

## How to Compile and Run fishsim

This is a short description how to compile and run the fishsim program. It describes briefly how to do this for Unix and one Windows compiler. It also describes how to change the simulation operating controls.

### Unix

1. In your home directory create a directory for the code. I recommend: `/home/blemasson/d/c/Cfish4` this will match my current version level

(use `mkdir` to make the directories)

2. Download the appropriate zipped tar file. These might be `Cf4sim05.tgz` (the source code and some data files) and `Dnvr_601_xyzuv.txt` (the text file of flows for the Denver experimental flume).

Copy these files to your fish directory: E.g. `cp /tmp/Fish/xxxx ~/d/c/Cfish4`

3. Expand the compressed tar file:

```
tar xzf Cfsim05.tgz
```

(Read unix docs on tar: "man tar" or "info tar")

4. Compile and link the program using make. Examine the set of files that have been expanded. Those of the form `mk*` are the make files for the simulation program. An example is: `mkCfsim0`. Make the program using

```
make -f mkCfsim0
```

(Read unix docs on make: `man make` or `info make`)

5. Edit the control file:

```
nedit Dcontrol.dat &
```

(Read the built-in nedit help. Read the comments in the `Dcontrol.dat` for the meaning of the variables. Also see below the Section Control File Documentation)

6. Run the program

```
fishsim
```

when prompted, type in the name of the control file. If your control file is called "DefltControl.dat" you can just hit return.

7. If your control file asks for DISLIN plots to the console, and asks for pausing, then the program will stop and wait for input from the terminal window you used to start fishsim.

If your control file requests no graphical output or instructs DISLIN to send the graphics to a file, the filename for output will be found in a small message written by DISLIN on the terminal window from which fishsim was run.

### Windows

1. See the instructions for downloading and installing the GNU C compiler for Windows called MinGW and the associated free plotting package for that compiler called DISLIN. You need

both of these working before you can compile and run `fishsim`.

2. `Fishsim` runs as a commandline application in Windows 9x and Windows 2000. (It has not been tested in Windows XP.) It does not use the Windows GUI, and it probably never will. Get over it.
3. Start a command line window. Do: Start ->Run and in the run line type: `cmd /K c:\autoexec.bat` (The file `autoexec.bat` must contain the information described below for the installation of the compiler and graphics library.)
4. Create a directory for the code. I recommend something like: `c:\Documents and Settings\YOUR_NAME\c\fish`

but you can do this anywhere.

(Use `mkdir` to make the directories.)

5. Download the appropriate ZIP file. Currently these might be `ver7.zip` that contains the source and all data files, including the large (7 MB) text file of flows for the Denver experimental flume).

Copy these files to your fish directory either using the DOS `copy` command or the Windows file manager.

6. Expand the compressed tar file. Use whatever Windows unzipper application you have installed.
7. Compile and link the program using `make`. Examine the set of files that have been expanded. Those of the form `mk*` are the make files for the simulation program. An example is: `mkCfsim0`. Make the program using

```
mingw32-make -f mkCfsim0
```

If none of your source code is corrupted, this should compile cleanly.

8. Edit the control file:

Use any Windows text editor, e.g. Notepad

See the documentation below for what the lines of the control file mean.

9. Run the program

```
fishsim
```

when prompted, type in the name of the control file. If your control file is called "DefltControl.dat" you can just hit return.

10. If your control file asks for DISLIN plots to the console, and asks for pausing, then the program will stop and wait for input from the terminal window you used to start `fishsim`.

If your control file requests no graphical output or instructs DISLIN to send the graphics to a file, the filename for output will be found in a small message written by DISLIN on the terminal window from which `fishsim` was run.

## The Control File

Fishsim uses a single control file to define all of the inputs needed. This minimizes the command-line input required of the user. The user uses an external text editor to change the operation of the code.

Here is an example of a control file. Following each line in italics is a description of that line. The italicized material is not part of the control file itself. The general form of the control file is 3 columns (except the last line). Column 1 is a comment that is a short description of the data item. Column 2 is the data that is read by fishsim. Column 3 is another comment field that gives a longer description and can hold default values of the data field. NOTICE: There must be NO SPACES in the first or last comment columns. Use underscores for readability.

```
Xvel_file_name      Dnvr_601_xyzuv.txt  %(x,y)_pos_and_(u,v)_infile
    The name of the text file containing the fluid flows in the x direction
Yvel_file_name      Dnvr_601_xyzuv.txt  %y-dir_velocities_(downstream)
    The name of the text file containing the fluid flows in the y direction
Barriers_name       Dnv_bars.txt      %barriers_locs_in_channel
    The name of the text file containing the locations of the barriers
Parameters          input6.dat      %fish_parameter_file
    The name of the text file containing fish parameters
Debug_name          debug.log      %debug_file
    The name of the text file containing the debugging output
Num_reps            1              %#MC_reps
    The number of Monte Carlo runs when doing efficiency simulations
Num_fish/rep        1              %#fish/rep
    The number of fish to run sequentially (not schooling)
deltat              0.01           %delta_time(seconds)[0.1][0.05]
    The nominal time step when the fish is in prolonged swimming mode
tlast               15.0           %max_simulation_time(seconds)[180]
    The number of seconds to simulate
plotdelt            0.5           %delta_time_for_graph_plot[1.0]
    The number of seconds between plotting episodes
printdelt           0.1           %delta_time_for_printing[1.0]
    The number of seconds between printing episodes
output              8              %output_level[6=plot]
    The level of output desired; must be greater than 6 for plots
debug               0              %debugging_level
    The level of debugging desired; large numbers imply lots of output
Random_seed         -1            %random_seed[12345]
    The random seed for Monte Carlo runs; -1 implies no random numbers
DISLIN              1              %1=dislin_plot,0=not
    Flag for DISLIN plotting
PlotXmin            0.0           %Plot_min_for_flume_Xaxis[0.0][14.0][2.75]
    Plotting window: min X (in meters)
PlotXmax            4.88          %Plot_max_for_flume_Xaxis[18.6][17.0][3.10]
    Plotting window: max X (in meters)
PlotYmin            0.0           %Plot_min_for_flume_Yaxis[0.0][1.3][2/0]
    Plotting window: min Y (in meters)
```

```

PlotYmax          3.2          %Plot_max_for_flume_Yaxis[2.4][1.6][2.25]
    Plotting window: max Y (in meters)
SkipFlow          8            %plot_every_nth_flow_vector
    Number of flow vectors to skip for plots; depends on plotting window size
ScaleFlow         0.05        %DISLIN:scale_flow_vectors
    Amount to scale the flow vectors for plotting; depends on plotting window size
PlotType          P            %Plot_fish_as_Path_or_Fish[F]
    Flag to plot object as connected arrows (P) or as a directed fish (F, broken)
PXL_INCH          273         %DISLIN:pixels/inch_for_PS_output
    Resolution for plotting to a graphics file
DEVICE            cons        %DISLIN:output_device["cons"=screen]\
                                ["pdf"=pdf]["post"=greyPS]["pscl"=clrPS]
    String Flag to direct plotting output to screen or graphics files
BACKGROUND        none        %DISLIN:background["norev"=whiteonblk]["revers"]
    String Flag for plotting background: for cons use norev, pdf use revers
ORIENT            none        %DISLIN:orientation[none]
    String Flag to control plotting orientation; leave as none
FILEOUT           F08_near     %filename_for_non-screen_output_(no_extension)
    Name of file for graphic file output
PzTime_s,_e      -1 -1       %start_and_end_times_for-pauses(-1=>no_pause)
    Start and Stop times to step through graphic display interactively

```