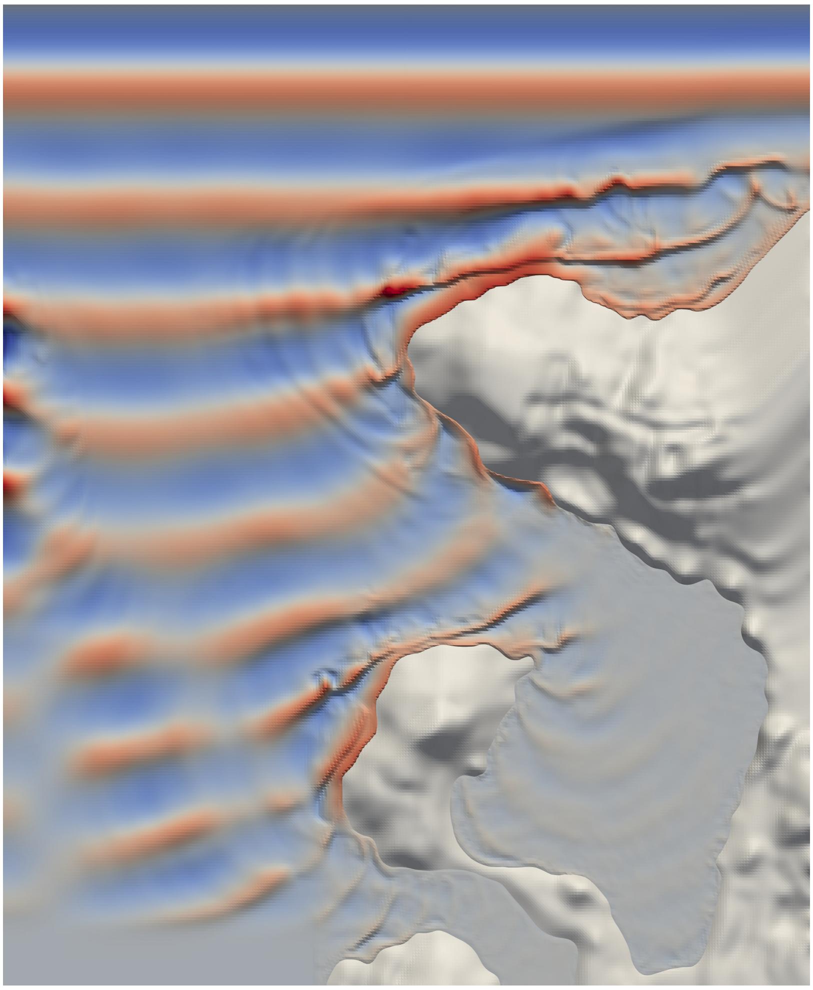
DIVEMesh :: User Guide



DIVEMesh 24.12

Contact

Hans Bihs

Professor
Marine Civil Engineering
NTNU Trondheim

hans.bihs@reef3d.com

Contents

Chapter 1

The ‘control.txt‘ file

1.1 B :: Boundary

B 1 double cell size dx

default: 0.0

B 2 int number of cells in x-, y- and z-direction

When activated, it will overwrite B 1.

default: 0 0 0

B 10 * double rectangular domain; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

B 101 int type of grid stretching function in x-direction

0 OFF

1 center focus (sinh, using B111)

2 wall focus (tanh, using B111)

5 point focus using sinh (requires B 114)

6 point focus using exponents (requires B 114)

8 three zones with constant dx (requires B 121)

9 three zones with stretching (requires B 124)

10 input from file (requires x-spacing.dat file with grid points between 0 and 1)

11 cell size based (requires B 127)

default: 0

B 102 int type of grid stretching function in y-direction

0 OFF
1 center focus (sinh, using B112)
2 wall focus (tanh, using B112)
5 point focus using sinh (requires B 115)
6 point focus using exponents (requires B 115)
8 three zones with constant dx (requires B 122)
9 three zones with stretching (requires B 125)
10 input from file (requires y-spacing.dat file with grid points between 0 and 1)
11 cell size based (requires B 128)
default: 0

B 103 int type of grid stretching function in z-direction

0 OFF
1 center focus (sinh, using B113)
2 wall focus (tanh, using B113)
3 lid focus (expontential, using B113)
4 lid focus (sinusoidal)
5 point focus using sinh (requires B 116)
6 point focus using exponents (requires B 116)
8 three zones with constant dx (requires B 123)
9 three zones with stretching (requires B 126)
10 input from file (requires z-spacing.dat file with grid points between 0 and 1)
11 cell size based (requires B 129)
default: 0

B 111 double stretching factor in x-direction

default: 1.0

B 112 double stretching factor in y-direction

default: 1.0

B 113 double stretching factor in z-direction

default: 1.0

B 114 double focus point for x-direction stretching B 101 5

default: 0.0

B 115 double focus point for y-direction stretching B 102 5

default: 0.0

B 116 double focus point for z-direction stretching B 103 5

default: 0.0

B 121 input for the three zones of B 101 8: int N1, double x1, int N2, double x2, int N3

The sum of the given N1,N2,N3 result in the number of elements in x-direction.

default: na

B 122 input for the three zones of B 101 8: int N1, double y1, int N2, double y2, int N3

The sum of the given N1,N2,N3 result in the number of elements in y-direction.

default: na

B 123 input for the three zones of B 101 8: int N1, double z1, int N2, double z2, int N3

The sum of the given N1,N2,N3 result in the number of elements in z-direction.

default: na

B 124 input for the three zones of B 101 9: int N1, double x1, double f1, int N2, double xf, double f2, int N3, double x2, double f3

The sum of the given N1,N2,N3 result in the number of elements in x-direction. The factors f1 and f3 are the linear stretching factors for the layers. The parameter x1 gives the border between the first and second layer, x2 between second and third. The parameter xf gives the location of the stretching focus in the second layer, where sinh stretching based on the factor f2 is used.

default: na

B 125 input for the three zones of B 102 9: int N1, double y1, double f1, int N2, double yf, double f2, int N3, double y2, double f3

see B 124.

default: na

B 126 input for the three zones of B 103 9: int N1, double z1, double f1, int N2, double zf, double f2, int N3, double z2, double f3

see B 124.

default: na

B 127 input for the cell size based stretching of B 101 11: double Δx_{min}, double Δx_{max}, double x_f, double δ_f, double r

A constant cell size of Δx_{min} is generated in the focus zone of length δ_f around x_f. The cells are stretched with the cell ratio r between the ends of the focus zone and the domain boundaries until the maximum cell size of Δx_{max} is reached.

default: na

B 128 input for the cell size based stretching of B 102 11: double Δy_{min}, double Δy_{max}, double y_f, double δ_f, double r

see B 127

default: na

B 129 input for the cell size based stretching of B 103 11: double Δz_{min}, double Δz_{max}, double z_f, double δ_f, double r

see B 127

default: na

B 130 int print grid to "gridSpacing.vtk"

0 OFF
1 ON
default: 0

1.2 C :: Channel

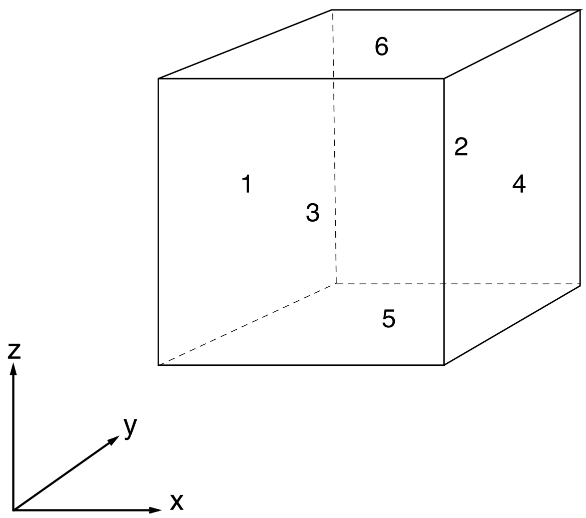


Figure 1.1: Definition of cell sides within DIVEMesh and REEF3D.

C 11 int Boundary Condition on Surfside 1

1 inflow

3 symmetry plane

6 wave generation

7 numerical beach

21 wall

default: 21

C 12 int Boundary Condition on Surfside 2

3 symmetry plane

6 wave generation

7 numerical beach

21 wall

default: 21

C 13 int Boundary Condition on Surfside 3

3 symmetry plane

6 wave generation

7 numerical beach

21 wall

default: 21

C 14 int Boundary Condition on Surfside 4

2 outflow

3 symmetry plane

6 wave generation

7 numerical beach

21 wall

default: 21

C 15 int Boundary Condition on Surfside 5

3 symmetry plane

21 wall

default: 21

C 16 int Boundary Condition on Surfside 6

3 symmetry plane

21 wall

default: 21

C 21 int Periodic boundary conditions in x-direction

0 OFF

1 ON

default: 0

C 22 int Periodic boundary conditions in y-direction

0 OFF

1 ON

default: 0

C 23 int Periodic boundary conditions in z-direction

0 OFF

1 ON

default: 0

1.3 D :: Data Interpolation

D 10 int turn data interpolation on/off

0 OFF

1 ON

default: 0

D 11 double Δx, Δy, Δx

default: 0.0 ; 0.0 ; 0.0

D 12 double factor x-coordinate, factor y-coordinate, factor z-coordinate,

default: 0.0 ; 0.0 ; 0.0

D 13 int Read data every i^{th} iteration

default: 1

D 14 int Type of interpolation

1 inverse distance

2 kriging

default: 1

D 15 int Number of smoothing iterations

default: 0

D 16 double Factor for data smoothing

default: 0.5

D 17 double Factor for data inverse distance

default: 16.0

D 18 double Range factor for kriging

default: 0.3

D 19 int Read dummy letter in front of 3D coordinates

0 OFF

1 ON

default: 0

D 23 int reverse sign of data variable in dmdata.dat

0 OFF

1 ON

default: 0

D 24 double distance limiter for inverse distance

default: na

1.4 G :: Geodat

G 9 int geodata for topo or solid

1 topo

2 solid

default: 1

G 10 int turn geodat on/off

0 OFF

1 ON

default: 0

G 11 double Δx, Δy, Δz

default: 0.0 ; 0.0 ; 0.0

G 12 double factor x-coordinate, factor y-coordinate, factor z-coordinate,

default: 0.0 ; 0.0 ; 0.0

G 13 double rotation angle of geo coordinates around vertical axis

default: 0.0

G 14 double x-coordinate and y-coordinate of origin for the rotation angle of geo coordinates around vertical axis

default: 0.0 ; 0.0

G 15 int interpolation scheme

Global inverse distance interpolation requires long computation times for large geodat sets. The recommended interpolation scheme for relatively homogeneously spaced geodat points is local inverse distance interpolation. Kriging is recommended for inhomogeneously spaced geodat points, e.g. geodat points from digitized map contour lines. Kriging becomes unstable when geodat points are spaced too close to each other, so for this option G 36 is turned on automatically with the default value of 1.0 (that value can be changed). Kriging becomes inefficient for geodat sets larger than 10,000 points, thus G 37 is recommended for kringing.

1 global inverse distance interpolation

2 local inverse distance interpolation

3 kriging

default: 2

G 17 int d_{ij} lower limit for local inverse distance interpolation

default:

G 19 int read a letter in front of the coordinates in the geo.dat file

0 OFF

1 ON

default: 0

G 20 int use automatic grid size

0 OFF

1 ON

default: 0

G 21 double Margins for automatic grid size $x_{start}, y_{start}, z_{start}$

default: 0.0 ; 0.0 ; 0.0

G 22 double Margins for automatic grid size $x_{end}, y_{end}, z_{end}$

default: 0.0 ; 0.0 ; 0.0

G 23 int reverse sign of vertical coordinate in geo.dat file

0 OFF

1 ON

default: 0

G 24 double raise topography above the level h by dz

default: 0.0 ; 0.0

G 25 double multiply topography above the level h by factor fz

default: 0.0 ; 0.0

G 26 double remove all geodat points below the bed level h

default: 0.0

G 27 double remove all geodat points above the bed level h

default: 0.0

G 31 int Number of smoothing iterations

default: 0

G 32 double Factor for topography data smoothing

default: 0.5

G 35 double Factor for data inverse distance

default: 16.0

G 36 double sampling geodat points: factor times horizontal average mesh size for duplicate geodat point identification

default: 1.0

G 37 int sampling geodat points: maximum number of geodat after random removal

default: 1e8

G 38 int skip horizontal cells for geodat interpolation algorithm

default: 1

G 39 int remove out-of-bounds geodata points

0 OFF

1 ON

default: 1

G 41 int print SWAN bottom file from interpolated geo points

0 OFF

1 ON

default: 0

G 51 int automatic holecheck and holefill for incomplete geodata sets

G 52 is used for elevation for areas without points.

0 OFF

1 ON

default: 0

G 52 double base topography value for local inverse distance interpolation

default: 0.0

G 53 double automatic search radius factor (times dx)

default: 0.0

1.5 H :: Hydrodynamic Coupling

H 10 int Turn on hydrodynamic coupling procedure

This option requires the presence of the corresponding state files and state file folders.

0 OFF

2 SFLOW to CFD

4 FNPF to CFD

5 NHFLOW to CFD

default: 0

H 21 double X-location in FNPF NWT which is origin in CFD NWT

default: 0.0

H 22 double Y-location in FNPF NWT which is origin in CFD NWT

default: 0.0

H 23 double Z-location in FNPF NWT which is origin in CFD NWT

default: 0.0

H 31 double start time coupling time series

default: -10.0^{19}

H 32 double end time coupling time series

default: 10.0^{19}

H 33 int start iteration coupling time series

default: -21^{8}

H 34 int end iteration coupling time series

default: 21^{8}

1.6 M :: MPI

M 10 int Number of processes

default: 1

M 11 int Partition in x-direction

0 OFF

1 ON

default: 1

M 12 int Partition in y-direction

0 OFF

1 ON

default: 1

M 13 int Partition in z-direction

0 OFF

1 ON

default: 0

M 20 int Decomposition method

1 Standard rectangular base domain

2 Improved rectangular base domain

default: 1

M 31 double Variance parameter for decomposition method 2

default: 0.1

1.7 S :: Solid

S 1 int read STL file and generate solid. Name the file as ‘solid.stl’

IMPORTANT: the STL file needs to be in ASCII format! Most CAD programs export to a binary STL file. It is possible to use e.g. Paraview for STL conversion from binary to ASCII.

0 OFF

1 ON

default: 0

S 2 int auto-generate mesh extend based on STL max/min coordinates

0 OFF

1 ON

default: 0

S 3 double margins for auto-generated mesh $xm_{start}, xm_{end}, ym_{start}, ym_{end}, zm_{start}, zm_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 4 double Scale STL geometry

default: 1.0

S 5 double 3D rotation; $x_{origin}, y_{origin}, z_{origin}, \phi, \theta, \psi$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 6 int print transformed STL model to ”REEF3D_Solid.stl”

0 OFF

1 ON

default: 0

S 7 double translation / change origin of STL model dx, dy, dz

default: 0.0 ; 0.0 ; 0.0

S 8 double Turn STL geometry in horizontal xy-plane in degree ^{o}

default: 0.0

S 9 int Invert inside/outside for STL geometry

1 regular

2 invert

default: 1

S 10 * double rectangular object; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 11 * double rectangular object array; $x_{origin}, y_{origin}, z_{origin}$, box length L, gap G, number of objects in each direction n_i, n_j, n_k

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 12 * double beam with rectangular profile with flexible orientation; $x_{start}, y_{start}, z_{start}, x_{end}, y_{end}, z_{end}$, b, h

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 15 int rotate STL geometry together with the geo.dat points

0 OFF

1 ON

default: 0

S 32 * double cylinder in y-direction; $x_{center}, z_{center}, radius$

default: 0.0 ; 0.0 ; 0.0

S 33 * double cylinder in z-direction; $x_{center}, y_{center}, radius$

default: 0.0 ; 0.0 ; 0.0

S 37 * double cylinder with flexible orientation and front face orthogonal to the cylinder axis $x_{start}, y_{start}, z_{start}, radius_{start}, x_{end}, y_{end}, z_{end}, radius_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 41 * double cone in x-direction; $y_{center}, z_{center}, x_{start}, x_{end}, radius_{start}, radius_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 42 * double cone in y-direction; $x_{center}, z_{center}, y_{start}, y_{end}, radius_{start}, radius_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 43 * double cone in z-direction; $x_{center}, y_{center}, z_{start}, z_{end}, radius_{start}, radius_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 51 * double sphere; $x_{center}, y_{center}, z_{center}, radius$

default: 0.0 ; 0.0 ; 0.0 ; 0.0

S 52 * double ellipsoid; $x_{center}, y_{center}, z_{center}, a_{axis}, b_{axis}, c_{axis}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 53 * double semi ellipsoid with vertical base; $x_{center}, y_{center}, z_{center}, a_{axis}, b_{axis}, c_{axis}, h_{base}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 54 * double semi ellipsoid with vertical base with rotation around the center; $x_{center}, y_{center}, z_{center}, a_{axis}, b_{axis}, c_{axis}, h_{base}, \phi, \theta, \psi$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 61 * double wedge object in x-direction; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 62 * double wedge object in y-direction; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 63 * double wedge object in z-direction; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 81 * double tetrahedon object, each of the 4 points is given by the coordinates $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4$

default: [4x] 0.0 ; 0.0 ; 0.0

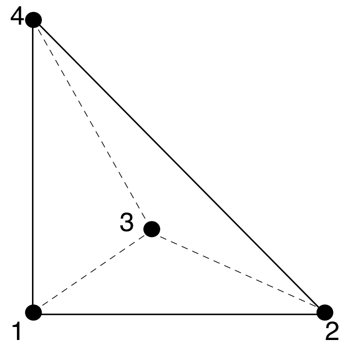


Figure 1.2: Definition of the tetrahedon points.

S 82 * double pyramid object, each of the 5 points is given by the coordinates $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4$
x_5, y_5, z_5

default: [5x] 0.0 ; 0.0 ; 0.0

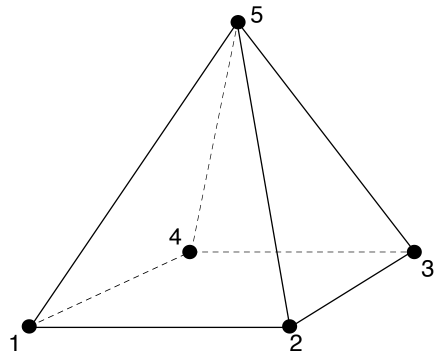


Figure 1.3: Definition of the pyramid points.

S 83 * double wedge object, each of the 6 points is given by the coordinates $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4,$
$x_5, y_5, z_5, x_6, y_6, z_6$

default: [6x] 0.0 ; 0.0 ; 0.0

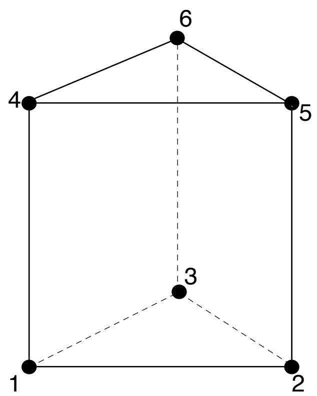


Figure 1.4: Definition of the wedge points.

S 84 * double hexahedron object, each of the 8 points is given by the coordinates $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4$
$x_5, y_5, z_5, x_6, y_6, z_6, x_7, y_7, z_7, x_8, y_8, z_8$

default: [8x] 0.0 ; 0.0 ; 0.0

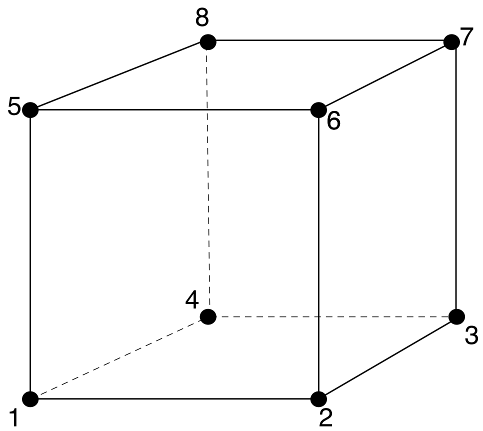


Figure 1.5: Definition of the hexahedron points.

S 121 * double vertical ogee weir, coordinates of upstream bottom corner, width, downstream height and hydraulic head x, y, z, b, P_d, H_0

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 122 * double vertical ogee weir, K, n, x_c and y_c

default: 0.5 ; 1.85 ; 0.22 ; 0.075

S 123 * double vertical ogee weir, R_1 and R_2

default: 0.45 ; 0.2

S 131 * double semi-circular cylinder in y-direction, x_c, z_c, y_{start}, y_{end}, $radius_{start}$, $radius_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 141 * double arch bridge; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, radius$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

S 301 int fluvial box geometry.

The inverted fluvial box geometry only works for special cases, where the inflow is on side 1 and the outflow is on side 4.

1 regular

2 inverted

default: 1

S 305 double fluvial box resolution, factor time dx results in ds (smaller values gives higher resolution); $factor$

default: 0.5

S 306 double fluvial box width; b

default: 1.0

S 307 double fluvial box flow heigth; fh, bed height bh

default: 0.5 ; 0.2

S 308 double fluvial box move origin; Δx, Δy, Δz,

default: 0.0 ; 0.0 ; 0.0

S 309 double fluvial box margins; Δx, Δy, Δz,

default: 0.0 ; 0.0 ; 0.0

S 310 * double fluvial box: straight segment; $length$

default: 0.0

S 320 * double fluvial box: left bend; $radius, angle\,\phi$

default: 0.0

S 330 * double fluvial box: right bend; $radius, angle\,\phi$

default: 0.0

S 340 * double fluvial box: meander (sine-generated curve); θ, L, N, ds

default: 0.0 ; 0.0 ; 0.0 ; 0.0

1.8 T :: Topo

T 10 * double rectangular object; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 11 * double rectangular object array; $x_{origin}, y_{origin}, z_{origin}$, box length L, gap G, number of objects in each direction n_i, n_j, n_k

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 32 * double cylinder in y-direction; $x_{center}, z_{center}, radius$

default: 0.0 ; 0.0 ; 0.0

T 33 * double cylinder in z-direction; $x_{center}, y_{center}, radius$

default: 0.0 ; 0.0 ; 0.0

T 37 * double cylinder with flexible orientation and front face orthogonal to the cylinder axis $x_{start}, y_{start}, z_{start}, radius_{start}, x_{end}, y_{end}, z_{end}, radius_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 41 * double cone in x-direction; $y_{center}, z_{center}, x_{start}, x_{end}, radius_{start}, radius_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 42 * double cone in y-direction; $x_{center}, z_{center}, y_{start}, y_{end}, radius_{start}, radius_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 43 * double cone in z-direction; $x_{center}, y_{center}, z_{start}, z_{end}, radius_{start}, radius_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 51 * double sphere; $x_{center}, y_{center}, z_{center}, radius$

default: 0.0 ; 0.0 ; 0.0 ; 0.0

T 52 * double ellipsoid; $x_{center}, y_{center}, z_{center}, a_{axis}, b_{axis}, c_{axis}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 53 * double semi ellipsoid with vertical base; $x_{center}, y_{center}, z_{center}, a_{axis}, b_{axis}, c_{axis}, h_{base}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 54 * double semi ellipsoid with vertical base with rotation around the center; $x_{center}, y_{center}, z_{center}, a_{axis}, b_{axis}, c_{axis}, h_{base}, \phi, \theta, \psi$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 61 * double wedge object in x-direction; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 62 * double wedge object in y-direction; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 63 * double wedge object in z-direction; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 81 * double tetrahedon object, each of the 4 points is given by the coordinates $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4$

default: [4x] 0.0 ; 0.0 ; 0.0

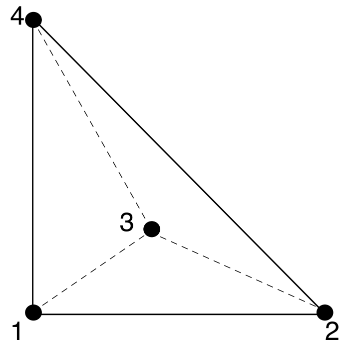


Figure 1.6: Definition of the tetrahedon points.

T 82 * double pyramid object, each of the 5 points is given by the coordinates $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4,$
x_5, y_5, z_5

default: [5x] 0.0 ; 0.0 ; 0.0

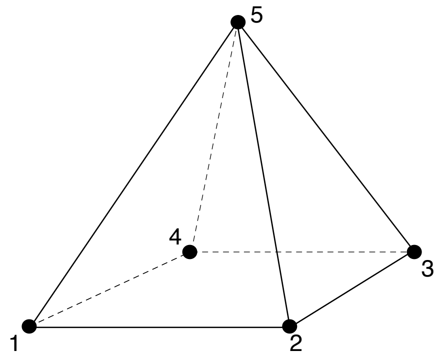


Figure 1.7: Definition of the pyramid points.

T 83 * double wedge object, each of the 6 points is given by the coordinates $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4,$
$x_5, y_5, z_5, x_6, y_6, z_6$

default: [6x] 0.0 ; 0.0 ; 0.0

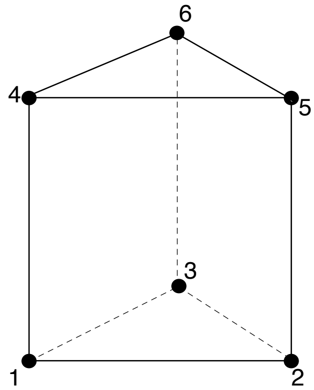


Figure 1.8: Definition of the wedge points.

T 84 * double hexahedron object, each of the 8 points is given by the coordinates $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y$
$x_5, y_5, z_5, x_6, y_6, z_6, x_7, y_7, z_7, x_8, y_8, z_8$

default: [8x] 0.0 ; 0.0 ; 0.0

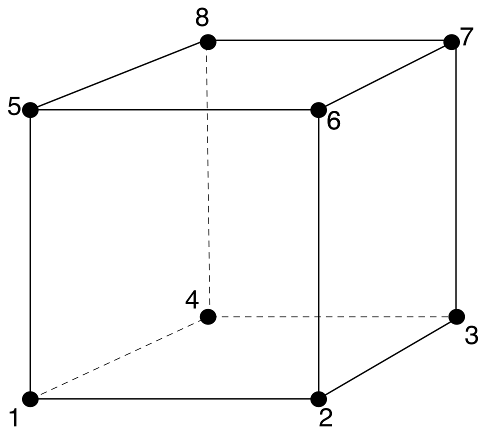


Figure 1.9: Definition of the hexahedron points.

T 121 * double vertical ogee weir, coordinates of upstream bottom corner, width, downstream height and hydraulic head x, y, z, b, P_d, H_0

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 122 * double vertical ogee weir, K, n, x_c and y_c

default: 0.5 ; 1.85 ; 0.22 ; 0.075

T 123 * double vertical ogee weir, R_1 and R_2

default: 0.45 ; 0.2

T 131 * double semi-circular cylinder in y-direction, x_c, z_c, y_{start}, y_{end}, $radius_{start}$, $radius_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 141 * double arch bridge; $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, radius$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

T 301 int fluvial box geometry.

The inverted fluvial box geometry only works for special cases, where the inflow is on side 1 and the outflow is on side 4.

1 regular

2 inverted

default: 1

T 305 double fluvial box resolution, factor time dx results in ds (smaller values gives higher resolution); $factor$

default: 0.5

T 306 double fluvial box width; b

default: 1.0

T 307 double fluvial box flow heigth; fh, bed height bh

default: 0.5 ; 0.2

T 308 double fluvial box move origin; Δx, Δy, Δz,

default: 0.0 ; 0.0 ; 0.0

T 309 double fluvial box margins; Δx, Δy, Δz,

default: 0.0 ; 0.0 ; 0.0

T 310 * double fluvial box: straight segment; $length$

default: 0.0

T 320 * double fluvial box: left bend; $radius, angle\,\phi$

default: 0.0

T 330 * double fluvial box: right bend; $radius, angle\,\phi$

default: 0.0

T 340 * double fluvial box: meander (sine-generated curve); θ, L, N, ds

default: 0.0 ; 0.0 ; 0.0 ; 0.0