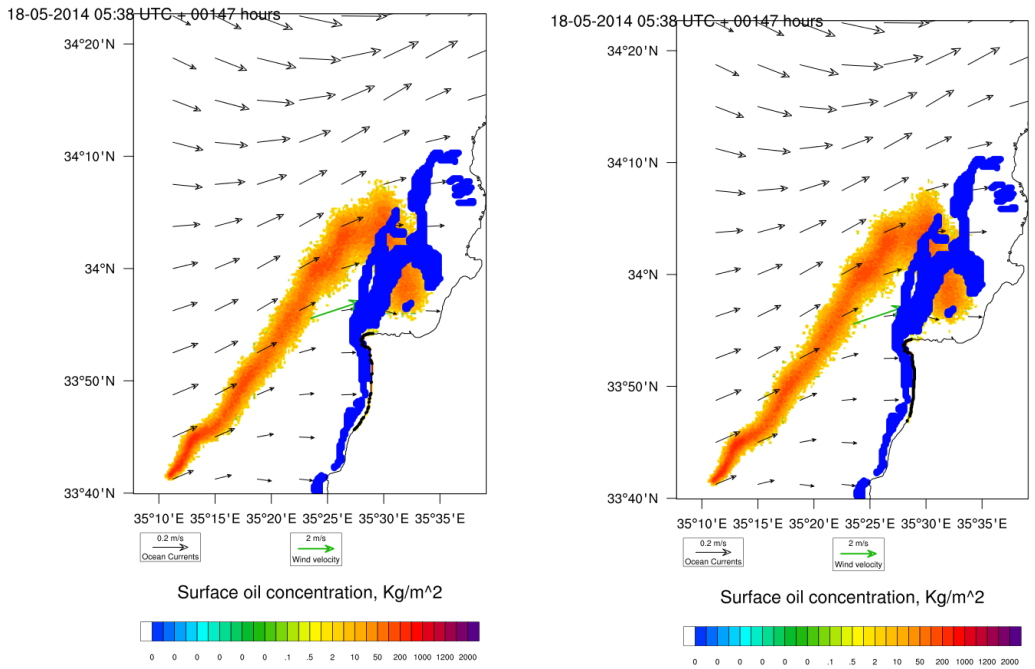


Fig. 7.2: Output maps after 147 hours of simulation performed with the Euler Forward and the 4^{th} order Runge-Kutta numerical schemes (respectively (a) and (b)). They represent sea surface oil-slick concentration, coastal oil impact (bold black coast line), current fields (black arrows), wind in the gravity centre of each slick (green arrows) and satellite detected observations on 19-07-2006 at 10:35 (blue slicks).

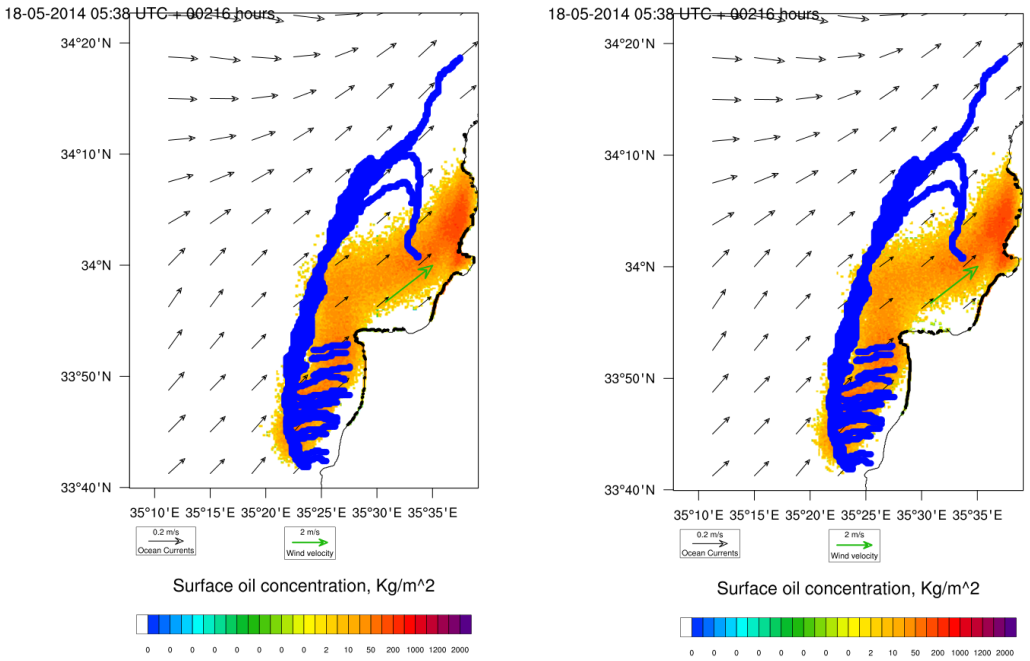


Fig. 7.3: Output maps after 216 hours of simulation performed with the Euler Forward and the 4^{th} order Runge-Kutta numerical schemes (respectively (a) and (b)). They represent sea surface oil-slick concentration, coastal oil impact (bold black coast line), current fields (black arrows), wind in the gravity centre of each slick (green arrows) and satellite detected observations on 22-07-2006 at 07:55 (blue slicks).

7.3 Elba island - Serious Game test case

This test case comprehends 2 experiments: the first simulates 7 satellite detected oil slicks on May 17^{th} 2014 certified by Italian Coast Guard while the second simulates 6 drifters released inside the oil slicks. The data to validate the result have been collected during the MEDEDESS4MS Elba Island Serious Game ((`http://www.medess4ms.eu/`).

The data needed to run this test case can be downloaded through the `Serious_Game_test_case` link at the MEDSLIK-II `Version_1.02` homepage.

Figure 7.4 shows the EMSA satellite detected slicks image provided by the Italian Coast Guard.

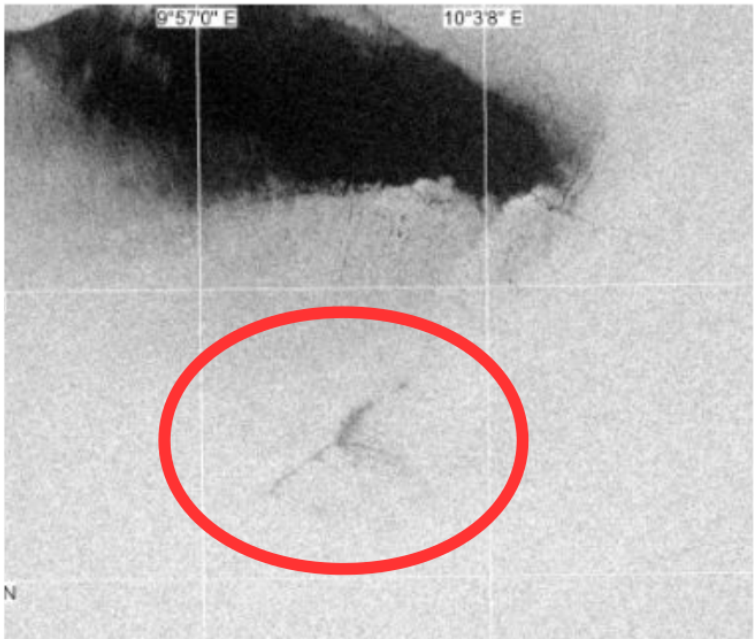


Fig. 7.4: Elba Island MEDESS4MS Serious Game EMSA satellite detected slicks image provided by the Italian Coast Guard

.

OIL SLICKS SIMULATION

The procedure to run this test case simulation is identical to the previous one, except for the name of the simulation and the wave data:

```
cd ${DOWNLOAD_DIRECTORY}
tar -xvzf Serious_Game_test_case.tar.gz
cd Serious_Game_test_case
cp OCE/* ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/OCE/
cp MET/* ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/MET/
cp WAV/* ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/WAV/
gunzip ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/*/*
cp SET-UP/medslik_inputfile_euler.txt ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/medslik_inputfile.txt
```

or

```
cp SET-UP/medslik_inputfile_runge.txt ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/medslik_inputfile.txt
```

and

```
cp SET-UP/medslik5.par ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
cd ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
./medslik_II.sh
```

The output directories will be:

`${INSTALLATION_DIR}/MEDSLIK-II_v1.02/OUTPUT/EULER`
`${INSTALLATION_DIR}/MEDSLIK-II_v1.02/OUTPUT/RUNGE`

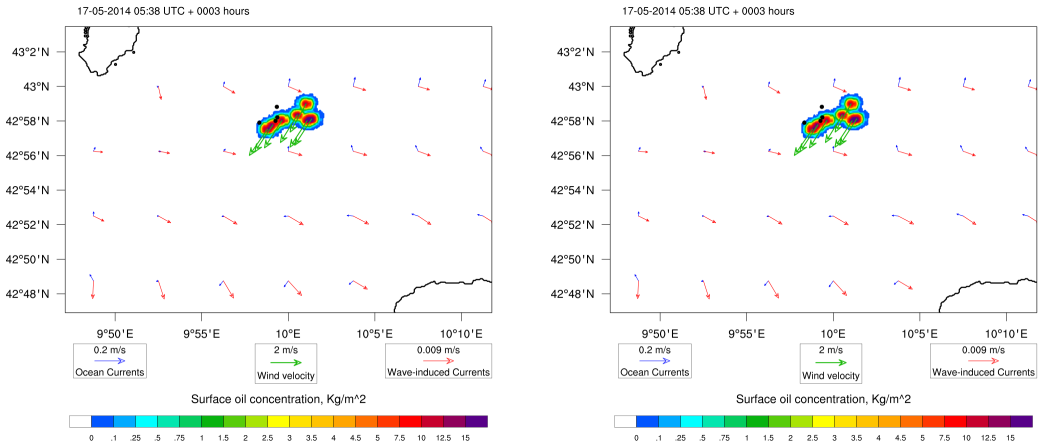
Inside the `${DOWNLOAD_DIRECTORY}/Serious_Game_test_case/OUTPUT/OIL_EXP` directory you can find the reference output and plot files which can be compared to the local system outputs to check the correct operation of the model.

The oil initial conditions, physical and numerical parameters and meteo-oceanographic forcings used in this test case simulation are listed in table 7.3.

Figures 7.5 and 7.6 represents output maps after 3 and 6 hours of simulation, respectively, both performed with the Euler Forward and the 4^{th} order Runge-Kutta schemes, respectively (a) and (b) in each figure.

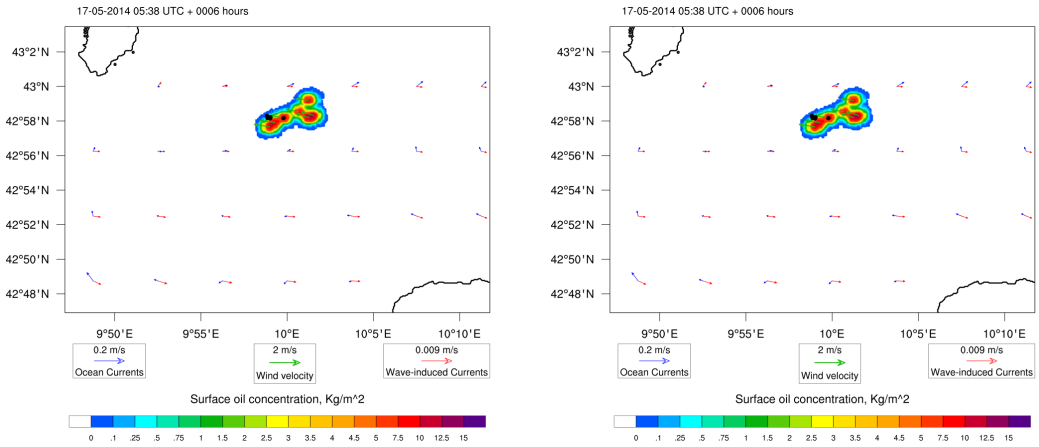
Table 7.3: Oil initial conditions, physical-numerical parameters and meteo-oceanographic forcings used in Serious Game test case

Start date	17-05-2014
Start hour	05:38
Simulation length	24 hours
Ocean Forcing	MFS hourly mean fields
Wind Forcing	ECMWF 1/4 degree resol.
Wave Forcing	MFS-WW3 hourly fields (snapshot)
Physical processes	NONE
Number of Spills	7
API	22
TONS of 1^{st} slick	34 tons
TONS of 2^{nd} slick	1 tons
TONS of 3^{rd} slick	6 tons
TONS of 4^{th} slick	8 tons
TONS of 5^{th} slick	3 tons
TONS of 6^{th} slick	6 tons
TONS of 7^{th} slick	5 tons
NUM SCHEME	EULER and RUNGE-KUTTA



(a) (b)

Fig. 7.5: Output maps after 3 hours of simulation performed with the Euler Forward and the 4^{th} order Runge-Kutta numerical schemes (respectively (a) and (b)). They represent sea surface oil-slick concentration, current fields (black arrows), wind in the gravity centre of each slick (green arrows), wave-induced current fields (red arrows) and the locations where oil slick samples were taken on 17-05-2014 at 08:00 (black dots).



(a) (b)

Fig. 7.6: Output maps after 6 hours of simulation performed with the Euler Forward and the 4^{th} order Runge-Kutta numerical schemes (respectively (a) and (b)). They represent sea surface oil-slick concentration, current fields (black arrows), wind in the gravity centre of each slick (green arrows), wave-induced current fields (red arrows) and the locations where 6 drifters were released inside the oil slick on 17/05/2014 at 11:15 (black dots).

DRIFTER SIMULATIONS

In order to run this test case it is needed to follow the instructions listed below:

```
cd ${DOWNLOAD_DIRECTORY}
tar -xvzf Serious_Game_test_case.tar.gz
cd Serious_Game_test_case
cp OCE/* ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/OCE/
cp MET/* ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/MET/
cp WAV/* ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/WAV/
gunzip ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/DATA/*/*
cp SET-UP/medslik5_0.par ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
cp SET-UP/medslik5_1.par ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
cp SET-UP/medslik5_2.par ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
cp -r drifters_observations ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
cd ${INSTALLATION_DIR}/MEDSLIK-II_v1.02/RUN/
```

The `medslik5_0.par` is the parameter file to run a simulation without activating the wave-induced currents physical process.

The `medslik5_1.par` is the parameter file to run a simulation with the MEDSLIK-II JONSWAP parameterization for the wave-induced currents physical process activated.

The `medslik5_2.par` is the parameter file to run a simulation with the wave-induced current fields provided by wave model output files.

The `SG_drifter_test_case` can be started just executing the following command:

```
./RUN_drifter_TESTCASE.sh
```

This script will excute 6 drifter simulations using both the Euler Forward and the 4^{th} order Runge-Kutta numerical schemes: simulations with the Euler Forward scheme activated are controlled by the script `euler_RUN_drifter_TESTCASE.sh` while simulations with the 4^{th} order Runge-Kutta scheme are launched by the `runge_RUN_drifter_TESTCASE.sh`

The output directories will be:

```
${INSTALLATION_DIR}/MEDSLIK-II_v1.02/OUTPUT/EULER_SG_drifter_test_case
${INSTALLATION_DIR}/MEDSLIK-II_v1.02/OUTPUT/RUNGE_SG_drifter_test_case
```

Inside the `${DOWNLOAD_DIRECTORY}/Serious_Game_test_case/OUTPUT/DRIFTER_EXP` directory you can find the reference output and plot files which can be compared to the local system outputs to check the correct operation of the model.

Figure 7.7 shows the results for simulations with the Euler Forward scheme activated. In figure (a) the 48 hours simulated trajectories are compared to the BEACON real one. Sub-figure (b) shows the results for a 48 hours simulation compared to the Italian Coast Guard SLDMB (Self-Located Datum Marker Buoy) real trajectory. In figure (c) and (d) the 24 hours simulated trajectories are compared to 2 CODE and 2 I-SPHERE real trajectories, respectively.

Figures 7.8 shows the results for the same kind of experiments except for the fact that simulations are performed with the 4^{th} order Runge-Kutta scheme activated instead of the Euler one.

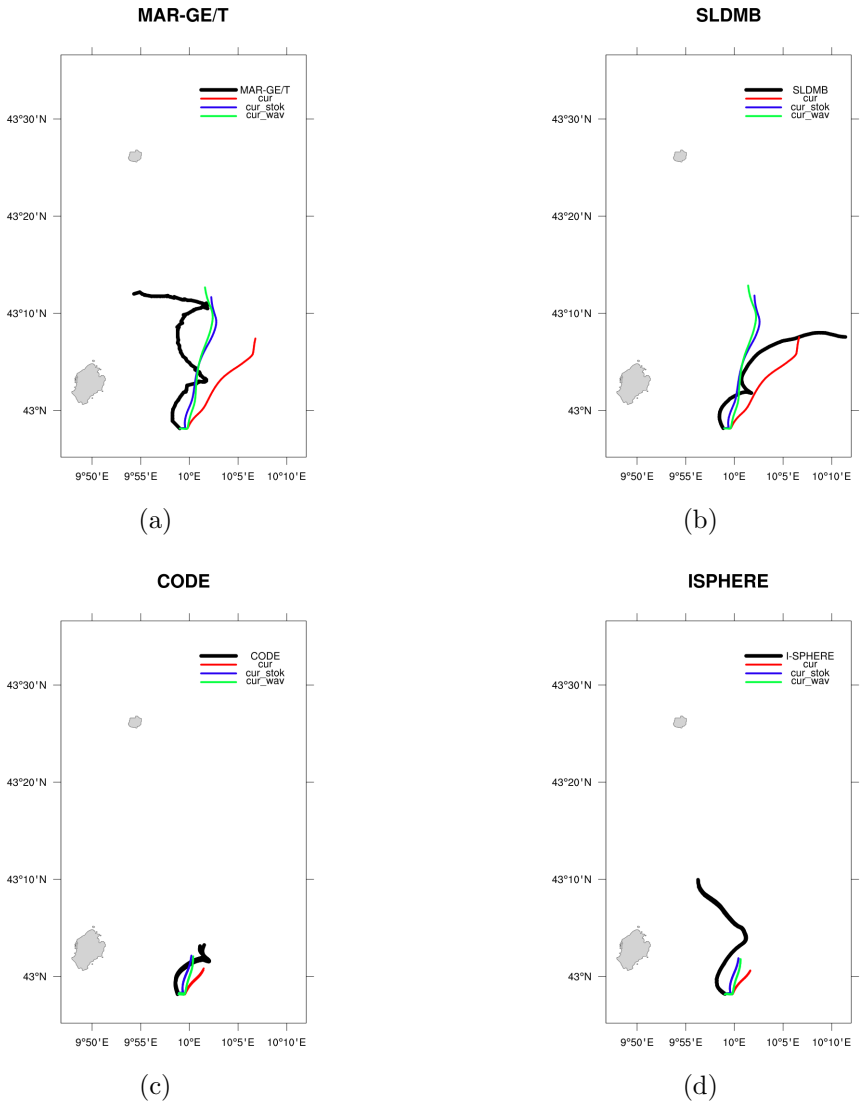


Fig. 7.7: Results for simulations with the Euler Forward scheme activated. In (a) the 48 hours simulated trajectories are compared to the BEACON real one. In (b) the results for 48 hours simulations are compared to the Italian Coast Guard SLDMB (Self-Located Datum Marker Buoy) real trajectory. In figure (c) and (d) the 24 hours simulated trajectories are compared to 2 CODE and 2 I-SPHERE real trajectories, respectively.

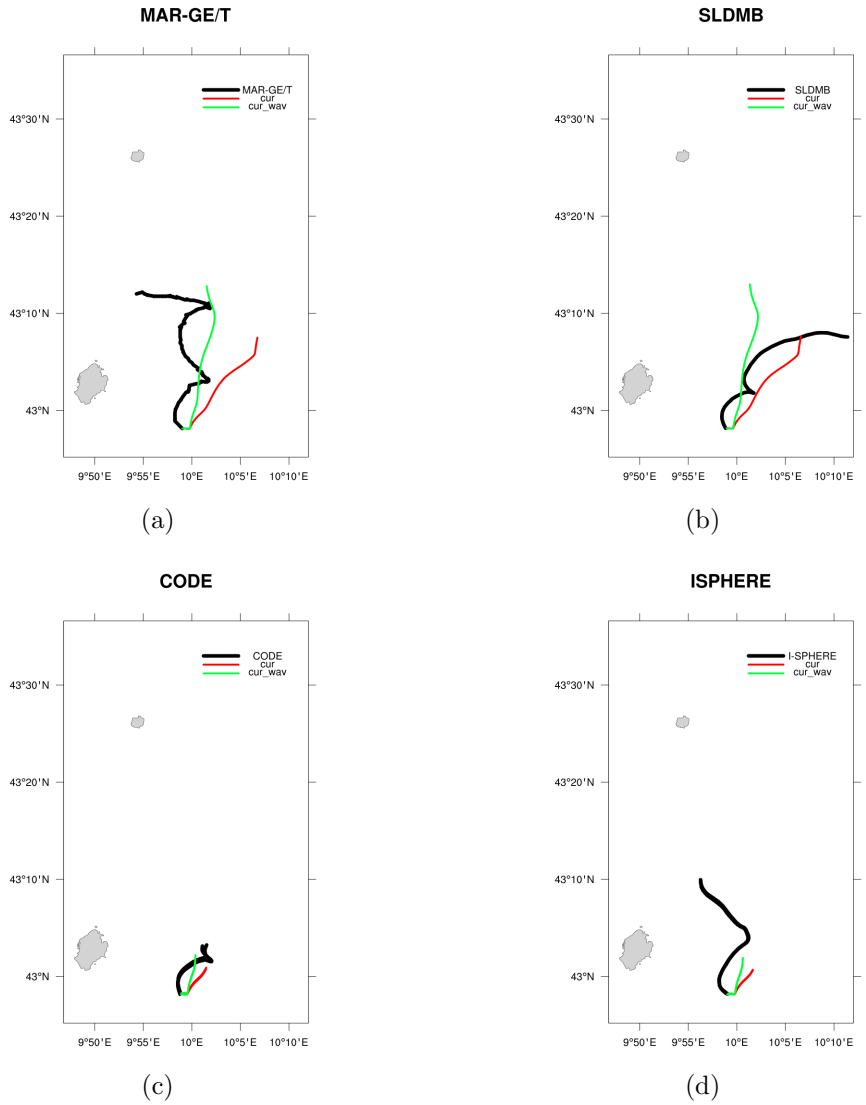


Fig. 7.8: Results for simulations with the 4^{th} order Runge-Kutta scheme activated. In (a) the 48 hours simulated trajectories are compared to the BEACON real one. In (b) the results for 48 hours simulations are compared to the Italian Coast Guard SLDMB (Self-Located Datum Marker Buoy) real trajectory. In figure (c) and (d) the 24 hours simulated trajectories are compared to 2 CODE and 2 I-SPHERE real trajectories, respectively.

Bibliography

[1] G. Coppini, M. De Dominicis, G. Zodiatis, R. Lardner, N. Pinardi, R. Santoleri, S. Colella, F. Bignami, D. R. Hayes, D. Soloviev, G. Georgiou, and G. Kallos. Hindcast of oil-spill pollution during the lebanon crisis in the eastern mediterranean, julyaugust 2006. *Mar. Pollut. Bullet.*, 62(1):140–153, 2011.

[2] M. De Dominicis, S. Falchetti, F. Trotta, N. Pinardi, L. Giacomelli, E. Napolitano, L. Fazioli, R. Sorgente, P. Haley, P. Lermusiaux, F. Martins, and M. Cocco. A relocatable ocean model in support of environmental emergencies - the costa concordia emergency case. *Geosciences Model Development*, in press.

[3] M. De Dominicis, N. Pinardi, G. Zodiatis, and R. Archetti. Medslik-ii, a lagrangian marine surface oil spill model for short-term forecasting - part 2: Numerical simulations and validations. *Geosciences Model Development*, 6:1871–1888, 2013.

[4] M. De Dominicis, N. Pinardi, G. Zodiatis, and R. Lardner. Medslik-ii, a lagrangian marine surface oil spill model for short-term forecasting - part 1: Theory. *Geosciences Model Development*, 6:1851–1869, 2013.

[5] United States Environmental Protection Agency (EPA) Office of Emergency and Remedial Response (OSWE). *Understanding Oil Spills And Oil Spill Response*. United States Environmental Protection Agency (EPA), December 1999.

Appendix A

License and Terms

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these

rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

Terms and Conditions

0. Definitions.

   "This License" refers to version 3 of the GNU General Public License.

   "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

   The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

   A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

   The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of

packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not

control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

   No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

   When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

   You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

   You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

   You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   (a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

(b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

(c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

(d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

(a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

(b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to

give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

(c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

(d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

(e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of

the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be

treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

(a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

(b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

(c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

(d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

(e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

(f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further

restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

   You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

   You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring

solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

    Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

    An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

    You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

    A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

    A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition,

"control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third

party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program

specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPER-

ATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

    If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

End of Terms and Conditions