SI3D

USER MANUAL

Department of Physical and Computational Limnology

Universidad de Granada. Instituto del Agua

November 2011

Table of Contents

Chapter 1

Compilation and Cluster connexion

1. SI3D Compilation

To compile si3d or parallel si3d (psi3d) is necessary Intel Fortran Compiler, you can download it with a non-comercial license in:

http://software.intel.com/en-us/articles/non-commercial-software-download/

Psi3d needs OpenMP library too, but it is included with the compiler.

If Archimedes cluster is used, Intel Fortran Compiler (11.1) is already installed and ready for use.

Si3d or psi3d use the next files:

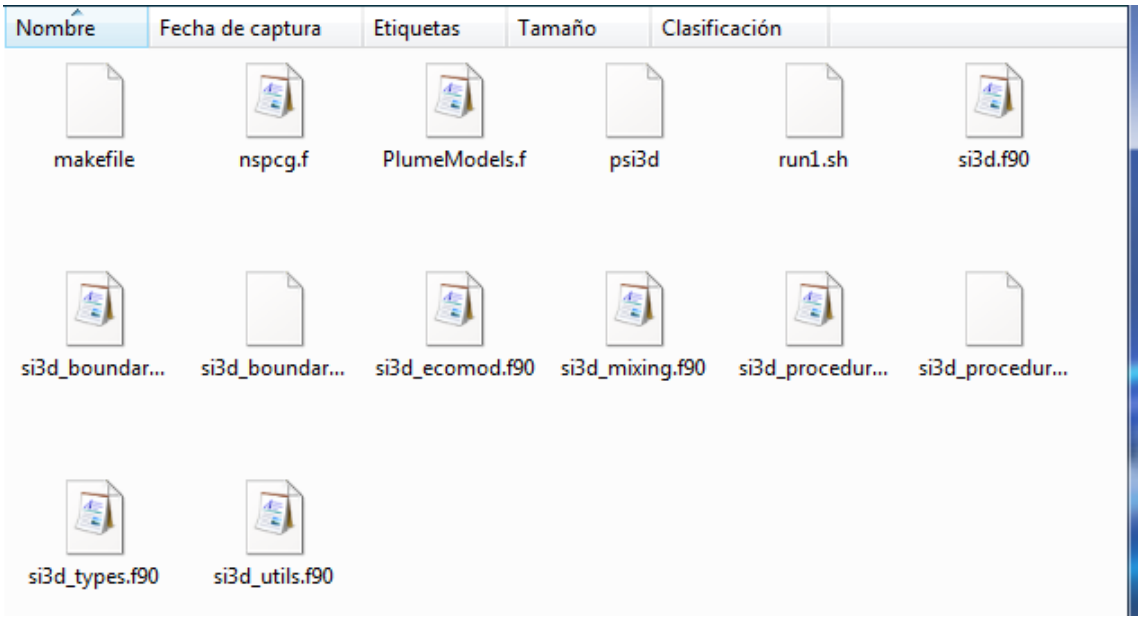


Figure .1.1. si3d files.

Files *.f90 include the si3d code in Fortran 90, makefile includes the number and the name of the files that will be compiled. Makefile can be modified to add or erase some files in the compilation. Default option include the files *.f90 show in the Figure 1.

run1.sh is used to compile si3d or psi3d and GOTM library, GOTM library is used by si3d and must be compile at least one. In run1.sh only must be modified:

- SI3DDIR → Full path to Si3d directory.
- GOTMDIR → Full path to GOTM directory.
- IFGOTM → True/False False if GOTM has been compiled at least one.

run1.sh will need execution permissions, type in the bash in si3d directory $**chmod 755 run1.sh** to change the permissions (only first time, the change will be permanent).

Finally, to compile si3d or psi3d only type in the bash in si3d directory **$run1.sh**

If all is ok a file named si3d or psi3d will be generated, that is the executable, which must be moved to model directory.

2. Archimedes cluster

A computer cluster is a group of linked computers, working together closely thus in many respects forming a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks.

Archimedes has 8 components or nodes, connected through local area Gigabit-Ethernet. Each node can communicate with another node and head node can communicate with other workstation through internet.

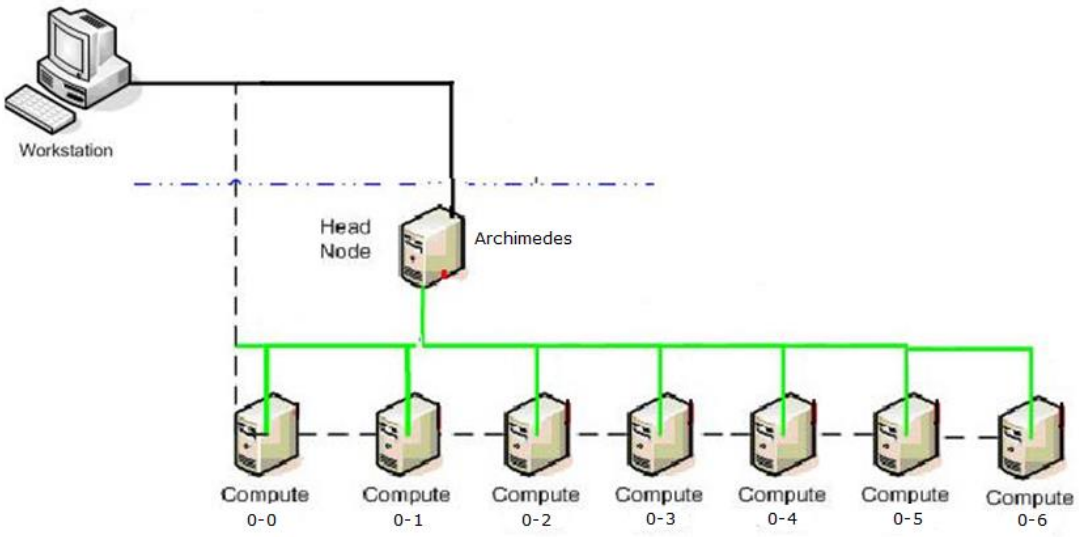


Figure1.2.

To connect to Archimedes, SSH protocol is used; the method used depends on the operating system. Besides, a VNC method can be used too to work in a graphic mode.

Linux, to connect Nautilus:

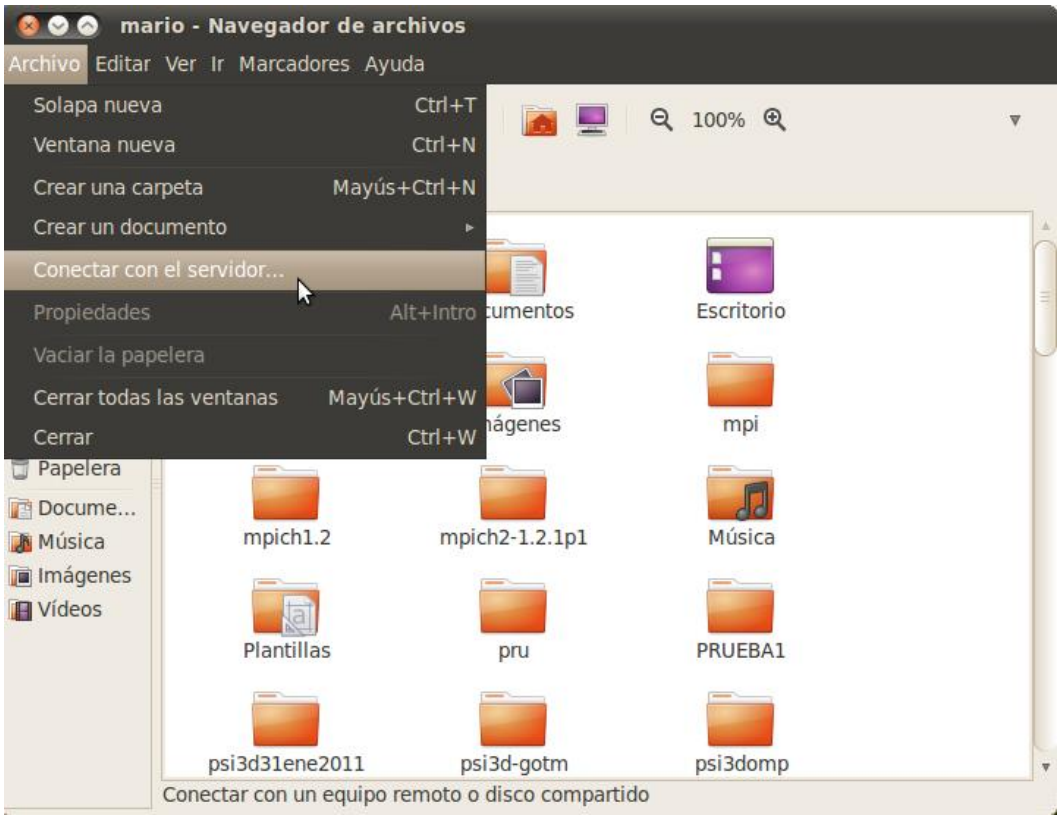


Figure 1.3.

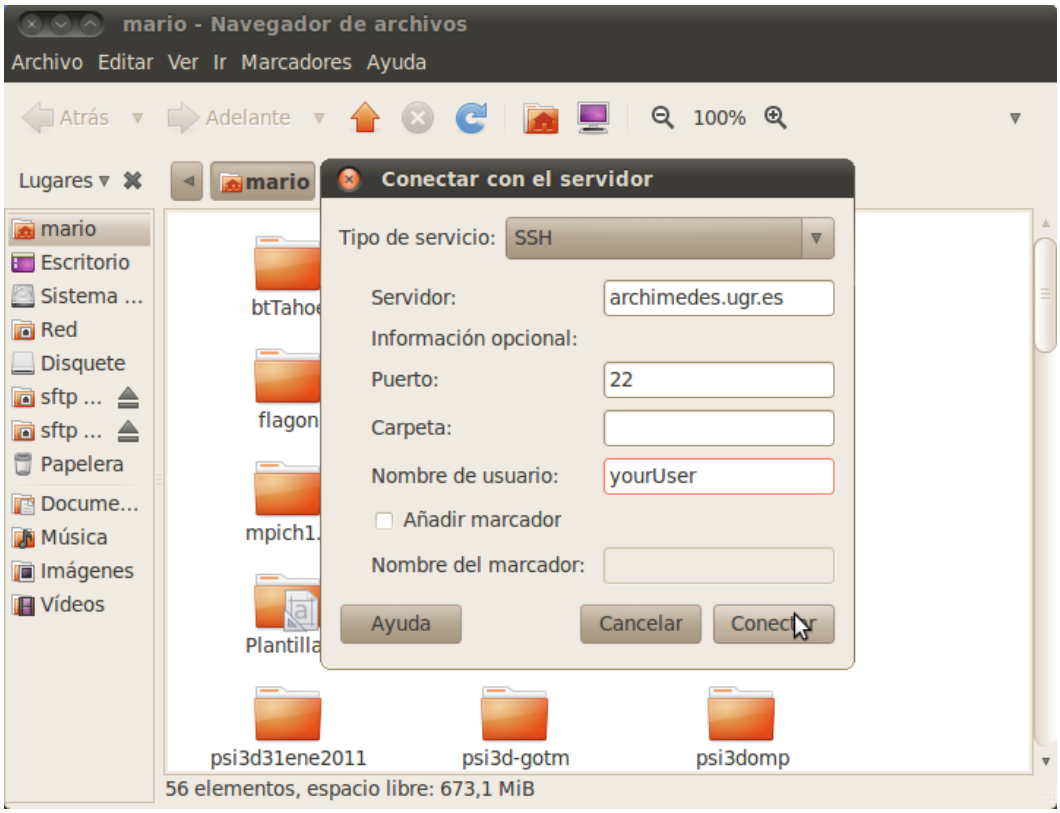


Figure 1.4.

And to connect a terminal:

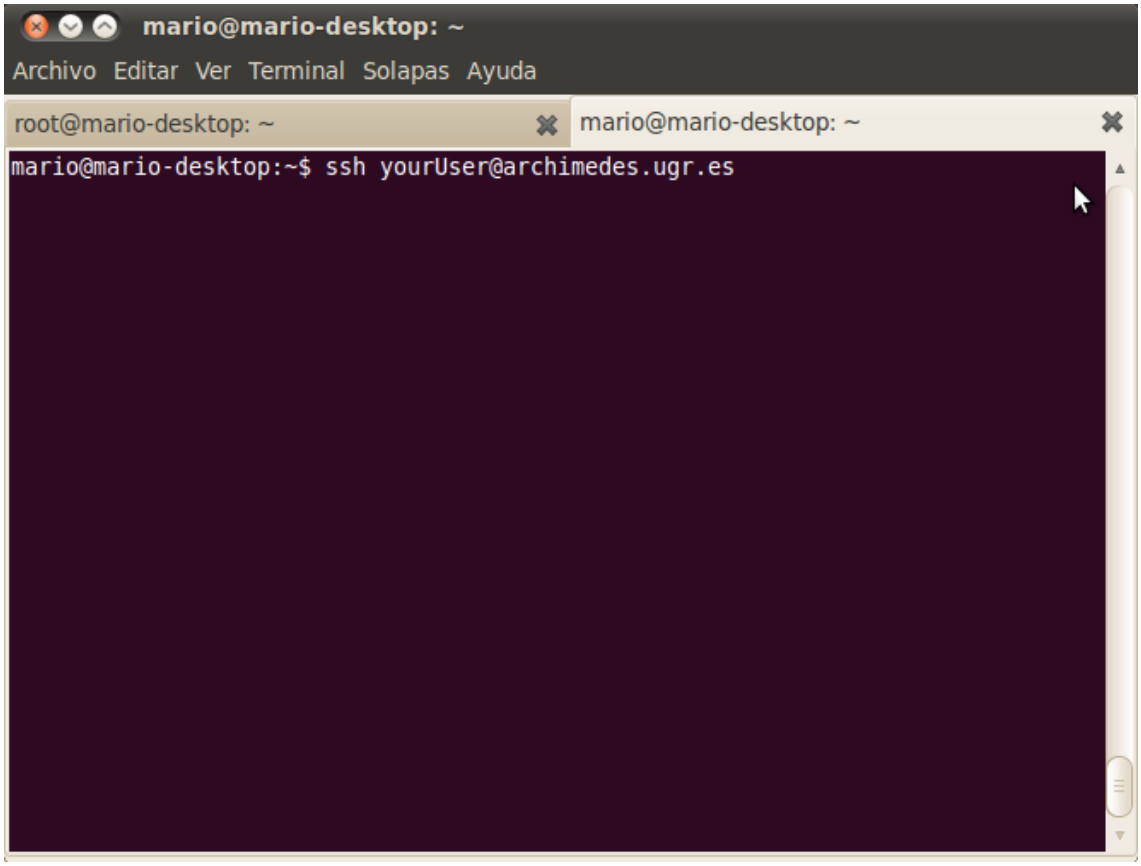


Figure 1.5.

Windows, a direct method or a graphic mode can be used.

1- *Direct method*

Download free software, putty and winscp

http://www.**putty**.org

http://winscp.net/eng/download.php

Putty is used to connect a terminal in Archimedes with SSH protocol, to connect to Archimedes using Putty, then ask you login (your user) and password:

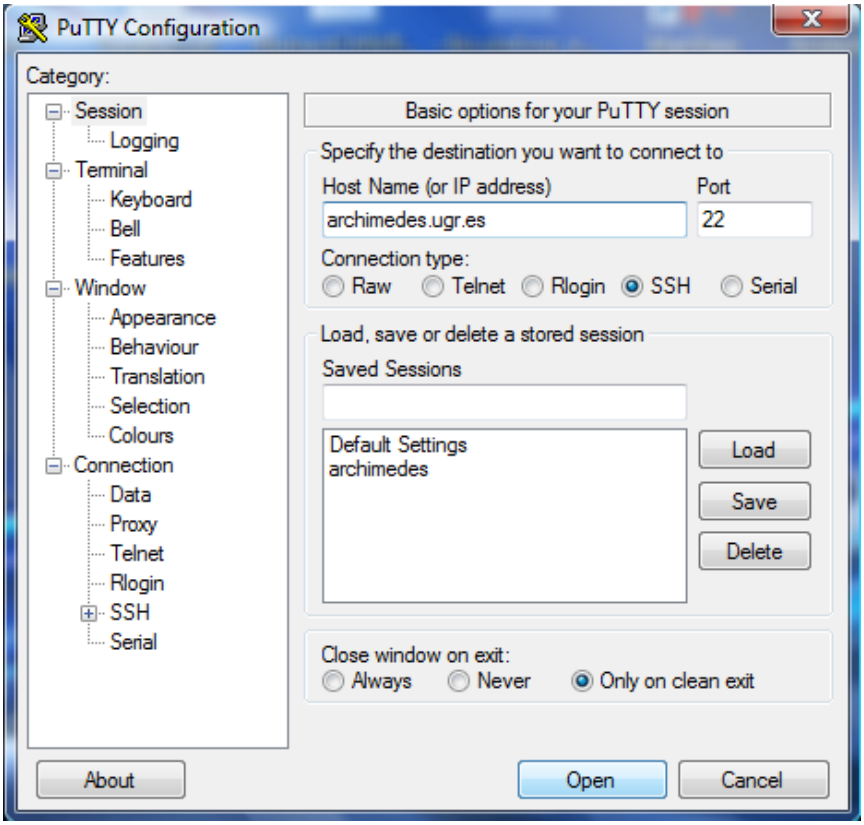


Figure 1.6.

WinSCP is used to move and copy files between a workstation and Archimedes with a SFTP protocol, to connect to Archimedes using WinSCP, fist a new connection is created.

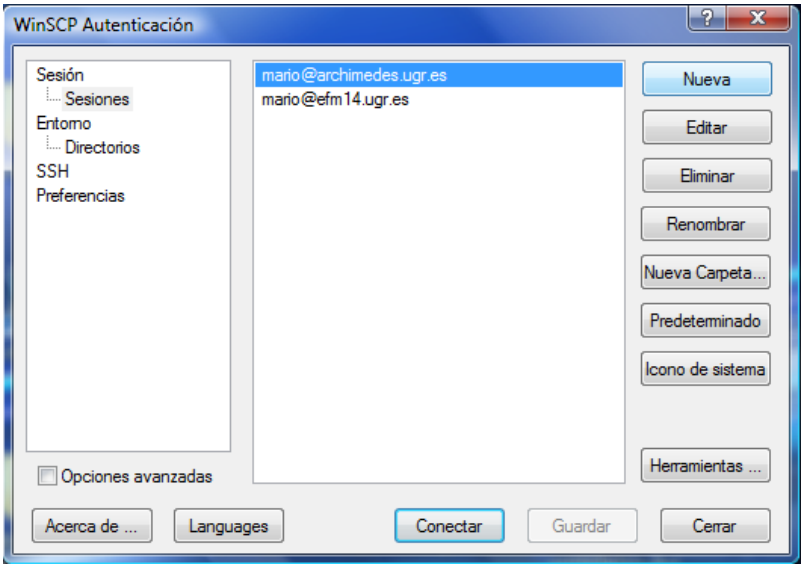


Figure 1.7.

Then the hostname, login and password are typed and click in connect.

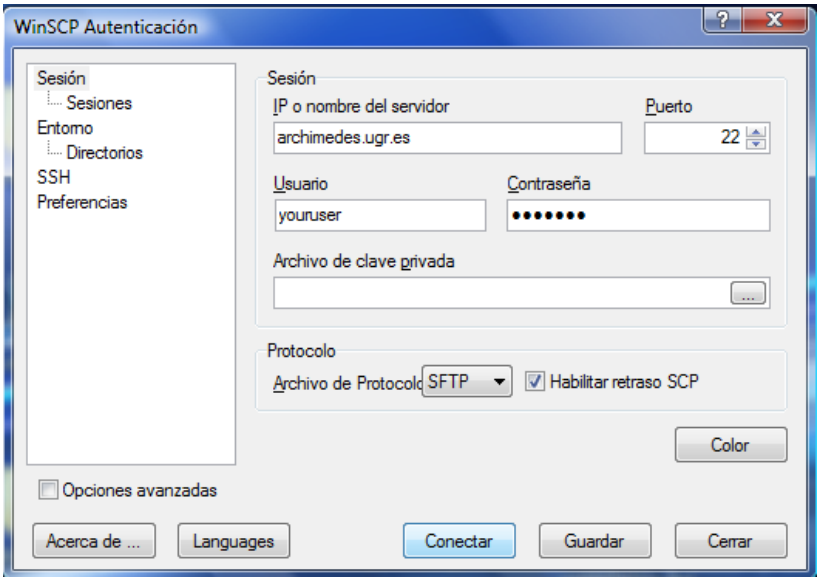


Figure 1.8.

2- *Graphic Mode.*

Download free software, putty and VNCViewer.

Putty is explained in the previous method

http://www.real**vnc**.com/products/**download**.htm

VNCViewer uses a client/server protocol. First, the server must be launched in Archimedes, putty is used for this.

In the terminal opened in Putty, type **$vncserver &** and write a password too.

Then VNCViewer is executed to launch the client, type hostname and display number, terminal with putty must remain open too.

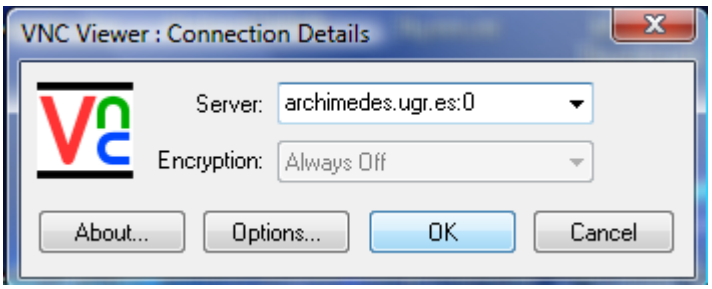


Figure 1.9.

Chapter 2

Input files

1. Introduction

The intent of this chapter is to provide guidance to researchers who wish to use the SI3D model. This chapter describes in detail the information and input files necessary to run the hydrodynamic aspects of the model.

2. Run-time instructions

To run SI3D the si3d.exe (or psi3d.exe) is executed from the directory containing the input files necessary for the run. This is the same directory where the output files will be created. The following is a list of input files required for the model run:

a. Main input file, as described in section 3.

b. Bathymetry file, as described in section 4

c. Initial conditions file, as described in section 5.

d. Open boundary condition file(s), as described in section 6.

e. Surface boundary condition file, as described in section 7.

3. Main input files

The main input file used for the SI3D model, *si3d_inp.txt*, defines important model parameters and determines how the model will be solved. The file also defines how model output is requested. As described in the following tables, the file is divided into twelve separate groups (example in inputs/main folder)

The first group includes a title for the run.

The second group defines the **start date and time** of the model run.

File name	SI3D name	Format	Description
Year	iyr0	G20.2	Year for start date of run
Month	imon0	G20.2	Month for start date of run
Day	iday0	G20.2	Day for start date of run
Hour	ihr0	G20.2	Hour for start of run

The third group defines **space-time domains, cell size and time steps**

File name	SI3D name	Format	Description
xl	xl	G20.2	Length of domain in EW direction (m)
yl	yl	G20.2	Length of domain in NS direction (m)
zl	zl	G20.2	Length of domain in vertical direction(m)
tl	tl	G20.2	Length of time simulated (s)
idx	idx	G20.2	Cell size in EW direction (m)
idy	idy	G20.2	Cell size in NS direction (m)
idz	idz	G20.2	Cell size in vertical direction (m)
dzmin	dzmin	G20.2	Minimum cell size in vertical direction(m)
datadj	datadj	G20.2	Adjustment of datum, by default datum=0
zeta0	zeta0	G20.2	Initial location of water surface (m)
idt	idt	G20.2	Time step (s)
ibathyf	ibathyf	G20.2	Defines how bathy file is read. 0 (general) 1 (DWSC) 2 (SHR). This depends on how bathy file is created

The fourth group contains **parameters that control the solution algorithm**.

File name	SI3D name	Format	Description
itrap	itrap	G20.2	Defines if trapezoidal iteration is used. If itrap = 1, iterations performed, if itrap = 0, single step performed.
niter	niter	G20.2	Number of trapezoidal iterations performed if itrap =1.
smooth	smooth	G20.2	Defines how leapfrog solution is smoothed. If ismooth = 1, smoothing performed, if ismooth = 0, no smoothing.
beta	beta	G20.2	Beta parameters for smoothing filter (0.05 – 2.0 recommended)
itrub	itrub	G20.2	Defines how vertical eddy viscosity is determined. If iturb = 0, constant eddy viscosity is assumed, if turb = 1, 2-eq model is used. (Mellor&Yamada)
az0	az0	G20.2	If iturb = 0, defines constant vertical eddy viscosity.
dz0	dz0	G20.2	If iturb = 0, defines constant vertical eddy

			diffusivity.
iadv	iadv	G20.2	Defines if momentum advection is on or off. If iadv = 0, it is off, if iadv = 1, it is on.
itrmom	itrmom	G20.2	Algorithm for momentum horizontal advection. If itrmom = 1, centered difference used, if itrmom = 2, upwind is used.
ihd	ihd	G20.2	Defines if horizontal diffusion is on or off. If ihd = 0, it is off, if ihd = 1, it is on.
ax0	ax0	G20.2	Horizontal eddy diffusivity/viscosity in EW direction (m^2s^{-1})
ay0	ay0	G20.2	Horizontal eddy diffusivity/viscosity in NS direction (m^2s^{-1})
f	f	G20.2	Coriolis parameter (s^{-1})
theta	theta	G20.2	Weighting parameter for semi-implicit solution (range is 0.5 - 1.0)
ibc	ibc	G20.2	Defines if baroclinic terms included in momentum solution. If ibc = 1, they are included, if ibc = 0, they are NOT included
isal	isal	G20.2	Defines if scalar transport equation is solved. If isal = 1, equation is included, if isal = 0, it is not.
itrsch	itrsch	G20.2	Defines algorithm used to solve scalar advection. itrsch = 1 is centered, 2 is upwind, 3 is centered at k = k1z+1 and 4 is flux-limiter
cd	cd	G20.2	Bottom drag coefficient.
ifsbc	ifsurfbc	G20.2	Defines how model determines surface heat flux. ifsurbc = 0, no heat flux, = 1, heat fluxes read from file (preprocess), = 2, heat fluxes calculated in model (runtime).
dtsbc	dtsurfbc	G20.2	Time in seconds between records in surfbc.txt
cw	cw	G20.2	Wind drag coefficient, used if ifsurfbc = 0.
ws	ws	G20.2	Wind speed (constant in m/s) when ifsbc = 0.
phi	phi	G20.2	Wind direction (constant in degrees) when ifsbc = 0.
idbg	idbg	G20.2	Check messages for debugging. If idbg = 0, no messages displayed during run, idbg = 1 displays messages.
nthread	nthread		Number of threads (parallel version)

The fifth group specifies output information for **time files (tf*_*.txt)**

File name	SI3D name	Format	Description
ipt	ipt	I20	Number of time steps between consecutive output to time file. The remaining lines in this group are only read and should only be included when ipt > 0
nnodes	nnodes	I20	Number of nodes where time file is requested. Maximum number of nodes is 20.
inodes	inodes	I5	Lists the i-location of nodes where time file is requested. The number of entries on this line should match nnodes.
jnodes	jnodes	I5	Lists the j-location of nodes where time file is requested. The number of entries on this line should match nnodes.

The sixth group specifies information for model output to **H-plane** files **(plane_*)**

File name	SI3D name	Format	Description
iht	iop	I20	Number of time steps between consecutive output to plane file. Remaining lines in this group are only read and should only be included if iop>0
nplanes	nplanes	I20	The number of planes where solution is output. The maximum number of planes the model will accept is 10.
plane1	pout(j)	I20	The k-value for the plane at which a plane file is requested. The is lines is repeated for j = 1 to j = n-planes

The seventh group specifies the information necessary for the model to create **cross-section (section_*)**

File name	SI3D name	Format	Description
ixt	iox	I20	Number of times steps between consecutive writes to cross-section files. Remaining lines in this group are only read and should only be included if iox>0
nsections	n_section	I20	The number of sections requested. The maximum number of sections is 10.
SectionID			This line is ignored by the model and is for user reference. This line and the following three sets of lines are repeated for j = 1 to j = n_sections
n_cells	n_sectioncells(j)	I20	The number of cells for each section.
xinodes	xinode(j,nn)	I5	The i location values for the first cross-section. The model reads 10 nodes per line. If

			more than 10 nodes are included in the cross-section, they are placed on the next line. The model will read the node values for nn =1,n_sectioncells.
xjnodes	xjnode(j,nn)	I5	The j location values for the first cross-section. The model reads 10 nodes per line. If more than 10 nodes are included in the cross-section, they are placed on the next line. The model will read the node values for nn = 1,n_sectioncells.

The eighth group defines information regarding to **3D space files**, which record temperature and the three velocity components **(ptrack_hydro.bnr)**

File name	SI3D name	Format	Description
ipxml	ipxml	I20	Number of time step between consecutive output to 3D space file. It must be negative
itspf	itspf	I20	Number of time steps before first output to 3D space file

The nineth group defines information regarding the **open boundaries** for the model **(openbc0*.txt)**

File name	SI3D name	Format	Description
nopen	nopen	I20	The number of open boundaries in the model
dts	dtsecopenbc	G20.2	The number of seconds between records in the openbc files.
OpenBdry1			Comment line; ignored by model. This line and the following six lines are repeated for each of the open boundaries (nn = 1 to nn = nopen).
iside	iside(nn)	I20	Determines the side of the domain the boundary lies on. 1 = West; 2 = North; 3 = East; 4 = South boundary.
itype	itype(nn)	I20	Determines the type of boundary. 1 = water surface elevation; 2 = surface flow; 3 = sub-surface flow.
isbc	isbc(nn)	I20	i-location for first cell along open boundary.
jsbc	jsbc(nn)	I20	j-location for first cell along open boundary.
iebc	iebc(nn)	I20	i-location for last cell along open boundary.
jebc	jebc(nn)	I20	j-location for last cell along open boundary.

The tenth group defines information for **open boundary conditions for nested**

File name	SI3D name	Format	Description
nxBDO	NxBDO	I20	Number of boundaries that will be saved for nested
iob	iob	I20	Time steps between consecutive records to output files
xxb	xxb	I20	Scaling in the x-direction
OpenBdry 1			Comment line; ignored by model. This line and the following six lines are repeated for each of the open boundaries (nn = 1 to nn = nopen).
iside	iside(nn)	I20	Determines the side of the domain the boundary lies on. 1 = West; 2 = North; 3 = East; 4 = South boundary.
isbc	isbc(nn)	I20	i-location for first cell along open boundary.
jsbc	jsbc(nn)	I20	j-location for first cell along open boundary.
iebc	iebc(nn)	I20	i-location for last cell along open boundary.
jebc	jebc(nn)	I20	j-location for last cell along open boundary.

The eleventh group defines the number of **tracers** modelled **(tracer_*)**

File name	SI3D name	Format	Description
ntr	ntr	I20	The number of tracers to simulate If ntr>0, the following lines are also read.
ecomod		0	
iotr	iotr	I20	The number of time steps between consecutive output to tracer file.

The twelfth group defines the specifications for **pss boundary conditions (pps0*.txt)**

File name	SI3D name	Format	Description
iopss	iopss	I20	The number of columns with point source-sinks
nodev	nodev	I20	Number of devices acting as source-sinks
iodt	iodt	I20	Time in seconds between consecutive records in the pss file
#			Comment line
iopssLoc		I20	This line is repeated as many columns have been defined at iopss. It contains the i-location, j-location and number of device

4. Bathymetry file

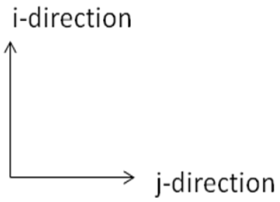
Folder: inputs/bathy

Script: bathy4si3d.m

Example: h

The bathymetry file, called *h* by the model, contains the information regarding the geometry of the domain. The bathymetry file is reading in the subroutine *bathy*.

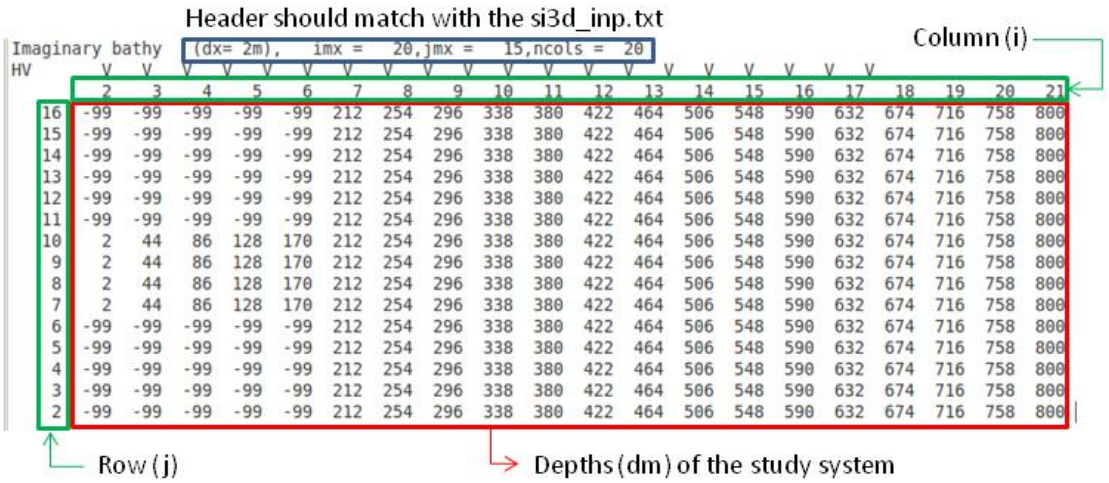
It is important to notice that the model Cartesian reference system calls “i” to the vertical direction and “j” to the horizontal direction, so each water cell location can be determined be a pair of values (i,j). You must be aware that it has been added an extra row at the north and south boundaries of the domain, and an extra column at the east and west boundaries of the domain. The origin of the system is the bottom left corner.



The first line of the *h* file header contains *dx*, the grid dimensions, *imx*, the number of cells in the i-direction, *jmx*, the number of cells in the j-direction, and ncolums, the number of columns at each row (=*imx*). Please, notice the limited space for each number at this line of the header (see bathy subroutine)

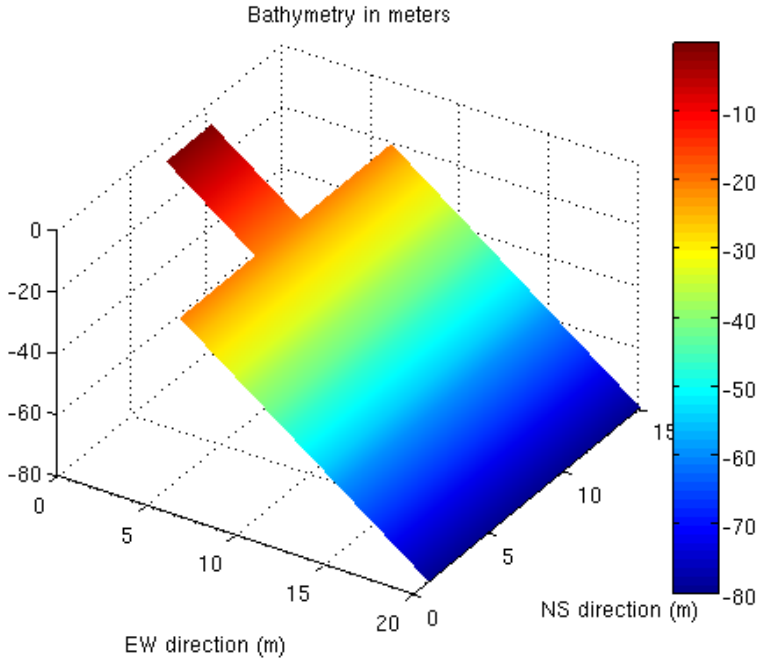
The second and third line of the header and first column are not read by the model. The third header line designates the number of the cell in the i-direction (i) and the first column represents the number of the cell in the j-direction (j). The matrix delimited by these two vectors contains the depths of our study system, which must be written in decimeters. Dry cells are represented by the value -99.

Example file: h file (with the 3D plot of the imaginary bathymetry)



Imaginary bathy (dx= 2m), imx = 20,jmx = 15,ncols = 20

HV	V 2	V 3	V 4	V 5	V 6	V 7	V 8	V 9	V 10	V 11	V 12	V 13	V 14	V 15	V 16	V 17	V 18	V 19	V 20	V 21
16	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
15	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
14	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
13	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
12	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
11	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
10	2	44	86	128	170	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
9	2	44	86	128	170	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
8	2	44	86	128	170	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
7	2	44	86	128	170	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
6	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
5	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
4	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
3	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800
2	-99	-99	-99	-99	-99	212	254	296	338	380	422	464	506	548	590	632	674	716	758	800



You should be aware that the bathymetry data at the *h* file is related with the dimensions that you select at the *si3d_inp.txt* file. As shown the Fig.2.1, *xl* must be equal to *imx*dx* and *yl* must be equal to *jmx*dy*. You should notice that the time step (*dt*) should be chosen according to the Couran condition,

$$dt < \frac{dx}{u_{max}}$$

```
-----------------------------------------------------
Space-time domains, cell size & time steps
-----------------------------------------------------
xl          !      40.        ! Length of domain (m) in EW direction
yl          !      30.        !   "    "    "    (m) in NS       "
zl          !      10         !   "    "    "    (m) in vertical
tl          !    3600.        ! Length of time (s) simulated (6dias + 22 horas,)
idx         !      2.0        ! Cell size (m) in EW direction
idy         !      2.0        ! Cell size (m) in NS direction
idz         !      0.1        ! Cell size (m) in vertical
dzmin       !      0.05       ! Min. cell size (m) in vertical
datadj      !      0.0        ! Adjustment of datum by default datum = 0.0
zeta0       !      0.0        ! Initial location of water surface
idt         !      1.0        ! Time step (s)
ibathf      !      -1         ! How bathy file is read 0(General) 1(SDWSC) 2(SHR)
-----------------------------------------------------
```

Fig. 2.1. Space domain at si3d_inp.txt file

5. Initial condition file

To start the simulations, and, therefore, to satisfy the two-level scheme of the semi-implicit, leapfrog trapezoidal scheme we must initialize the scalars and the velocity fields in our computational domain (initial conditions). At the beginning, we assume we have a quiescent domain (u = v = w =0 m/s), then, we only have to define the initial value of the scalars (temperature). We introduce a unique temperature profile which is characteristic of the whole domain, and therefore we assume, initially, that there are no horizontal gradients (horizontally homogeneous domain). Besides, in the case we want to take tracer (conservative substances) experiments during our simulations, we must initialize the values (background concentrations) of these tracers: one vertical profile per each of the tracers.

The input file used to initialize temperature is the *si3d_init.txt*. There are two ways to initialize temperature: It can be constant in the whole water column (homogeneous system) or it can be variable (stratified system)

5.1. Temperature profile with constant Δz.

With this file we introduce a value of temperature (and tracers) per each Δz we define in the input file. Therefore Δz defines the maximum resolution of our initial profiles. Fig. 2.1 shows and example of a *si3d_init.txt* file, where the first 5 lines are leaved for string characters (headlines) and from that line on, the vertical profiles are set. Note how at the depth profile we account for two extra cells (one on the surface and the other one at the bottom). These are the top and bottom cells of the water column which have a thickness Δz which will vary in time and space due to changes in water surface elevation and the irregular bathymetry. In fact, the code does not read the values of depth you introduce at the *si3d_init.txt* file, but counts the number of lines, and so Δz (from surface to bottom) and assigns to all the corresponding cells at this depth (line in the init file) the value of temperature (and tracers) at the given line.

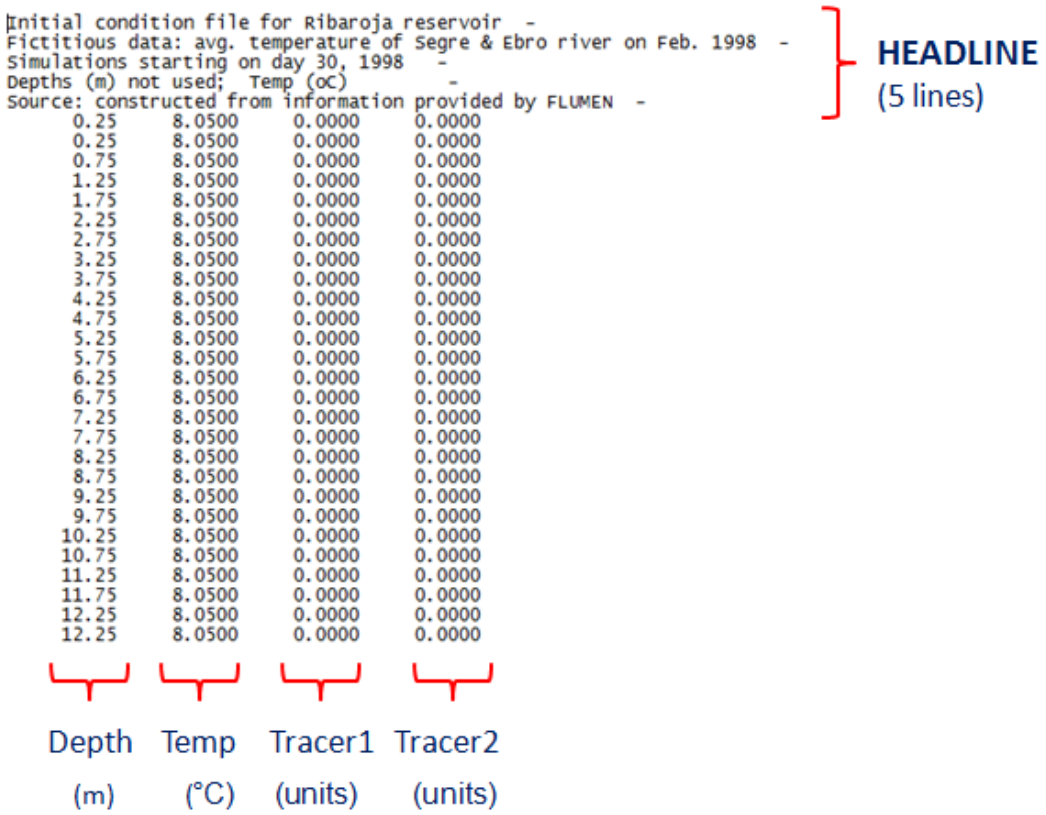


Fig.2.1. Example of a si3d_init.txt file for a domain with maximum depth ~12 m and a two tracer releases experiment with background concentrations of 0 units and a constant temperature vertical profile. (If no tracers are simulated, their corresponding columns are not needed)

5.2. Variable depth and temperature profile

Folder: init_cond/variableT&dz

Scripts: GenerateLAYERS.m and initcond4si3d.m

Example: si3d_layer.txt and si3d_init.txt

You may want to obtain a vertical resolution from the model with an irregular distance between layers (Δz is not constant). In this case you have to set the variable *ibathyf* = -1 at the *si3_inp.txt* file and create a *si3d_layer.txt* file which should be included as input file for running the model. If you choose this option, the script that you use to create the *si3d_init.txt* must start with the rule that you have used to create the layer. Besides, the maximum depth that you set at those scripts must be the same that you write at *zl* in the "Space domain" section at the *si3d_inp.txt* file. In addition, you may require to define the initial temperature profile according to the data recorded by any probe (e.g. by a CTD). Examples of the *si3d_layer.txt* and *si3d_init.txt* files are

shown below. The folder init_cond/variableT&dz contains the scripts and data required to generate the files

Example file: si3d_layer.txt

```
Depths to top of layers in Si3D Grid for Lake Beznar
** used if ibathyf in si3d_inp.txt is set to < 0
-----------------------------------------------------
   km1   =        36
      1.00  -100.0000
      2.00  -100.0000
      3.00     0.1000
      4.00     0.2060
      5.00     0.3184
      6.00     0.4375
      7.00     0.5637
      8.00     0.6975
      9.00     0.8394
     10.00     0.9897
     11.00     1.1491
     12.00     1.3181
     13.00     1.4972
     14.00     1.6870
     15.00     1.8882
     16.00     2.1015
     17.00     2.3276
     18.00     2.5673
     19.00     2.8213
     20.00     3.0906
     21.00     3.3760
     22.00     3.6786
     23.00     3.9993
     24.00     4.3392
     25.00     4.6996
     26.00     5.0816
     27.00     5.4865
     28.00     5.9156
     29.00     6.3706
     30.00     6.8528
     31.00     7.3640
     32.00     7.9058
     33.00     8.4802
     34.00     9.0890
     35.00     9.7343
     36.00    10.4184
```

Layer Depth

Example file: si3d_init.txt (not full data)

```
Initial condition file for si3d model  -
Data for Beznar Reservoir 2009         -
Simulations starting on day 248        -
Depths (m)   Temp (oC)                 -
Source: NEMO data                      -
----------------------------------------
     -0.05    25.5956     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -0.05    25.5956     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -0.15    25.5907     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -0.26    25.5854     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -0.38    25.5799     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -0.50    25.5740     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -0.63    25.5677     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -0.77    25.5611     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -0.91    25.5541     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -1.07    25.5096     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -1.23    25.4140     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -1.41    25.3128     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -1.59    25.2054     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -1.79    25.0916     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -1.99    24.9710     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -2.21    24.9783     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -2.45    24.9895     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -2.69    25.0013     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -2.96    25.0139     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -3.23    25.0104     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -3.53    25.0033     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -3.84    24.9959     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -4.17    24.9879     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -4.52    24.9795     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -4.89    24.9706     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -5.28    24.9541     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -5.70    24.9336     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -6.14    24.9190     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -6.61    24.9190     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -7.11    24.9086     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -7.63    24.8580     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -8.19    24.7668     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -8.78    24.5947     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     -9.41    24.4826     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    -10.08    24.3844     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    -10.08    24.3844     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
```

Depth Temp

6. Boundary condition files

Boundary conditions are the set of conditions specified for the behavior of the solution to a set of differential equations at the boundary of its domain. Therefore, here we introduce the velocity fields into and out of the domain or we force the model with a specific water surface elevation (wse). Besides, as in *the si3d_init.txt* file, we must introduce the value of the scalars (and tracers) at the boundaries. The most common way to set the velocity fields in or out of the domain is by setting flow discharges (and so considering a homogeneous velocity field at the corresponding inflow/outflow section). The code then gives the corresponding velocity to each boundary cell according to its area $\Delta x \Delta z$ (or $\Delta y \Delta z$). We have two methods to introduce a homogenous flow into or out of the domain and its choice depends on the inclination of the boundary

with respect to the x and y axes of the Cartesian grid in the bathy file. When the boundary is perpendicular (or parallel) to the x axis we introduce the boundary as an *openbc_.txt* file in the rest of cases we introduce the boundary as a *pss_.txt file*.

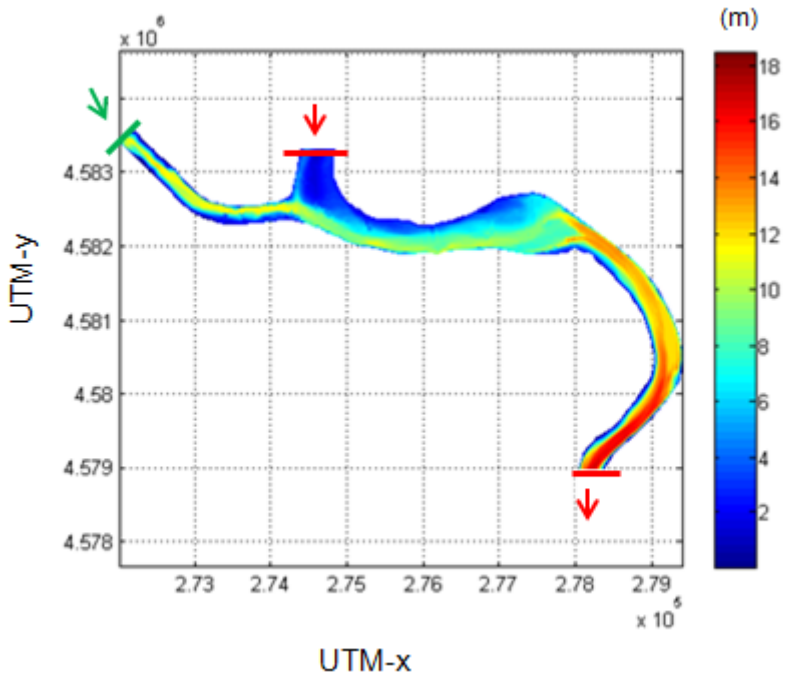


Fig.2.2. Example of a computational domain with two openbc_.txt type boundaries (North and South boundaries) and one pss_.txt type boundary (West boundary).

6.1. Open Boundary conditions: openbc(number).txt

With this configuration we can force the model with inflows or outflows (discharges) or restrict the oscillation of the wse at each boundary. Boundaries in this case, must be perpendicular (East and West boundaries) or parallel (North and South boundaries) to the x axis of the Cartesian grid in the bathy file.

a. Flow boundaries:

Folder: boundary_cond

Scripts: Openbc_Flowboundary.m

We assume here that the lateral and vertical components of the velocity fields are negligible and, therefore, the flows into and out of the domain are always perpendicular to the boundary. In order to correctly introduce discharges we have to write them with the appropriate sign: since E-W velocities are positive towards the east and N-S velocities towards the north, inflows must be positive at West and South boundaries and negatives at the North and East boundaries. The opposite is valid for outflows. At each of the *openbc_txt* file we include a value of discharge and temperature (and tracers if needed) for each of the Δt in the boundaries. This $\Delta t_{boundary}$ is given in the si3d_inp.txt file (see below) and accounts for the time during which discharges and scalars at a given boundary have constant values, which are equal to the corresponding values in each line. This $\Delta t_{boundary}$ is independent of the computational time step (although it cannot be lower than the computational time step) and therefore, during several computational time steps discharges and scalars can remain the same at a given boundary. As it was the case of the depth in the si3d_init.txt file, the code does neither read the time file in the openbc_.txt file (first column), but it counts lines, and assign the corresponding value of discharge and temperature to the corresponding computational time. For example if $\Delta t_{boundary}$ = 1 hour, the code will read a new line each hour and within this hour the values are homogeneous and equal to the values at this line. Fig. 2.3 shows and example of a flow *openbc_.txt* file.

In the case of outflow boundary conditions we do probably not want to introduce a known value for the scalars or the tracer fields, but that the boundary adopts the value of the immediately adjacent cell in the domain. In these cases we fill the columns of temperature and tracers with a constant (through time) value of -20.

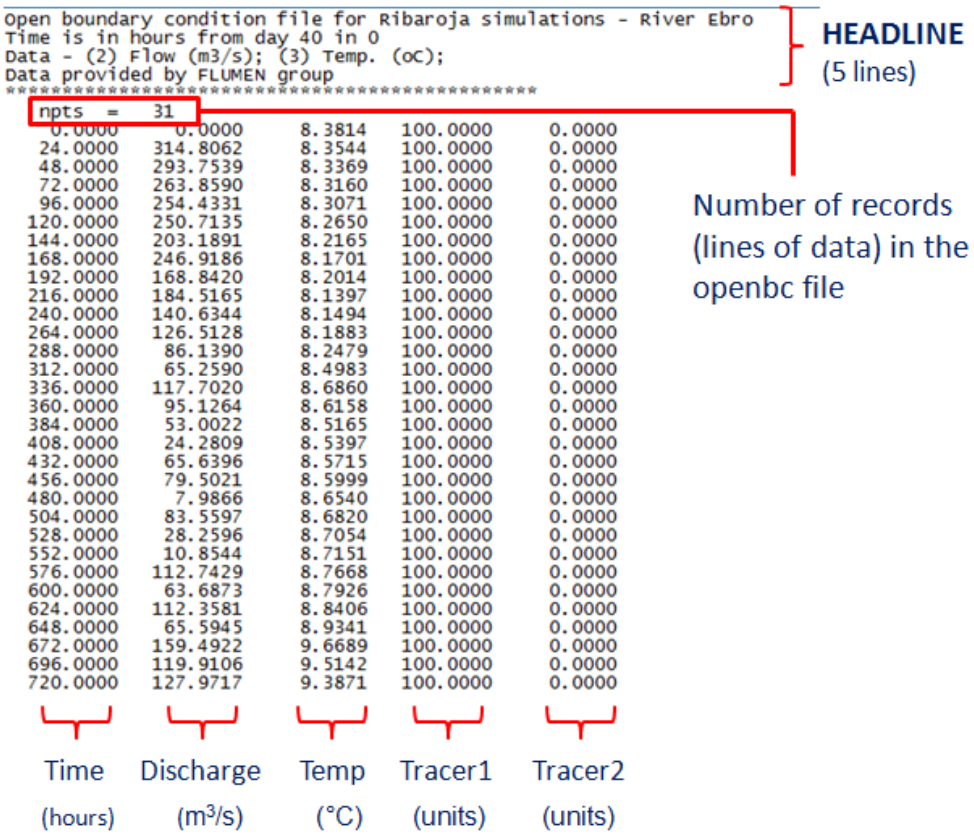


Fig. 2.3 openbc_.txt file for the case of a flow boundary condition with two tracer releases experiments (If no tracers are simulated, their corresponding columns are not needed). This is an example of a West boundary condition with flow into the domain (positive values)

b. Wse boundaries:

Folder: boundary_cond

Scripts: Openbc_WSEboundary.m

We can also restrict the oscillation of the wse or completely neglect them (constant wse) at a given boundary by using the openbc_.txt file. This file looks exactly the same as the file shows in fig. 2.3 except for the column of discharges which is replaced for the column of oscillations of the wse (m above, >+, or below ,<0, the mean water surface elevation).

c. Include *openbc_*txt in *si3d_inp.txt* file:

At the si3d_inp.txt file we need to include the number of openbc-type boundaries we have in our domain, $\Delta t_{boundary}$ and their locations (i & j cells). Fig. 2.4 shows and

example of a portion of the si3d_inp.txt file where these parameters are specified. In this example we have three openbc-type boundaries with $\Delta t_{boundary}$ = 600s. At each of the boundaries we specify if it is at the North, South, East or West; if it is a flow or a wse type boundary and the horizontal location (i,j) of the initial and final wet columns of the boundaries.

```
-------------------------------------------------------
Open boundary conditions specifications
-------------------------------------------------------
 nopen       !        3           ! No. of open boundaries
 dtsecopenbc !      600.          ! Time (s) betwen consecutive records in openb boundary conditions files
 OpenBdry 1  !   ##########       ! Comment line - Three lines for each open boundary
 iside       !        2           ! 1 = West; 2 = North; 3 = East; 4 = South boundary
 itype       !        2           ! 1 = wse ; 2 = surface flow; 3 = subsurface flow
 isbc        !      240           ! i- location for first cell along open boundary
 jsbc        !      458           ! j- location for first cell along open boundary
 iebc        !      265           ! i- location for last cell along open boundary or k- top cell (if itype = 3)
 jebc        !      458           ! j- location ofr last cell alont open boundary or k- bot cell (if itype = 3)
 OpenBdry 1  !   ##########       ! Comment line - Three lines for each open boundary
 iside       !        2           ! 1 = West; 2 = North; 3 = East; 4 = South boundary
 itype       !        2           ! 1 = wse ; 2 = surface flow; 3 = subsurface flow
 isbc        !      268           ! i- location for first cell along open boundary
 jsbc        !      458           ! j- location for first cell along open boundary
 iebc        !      284           ! i- location for last cell along open boundary or k- top cell (if itype = 3)
 jebc        !      458           ! j- location ofr last cell alont open boundary or k- bot cell (if itype = 3)
 OpenBdry 3  !   ##########       ! Comment line - Three lines for each open boundary
 iside       !        4           ! 1 = West; 2 = North; 3 = East; 4 = South boundary
 itype       !        1           ! 1 = wse ; 2 = surface flow; 3 = subsurface flow
 isbc        !      604           ! i- location for first cell along open boundary
 jsbc        !        5           ! j- location for first cell along open boundary
 iebc        !      636           ! i- location for last cell along open boundary or k- top cell (if itype = 3)
 jebc        !        5           ! j- location ofr last cell alont open boundary or k- bot cell (if itype = 3)
-------------------------------------------------------
```

Fig 2.4. Example of a domain with three openbc-type boundaries (two at the North and one at the south)

6.2. Point source-sink boundary conditions: pss(number).txt

Folder: boundary_cond

Scripts: pssgenerator.m

In this case, we can only simulate flow type boundaries (restriction of the oscillation of the wse is not possible in this case). Fig. 2.5 shows an example of a *pss_.txt file*. The pss_.txt files have several utilities besides the open boundary condition one. In order to the code treat it as a boundary condition we must type -2 in the “Type” line. We introduce the number of records (lines of data) in the pss file, the face of the cells through which the flow enters or leave the domain (see below), the discharge threshold above which this boundary is activated and finally the columns of time, discharges (positive or negative according to the sign of the velocity fields), temperature and tracers as in the openbc case.

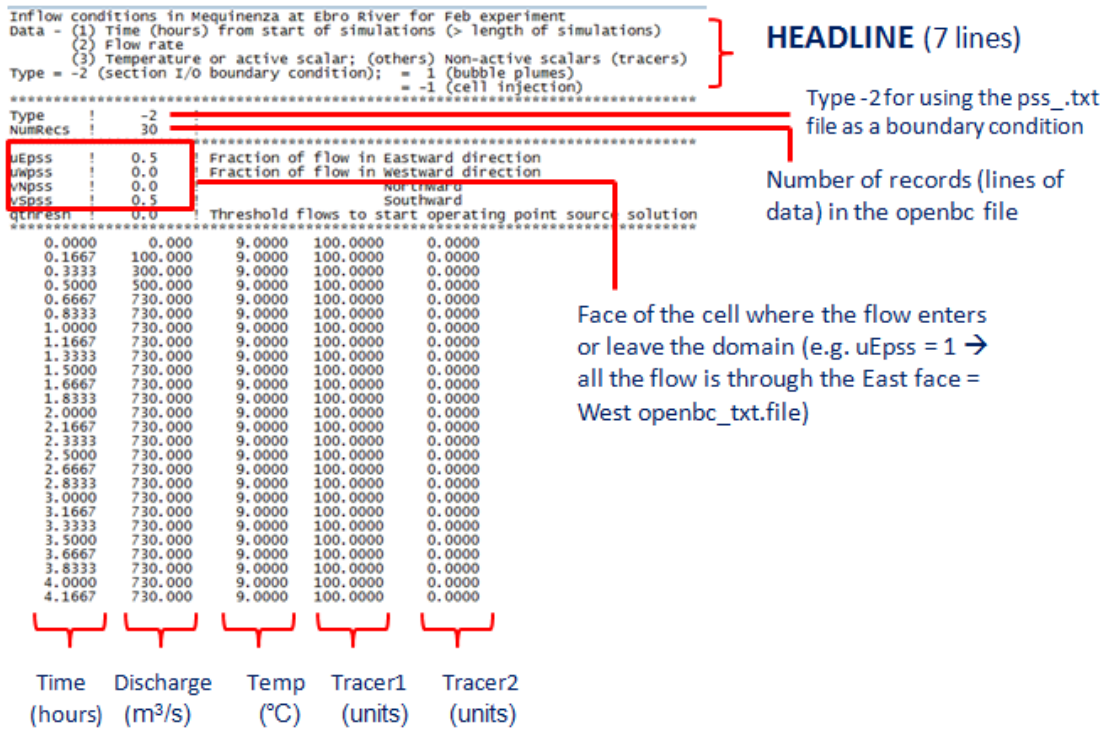


Fig. 2.5 Example of a pss_.txt file where the flow enters the domain with an angle of 45° (0.5 East face + 0.5 South face) with respect to the x axis of the Cartesian grid in the bathy file

In order to include the face of the cell through which the flow enters or leave the domain we have to take into account the following criteria: uEpss, uWpss, vNpss and vSpss represent the E, W, N and S faces of the water cells (columns) at a given boundary. Therefore we introduce horizontal flows (the vertical component of the velocity is always 0 at the boundary). A value of 1 for this parameter implies the flow passes only through this face, and so the rest of parameters must have a value of 0 (no flux through these faces). Values in between 0 and 1 represent the fraction of each face through which the flow passes. We can combine these fractions to adjust the boundary to our objectives but taking into account that the overall value of the 4 faces must always sum to 1. Fig.2.6 shows some examples.

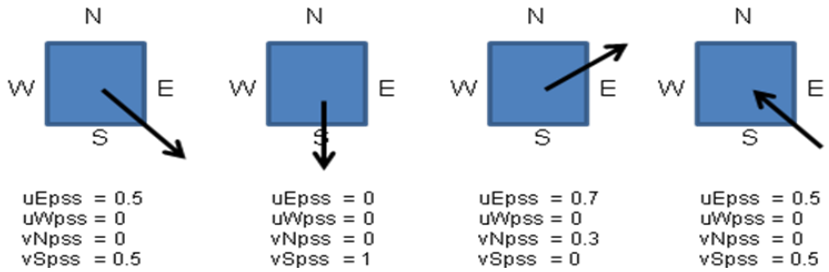


Fig. 2.6. Examples of values of uEpss, uWpss, vNpss and vSpss and the resultant direction of the flow (Note that they are the same if directed in or out the domain, we must specify this with the sign of the discharges)

Include *pss_txt* in *si3d_inp.txt* file:

Fig. 2.7 shows an example of the portion of the si3d_inp.txt file where we define the location of the boundary file and $\Delta t_{boundary}$. In this case the configuration differs slightly to those for the openbc.txt files. We have to introduce the total number of water columns that acts as sources/sinks. For example, if we have two pss-type boundaries with 10 columns each, we introduce at the "iopss" line 20. *Nodev* refers to the number of boundaries (in the previous example *Nodev* = 2), and iodt = $\Delta t_{boundary}$. After this, we introduce the location (i,j) of each of the water columns in the boundaries (*iopssLoc*) followed by a number which identify the number of the boundary which they form part of. In fig. 2.7 all the columns are located at the same boundary so the third column at iopssLoc is always equal to 1.

```
------------------------------------------------------
Specification for Point Sources and Sinks
------------------------------------------------------
 iopss       !       10                          ! No. of columns with point sources-sinks
 Nodev       !        1                          ! No. of devices acting as sources-sinks
 iodt        !      600.                         ! Time (secs) between consecutive records in io files
 #########   ! i- j- device No. for each column
 iopssLoc   4 465   1
 iopssLoc   5 466   1
 iopssLoc   6 467   1
 iopssLoc   7 468   1
 iopssLoc   8 470   1 ! Type = 1 (linear plume in NS direction)
 iopssLoc   9 471   1 ! Type = 2 (linear plume in EW direction)
 iopssLoc  10 472   1
 iopssLoc  11 473   1
 iopssLoc  12 474   1
 iopssLoc  13 475   1
```

Fig 2.7. Example of a domain with one pss-type boundary condition.

7. Surface boundary condition files

There are four options available for the surface boundary conditions, and the form is specified by the variable *ifsbc* at the "Parameters controlling solution algorithm" in the *si3d_inp.txt*. The options for *ifsbc* are 0, 1, 2 and 11. These four options can be classified depending on whether the values of the surface boundary conditions provided to the model are constant or variable. Constant surface boundary conditions dot not require a *surfbc.txt* input file (*ifsbc* = 0), while variable surface boundary conditions (*ifsbc* = 1, 2, or 11) need that file.

7.1. Constant surface boundary conditions - No surfbc.txt file required

When *ifsbc* = 0, no heat flux is modeled and the surface boundary conditions are set to a constant value. In this case, it is not necessary a surface boundary condition file. You should define a fixed value of the wind drag coefficient (*cw*), wind speed (*ws*, in ms^{-1}), and wind direction (phi, in degress) at the "Parameters controlling solution algoritm" in the *si3d_inp.txt*.

```
-----------------------------------------------------
Parameters controlling solution algorithm
-----------------------------------------------------
itrap       !       1          ! 1 = trapezoidal iteration is used; single step LF if 0
niter       !       2          ! No. of trapezoidal iterations
smooth      !       0          ! Smoothing of LF solution 1 = Smooth; 0 = DO NOT smooth
beta        !    0.05          ! Beta parameter for smoothing filter (0.05-0.2 recommended)
iturb       !       1          ! 0 = constant vert. eddy visc. 1 = 2-eq. model (Mellor&Yamada)
az0         !   1.E-3          ! Constant vert. eddy viscosity (if iturb = 0)
dz0         !   1.E-3          ! Constant vert. eddy diffusivity (if iturb = 0)
iadv        !       1          ! 0 = mom. advection OFF; 1 = mom. advection ON
itrmom      !       2          ! Algorithm for mom. horizontal advection 1=centered; 2=upwind
ihd         !       1          ! 0 = horizontal diffusion (of mom. or scalar) OFF 1 = horiz. diffusion ON  2 = Smagorisnki
ax0         !   1.E-1          ! Horizontal eddy diffusivity/viscosity in EW-direction (ms-2)
ay0         !   1.E-1          ! Horizontal eddy diffusivity/viscosity in NS-direction(ms-2)
f           !   9.1e-5         ! f Coriolis parameter (s-1)
theta       !       1.         ! Weighting parameter for semi-implicit solution (0.5-1) - Not used
ibc         !       1          ! 1 = Baroclinic terms in included in mom. solution (0 otherwise)
isal        !       1          ! 1 = Solve scalar transport equation (0 otherwise)
itrsch      !       4          ! Scalar advection 1=centered(C);2=upwind(U);3=U at k=k1z+1; (4)=Flux limiter
cd          !   0.002          ! Bottom drag coefficient (adimensional)
ifsbc       !       0          ! Surface BC; 0 = constant ; 1 = preprocess mode ; 2 = runtime mode
dtsbc       !      0.          ! Time step (s) between records in surbc.txt (when ifsbc = 1)
cw          !    1E-3          ! Wind drag coefficient (only if ifsbc = 0)
ws          !     5.0          ! Wind speed (constant in m/s) when ifsbc = 0
phi         !   270.0          ! Wind direction (constant in degress) when ifsbc = 0
idbg        !       0          ! 0 = no check messages 1 = check messages output for debug purposes
nthread     !       8          !
```

Fig 2.8. Example of constant wind definition at si3d_inp.txt.

7.2. Variable surface boundary conditions - surfbc.txt file required

7.2.1. Data from one station

Folders: met_cond/oneST/prepocess and met_cond/oneST/runtime

Scripts: surfbc4si3d_PREPROCESS.m and surfbc4si3d_RUNTIME.m

Example: surfbc.txt

If *ifsbc* is equal to 1 or 2, data provided on *surfbc.txt* correspond to only one meteorological station and is not need to specify the location of that station. The difference between these two options is that *ifsbc* = 1 corresponds to the *preprocess mode*, which means that the model reads surface heat fluxes from the surface boundary condition file, while *ifsbc* = 2 correspond to the *runtime mode*, which means that the model calculates the surface heat fluxes and reads meteorological conditions from the surface boundary condition file. The first seven lines of the surface boundary condition file are the same for either case. Lines 1 to 6 are header lines that are ignored by the model. Line seven contains the number of records in the file. The remainder of the file contains the time series of data necessary for the model. Data from the *surfbc.txt* file is reading at intervals of time define by *dtsbc* (in seconds) at the "Parameters controlling solution algorithm" in the *si3d_inp.tx*". The following Table 2.1 and 2.2 summarize the information necessary for either case. Examples of each mode are also presented.

Table 2.1. Surface foundary condition file for ifsbc = 1 (prerpocess mode)

SI3D name	Variable	Units
-	Time or frames	Hour or none
eta	Extinction coefficient	m^{-1}
Qsw	Shortwave radiation, corrected for albedo	KJ m^{-2} s^{-1}
Qn	Net radiation, includes shorwave radiation	KJ m^{-2} s^{-1}
cdw	Wind drag coefficient	-
uair	u-direction wind speed	ms^{-1}
vair	v-direction wind speed	ms^{-1}

Table 2. 2. Surface boundary condition file for ifsbc = 2 (runtime mode)

SI3D name	Variable	Units
-	Time or frames	Hour or none
eta	Extinction coefficient	m^{-1}
Qsw	Shortwave radiation, corrected for albedo	KJ m^{-2} s^{-1}
Ta	Air temperature	ºC
Pa	Atmospheric pressure	Pascals
Rh	Relative humidity	Fraction
cc	Cloud cover	Fraction
cdw	Wind drag coefficient	-
uair	u-direction wind speed	ms^{-1}
vair	v-direction wind speed	ms^{-1}

Example file: surfbc.txt (not full data) for one station and runtime mode

```
Surface boundary condition file for si3d model
Beznar Reservoir simulations
Time is in given in hours from 16:30 hrs on julian day 248,2009
   Time in   // Data format is (10X,G11.2,...) Time attc Hsw Ta Pa hr cc cw ua va
   5-min    // SOURCE = Beznar Met Station 2009 - ACC)
 intervals  (Note : file prepared on 11-Nov-2011
   npts = 2020
    0.0000     0.5188   518.7870    27.8500 96045.0000     0.5430     0.2054     0.0015    -6.2075     5.5832
    0.0833     0.5188   503.6228    27.8090 96045.0000     0.5503     0.2054     0.0015    -6.6283     5.8536
    0.1667     0.5188   488.4586    27.7680 96045.0000     0.5577     0.2054     0.0015    -7.0491     6.1241
    0.2500     0.5188   473.2944    27.7270 96045.0000     0.5650     0.2054     0.0015    -7.4698     6.3945
    0.3333     0.5188   456.3490    27.6693 96045.0000     0.5637     0.2054     0.0015    -6.5627     6.0830
    0.4167     0.5188   439.4037    27.6117 96045.0000     0.5623     0.2054     0.0015    -5.6555     5.7716
    0.5000     0.5188   422.4583    27.5540 96045.0000     0.5610     0.2054     0.0015    -4.7484     5.4601
    0.5833     0.5188   405.1692    27.5050 96045.0000     0.5640     0.2054     0.0015    -4.7889     5.5066
    0.6667     0.5188   387.8801    27.4560 96045.0000     0.5670     0.2054     0.0015    -4.8293     5.5531
    0.7500     0.5188   370.5911    27.4070 96045.0000     0.5700     0.2054     0.0015    -4.8698     5.5997
    0.8333     0.5188   352.6736    27.2840 96045.0000     0.5787     0.2054     0.0015    -4.9105     5.6465
    0.9167     0.5188   334.7561    27.1610 96045.0000     0.5873     0.2054     0.0015    -4.9512     5.6932
    1.0000     0.5188   316.8386    27.0380 96045.0000     0.5960     0.2054     0.0015    -4.9918     5.7400
    1.0833     0.5188   296.8624    26.9157 96048.3333     0.5997     0.2054     0.0015    -4.9726     5.9164
    1.1667     0.5188   276.8861    26.7933 96051.6667     0.6033     0.2054     0.0015    -4.9533     6.0928
    1.2500     0.5188   256.9098    26.6710 96055.0000     0.6070     0.2054     0.0015    -4.9341     6.2692
    1.3333     0.5188   233.0611    26.6547 96055.0000     0.6093     0.2054     0.0015    -4.9576     5.8347
    1.4167     0.5188   209.2123    26.6383 96055.0000     0.6117     0.2054     0.0015    -4.9811     5.4001
    1.5000     0.5188   185.3635    26.6220 96055.0000     0.6140     0.2054     0.0015    -5.0046     4.9656
    1.5833     0.5188   134.8545    26.4837 96058.3333     0.6193     0.2054     0.0015    -5.3997     5.3577
    1.6667     0.5188    84.3455    26.3453 96061.6667     0.6247     0.2054     0.0015    -5.7949     5.7497
    1.7500     0.5188    33.8365    26.2070 96065.0000     0.6300     0.2054     0.0015    -6.1900     6.1418
    1.8333     0.5188    22.5576    26.0850 96068.3333     0.6347     0.2054     0.0015    -6.2420     5.9971
    1.9167     0.5188    11.2788    25.9630 96071.6667     0.6393     0.2054     0.0015    -6.2939     5.8523
    2.0000     0.5188     0.0000    25.8410 96075.0000     0.6440     0.2054     0.0015    -6.3458     5.7076
    2.0833     0.5188     0.0000    25.7117 96078.3333     0.6500     0.2054     0.0015    -6.6131     6.0560
    2.1667     0.5188     0.0000    25.5823 96081.6667     0.6560     0.2054     0.0015    -6.8803     6.4044
    2.2500     0.5188     0.0000    25.4530 96085.0000     0.6620     0.2054     0.0015    -7.1476     6.7528
    2.3333     0.5188     0.0000    25.1943 96088.3333     0.6793     0.2054     0.0015    -6.5393     6.6459
    2.4167     0.5188     0.0000    24.9357 96091.6667     0.6967     0.2054     0.0015    -5.9310     6.5391
    2.5000     0.5188     0.0000    24.6770 96095.0000     0.7140     0.2054     0.0015    -5.3228     6.4323
    2.5833     0.5188     0.0000    24.6047 96101.6667     0.7293     0.2054     0.0015    -5.1167     6.2807
    2.6667     0.5188     0.0000    24.5323 96108.3333     0.7447     0.2054     0.0015    -4.9107     6.1292
    2.7500     0.5188     0.0000    24.4600 96115.0000     0.7600     0.2054     0.0015    -4.7046     5.9777
    2.8333     0.5188     0.0000    24.3957 96118.3333     0.7640     0.2054     0.0015    -5.3754     6.2066
    2.9167     0.5188     0.0000    24.3313 96121.6667     0.7680     0.2054     0.0015    -6.0461     6.4355
    3.0000     0.5188     0.0000    24.2670 96125.0000     0.7720     0.2054     0.0015    -6.7168     6.6644
    3.0833     0.5188     0.0000    24.1947 96128.3333     0.7753     0.2054     0.0015    -6.4560     6.3118
```

7.2.2. Data from several stations

Folder: met_cond/severalST

Scripts: surfBC_severalst.m

Example: surfbc.txt

If you use ifsbc = 11, you can use the information of several meteorological stations to force the model. Again, lines 1 to 6 are header lines that are ignored by the model. Line seven contains the number of stations and line eight the pairs (i, j) coordinates in the SI3D grid of the stations location. In this case, the number of records in the file is contained in line nine. The remainder of the file contains the time series of data necessary for the model. The following Table 2.3 summarizes the information necessary. You must be aware that time, extinction coefficient and atmospheric pressure are the same for all the stations and that information is only written once, while the rest of variables are different and repeated for each station. As it was mentioned before, data from the *surfbc.txt* file is reading at intervals of time define by *dtsbc* (in seconds) at the "Parameters controlling solution algoritm" in the *si3d_inp.txt*.

Table 2.3. Surface boundary condition file for ifsbc = 11

SI3D name	**Variable**	**Units**
-	Time or frames	Hour or none
eta	Extinction coefficient	m^{-1}
Pa	Atmospheric pressure	Pascals
Sw	Penetrative component of heat flux	KJ m^{-2} s^{-1}
Ta	Air temperature	^{o}C
Rh	Relative humidity	Fraction
lw	Longwave radiation	KJ m^{-2} s^{-1}
uair	u-direction wind speed	ms^{-1}
vair	v-direction wind speed	ms^{-1}

Example file: surfbc.txt (not all records and only 2 stations) for several stations

```
Surface boundary condition file for si3d model
Lake Tahoe simulations
Time is in given in hours (Grenwhich time) from 00:00 hrs on julian day 185, 2008)
   Time in   // Data format is (10X,G11.2,...) Time attc Pa Swn Ta Rh Lwin ua va
   60-min    // SOURCE = UCDavis )
 intervals  (Note : file prepared on 11-Nov-2011)
 No. Stats 10
 Grid Locs      96.00      65.00      29.00      45.00       9.00     115.00      54.00     172.00      42.00      62.00      92.00      15.00      22.00     138.00
  pnts = 1585
```

Time attc	Pa	Swn	Ta	Rh	Lwin	ua	va	Swn	Ta	Rh	Lwin	ua	va
0.0000	0.0500	81200.000	90.4007	22.0583	0.1995	324.5000	1.1817	2.1496	90.4007	22.0583	0.1995	324.5000	2.0800	-0.2607
0.1667	0.0500	81200.000	85.2313	22.0551	0.1991	323.6333	1.2253	0.4508	85.2313	22.0551	0.1991	323.6333	2.0686	-0.2956
0.3333	0.0500	81200.000	80.0619	22.0519	0.1987	320.4667	0.9047	1.2377	80.0619	22.0519	0.1987	320.4667	2.0566	-0.3302
0.5000	0.0500	81200.000	74.8925	22.0486	0.1982	318.4000	1.9839	0.6792	74.8925	22.0486	0.1982	318.4000	2.0440	-0.3645
0.6667	0.0500	81200.000	69.7232	22.0454	0.1978	318.4333	2.1436	0.7006	69.7232	22.0454	0.1978	318.4333	2.0309	-0.3984
0.8333	0.0500	81200.000	64.5538	22.0422	0.1974	318.4667	2.3042	0.7177	64.5538	22.0422	0.1974	318.4667	2.0172	-0.4320
1.0000	0.0500	81200.000	59.3844	22.0390	0.1969	318.5000	2.4658	0.7304	59.3844	22.0390	0.1969	318.5000	2.0030	-0.4652
1.1667	0.0500	81200.000	54.2150	22.0358	0.1965	315.8333	2.1328	0.0410	54.2150	22.0358	0.1965	315.8333	1.9882	-0.4981
1.3333	0.0500	81200.000	49.0456	22.0325	0.1960	313.5667	2.1462	-0.1225	49.0456	22.0325	0.1960	313.5667	1.9729	-0.5306
1.5000	0.0500	81200.000	43.8763	22.0293	0.1956	311.5000	2.1309	0.3299	43.8763	22.0293	0.1956	311.5000	1.9570	-0.5627
1.6667	0.0500	81200.000	38.7069	22.0261	0.1952	311.4667	2.0057	0.1485	38.7069	22.0261	0.1952	311.4667	1.9407	-0.5944
1.8333	0.0500	81200.000	33.5375	22.0229	0.1947	311.4333	1.8661	-0.0109	33.5375	22.0229	0.1947	311.4333	1.9238	-0.6258
2.0000	0.0500	81200.000	28.3681	22.0197	0.1943	311.4000	1.7148	-0.1470	28.3681	22.0197	0.1943	311.4000	1.9064	-0.6567
2.1667	0.0500	81200.000	23.1988	22.0164	0.1939	312.6000	1.7417	-0.2077	23.1988	22.0164	0.1939	312.6000	1.8885	-0.6872
2.3333	0.0500	81200.000	18.0294	22.0132	0.1934	308.3333	1.4462	1.2308	18.0294	22.0132	0.1934	308.3333	1.8702	-0.7173
2.5000	0.0500	81200.000	12.8600	22.0100	0.1930	302.9000	-1.9743	2.8196	12.8600	22.0100	0.1930	302.9000	1.8514	-0.7470
2.6667	0.0500	81200.000	9.9833	21.9233	0.1973	301.6000	-1.5913	2.3981	9.9833	21.9233	0.1973	301.6000	1.8321	-0.7762
2.8333	0.0500	81200.000	7.1067	21.8367	0.2017	300.3000	-1.2307	1.9594	7.1067	21.8367	0.2017	300.3000	1.8123	-0.8050
3.0000	0.0500	81200.000	4.2300	21.7500	0.2060	299.0000	-0.8933	1.5045	4.2300	21.7500	0.2060	299.0000	1.7921	-0.8333
3.1667	0.0500	81200.000	0.0000	21.3100	0.2427	293.6333	-2.2321	1.9864	0.0000	21.3100	0.2427	293.6333	1.7715	-0.8612
3.3333	0.0500	81200.000	0.0000	20.6367	0.2210	291.5000	-2.7958	2.3129	0.0000	20.6367	0.2210	291.5000	1.7504	-0.8886
3.5000	0.0500	81200.000	0.0000	19.5800	0.2350	292.0000	-4.1857	2.4166	0.0000	19.5800	0.2350	292.0000	1.7290	-0.9156
3.6667	0.0500	81200.000	0.0000	19.1333	0.2403	290.8333	-3.9677	2.4825	0.0000	19.1333	0.2403	290.8333	1.7071	-0.9420
3.8333	0.0500	81200.000	0.0000	18.6867	0.2457	289.6667	-3.7504	2.5360	0.0000	18.6867	0.2457	289.6667	1.6848	-0.9680
4.0000	0.0500	81200.000	0.0000	18.2400	0.2510	288.5000	-3.5344	2.5773	0.0000	18.2400	0.2510	288.5000	1.6621	-0.9935
4.1667	0.0500	81200.000	0.0000	16.7367	0.2740	293.6667	-3.9499	1.9237	0.0000	16.7367	0.2740	293.6667	1.6391	-1.0186
4.3333	0.0500	81200.000	0.0000	15.8933	0.2720	291.0000	-4.2772	1.8038	0.0000	15.8933	0.2720	291.0000	1.6157	-1.0431
4.5000	0.0500	81200.000	0.0000	15.2200	0.2710	288.6000	-3.7482	2.1990	0.0000	15.2200	0.2710	288.6000	1.5919	-1.0671
4.6667	0.0500	81200.000	0.0000	15.1267	0.2730	288.6000	-3.8339	2.0357	0.0000	15.1267	0.2730	288.6000	1.5678	-1.0906
4.8333	0.0500	81200.000	0.0000	15.0333	0.2750	288.6000	-3.9126	1.8690	0.0000	15.0333	0.2750	288.6000	1.5433	-1.1136
5.0000	0.0500	81200.000	0.0000	14.9400	0.2770	288.6000	-3.9840	1.6993	0.0000	14.9400	0.2770	288.6000	1.5186	-1.1360
5.1667	0.0500	81200.000	0.0000	14.0333	0.2940	289.0333	-3.4457	1.3089	0.0000	14.0333	0.2940	289.0333	1.8046	-2.0833
5.3333	0.0500	81200.000	0.0000	13.5733	0.2963	288.1667	-3.4628	1.3177	0.0000	13.5733	0.2963	288.1667	2.3608	-1.0005
5.5000	0.0500	81200.000	0.0000	13.1500	0.3070	288.7000	-3.7175	1.7335	0.0000	13.1500	0.3070	288.7000	1.9487	-1.1755
5.6667	0.0500	81200.000	0.0000	12.9733	0.3077	288.1667	-3.6932	1.7065	0.0000	12.9733	0.3077	288.1667	2.0266	-1.3178

Common variables | Station 1 | Station 2

Chapter 3

Outputs: Code and Representation

1. Output routines

Si3d outputs subroutines are presented here, these subroutines are in si3d_utils.f90.

1- Output files is created and opened

```
CHARACTER(LEN=12) :: output_file="output.txt"
OPEN (UNIT=i6, FILE=output_file, IOSTAT=ios)
IF(ios /= 0) CALL open_error ( "Error opening & &"//output_file, ios )
```

2- To write in opened file, first a format is specified and then the write sentence with the variables is used, for example:

```
WRITE (UNIT=i6, FMT='("Run number = ",A8,A4,",Start &  & date of run: ",I2,"/",I2,"/",I4," at ",I4.4," & &hours")')date, time(1:4),imon,iday,iyr,ihr
```

3- Some variables used to control the outputs.

n	→ Specifies the time step number.
(i,j)	→ Water column Coordinates in 2D like h bathymetry file.
(i,j,k)	→ Water column Coordinates in 2D and depth of the water column.
l	→ Water column Coordinate in 1D, only not dry columns have this id.
(l,k)	→ Water column Coordinate in 1D and depth of the not-dry water column.
kmz(l)	→ Specifies depth of the not-dry water columns.
jm	→ Specifies number of files in 2D coordinates.
im	→ Specifies number of columns in 2D coordinates.
lm	→ Specifies number of not-dry columns water in 1D coordinates.
cm	→Specifies number of columns water in 1D coordinates.

ij2l(i,j) → Coordinates i,j 2D are converted in l 1D.

l2i(l) → Obtain the coordinate i 2D from l 1D coordinate.

l2j(l) → Obtain the coordinate j 2D from l 1D coordinate.

l2c(c) → Obtain the number of not-dry column water in 1D coordinates.

mask2d(i,j) → Return true if the coordinate i,j 2D specified is a not-dry water column

mask(c) → Return true if the coordinate c 1D specified is a not-dry water column

4- Subroutines used to control the outputs.

a) SUBROUTINE outr

To output model run parameters & performance measures to file.

b) SUBROUTINE outt

To write output to timefile(s). A separate timefile is opened for each node where output is requested.

c) SUBROUTINE outw

To write the wind field provided to the model as boundary condition.

d) SUBROUTINE outv

To write output a cross section to a binary file.

e) SUBROUTINE outNB

To write transport and scalar values at a cross section to a binary file to be used as boundary conditions in nesting procedure.

f) SUBROUTINE outh

To write output at a specific layer in binary format

g) SUBROUTINE outz

To write tracer concentrations in computational domain

h) SUBROUTINE outp

To write the complete solution in the computational domain. The resulting file is used to drive particle tracking simulations with PTRACK-TOOL.

i) SUBROUTINE outs

To write ascii output in xml format to a file used for velocity and particle-tracking animations with the Gr application. The file, called 'spacefile.xml', is essentially a header file for the sequential binary files (spacefile3d.bin and spacefile2d.bin) written out in SUB outs_bin. The binary files contain the 2d

and 3d data from all the wet spatial nodes in the solution at snapshots in time.

j) SUBROUTINE outs_bin

To write output in binary format to two spacefiles for velocity and particle-tracking animations with the Gr application. The two files are: spacefile3d.bin and spacefile2d.bin. 3-D variables are stored in the '3d' file and 2-d variables in the '2d' file. The ordering of nodes written to the file must agree with the si3d_bathy.xml file written out in SUB outg. The velocities written to the file are taken from the faces of the grid cell control volumes. The two unformatted files written to in this subroutine are sequential. They are written as 'streaming' data (no record breaks in the file) using the 'little-endian' bit order for the storage layout. The data values themselves are stored as 2-byte integers. Each call to the subroutine outputs one time step of data to the spacefiles. The CALL and file OPEN statements for this subroutine are within SUB outs_xml.

k) SUBROUTINE outg

To process and output bathymetry to a file for use in graphics post-processing and particle tracking. The bathymetry output by this subroutine is processed from the original bathymetry read into the program in SUB bathy. The processed bathymetry is similar to that processed in SUB bathy for the actual model calculations except that depths are defined at the corners of each grid cell rather than the mid-sides.

2. Representation of output files (MATLAB scripts)

As it was mentioned when *si3d_input.txt* was described, there are different types of outputs from the model. In this section, it is listed the types of SI3D output files, the contained information and the names of the MATLAB scripts used to process each sort of information. In addition, there is a distinction between the scripts which read all the fames and only one frame of time.

Table 3.1. Outputs and MATLAB scripts

SI3D output file	**Information**	**MATLAB scripts**
tf*_*.txt	Time series (points)	outputs/allFrames/plot_profile_timeseriesTRACER_DZ.m
plane_*	H-planes	outputs/oneFrame/LoadPlotOneSheetFile.m outputs/allFrames/LoadPlotSheetFile3L.m
section_*	Cross sections	outputs/allFrames/LoadPlotSectionFile3D.m
tracer_*	Tracers	outputs/oneFrame/LoadPlotOneTracerFile.m outputs/allFrames/LoadPlotTracerFile3L.m outputs/allFrames/PlotTracerFile3L_xyz.m