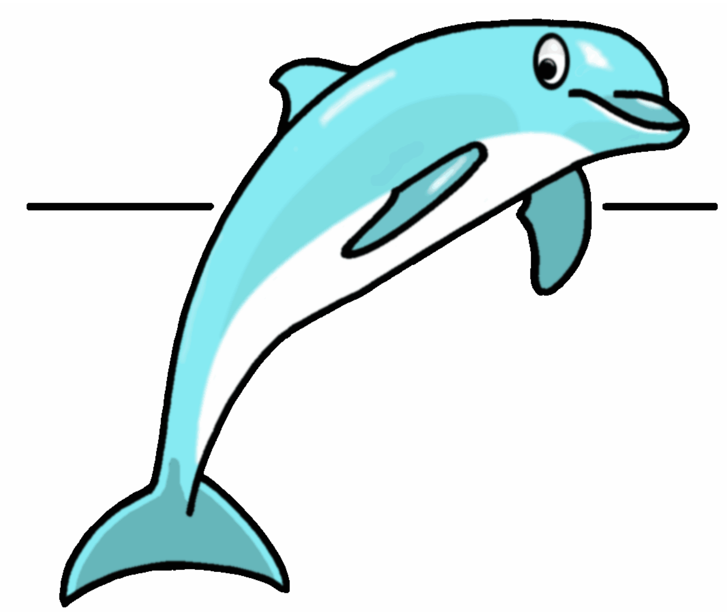
CYCLONE
Fluid Dynamics

Dolfyn INput guide

vs 0.6xx DRAFT



CFD-151201
19th December 2015

Cyclone Fluid Dynamics B.V.
Sweelinckklaan 4 Tel.: +31-40-22 30 491 Web: www.cyclone.nl
NL–5583 XM Waalre Fax.: +31-40-22 30 490 email: info@cyclone.nl

CYCLONE
Fluid Dynamics

Dolfyn INput guide

vs 0.6xx DRAFT

Cyclone Fluid Dynamics BV

Author:
H.W. Krüs

CFD-151201
Cyclone Fluid Dynamics B.V.
19th December 2015

Cyclone Fluid Dynamics B.V.
Sweelinckklaan 4 Tel.: +31-40-22 30 491 Web: www.cyclone.nl
NL–5583 XM Waalre Fax.: +31-40-22 30 490 email: info@cyclone.nl

Contents

1 Introduction

Commands can be grouped into some categories:

1. General commands
2. Control parameters
3. Fluid properties
4. Postprocessing
5. Boundary conditions

1.1 Syntax

The command line syntax is:

Call:	**command,** *value [, option 1 | option 2]*
Default(s):	default(s)
Example:	command, command example

Arguments are separatated by one comma or one or more blank spaces (multiple blank spaces count as one, and one comma and one or more spaces count as one comma).

2 General commands

2.1 Title

Call:	**title,** *string*
Default(s):	(empty)
Example:	title, A dolfyn simulation

Adds a title to various files. One line only.

2.2 Steps

Call:	**steps,** *number [, resmax]*
Default(s):	100, 1.e-4
Example:	steps, 400, 1.e-5

Number of iteration steps (steady state) or number of time steps (time dependent). Optional the final residu.

2.3 Thermal model

Call:	**thermal,** *off | on*
Default(s):	off
Example:	thermal, on

Switch thermal model on or off.

2.4 Turbulence model

Call:	**turbulence,**	*off | ke [, length] | rng [, length] |* *chen [, length] | kl [, length] | mmk [, length]*
Default(s):	off	
Example:	turbulence, ke	

Choose one of:

off (no turbulence model)

ke standard k-ε turbulence model

rng the RNG k-ε turbulence model

che Chen's variant of the standard k-ε turbulence model

kl Kato & Launder's k-ε turbulence model

mmk Murakami, Mochida and Kondo k-ε turbulence model

The value of *length* is used during initialisation (first estimate of the turbulent dissipation ε).

2.5 Time dependency

Call:	**transient,** *timestep [, quad [, blend]]*
Default(s):	off
Example:	transient, 0.1

Selects time dependent options standard implicit Euler (only one time step is saved) or a second order Euler scheme ('quad', saves two time steps). 'Blend' sets the blending between the two schemes.

2.6 Restart

Call:	**restart,** *(empty) | no | off | initial [| cell flux] | reset*
Default(s):	no
Example:	restart, reset

A simulation can be restarted. The following options are avialable:

(empty) Restart the simulation.

no | *off* No restart. Start again ignoring everything.

initial Use a restart file as initial guess for a new simulation.

initial cell flux Use a restart file as initial guess for a new simulation but ignores the stored fluxes.

reset Restart but reset all counters (iteration or time step to zero).

2.7 Comments

Call:	*# string*
Default(s):	(empty)
Example:	# just some comment.

Everything following a # will be ignored. A comment can follow a command.

2.8 Save

Call:	**save,** *every, number | time, number | ...*
	iteration, number | cpu, time [s|m|h]
Default(s):	iteration, 500
Example:	save, iter, 100

Set how often a restart file should be saved. Choose a suitable value, not too short, nor too long. Options are:

every Every n steps.

time After 'delta time' for transient simulations.

iteration Every number of iteration steps.

cpu After an amount of cpu time used.

2.9 Output

Call:	**output,** *every, number | time, number | ...*
	iteration, number
Default(s):	(none)
Example:	output, iter, 100

Set how often a postprocessing file should be written. Options are:

every Every n steps.

time After 'delta time' for transient simulations.

iteration Every number of iteration steps.

2.10 Use

2.10.1 Patches, particles, sensors

Call:	**use,** *patches, scalars | particles, number | sensors, number*
Default(s):	(none)
Example:	use, particles, 100

Switches extra features on. Currently available are:

patches Using patches.

particles Using particles.

sensors Using sensors.

2.11 Dimension

Call:	**dimension,** *scalars, number*
Default(s):	(none)
Example:	dimension, scalars, 4

Allocate space for extra scalar transport equations.

2.11.1 Gauss

Call:	**use,** *Gauss, iterations*
Default(s):	2
Example:	use, Gauss, 4

Switches to the Gauss method for the calculation of the gradients. The number of iterations sets the number of passes.

2.11.2 Least squares

Call:	**use,** *least squares*
Default(s):	(none)
Example:	use, least squares

Switches to the least squares method for the calculation of the gradients.

2.11.3 Lapack

Call:	**use,** *LAPACK*
Default(s):	(none)
Example:	use, LAPACK

Switches to the LAPACK subroutine SGESV to be used in the calculation of the gradients using the Least Squares method.

2.11.4 Fix ABL

Call:	**use,** *FixABL*,u,v,w,k,ε
Default(s):	0.0,0.0,0.0,0.0,0.0
Example:	use, FixABL, 10.0,0.0,0.0,1.0,0.01

Special subroutine for Atmosferic Boundary Layers. Use in conjunction with UserInitialField and UserInlet.

2.12 Debug

Call:	**debug,** *number*
Default(s):	0
Example:	debug, 2

Increases the verbosity of the output (both to the console and the debug file). Only useful for developers.

2.13 Check out

Call:	**check,** *variable, range | average, real1, real2 [, report]*
Variable:	u | v | w | p | k | eps | T
Default(s):	(none)
Example:	(see below)

For debugging, testing and check out purposes. Two forms are available: range and average. First variant only checks the minimum and maximum of the cell centered variable values. The latter computes a volume weighted average (using a bandwidth to check against).

If a check fails it will be reported in the form " *** Test Variable V FAILED ***" which can be detected in a check out script. Of course when one combines "limit" with "check" no failures will appear; this combination is therefore not recommended for testing purposes. The command is silent successfully, unless "report" has been appended.

Example:

```
check u range   0.00  1.0   report
check v range  -1.00  0.0
check w average 0.0   1.e-4
check p range  -0.45  0.75
```

2.14 Math

Call:	**set,** *variable, expression*
Default(s):	(none)
Example:	set T 273 + 500.

and

Call:	**math,** *variable, [function | expression] | degrees*
Default(s):	(none)
Example:	math u cos $angle * $uoo

Use 'set' in order to define a variable and 'math' to do some math with a variable. Retrieve a variable with the dollar sign ('$').

An example:

```
set T 273 + 500.
set rho 100000. / 287. / $T

vislam  36.4e-06
density $rho

set  angle 30
set  uoo 1.0
math degrees
math u cos $angle * $uoo
math v sin $angle * $uoo
```

Available math functions are: cos, sin, tan, abs, exp, log, ln, sqrt.

2.15 User

Call:	**user,** *user1 [, user2 [, user3 [, user4]]]*
Default(s):	(none)
Example:	user 1.0 2.0

Simple method to set user specifc data optionally needed in the user defined subroutines. Currently only four reals or integers.

3 Control parameters

3.1 Monitor

Call: **monitor,** *cell, cell_number | coor,* x, y, z
Default(s): 1
Example: monitor, cell, 2004

Picks a cel to be monitored during the run.

Note: legacy form "monitor, cell_number" remains avialable.

3.2 Blending factors

Call: **gamma,** *array*
Default(s): 0.0, 0.0, 0.0
Example: gamma, 0.95, , 0.25

Assigns the blending factors ('gamma'). The values for u, v, and w are equal, as well as for k and ε. The second value (for pressure) is useless, but present to be consistent with other commands.

A special extended version is available which allows all variables to be set:
Call: **gamma*,** *array*
Default(s): 0.0, 0.0, 0.0
Example: gamma*, 0.95,0.95,0.95, , 0.25

3.3 Alternative differencing schemes

Call: **scheme,** *variable,scheme,blend*
Default(s): (none)
Example: scheme, T, gamma, 0.8

Default is still blending of a central differencing scheme (CD1) with standard upwind differencing (UD) (command 'gamma').

The LUD, MinMod and Gamma schemes are based on the Convective Boundness Criterion. The choices are:

UD Standard upwind differencing.

CD Central differencing based on weighted distances (CD1).

CD2 An alternative to CD1 based on averaging the result of two estimates using the gradient.

CD3 Very simple averaging both sides of the face (ignoring every possible correction). For testing puposes only, however might be useful when awkward meshed have to be used.

LUD Linear upwind differencing (CBC based).

LUX Linear upwind straight on the rocks (to be used with slope limiters).

MIN A minmod scheme which blends UD, LUD and CD1.

GAMMA Jasak's scheme which blends UD and CD1.

Still work in progress.

3.4 Relaxation factors

Call:	**relax,** *array*
Default(s):	0.5, 0.2, 0.5, 0.95
Example:	relax, 0.6, 0.3, 0.6, 0.95

Sets the relaxation factors for the velocity components, the pressure and the scalar transport equations (turbulence components, temperature and scalars).

A special extended version is available which allows all variables to be set:

Call:	**relax*,** *array*
Default(s):	0.5,0.5,0.5, 0.2, 0.5,0.5, 0.5,0.5
Example:	relax, 0.6, 0.6, 0.6, 0.3, 0.4,0.4, 0.9

3.5 Selecting gradient method

Call:	**grad,** *variable,ls | gauss[,passes]*
Variable:	u | v | w | uvw | p | k | eps | kep| T | sca,[all | id]
Default(s):	(none)
Example:	grad, p, ls
	grad, gauss, T, 4

Default is Gauss' method for all the gradients and the alternative is the least squares method. This command allows to set or change it for individual variables.

3.6 Slope limiters

Call:	**slope,** *variable,[off|BJ|VN|VA|P1][f|c|n]*
Variable:	u | v | w | uvw | p | k | eps | kep | T | sca,[all | id]
Default(s):	(none)
Example:	slope, T, off
	slope, UVW, vnf

Slope limiters are needed for all tet and (very) bad meshes.

Choices are:

off Switch slope limiter off.

BJ Using the method by Barth & Jespersen (original).

VN Using the method by Venkatarishnan (BJ refined).

VA Using the Van Albada limiter (included only for testing purposes).

P1 Using an adapted polynomial.

The limiter can be tested on variuous points:

c At cell centres (conservative estimate with a damping effect).

f Using face centres (allow for a tiny overshoot).

n Using the cell nodes (considerable more effort and memory, the final result is in between 'f' and 'c'.

Using slope limiters in combination with the LUX linear upwind scheme allows for second order acuracy on all types of meshes (including all tet meshes). In such cases a good set of commands might be:

```
scheme UVW LUX

slope UVW vnf
slope p vnf
```

Optionally you can select least gradients for the pressure using 'grad,p,ls'. Note that the rest is left to the default upwind (UD) scheme; as the k-ε-model is dominated by sources this is not a harsh restriction.

3.7 Relative solver accuracy

Call:	**rtol,** *array*
Default(s):	0.1, 0.05, 0.1
Example:	rtol, 0.1, 0.01, 0.1

The relative solver accuracy of the linear solver *per inner iteration step*.

3.8 Switches

Call:	**switch,** *courant, number | maxouter, number | ...* *pp, number | cds , cd1 | cd2 | cd3 | lambda, off | test on|off*
Default(s):	(none)
Example:	switch,courant,0.35

Sets the particle Courant number (number of steps of a particle within a cell expressed as a fraction of cell length).

3.9 Initialisation

Call:	**init,** *field, array | user | fact, factor | steps, number of steps*
Default(s):	(field) 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 293.0
Example:	init, field, 1.0

There are several initialisation options at the beginning of a run:

field,array Sets the velocity components, pressure etc..

user Use a user written subroutine (subroutine UserInitialField)

fact, factor A temporary factor for the laminar viscosity.

steps, number of steps Number of initilisation steps.

3.10 Pressure iteration

Call:	**pcor,** *max, maximum | fac, factor*
Default(s):	(max) 4, (fac) 0.25
Example:	pcor, 8

Two parameters for the pressure iteration

max, maximum Maximum number of pressure corrections.

fact, factor Reduction factor in the pressure iteration.

3.11 Equations

Call:	**solver,** *variable, on | off*
or	
	solver, *variable, sparse | bcg | direct | user*
Variable:	u | v | w | p | k | eps | T
Default(s):	(none)
Example:	solver, w, off

Switch the solution of a transport equation (or pressure) on or off:

u U velocity component.

v V velocity component.

w W velocity component.

p Pressure.

k Turbulent kinetic energy.

eps Turbulent dissipation.

T Temperature.

Optionally other linear solvers can be activated (not implemented yet).

3.12 Limit

Call:	**limit,** *variable, off | lower | upper*
Variable:	u | v | w | p | k | eps | T
Default(s):	limit k lower 1.0e-09
	limit eps lower 1.0e-12
Example:	limit T lower 293.0

Limit scalars to enforce them to be positive or within a range. Use only when needed.

4 Fluid properties

4.1 Density

Call:	**density,** *density | constant, density | gaslaw, R*
Default(s):	1.2
Example:	density, 1.205

Sets the density.

4.2 Reference pressure

Call:	**pref,** *cell, cell_number | coor,* x, y, z
Default(s):	1
Example:	pref, cell, 2004

Sets where in the domain the (relative) pressure is '0'. Alle pressure are relative to this relative pressure.

Note: legacy form "pref, cell_number" remains avialable.

4.3 Gravity

Call:	**gravity,** g_x, g_y, g_z
Default(s):	0.0, 0.0, 0.0
Example:	gravity, 0.0, -9.81, 0.0

Sets the orientation of the gravity vector (only useful for thermal and/or particle analyses).

4.4 Expansion coefficient

Call:	**beta,** *number*
Default(s):	0.001
Example:	beta,0.003

Sets the expansion coefficient 'beta' (only useful for thermal analyses).

4.5 Laminar viscosity

Call:	**vislam,** *viscosity*
Default(s):	0.001
Example:	vislam, 18.6e-6

Sets the laminar, or dynamic, viscosity.

Please note that the 'kinematic viscosity' is the dynamic viscosity divided by the fluid density.

4.6 Specific heat capacity

Call:	**cp,** *cp*
Default(s):	1006.
Example:	cp, 1000.

Sets the specific heat capacity.

4.7 Prandtl number

Call:	**prandtl,** *number*
Default(s):	0.6905
Example:	prandtl, 7.

Sets the Prandtl number. See also *Heat conduction*.

4.8 Heat conduction

Call:	**conductivity,** *number*
Default(s):	0.02637
Example:	conductivity, 0.02

Sets the value of heat conduction.

Note: The Prandtl number and heat conduction are related:

$$Pr = \frac{\mu_{\mathrm{lam}} C_p}{\lambda}$$

Last call prevails.

4.9 Particles

Call: **particle,** *number, prop,density,diameter | number,init,1,x0,y0,z0,u0,v0,w0*
Default(s): (none)
Example: (see below)

Command to generate and release particles. Best illustrated by two examples:

Example 1:

```
set np 40
math deg
math a1 sin 45 * 2

use particles $np

part          1 prop 1000 100.e-6
gene $np - 1 1

part          1 init 1 0.0025 0.99 0.105   0.0   0.0 0.0
gene $np - 1 1,,,      0.01   0.00 0.000   0.0   0.0 0.0
...
```

Example 2:

```
set x0  -0.0265
set y0   0.0456
set z0  -0.20
set x1  -0.005
set nr 8
set dx $x1 - $x0 / $nr
set dy 0.0
set dz 0.0

use particles $nr

part      1 prop 1000. 20.e-6
gene $nr - 1 1

part          1   init 1 0.5 * $dx + $x0 $y0 $z0  0.0   0.0 0.0
gene $nr - 1 1,,,          $dx           $dy $dz  0.0   0.0 0.0
...
```

Note that the syntax is quite simple from ‘left to right’ using a ‘$’-sign to fetch a variable. Thus the result of ‘0.5 * $dx + $x0’ (adds half dx to x0) is different to the result of ’$x0 + 0.5 * $dx’ (adds 0.5 to x0 and multiplies the lot with dx).

4.10 Sensors

Call:	**sensors,** *number,init,1,x0,y0,z0*
Default(s):	(none)
Example:	(see below)

Command to generate sensors. Variables are interpolated to these points and the result is printed.

Example:

```
set ns 8
use sensors $ns
sens         1 1 0.06249 0.06249 0.0
gene $ns - 1 1,,0.125 0.0 0.0
```

5 Postprocessing

Several postprocessing options are available; choose one or more of them.

5.1 Print

Call:	**print,** *cell, range | wall, range | file, filename iteration, number*
Range:	user | all | start, end, increment
Default(s):	(none)
Example:	print, cell, 1,100,5

Print results. Options are:

cell Cell data. All cells, a cell range, or user defined.

wall Wall data. All walls, a wall range, or user defined.

file Output to file 'filename'.

5.2 OpenDX

Call:	**use,** *OpenDX*
Default(s):	(none)
Example:	use, opendx

Writes a '*.odx' file for OpenDX.

5.3 VTK

Call:	**use,** *VTK*
Default(s):	(none)
Example:	use, vtk

Writes '*.vtk' file for ParaView, VisIt, of MayaVi.

5.4 GMV

Call:	**use,** *GMV*
Default(s):	(none)
Example:	use, gmv

Writes a file for GMV (not ready yet).

5.5 Tecplot

Call:	**use,** *tecplot*
Default(s):	(none)
Example:	use, tecplot

Writes '*.dat' file for Tecplot.

5.6 CGX

Call:	**use,** *CGX*
Default(s):	(none)
Example:	use, cgx

Writes an '*.frd' output file for CGX (see *www.dhondt.de*).

5.7 Extra options

	post, *post, variable, cell / vert [, yes / no]*
Variable:	u l v l w l p l k l eps l T l sca l den l vis l lvi l vor
Default(s):	(see below)
Example:	post, T, vert, yes

Write extra nodal or cell data (if possible). Standard cell data is written for the solved transport equations only; nodal results have to be selected.

Vorticity of cells only.

5.8 Special OpenDX options

Call: **opendx,** *dump, steps*

or

opendx, *(normals, on | off) | (centers, on | off) | ...*
(massflux, on | off)

or

opendx, *post, variable, cell | vert [, yes | no]*

Variable: u | v | w | p | k | eps | T | sca | den | vis | lvi
Default(s): (none)
Example: opendx, post, T, vert

6 Boundary conditions

Boundary conditions are special because *all* boundary conditions have to be provided. The reason is that it thus enables simple switching options on or off.

The following options are currently avaliable:

1. **inlet** Inlet.
2. **outlet** Outlet.
3. **wall** Wall.
4. **symplane** Symmetry plane.

The numbering is arbitrary. The boundary region with index ’0’ is the default boundary set by the preprocessor when a boundary is found with no entry in the ’*.bnd’ file,

The calling sequence is arbitrary; the last call counts.

6.1 Inlet

Call:	**boundary,** *number | name, [, user]*	
	inlet	
	u, v, w	
	density	
	temperature	
	keps | inle	
	number, number	
Default(s):	(none)	
Example:	boundary, 1	
	inlet	
	5.0, 0.0, 0.0	# 3 velocity components
	1.2	# density
	293.0	# temperature (in Kelvin)
	keps	# k en ε selected
	1.e-4,1.e-4	# and the values for k en ε

The option *user* selects the user written subroutine ’UserInlet’. In this subroutine one can set all or some of the boundary conditions. Instaed of a number a name can be used (when a *.inp file was used by the preprocessor).

6.2 Outlet

Call:	**boundary,** *number | name*	
	outlet	
	number	
Default(s):	(none)	
Example:	boundary, 2	
	outlet	
	1.0	# relative amount for this outlet

Sets where the flow is allowed to flow out of the domain. For multiple outlets the sum of the relative amounts has to be equal to 1.0.

6.3 Wall

Call:	**boundary,** *number | name*	
	wall	
	noslip | slip | rough, z_0	
	u, v, w	
	adiabatic | fixed | flux	
	number, number	
Default(s):	(none)	
Example:	boundary, 3	
	wall	
	noslip	
	1.0, 0.0, 0.0	# 3 velocity components
	fixed	# option fixed temperature
	293.0,0.0	# temperature (in Kelvin), resistance R

Set the values for a wall. A wall can be frictionless ('slip'), or not ('noslip'), or even 'rough'. The latter allows also for a wall velocity in the plane of the wall ('moving wall'). Finally the thermal properties of the wall have to be specified (even for an isothermal simulation). Note that the thermal resistance R has to be specified.

6.4 Symmetry plane

Call:	**boundary,** *number | name*
	symp
Default(s):	(none)
Example:	boundary, 4
	symp

7 Example Dolfyn INput file

```
title, P 911

steps 2000 1.e-5
restart no

VisLam 1.81e-05
Density=1.2
Pref 858934
monitor, 1373

gamma,0.5
relax,0.6,0.25,0.6

turbulence,ke,0.1
init field,-30,,,0.0,0.000001,,293.

use vtk              # write vtk-file
post,u,vert,yes      # nodal velocities as well
#
# boundary conditions
#
boundary,0       # default
wall
noslip
0. 0.0 0.0
adiab

boundary,1       # inlet
inlet
-30.0 0.0 0.0    # u, v, w,
1.2              # density in
293.             # Tin
inle             # intensity i, length scale l
0.001 0.0004

boundary,2       # outlet
outlet
1.0

boundary,3       # wall 1
wall
noslip
0. 0.0 0.0
adiab

boundary,4       # wall 2
wall
noslip
0. 0.0 0.0
adiab
```

Index