FVCOM Clyde Sea Hydrodynamic Model User Guide and Technical Report

Alessandro Damiano Sabatino, Ian Thurlbeck

March 15, 2016

1 Introduction

The Clyde Sea model was developed initially to simulate the dispersal of the shellfish larvae in the Firth of Clyde. The main output of this project is a full hydrodynamic model baroclinic model with surge hindcasting capabilities.
The model was made using the Finite-Volume Community Ocean Model (FVCOM), that was developed by Chen et al. (2003) in the University of Massachussets-Darmouth, and successively upgraded by a joint team from UMassD and WHOI-MIT in 2006 (Chen et al., 2006a). The FVCOM is an unstructured grid, finite-volume, primitive equations hydrodynamic model. More general information about the model can be found in the FVCOM User Manual (Chen et al., 2006b), in which the general setting of the FVCOM, along with a detailed guide for the set-up of the model is provided. The version in which the Clyde Sea model was built is the 3.1.6.

2 FVCOM Inputs

FVCOM requires the following inputs, that should be located in the INPUT folder: a computational grid, in which the geographic information of the nodes and the elements of the model are specified. This can be done both in Cartesian or in Lat/Lon. The present model is in Cartesian coordinates (Ordnance Survey coordinates); a bathymetry file in which the information about the depth at any node is given; a file containing the latitude at each node (for the implementation in the model of the Coriolis force); a file containing information about the sponge layer; a file for the number of layers in the model; and the input files that contains the physical forcing for the model. In Table 1 are reported all the names and the specification of the file required.
In addition a run, with the general information of the model along with the

names of the input file, should be provided along with an *mpdhosts* file containing the names of the nodes in which the cluster should run if the parallel code is used.

File	Content
ASCII files	
xxx_run.nml	Run file, contains information about the timestep and the set-up
xxx_grd.dat	Contains information about grid, in particular the position of the node and the connectivity between nodes.
xxx_dep.dat	Contains the depth of each node
xxx_cor.dat	Contains the latitude of each node to calculate Coriolis effect
xxx_obc.dat	Number and position of the open boundary nodes.
xxx_spg.dat	Contains the information about sponge layer at open boundaries
xxx_sigma.dat	Specify the number of vertical layer of the model
River_Namelist.nml	Contains the information about the number and the position of the rivers in the model.
NetCDF files	
xxx_julian_obc.nc	Water elevation at open boundary nodes
xxx_wnd.nc	Contains information at each node of the wind, the mean sea level pressure, the temperature, the humidity, the evaporation/precipitation and the radiation heating fluxes
xxx_restart.nc	File used for the hotstart option in the run file. If a previous simulation is available for the first day of the specified simulation, the model can start with those conditions implemented.
xxx_riv.nc	River discharge of the rivers specified in *River_Namelist.nml*
xxx_tsobc.nc	Salinity and temperature conditions at open boundary for the advection of temperature inside the model from the open boundaries.

Table 1: List of files required for the FVCOM run

2.1   SMS generated unstructured grid

The FVCOM run on an unstructured grid. This means that the elements of the grid are not squared as the old hydrodynamic model, but are triangular. This have an obvious advantage due to the flexibiliy of the mesh in following complex coasts as well as having very high resolution (Ferziger and Perić, 2002). The FVCOM software does not have a grid generator software included in the code.

To generate the unstructured grid for the Clyde Sea model we used the Surface-water Modelling System (SMS) by Acquaveo. The SMS is a commercial software that is used for oceanographic application and is one of the software that is suggested in the FVCOM User Manual for the generation of the computational grid in FVCOM (Chen et al., 2006b).

FVCOM does have some strict requirements for the quality of the mesh, in order to have a stable computation of the oceanographic flow:

- Minimum interior angle of each element: 30°

- Maximum interior angle of each element: 120°

- Maximum slope: 0.1

- Element area change: 0.5

- Connecting elements: should be less than 8 for each node

In addition the open boundaries of the computational grid should be put far away from the area in study to avoid unwanted numerical noise in the area. The main input for the SMS software for the generation of the grid is the coastline. We used two different coastlines for the Clyde Sea model: the NOAA Global Self-consistent, Hierarchical, High-resolution Geography Database (GSHHG) (Wessel and Smith, 1996) that is freely available under the GNU Lesser General Public license and the high resolution SeaZone coastline, that is not freely available. The licence for the SeaZone coastline was purchased for this project by the Scottish Environmental Protection Agency (SEPA).

Since we choose the Cartesian coordinates to solve the equations, a conversion between latitude and longitude nodes of the SeaZone and the GSHHG data to some Cartesian system (in meters) was required. For this purpose we used the *MatLab ll2os* script, that converts the Latitude and the Longitude in Easting and Northing. The *ll2os* script and its inverse *os2ll* that is also used for the pre-processing and the post-processing of the input files are available from the MatWorks website. In addition, a pre-processing was needed to convert the file from the GSHHG format to a format that SMS could read. This was made using a *FORTRAN 77* script called *readcst_new.f* that was included in the FVCOM code.

Once the coastline is loaded in SMS, is possible to change the resolution of the coastline. This allow the user to specify areas with an enhanced resolution to have a better description of the flow.

For the Clyde Sea model we decided to fix the open boundaries in the Irish Sea and in the Atlantic Ocean and to have an enhanced resolution in the area near the Clyde river estuary and in the sealochs such as the Loch Fyne. The resulting grid is composed of 30644 nodes, 56131 elements and extends from about the Isle of Mull in the North, to the central part of Irish Sea to the South. The spatial resolution in the Clyde and in North Channel is very fine: 1 km for the North Channel area, 600-800 m for the southern part of Clyde Sea, 300-400 m

for the central part and about 100 m for the northern sealochs and the estuary. The resolution at the open boundaries is approximately 7 km. The number of open boundary nodes are 83. In Figure 1 the computational grid is showed.
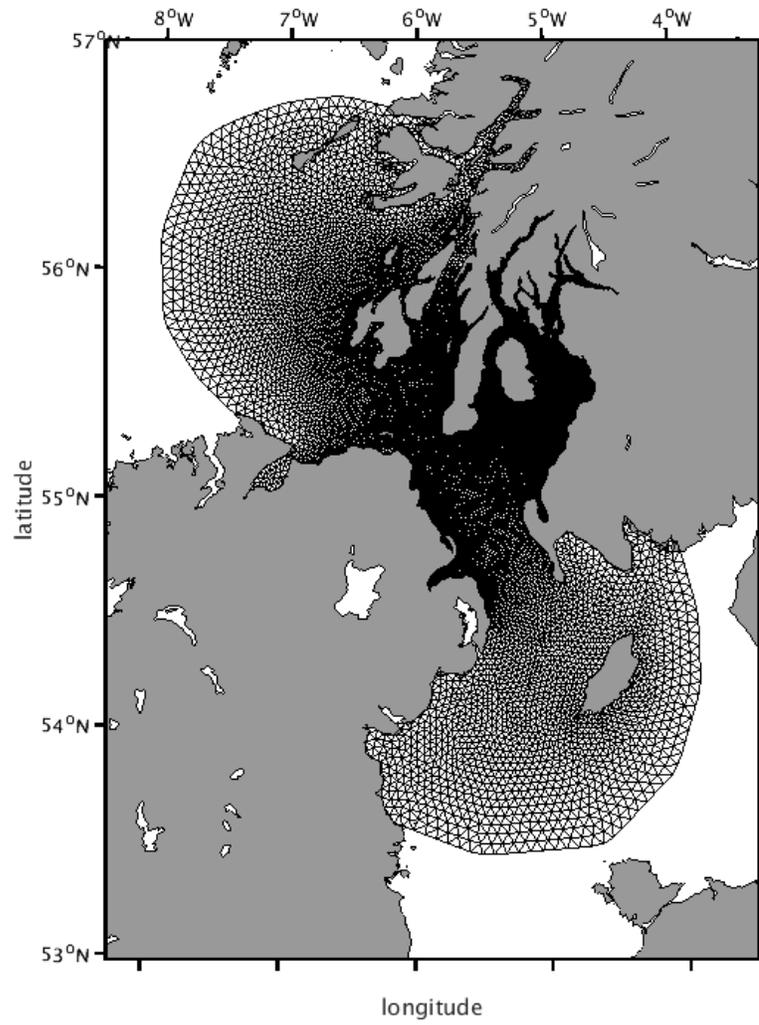
Figure 1: The computational grid for the Clyde Sea model generated with the SMS software

The main output of this procedure is the *xxx_grd.dat* file: the SMS in fact output a file that contains the geographic information of the nodes of the computational grid and the information about the connectivity of the nodes (triangular elements): each element is specified as a triplet of the number of the nodes.

The following step for the grid specification is the bathymetry. The SMS software is then used to generate the *xxx_dep.dat*: the information of the bathymetry are inserted in the SMS using the GUI. The bathymetry for the Clyde Sea model come from two sources: the General Bathymetric Chart of the Oceans (GEBCO), that is a freely available global-scale bathymetry with a resolution of 0.5' x 0.5' and the licensed SeaZone bathymetry, that covers area of the Clyde Sea at very high resolution. The SMS was used to interpolate these two sources on the computational grid. The resulting output is a file with three columns: the first two columns are the geographic locations of the nodes, while the third is the bathymetry, with the oceanographic convenction that the depth is positive going deeper.

Inside SMS is possible to export lines of files to text files. We selected the nodes at the open boundary and we exported the corresponding number of each node. Then these nodes are used to generate the *xxx_obc.dat* file. The last file that is generated in this procedure is the *xxx_cor.dat*: the location of the nodes in the *xxx_grd.dat* (or in the *xxx_dep.dat*, since are the same) are transformed in latitude/longitude using the *os2ll* MatLab script. The format of the *xxx_cor.dat* is the same as the *xxx_dep.dat*, with three column, but the third column is the latitude instead of the depth.

The choice of the number of vertical layers and their distribution troughout the water column can be done in the *xxx_sigma.dat*, that is a straigthforward text file.

2.2   OTPS generated tidal predictions

After the generation of the files for the grid specification, the next step is the generation of the main input files containing the forcing. Those files are in NetCDF format, that is a widely used format for geophysics and science in general, that is very easy to be generate using MatLab and R. The tidal forcing is implemented in the model using the open-source software OTPS (Oregon State University Tidal Prediction Software) that is based on satellite osbervations of the water level and on measurement from tide gauges (Egbert et al., 1994; Egbert and Erofeeva, 2002). Some regional solutions were developed for specific areas of the world. In particular the so-called 'European Shelf' encompasses with a good resolution (1/30° resolution) the Great Britain (Egbert et al., 2010).

The set-up of the NetCDF files *xxx_julian_obc.nc* is here described. First a *MatLab* script called *madness.m*, that need as input the *xxx_obc.dat* without the header, generates 83 files: one for each open boundary node. These 83 files contain the latitude and the longitude of the open boundary node and one year of date.

```
53.564005    -6.0134592    2005    1    1    0    0     0
53.564005    -6.0134592    2005    1    1    0    15    0
53.564005    -6.0134592    2005    1    1    0    30    0
53.564005    -6.0134592    2005    1    1    0    45    0
53.564005    -6.0134592    2005    1    1    1    0     0
53.564005    -6.0134592    2005    1    1    1    15    0
53.564005    -6.0134592    2005    1    1    1    30    0
53.564005    -6.0134592    2005    1    1    1    45    0
53.564005    -6.0134592    2005    1    1    2    0     0
53.564005    -6.0134592    2005    1    1    2    15    0
53.564005    -6.0134592    2005    1    1    2    30    0
```

Figure 2: An example of an output file from the *madness.m* MatLab script

The those 83 files are automatically put into a folder called *Obcf*, that have to be generated before running the script. To change the year of the script it can be done using the 'Find and Replace' tool in MatLab. Then a second folder should be created by the user called *Output* and a script called *run_all.sh* is runned from the Terminal. This file run the OTPS software and generate a 83 files with the tidal amplitude for all the 83 locations.

```
53.5640    -6.0135    01.01.2005 00:00:00    0.124    17.103
53.5640    -6.0135    01.01.2005 00:15:00    0.288    17.103
53.5640    -6.0135    01.01.2005 00:30:00    0.444    17.103
53.5640    -6.0135    01.01.2005 00:45:00    0.590    17.103
53.5640    -6.0135    01.01.2005 01:00:00    0.723    17.103
53.5640    -6.0135    01.01.2005 01:15:00    0.842    17.103
53.5640    -6.0135    01.01.2005 01:30:00    0.943    17.103
53.5640    -6.0135    01.01.2005 01:45:00    1.027    17.103
53.5640    -6.0135    01.01.2005 02:00:00    1.091    17.103
53.5640    -6.0135    01.01.2005 02:15:00    1.135    17.103
53.5640    -6.0135    01.01.2005 02:30:00    1.158    17.103
```

Figure 3: An example of the OTPS output file generated from the Figure 2 file

The Figure 3 shows the output file from the OTPS software. The fifth column of the file is the predicted tidal elevation in this point from the OTPS at that given time. Then, a new MatLab script called *untitled.m* that should be in the *Output* folder is run to unify the 83 files in one text file called *xxx_elj_obc.dat* that was the old text file used in the version 2 of FVCOM as input for the tidal forcing at the boundaries. This is an intermediate file that can be directly transformed in a NetCDF file for the tidal forcing or can be modified in order to take account for the surge wave from the boundary. For this second path some tide gauge data are used in order to account for the surge wave (Table 2)

The data for these tide gauges can be downloaded from the British Oceanographic Data Centre (BODC) website free of charge. An example of the tide gauges data is shown in the Figure 4.

After the number of the observation, the date and the time, there are two columns: one is the total elevation recorded, the second is the estimated residuals. This is the column that is considered in the set-up of the surge boundaries.

Tide Gauge	Latitude (°)	Longitude (°)
Southern boundary		
Holyhead	53.308	-4.631
Workington	54.651	-3.568
Llandudno	53.332	-3.825
Port Erin*	54.09	-4.77
Northern boundary		
Tobemory	56.623	-6.064
Port Rush*	55.206	-6.657
Port Ellen*	55.63	-6.19

Table 2: List of tide gauges used for calculating the residuals. * those tide gauges are inside the model boundary and were also used for the validation of the model

```
Port:            P049
Site:            Millport
Latitude:        55.74964
Longitude:       -4.90583
Start Date:      01JAN2006-00.00.00
End Date:        31DEC2006-23.45.00
Contributor:     National Oceanography Centre, Liverpool
Datum information: The data refer to Admiralty Chart Datum (ACD)
Parameter code:  ASLVBG02 = Surface elevation (unspecified datum) of the water body by bubbler tide gauge
  Cycle    Date      Time     ASLVBG02    Residual
 Number yyyy mm dd hh mi ssf      f          f
     1) 2006/01/01 00:00:00     3.2210     -0.0146
     2) 2006/01/01 00:15:00     3.2690     -0.0039
     3) 2006/01/01 00:30:00     3.2940      0.0029
     4) 2006/01/01 00:45:00     3.2920      0.0034
     5) 2006/01/01 01:00:00     3.2610     -0.0033
     6) 2006/01/01 01:15:00     3.2110     -0.0068
     7) 2006/01/01 01:30:00     3.1390     -0.0104
     8) 2006/01/01 01:45:00     3.0440     -0.0163
     9) 2006/01/01 02:00:00     2.9310     -0.0214
    10) 2006/01/01 02:15:00     2.7970     -0.0312
    11) 2006/01/01 02:30:00     2.6370     -0.0536
    12) 2006/01/01 02:45:00     2.4750     -0.0672
    13) 2006/01/01 03:00:00     2.3050     -0.0806
    14) 2006/01/01 03:15:00     2.1350     -0.0878
    15) 2006/01/01 03:30:00     1.9550     -0.1005
    16) 2006/01/01 03:45:00     1.7640     -0.1210
    17) 2006/01/01 04:00:00     1.5680     -0.1451
    18) 2006/01/01 04:15:00     1.3820     -0.1601
```

Figure 4: An example of a wave gauge data file downloaded from the BODC

It is possible to estimate directly also the residuals, without using the last column from the wave gauge file: using the penultimate column of the file is possible to estimate the harmonic components using the *ut_solv.m* script present in the *UTide* Matlab package (Codiga, 2011) and then reconstruct the tidal component only using the *ut_predict.m* script. In both cases, before running any MatLab script is necessary to change the $-99.000N$ data that could be present in the tide gauge file in NaN and also the ')' using the 'Find and Remove' option in $KWrite$ and also to remove the header. This is necessary because MatLab cannot read this kind of characters.

Then a MatLab script called *vissurge.m* is runned. This script load the files corresponding to the tide gauges in the Table 4 (you should just change the year in the script for the files that are loaded) and load the *xxx_elj_obc.dat* file, then average the surge wave for the four tide gauges in the southern boundary

and the three for the northern boundary and add this estimated surge wave to the corresponding south and north open boundary points and finally generates a file called *xxx_elj_obc3.dat* that is the modified tidal elevation file.

The last step is then to convert the .dat (*xxx_elj_obc3.dat* or *xxx_elj_obc.dat*) file in a NetCDF file and this is made by running the *new.m* file. This requires to have downloaded from the MatWorks website the *mjuliandate.m* and the *juliandate.m* MatLab scripts. The final output of the *new.m* is the *xxx_julian_obc.nc*.

2.3   Interaction with the atmosphere

After the generation of the tidal (or surge) boundary conditions, the successive step is to generate the conditions for the interaction between the atmosphere and the ocean. Those conditions are critical for the simulation of the complete hydrodynamics of the Clyde Sea. The following inputs is obtained from ECMWF:

- wind (at 10 m from the soil)

- mean sea-level pressure

- evaporation

- precipitation

Those input are from the ERA-Interim dataset (Dee et al., 2011), and have a spatial resolution of 0.125°x0.125° and a temporal resolution of 3 hours. The downloaded files are in NetCDF format. In addition to those files some other files for the interaction between atmosphere and ocean are provided by NOAA NCEP/NCAR reanalysis (Kalnay et al., 1996):

- Heat flux radiation (Downward longwave solar, Downward shortwave solar)

- 2 m atmospheric temperature

- relative humidity

Those input have a spatial resolution of 1°x1° and a temporal resolution of 6 hours. The procedure for the processing those files in order to be suitable as input for the Clyde Sea FVCOM model is similar for all the variables. For the wind and pressure a script for each forcing (called *wind2.m* and *pressure2.m*) is run: this script re-grids the data, using the MatLab command *griddata*, from the regular grid into the node locations for the pressure and on the centre of the triangular element for the wind information. FVCOM infact calculate the

9

currents not at each node but at each centre of the element location. So the first part of the *wind2.m* file calculate the position of the centre of each triangle using the two part of the *xxx_grd.dat* file: the node coordinates (that were put manually in a .dat file called *geopos.dat*) and the elements that are specified as connectivity between nodes (that were put manually in a .dat file called *node.dat*). Wind and pressure information then are shifted, since the ECMWF files for wind and pressure have the first half of the file at 6 hour resolution and the data in the middle of these 6 hours resolution is put at the end of the year. This means that the first 9 hours of 1-year reanalysis will be the 1st element (00:00), the 1461st element (03:00), the 2nd element (06:00), and the 1462nd element (09:00). A little for-if MatLab cycle is made at the end of the file in order to correct this problem. For the evaporation/precipitation the procedure is exactly the same as the wind and pressure (the MatLab file is called *evaprec.m*) and is the same for the heat fluxes (Downward longwave solar, Downward shortwave solar) and for the temperature and the humidity (*radiescion2.m*), with the difference that the for-if script is not present, since the NCEP/NCAR does not have the time shift present in the ECMWF files. For the 2m air temperature, the only operation to do is to subtract -273.15, since the NCEP/NCAR provides the temperature in kelvin and not in celsius, while the FVCOM requires the celsius degrees. The wind is the only forcing between the above that have to be gridded in the centre of each element instead of the node.

All of these forcings goes to only one NetCDF file, that is the wind file (it is possible to separate in FVCOM the radiation inputs or the pressure or the evaporation/precipitation in another NetCDF file, but we decided to put all together). The generation of the *xxx_wnd.nc* is made by running the *superforcingfile.m* MatLab script, that uniform the timestep re-timing the data (it is possible both to interpolate the forcing with a time resolution of 6 hours in 3 hours, or vice versa). Also in this case for the correct working of the file are required the *mjuliandate.m* and the *juliandate.m* MatLab scripts as well as the *node.m* and *geopos.m* files and the .dat output files from the previous MatLab script.

2.4 Open boundary temperature and salinity

In order to take account of the advection of the temperature and salinity from the outside. The only data available for the temperature at given location is the sea surface temperature from the ERA-Interim reanalysis. The main assumption that was made for the Clyde Sea model is that the water column is fully mixed in the open boundary locations, so we can use the same temperature for the bottom layer and for the top layer. For the salinity the same assumption is made but in this case with another stronger assumption: the salinity doe not change dramatically in time. This assumption allowed us to use monthly average for each boundary node based on 10-year POLCOM runs for the UK shelf (Edwards et al., 2012). These averages are stored in the folder *Salinity*.

The sea surface temperature and the monthly averaged files are, then, read by the *tempsalobc.m* MatLab script, that interpolate the temperature for each boundary node, generate a salinity output that changes every month and then generate the *xxx_tsobc.nc* file for the FVCOM run. It is important to notice that, since the temperature/salinity input specification for FVCOM is vertical, the number of layers for this file should be the same as the number of layers for the *xxx_sigma.dat*. If not, the FVCOM simulation will stop after the check of the input files.

2.5 Rivers

River input are the only source of freswater for the model with the precipitation flow. Gauged river flow were provided by SEPA after a formal request to the data management office. In the Table 3 the list of the rivers provided by the SEPA is given.

Name	Station	Temporal res.	Latitude (°)	Longitude (°)
Doon	Auchendrane	15 min	55.410	-4.627
Stinchar	Balnowlart	15 min	55.107	-4.968
Clyde	Daldowie	15 min	55.830	-4.122
Little Eichaig	Dalinlongart	15 min	55.995	-4.980
Carradale	Dippen	15 min	55.583	-5.497
Ayr	Mainholm	15 min	55.461	-4.593
Girvan	Robstone	15 min	55.259	-4.808
Irvine	Shewalton	15 min	55.597	-4.628

Table 3: Location of the river data provided by Scottish Environmental Protection Agency (SEPA) for the Clyde model

The files were only including the information about the flow of the rivers and not about the temperature of the discharge. However, many ungauged rivers are present in the Clyde Sea, in particular for the sealochs. The rivers that are reported in Table 3 are the rivers that were used for the results of the simulations in the Thesis and in the surge paper. However, to fill the gaps due to the absence of rivers in fjords, ungauged rivers will be added using models such as *G2G* or other models of this kind. A river file with an update inflow of the river is yet present in the model inputs as well as in the river MatLab scripts. No observations of the temperature and the salinity was available for the rivers, so the maximum air daily temperature was chosen as the temperature of discharge and the salinity was fixed at 15 PSU.

3 FVCOM Installation

FVCOM and all the support libraries must be compiled with MPI (Message Passing Interface) support using the same compiler suite throughout.

3.1 Compiler environment

Add the following to your ~/.bashrc file:

```
# Intel Compilers
if [ -f /opt/intel/ictce/3.2.2.013/ictvars.sh ]
then
        PATH=/opt/intel/Compiler/11.1/059/bin/intel64:$PATH
        export INTEL_LICENSE_FILE=/opt/intel/licenses
        . /opt/intel/ictce/3.2.2.013/ictvars.sh

        # pick up ifort compiled netcdf, etc.
        PATH=$PATH:/home/cais04/local/bin
        LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/cais04/local/lib
fi
```

After a logoff/logon the Intel compilers should be available:

```
  $ type mpicc
  mpicc is /opt/intel/impi/3.2.2.006/bin64/mpicc

  $ type ifort
  ifort is /opt/intel/Compiler/11.1/059/bin/intel64/ifort
```

3.2 Unpacking FVCOM

Create working directory, unpack the source code, then set file and directory permissions:

```
  $ cd ~
  $ mkdir -p fvcom
  $ cd fvcom
  $ tar zxvf /tmp/fvcom-3.1.6-sourcecode.tar.gz
  $ find FVCOM3.1.6 -type d -exec chmod 755 {} \;
  $ find FVCOM3.1.6 -type f -exec chmod 644 {} \;
```

This creates a FVCOM3.1.6 sub-directory.

3.3   Patching FVCOM

Apply the following patches by overwriting the original files:

Source file	Size	Date
mod_nctools.F	355454	12 Dec 2014
mod_nesting.F	132170	12 Dec 2014
swanmain.F	129697	12 Dec 2014

Table 4: Patch files for FVCOM 3.1.6

```
$ cp FVCOM3_patches/mod_nctools.F FVCOM3.1.6/FVCOM_source
$ cp FVCOM3_patches/mod_nesting.F FVCOM3.1.6/FVCOM_source
$ cp FVCOM3_patches/swanmain.F    FVCOM3.1.6/FVCOM_source
```

3.4   Compiling METIS library

For parallel runs we need the METIS graph partitioning library which is part
of the FVCOM source code. Switch to the METIS source directory and apply
a patch to fix the re-definition of log2:

```
$ cd ~/fvcom/FVCOM3.1.6/METIS_source/
$ patch < metis-4.0.patch
```

Edit the makefile and set:

```
CC = mpicc
```

Compile:

```
$ make clean
$ make
```

This produces library file **libmetis.a**

3.5   NETCDF library

NETCDF library version 3.6.3, unpack library, then compile and install:

```
$ mkdir -p ~/fvcom/libs
$ cd ~/fvcom/libs
$ tar zxvf /tmp/netcdf-3.6.3.tar.gz
$ cd netcdf-3.6.3
$ CC=icc ./configure --prefix=/import/scratch3/install
$ make
$ make install
```

the configuration process detects and uses the Intel ifort Fortran compiler as
well.

3.6   FVCOM compilation

Switch to the FVCOM source directory and copy the template **make.inc** file:

```
$ cd ~/fvcom/FVCOM3.1.6/FVCOM_source
$ cp make.inc_example make.inc
```

Edit the **make.inc** file and adjust a few basic settings:

```
# around line 69, remove -L, -I:
dashI= $(empty)
dashL= $(empty)

# around line 72, comment out these 2 lines:
#    LIBDIR       =  -L$(subst $(colon),$(dashL),$(LIBPATH))
#    INCDIR       =  -I$(subst $(colon),$(dashI),$(INCLUDEPATH))

# around line 85, adjust settings for julian library:
DTLIBS       = -ljulian -Llibs/julian
DTINCS       =

# around line 96, set netcdf library options:
IOLIBS = -L/import/scratch3/install/lib -lnetcdf
IOINCS = -I/import/scratch3/install/include

# around line 151, adjust settings for METIS:
PARLIB = -lmetis  -L../METIS_source

# around line 435, comment out:
# FLAG_41 = -DOLD_PETSC

# around line 509, set fortran compiler:
FC       = mpiifort
```

14

Build **julian** date support library:

```
$ cd libs
$ sh ./untar.sh julian
$ cd julian
$ make
$ cd ../..
```

creates static library **libjulian.a**

Now compile FVCOM:

```
$ make
$ cp fvcom /import/scratch3/install/bin/fvcom3
```

This is the base configuration of FVCOM. Keep this configuration:

```
$ cp make.inc make.inc.fvcom3
```

Now copy this configuration file to create a set of other configurations of FV-COM:

```
$ cp make.inc.fvcom3 make.inc.fvcom3h
$ cp make.inc.fvcom3 make.inc.fvcom3rp
$ cp make.inc.fvcom3 make.inc.fvcom3rww
$ cp make.inc.fvcom3 make.inc.fvcom3rw
$ cp make.inc.fvcom3 make.inc.fvcom3r
```

Adjust settings in each file to create each configuration:

Flag	Controls	fvcom3	fvcom3h	fvcom3r	fvcom3rp	fvcom3rw	fvcom3rww
FLAG_14	RIVERS	OFF	ON	ON	ON	ON	ON
FLAG_27	HEATING	OFF	ON	OFF	OFF	OFF	OFF
FLAG_28	AIR PRESSURE	OFF	ON	OFF	ON	OFF	OFF
FLAG_32	WAVES	OFF	OFF	OFF	OFF	ON	ON
FLAG_33	EXPLICIT	OFF	OFF	OFF	OFF	ON	ON
FLAG_33	WAVE ONLY	OFF	OFF	OFF	OFF	OFF	ON

Table 5: FVCOM Configurations

Copy each configuration file in turn on top of the existing **make.inc** and then recompiler FVCOM and install the FVCOM binary each time, e.g.:

```
$ cp make.inc.fvcom3h make.inc
$ make clean
$ make
$ cp fvcom /import/scratch3/install/bin/fvcom3h
```

So this generates the **fvcom3h** variant.

4 FVCOM Runs

4.1 Cluster information

MathStat cluster "nais" is approximately 26 nodes with single processors with between 6 and 8 cores. Parallel processing is provided by the MPI libraries within the Intel compiler suite.

4.2 FVCOM run file

After a procedure of calibration and validation, that is reported in the Ph.D thesis and in the surge paper. The final set-up of the hydrodynamic model is found. In Table 4 we report the final set-up for the model

Variable	Value
External Timestep[1]	1 s
Internal time split[1]	1
External Timestep[2]	0.2 s
Internal time split[2]	10
Smagorinsky constant	0.2
Horizontal Prandtl number	1.0
Vertical Prandtl number	1.0
Horizontal mixing coefficient	0.4
Vertical mixing coefficient	10^{-6}
Minimum Bottom Roughness	0.0024 m
Roughness length	0.001 m
Sponge layer	11 km
Damping coefficient	0.008
Boundary temperature nudging timescale	0.005
Boundary salinity nudging timescale	0.005

Table 6: Parameters of the model after the calibration procedure: for the timestep, two values are reported: [1] is for the barotropic Clyde Sea model, with no temperature and salinity included, while [2] is for the baroclinic model

These settings are included into the *xxx_run.nml* file, that controls the run of the FVCOM simulation. In the file cna be also specified the date in which the model runs (start and finish), type of run ('coldstart' or 'hotstart' if a previous simulation is available for the initial date) and the name of the files used for the simulation as well as the modules included for the simulation. An example of the *xxx_run.nml* is included in the folder.

It is possible to set-up the timestep in which the output are written into the NetCDF file. For the offline lagrangian code, is fundamental that the output should be with a 1 hour timestep due to the requirements of the lagrangian

code. The timestep of the output can be set-up in the runfile section called *NML_NETCDF*, with the option *NC_OUT_INTERVAL*. It is also possible to generate a restart file in order to use the hotstart option for a successive simulation. This can be specified in the *NML_RESTART* section, setting as T the *RST_ON* and setting the *RST_FIRST_OUT* as the timestep in which the restart output is needed. The restart output will be generated in NetCDF file and will contain the value of the variables at the time of the simulation specified in *RST_FIRST_OUT*.

4.3   mpdhosts file

This text file is fundamental for the specification of the node of the cluster in which the FVCOM model should be run. An example of this file is shown in Figure 5.
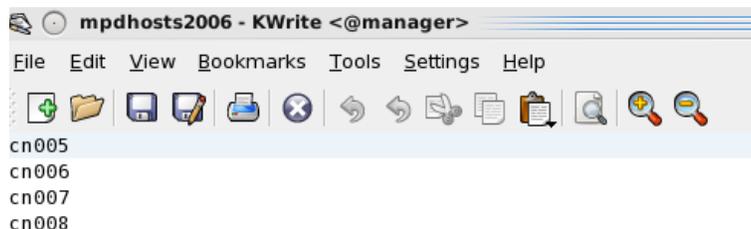

Figure 5: An example of the mpdhosts file

4.4   FVCOM Invocation

FVCOM in the cluster can be invoked from the folder in which the run file is present. In the same folder should be also present the *mpdhosts* file, containing the nodes in which the FVCOM simulation should be run. An example on how the FVCOM is invoked on the cluster is shown here in Figure 6.

Figure 6: The string of characters for running an FVCOM simulation

This string generates also a log file with all the information regarding the simulation and is useful since run in the background and run also if the terminal is closed or the pc is turned off.

References

C. Chen, H. Liu, and R. C. Beardsley. An unstructured grid, finite-volume, three-dimensional, primitive equations ocean model: application to coastal ocean and estuaries. *Journal of atmospheric and oceanic technology*, 20(1): 159–186, 2003.

C. Chen, R. C. Beardsley, and G. Cowles. Finite volume coastal ocean. *Oceanography*, 19(1):78, 2006a.

C. Chen, G. Cowles, and R. Beardsley. An unstructured grid, finite-volume coastal ocean model: FVCOM user manual. *SMAST/UMASSD*, 2006b.

D. L. Codiga. *Unified tidal analysis and prediction using the UTide Matlab functions*. Graduate School of Oceanography, University of Rhode Island Narragansett, RI, 2011.

D. Dee, S. Uppala, A. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. Balmaseda, G. Balsamo, P. Bauer, et al. The ERA-Interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 137(656):553–597, 2011.

K. Edwards, R. Barciela, and M. Butenschon. Validation of the nemo-ersem operational ecosystem model for the north west european continental shelf. *Ocean Science*, 8:983–1000, 2012.

G. D. Egbert and S. Y. Erofeeva. Efficient inverse modeling of barotropic ocean tides. *Journal of Atmospheric and Oceanic Technology*, 19(2):183–204, 2002.

G. D. Egbert, A. F. Bennett, and M. G. Foreman. Topex/poseidon tides estimated using a global inverse model. 1994.

G. D. Egbert, S. Y. Erofeeva, and R. D. Ray. Assimilation of altimetry data for nonlinear shallow-water tides: Quarter-diurnal tides of the northwest european shelf. *Continental Shelf Research*, 30(6):668–679, 2010.

J. H. Ferziger and M. Perić. *Computational methods for fluid dynamics*, volume 3. Springer Berlin, 2002.

E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, et al. The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American meteorological Society*, 77 (3):437–471, 1996.

P. Wessel and W. H. Smith. A global, self-consistent, hierarchical, high-resolution shoreline. *J. geophys. Res*, 101:8741–8743, 1996.