Memo

To
Arthur van Dam, Bert Jagers, Didrik Meijer, Jan Noort, Jaco Stout

Date
generated: 2019-05-05

Reference
003

Number of pages
xi

From
Jan Mooiman,
Arthur van Dam,
Bert Jagers

Direct line
+31 (0)88 335 8568

E-mail
jan.mooiman@deltares.nl,
arthur.vandam@deltares.nl,
bert.jagers@deltares.nl

Subject
D-Flow 1D NetCDF-format; CF-1.7 UGRID-0.0 Deltares-0.0

Copy to
—

Version control information

Location : `https://repos.deltares.nl/repos/ds/trunk/doc/user_manuals/white_papers/ugrid/dflow1d/ugrid_1d.tex`
Revision : 59828

Contents

1 Agreements

The following agreements are made for the attributes of a variable

Units

The units are presented according the Climate and Format convention, see http://cfconventions.org/Data/cf-standard-names/41/build/cf-standard-name-table.html, column "canonical units". Because the unit of Chézy does not fulfill the SI unit convention, the power of the length is not an integer ($C = [m^{\frac{1}{2}}/s] \equiv [m^{1/2}/s] \equiv [\sqrt{m}/s]$), we choose to represent the unit of Chézy by [m(0.5) s-1], [m1/2 s-1] or [m(1/2) s-1].

long_name

Long names should start with a capital character, because these names are used by labeling plots. See http://cfconventions.org/Data/cf-conventions/cf-conventions-1.7/cf-conventions.html#long-name

2 Requirements 1D network and mesh

1 Nodes nummers zijn niet noodzakelijk oplopend genummerd en kunnen een willekeurige ligging hebben in het netwerk, see Figure 5.
2 Edges zijn niet noodzakleijk oplopend genummerd en kunnen een willekeurige ligging hebben in het netwerk, see Figure 5.
3 Uitwisselingen tussen branches moet ook opgeslagen kunnen worden, en moeten derhalve opgenomen worden in de definitie.
4 Voor een splitsingspunt moeten voor elke branch de ingangsdebieten, -snelheden en bodemliggingen opgeslagen kunnen worden.

3 Keyword definition for 1D networks

Table 1: 1D network topology

Required topology attributes	**Value**	**Example**
cf_role	mesh_topology	
topology_dimension	1	
edge_dimension		nNetworkBranches
edge_geometry		network1D_geometry
edge_node_connectivity		network1D_edge_nodes
node_coordinates		network1D_nodes_x␣network1D_nodes_y
node_dimension		nNetworkNodes
Optional attributes		

long_name		Network topology

Table 2: 1D network geometry, referenced by the value of the attribute **edge_geometry** *attribute as defined inTable 1*

Required geometry attributes	**Value**	**Example**
geometry_type	line	
geometry_length		network1D_geom_length
node_coordinates		network1D_geom_x␣network1D_geom_y
node_count		nGeometryNodes
part_node_count		network1D_part_node_count
Optional attributes		
long_name		Network Geometry
node_name		Node name
edge_name		Branch name

Table 3: 1D computational mesh

Required mesh attributes	**Value**	**Example**
cf_role	mesh_topology	
topology_dimension	1	
coordinate_space	*defined by Table 1*	
branch_length		mesh1D_branch_length
edge_dimension		nMesh1DEdges
edge_node_connectivity		mesh1D_edge_nodes
node_coordinates		mesh1D_nodes_branch_id␣mesh1D_nodes_branch_
node_dimension		nMesh1DNodes
Optional attributes		
long_name		Mesh 1D
node_name		Mesh 1D node name
edge_name		Mesh 1D branch name

Vragen:

1 Is het attribute **edge_geometry** de enige entry naar de geometry? Kan een netwerk zonder geometry bestaan?
2 Twee lengten van een branch opgegeven, voor de geometry en de mesh. Willen we dit of zijn deze per definitie gelijk? Indien er geen geometry is opgegeven dan moet de branch lengte bij het mesh worden opgenomen.
3 Naamgevingen, branch namen in de geometrie kunnen verschillen van de naamgevingen in het rooster.
4 Lokatienamen moet ook volgens de chainage worden gedefinieerd. Bijv. het Oranjekanaal dat uitkomt in de Nieuwe Waterweg ligt op 12 km vanaf het splitsing bij Het Scheur. En niet

gerelateerd aan water, maar wel interessant voor de scheepvaart, bijv. Dirkzwager ligt op 6 km van Het Scheur.

4 Network1D (map) nc-file

Example of map-file, including bed levels. The bed levels have the extra requirement that at a connection node each branch has its own bed level, so multi valued at the connection node.

```
...

        double s1_1d(time, nMesh1DNodes) ;
                s1_1d:location = "node" ;
                s1_1d:long_name = "Water level" ;
                s1_1d:mesh = "mesh1D" ;  //  this is UGRID location pointing to mesh1D topology
                s1_1d:standard_name = "sea_surface_height_above_geoid" ;
                s1_1d:units = "m" ;
        double u_1d(time, nMesh1DEdges) ;
                u_1d:location = "edge" ;
                u_1d:long_name = "velocity along branch" ;
                u_1d:mesh = "mesh1D" ;
                u_1d:standard_name = "sea_water_speed" ;
                u_1d:units = "m s-1" ;
        double bl_1d(time, nMesh1DNodesOnBranches) ;
                bl_1d:location = "nodes_on_branch" ;
                bl_1d:long_name = "Bed level on branch" ;
                bl_1d:mesh = "mesh1D" ;
                bl_1d:standard_name = "bedlevel" ;
                bl_1d:units = "m" ;

...
```

nodes_on_branch All nodes defined on a branch, including begin and end point.
Retrieve this data from `mesh1D_edge_branch`. Retrieve the edges forming branch i from `mesh1D_edge_branch` and count these edges, say there are n edges, then the number of nodes on branch i is $n+1$.
Remarks/Questions:

1. Sorting of node numbers along chainage of the branch is needed!
2. How to administrate the bed levels at the connection nodes. For a junction node α, there are three different bed levels and just one water level. Do have the bed levels (bl) the subscript of the edge numbers? So we get $bl_{\alpha,I}$, $bl_{\alpha,II}$, $bl_{\alpha,III}$ and ζ_α. For water quality computations of concentration j (c_j) it could be: $c_{j,\alpha,I}$, $c_{j,\alpha,II}$, $c_{j,\alpha,III}$ and ζ_α.
3. Probably also administrate the water levels, $\zeta_{\alpha,I}$, $\zeta_{\alpha,II}$, $\zeta_{\alpha,III}$. Remark these water levels are all three the same, so redundancy of the water level values which is not desirable.

5 D-Flow 1D history nc-file

To be added station locations, see for an example the D-Flow FM history file.

```
...

double s1(time=ntimes, stations=nstations);
    coordinates = "station_x station_y"
    long_name = "Water level"
    standard_name = "sea_surface_height_above_geoid"
    units = "m"
double u(time=ntimes, stations=nstations);
    coordinates = "station_xu station_yu"
    long_name = "Velocity along branch"
    standard_name = "sea_water_speed"
    units = "m s-1"

...
```

Note: The coordinates of the station does not necessarily have the same coordinates as the mesh or network.

6 Example based on Figure 1

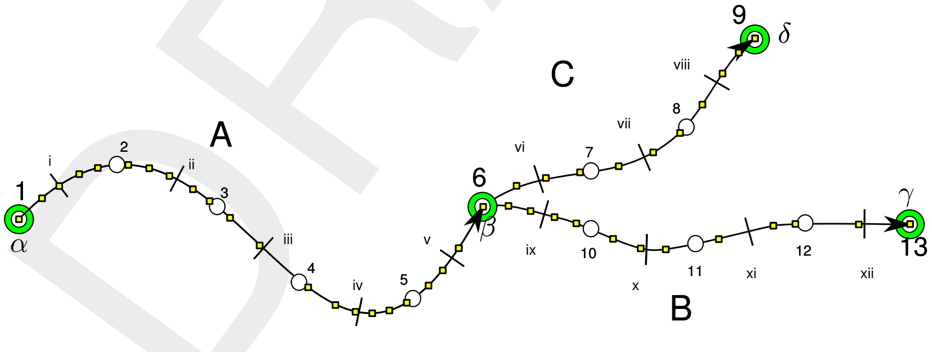

Figure 1: Simple computational 1D network;
3 branches (A, B, C),
4 network nodes (green circles, α, β, γ, δ);
13 nodes (open circles + green circles, ζ-points, latin numbers),
12 edges (u-points; roman numbers), 44 network nodes (yellow squares)

Dimensions

nNetworkBranches = 3; A, B, C
nNetworkNodes = 4; α, β, γ, δ
nBranchPointsTotal = 46; yellow squares per branch (including both end points)
nGeomPoints = 44; yellow squares (total number of distinct points)

nMeshNodes = 13 (13=6+5+4-(3-1); 3 branches are connected to node 6)
nMeshEdges = 12 (12=5+4+3); edges are between ζ-points

6.1 Network nodes

network1D_nodes_x(nNetworkNodes)
`-187.96667, 2195.7333, 4071.4928, 3445.4246`

network1D_nodes_y(nNetworkNodes)
`720.81667, 708.71667, 690.94861, 1540.1838`

network1D_edge_nodes(nNetworkBranches,Two)

I: 1 2
II: 2 3
III: 2 4

6.2 Geometry

network1D_part_node_count(nNetworkBranches)
`22, 13, 11`

So the branches contain the following geometry points:

```
Branch A: 1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
          18, 19, 20, 21, 22, 23, 24, 2;
Branch B: 2, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 3;
Branch C: 2, 36, 37, 38, 39, 40, 41, 42, 43, 44, 4.
```

Note that the node number 2 appear three times in this array.

network1D_geom(nNetworkBranches)
The three geometry lengths (the offset of the end point of a branch)

`length_A, ..., length_C`

The offset is in general not equal to the euclidical length between the nodes on the branch

network1D_geom_x(nNetworkNodes)
`x_1, ..., x_46`

network1D_geom_y(nNetworkNodes)
`y_1, ..., y_46`

7 Numerical discretization

mesh1D_node_branch_id(nMeshNodes)
`A, A, A, A, A, A, C, C, C, B, B, B, B`

mesh1D_node_branch_offset(nMeshNodes)
`0, 500, 1000, 1500, 2000, 2500` (on branche A, 6 nodes)

`700, 1400, 2100` (on branche B, 4 nodes; without node 6)
`400, 800, 1200, 1600` (on branche C, 3 nodes; without node 6)

mesh1D_edge_nodes(nMeshEdges,Two)

I:	1	2
II:	2	3
III:	3	4
IV:	4	5
V:	5	6
VI:	6	7
VII:	7	8
VIII:	8	9
IX:	6	10
X:	10	11
XI:	11	12
XII:	12	13

8 Example based on Figure 2

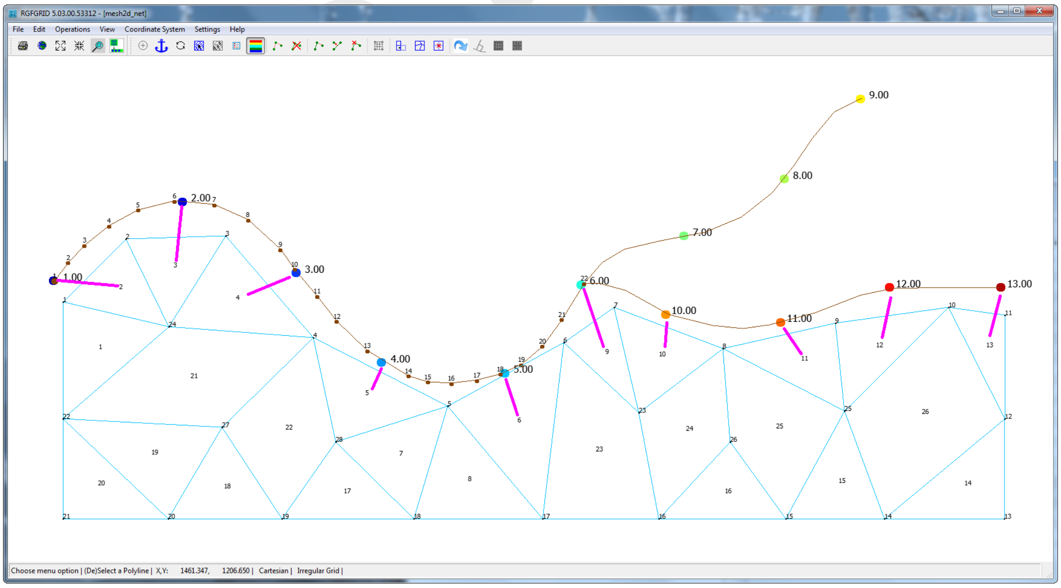

Figure 2: Composite mesh (1D and 2D),
1D Mesh*: coloured dots (=computational 1D mesh), ζ-points; numbered dots, 1d network geometry for branch A (Figure 1).*
2D Mesh*: light blue with node and face numbers.*
1D2D connections are pink coloured.

1D2D-links

1D2D-links are defined between de computational nodes of the 1D Mesh (ζ-points) and the faces of the 2D Mesh (ζ-points). The link table for Figure 2 read:

Table 4: 1D2D Link table based on Figure 2

1d node	2d face
1	2
2	3
3	4
4	5
5	6
6	9
10	10
11	11
12	12
13	13

9 Offset determination along branch, mesh as given in Figure 5

Gebruik van offset in een branch incl. begin- en eindpunt

Per branch moet de nodes worden opgegeven, zoals in onderstaand voorbeeld

```
Nodes at branch:
# A: 1, 2, 17, 4, 5, 6
# B: 1, 12, 13, 19, 20, 21, 8, 7
# C: 7, 22, 3, 10, 11
# D: 1, 18, 19, 14, 15, 16, 9, 7
mesh1d_geom_offset[:] =
   [0, 500, 1000, 1500, 2100, 2750,
    0, 700, 1400, 2100, 2800, 3150, 3500,
    0, 700, 1400, 2200, 3000,
    0, 350, 700, 1050, 1400, 1750, 2100, 2500]
```

Het is niet mogelijk om de offset per node op te geven, want bijv. node 7 komt in meerdere branches voor:

Node 7 is de laatste node van branch B en heeft een offset van 3500,
Node 7 is de eerste node van branch C en heeft een offset van 0,
Node 7 is de laatste node van branch D en heeft een offset van 2500.

Gebruik van Fill_Value op connection node

De branch offset van connection nodes wordt op Fill_Value gezet, via de edge-node en edge-branch relatie is bekent welke offset nodig is voor het rekenhart. Dus als de offset een Fill_Value is dan moet het een connection node zijn.

Extra exchanges at a connection node

Per connection node extra verbindingen toevoegen die ook een fillvalue hebben, daarop kan later de uitwisseling geschreven worden tussen bijvoorbeeld delwaq-cellen.

3 extra verbindingen bij splitsingspunt
6 extra verbindingen bij kruising van branches
...

De algemene formule om het aantal verbindingen te bepalen per connection node luidt:

$$x_i = \frac{1}{2}n(n-1) \tag{1}$$

met

x_i	extra aantal extra verbindingen op de connection node i
n	aantal edges dat uitkomt op een connection node

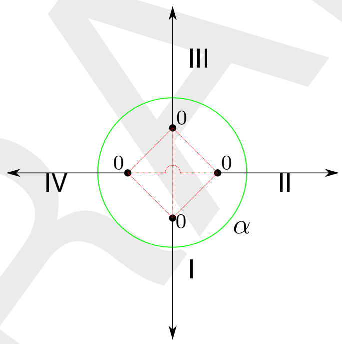

Figure 3: Example of connection node α between 4 edges (I, II, III, IV), containing 6 $(= \frac{1}{2}4(4-1))$ intermediate connections, 0 indicate the chainage on each branch.

The table belonging to Figure 3 read:

Table 5: Connections between edges over a connection node

Node	From edge	To edge
α	I	II
α	I	III
α	I	IV
α	II	III
α	II	IV
α	III	IV

In het rekengrid van Figure 5 zijn er 26 verbindingen. Elke node is een verbinding tussen twee edges, dat zijn er 22 en nog 4 voor de nodes 1 en 7.

10 Figures with example numbering of nodes and edges

For each of the figures in this section a NetCDF-file is available.

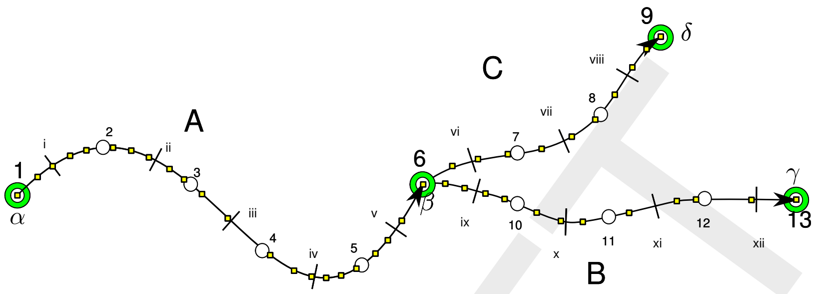

Figure 4: Simple 1D network and mesh, all branches do have intermediate ζ-points; 3 branches (A, B, C), 4 network nodes (green circles, α, β, γ, δ), 13 ζ-points.

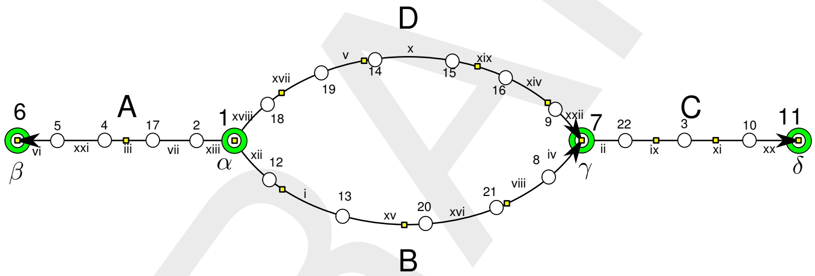

Figure 5: Simple 1D network and mesh, all branches do have intermediate ζ-points; 4 branches (A, B, C, D), 4 network nodes (green circles, α, β, γ, δ), 22 ζ-points (latin numbers), 22 edges (roman numbers).

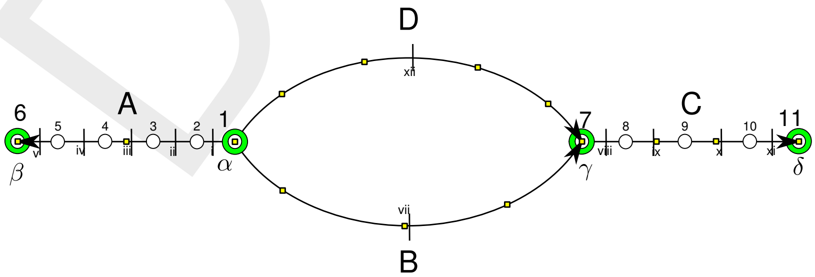

Figure 6: Simple 1D network and mesh, 2 branches does not have intermediate ζ-points (so only end points); 4 branches (A, B, C, D), 4 network nodes (green circles, α, β, γ, δ), 11 ζ-points.

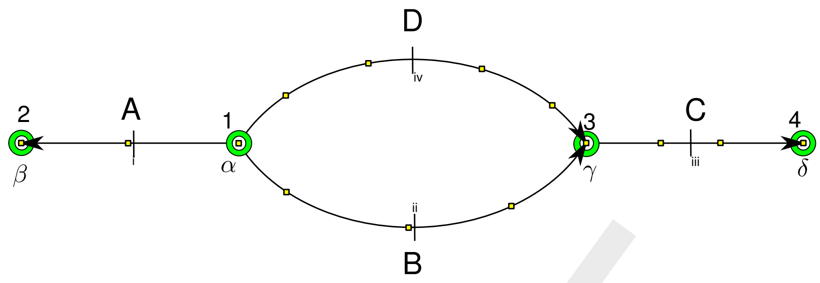

Figure 7: Simple 1D network and mesh, all branches do not have intermediate ζ-points (so only end points); 4 branches (A, B, C, D), 4 network nodes (green circles, α, β, γ, δ), 4 ζ-points.

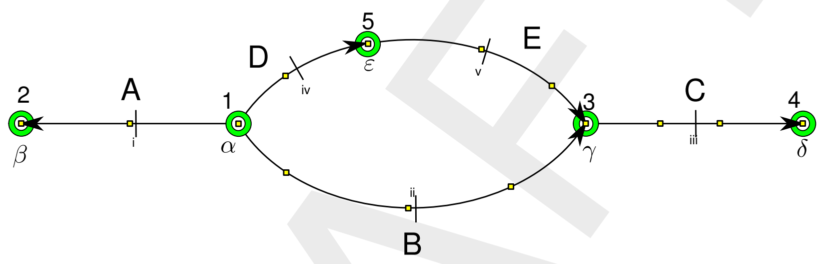

Figure 8: Simple 1D network and mesh, all branches do not have intermediate ζ-points (so only end points) and connection point ε does only have two branches connect. 5 branches (A, B, C, D, E), 5 network nodes (green circles, α, β, γ, δ, ε), 5 ζ-points.