

Unstructured grids in D-Flow FM

D-Flow FM (formerly Unstruc) is a shallow water flow solver on flexible meshes (also known as 'unstructured grids'). The output data of D-Flow are stored in NetCDF files. Since a convention for storing unstructured grid data in NetCDF files didn't exist, we have started to define an extension to the CF standard. Meanwhile this effort has been linked to other international initiatives, and the latest proposal for the conventions can be found on [this unstructured grid page](#). Although we lead and follow these developments, we also want to keep the output files relatively stable, hence the D-Flow FM output files don't always follow the proposed conventions exactly. This page describes the format of the NetCDF files that are currently produced by D-Flow FM.

Unstructured grid as a net

- The unstructured grid is a set of net nodes (coordinates) and a set of net links connecting them.
- Flow data is defined on net cells (naming might change). Cells have a (circum-)center (water level point). The entire cell is defined through its contour line (`bounds: NetcellContour_x/y`). The contour line contains the coordinates of surrounding net nodes for 2D cells and generated coordinates for 1D channels/manholes/etc.

Examples

- [D-Flow FM net file with 2D cells on lon/lat-grid.](#)
- [D-Flow FM map file with 2D cells on lon/lat-grid.](#)
- [D-Flow FM map file with staggered solution variables \(concept\)](#)
- [D-Flow FM map file with z-layers \(concept\)](#)
- [D-Flow FM map file with sigma coords \(concept\)](#)
- [D-Flow FM map file with subdomains for variable nr of layers \(flexible layers, concept\)](#)

D-Flow FM net file with 2D cells on lon/lat-grid.

New version (Mar '10) with additional NetLinkType 0 (closed 2D link/thin dams), and netcell (=NetElem) data + boundary data (intended for use in RGFRID)

```

netcdf courantc_net {
dimensions:
    nNetNode = 178907 ;
    nNetLink = 363286 ;
    nNetLinkPts = 2 ;
    nBndLink = 14337 ;
    nNetElem = 184189 ;
    nNetElemMaxNode = 7 ;
variables:
    double NetNode_x(nNetNode) ;
        NetNode_x:units = "degrees_east" ;
        NetNode_x:standard_name = "longitude" ;
        NetNode_x:long_name = "netnodal lon-coordinate" ;
    double NetNode_y(nNetNode) ;
        NetNode_y:units = "degrees_north" ;
        NetNode_y:standard_name = "latitude" ;
        NetNode_y:long_name = "netnodal lat-coordinate" ;
    double NetNode_z(nNetNode) ;
    int NetLink(nNetLink, nNetLinkPts) ;
        NetLink:standard_name = "netlink" ;
        NetLink:long_name = "link between two netnodes" ;
    int NetLinkType(nNetLink) ;
        NetLinkType:long_name = "type of netlink" ;
        NetLinkType:valid_range = 0, 2 ;
        NetLinkType:flag_values = 0, 1, 2 ;
        NetLinkType:flag_meanings = "closed_link_between_2D_nodes link_between_1D_nodes
link_between_2D_nodes" ;
    int NetElemNode(nNetElem, nNetElemMaxNode) ;
        NetElemNode:long_name = "Mapping from net cell to net nodes." ;
    int BndLink(nBndLink) ;
        BndLink:long_name = "Netlinks that compose the net boundary." ;
    int NetElemLink(nNetElemLink, nNetElemLinkPts) ;
        NetElemLink:long_name = "link between two netelems" ;

// global attributes:
    :institution = "Deltares" ;
    :references = "http://www.deltares.nl" ;
    :source = "UNSTRUC v1.0.11.10693:1069, model" ;
    :history = "Created on 2010-03-05T09:41:31+0100, UNSTRUC" ;
    :Conventions = "CF-1.4:Deltares-0.1" ;

data:

    NetNode_x = 10.7736671537707, 10.775, 10.7783013103449, 10.775, 10.775,
        10.80625, 10.7839111680939, 10.80625, 10.8375, 10.80625, 10.80625,
// ...
    NetLink =
        23381, 23382,
        32338, 32341,
// ...
    NetLinkType = 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2,
// ...
    NetElemNode =
        1, 2, 5, _, _, _, _ ,
        1, 5, 4, _, _, _, _ ,
// ...
    BndLink = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
        20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
// ...

```

D-Flow FM map file with 2D cells on lon/lat-grid.

```

netcdf test_map {
dimensions:
    nNetNode = 178907 ;
    nNetLink = 363472 ;

```

```

nNetLinkPts = 2 ;
nNetCell = 184324 ;
nNetCellMaxNode = 7 ;
nNetCellContourPts = 99 ;
nNetCellLink = 349141 ;
nNetCellLinkPts = 2 ;
time = UNLIMITED ; // (1 currently)
variables:
double NetNode_x(nNetNode) ;
    NetNode_x:units = "degrees_east" ;
    NetNode_x:standard_name = "longitude" ;
    NetNode_x:long_name = "netnodal lon-coordinate" ;
double NetNode_y(nNetNode) ;
    NetNode_y:units = "degrees_north" ;
    NetNode_y:standard_name = "latitude" ;
    NetNode_y:long_name = "netnodal lat-coordinate" ;
double NetNode_z(nNetNode) ;
int NetLink(nNetLink, nNetLinkPts) ;
    NetLink:standard_name = "netlink" ;
    NetLink:long_name = "link between two netnodes" ;
int NetLinkType(nNetLink) ;
    NetLinkType:standard_name = "netlink_type" ;
    NetLinkType:long_name = "type of netlink" ;
    NetLinkType:valid_range = 1, 2 ;
    NetLinkType:flag_values = 1, 2 ;
    NetLinkType:flag_meanings = "link_between_1D_nodes link_between_2D_nodes" ;
double time(time) ;
    time:units = "seconds since 2010-01-01 00:00:00" ;
double NetCell_xc(nNetCell) ;
    NetCell_xc:units = "degrees_east" ;
    NetCell_xc:standard_name = "longitude" ;
    NetCell_xc:long_name = "Net cell circumcenter x" ;
    NetCell_xc:bounds = "NetCellContour_x" ;
double NetCell_yc(nNetCell) ;
    NetCell_yc:units = "degrees_north" ;
    NetCell_yc:standard_name = "latitude" ;
    NetCell_yc:long_name = "Net cell circumcenter y" ;
    NetCell_yc:bounds = "NetCellContour_y" ;
double NetCellContour_x(nNetCell, nNetCellContourPts) ;
    NetCellContour_x:units = "degrees_east" ;
    NetCellContour_x:standard_name = "longitude" ;
    NetCellContour_x:long_name = "List of x-points forming netcell" ;
double NetCellContour_y(nNetCell, nNetCellContourPts) ;
    NetCellContour_y:units = "degrees_north" ;
    NetCellContour_y:standard_name = "latitude" ;
    NetCellContour_y:long_name = "List of y-points forming netcell" ;
int NetCellNode(nNetCell, nNetCellMaxNode) ;
    NetCellNode:standard_name = "netcell_node" ;
    NetCellNode:long_name = "Mapping from net cell to net nodes (first column is node count, will
change)." ;
    NetCellNode:netcell_coords = "NetCell_xc NetCell_yc" ;
int NetCellLink(nNetCellLink, nNetCellLinkPts) ;
double s1(time, nNetCell) ;
    s1:cell_methods = "area: mean" ;
    s1:coordinates = "NetCell_xc Netcell_yc" ;
    s1:grid_mapping = "crs" ;
double ucx(time, nNetCell) ;
    ucx:cell_methods = "area: point" ;
    ucx:coordinates = "NetCell_xc Netcell_yc" ;
    ucx:grid_mapping = "crs" ;
double ucy(time, nNetCell) ;
    ucy:cell_methods = "area: point" ;
    ucy:coordinates = "NetCell_xc Netcell_yc" ;
    ucy:grid_mapping = "crs" ;
int crs ;
    crs:grid_mapping_name = "latitude_longitude" ;
    crs:longitude_of_prime_meridian = 0.f ;
    crs:semi_major_axis = 6378137.f ;
    crs:inverse_flattening = 298.2572f ;

// global attributes:

```

```

        :institution = "Deltares" ;
        :references = "http://www.deltares.nl" ;
        :source = "UNSTRUC v1.0.11.10021:1007, model csmcourant" ;
        :history = "Created on 2010-01-06T15:35:40+0100, UNSTRUC" ;
        :Conventions = "CF-1.4/Deltares-0.1" ;

data:

NetNode_x = 10.7736671537707, 10.775, 10.7783013103449, 10.775, 10.775,
            10.80625, 10.7839111680939, 10.80625, 10.8375, 10.80625, 10.80625,
// ...
NetLink =
    80895, 80882,
    167568, 167573,
    108907, 109422,
// ...
NetLinkType = 1, 1, 1, 1, 1, 1, 1, 1,
// ...
time = 0.0300439515169901 ;
NetCell_xc = 10.7743335768854, 10.7743335913705, 10.7906249344082,

NetCellContour_x =
    10.7736671537707, 10.775, 10.775, _, _, _, _, _, // ...
    10.775, 10.7783013103449, 10.80625, 10.80625, 10.775, _, _, _, _, _, _
// ...
NetCellNode =
    3, 1, 2, 5, _, _, _
    3, 1, 5, 4, _, _, _
    5, 2, 3, 6, 8, 5, _
// ...
NetCellLink =
    1, 2,
    1, 3,
// ...
s1 =
    4.78405906502741e-001, 5.63697801327623e-001, 2.44636456454523e-001,
    7.54911502823542e-001, 2.32821367302938e-001, 1.96039340036838e-001,
    5.73297867243609e-001, 2.28008030376163e-001, 3.37509917682617e-001,
// ...
crs = 4326 ;

```

D-Flow FM map file with staggered solution variables (concept)

TODO: replace normal velocity by u/v at edge?

```

netcdf test_map {
dimensions:
    nNetNode = 706 ;
    nNetLink = 1313 ;
    nNetLinkPts = 2 ;
    nBndLink = 104 ;
    nNetElem = 610 ;
    nNetElemMaxNode = 7 ;
    nFlowElem = 665 ;
    nFlowElemMaxNode = 6 ;
    nFlowElemContourPts = 99 ;
    nNetElemLinkID = 65 ;
    nNetElemLink = 1078 ;
    nNetElemLinkPts = 2 ;
    nFlowLink = 1143 ;
    nFlowLinkPts = 2 ;
    time = UNLIMITED ; // (1 currently)
variables:
    double NetNode_x(nNetNode) ;
        NetNode_x:units = "m" ;
        NetNode_x:standard_name = "projection_x_coordinate" ;
        NetNode_x:long_name = "netnodal x-coordinate" ;
    double NetNode_y(nNetNode) ;

```

```

        NetNode_y:units = "m" ;
        NetNode_y:standard_name = "projection_y_coordinate" ;
        NetNode_y:long_name = "netnodal y-coordinate" ;
double NetNode_z(nNetNode) ;
int NetLink(nNetLink, nNetLinkPts) ;
        NetLink:standard_name = "netlink" ;
        NetLink:long_name = "link between two netnodes" ;
int NetLinkType(nNetLink) ;
        NetLinkType:long_name = "type of netlink" ;
        NetLinkType:valid_range = 0, 2 ;
        NetLinkType:flag_values = 0, 1, 2 ;
        NetLinkType:flag_meanings = "closed_link_between_2D_nodes link_between_1D_nodes
link_between_2D_nodes" ;
int NetElemNode(nNetElem, nNetElemMaxNode) ;
        NetElemNode:long_name = "Mapping from net cell to net nodes." ;
int BndLink(nBndLink) ;
        BndLink:long_name = "Netlinks that compose the net boundary." ;
double time(time) ;
        time:units = "seconds since 1992-08-31 00:00:00" ;
double FlowElem_xcc(nFlowElem) ;
        FlowElem_xcc:units = "m" ;
        FlowElem_xcc:standard_name = "projection_x_coordinate" ;
        FlowElem_xcc:long_name = "Flow element circumcenter x" ;
        FlowElem_xcc:bounds = "FlowElemContour_x" ;
double FlowElem_ycc(nFlowElem) ;
        FlowElem_ycc:units = "m" ;
        FlowElem_ycc:standard_name = "projection_y_coordinate" ;
        FlowElem_ycc:long_name = "Flow element circumcenter y" ;
        FlowElem_ycc:bounds = "FlowElemContour_y" ;
double FlowElemContour_x(nFlowElem, nFlowElemContourPts) ;
        FlowElemContour_x:units = "m" ;
        FlowElemContour_x:standard_name = "projection_x_coordinate" ;
        FlowElemContour_x:long_name = "List of x-points forming flow element" ;
double FlowElemContour_y(nFlowElem, nFlowElemContourPts) ;
        FlowElemContour_y:units = "m" ;
        FlowElemContour_y:standard_name = "projection_y_coordinate" ;
        FlowElemContour_y:long_name = "List of y-points forming flow element" ;
int FlowLink(nFlowLink, nFlowLinkPts) ;
        FlowLink:long_name = "link/interface between two flow elements" ;
int FlowLinkType(nFlowLink) ;
        FlowLinkType:long_name = "type of flowlink" ;
        FlowLinkType:valid_range = 1, 2 ;
        FlowLinkType:flag_values = 1, 2 ;
        FlowLinkType:flag_meanings = "link_between_1D_flow_elements link_between_2D_flow_elements" ;
double FlowLink_xu(nFlowLink) ;
        FlowLink_xu:long_name = "Center coordinate of net link (velocity point)." ;
        FlowLink_xu:units = "m" ;
        FlowLink_xu:standard_name = "projection_x_coordinate" ;
double FlowLink_yu(nFlowLink) ;
        FlowLink_yu:long_name = "Center coordinate of net link (velocity point)." ;
        FlowLink_yu:units = "m" ;
        FlowLink_yu:standard_name = "projection_y_coordinate" ;
double sl(time, nFlowElem) ;
        sl:coordinates = "FlowElem_xcc FlowElem_ycc" ;
double ucx(time, nFlowElem) ;
        ucx:coordinates = "FlowElem_xcc FlowElem_ycc" ;
double ucy(time, nFlowElem) ;
        ucy:coordinates = "FlowElem_xcc FlowElem_ycc" ;
double unorm(time, nFlowLink) ;
        unorm:standard_name = "normal_velocity" ;
        unorm:units = "m s-1" ;
        unorm:interfaces = "FlowLink" ;
        unorm:coordinates = "FlowLink_xu FlowLink_yu" ;

// global attributes:
        :institution = "Deltares" ;
        :references = "http://www.deltares.nl" ;
        :source = "UNSTRUC v1.0.11.10693:1069, model" ;
        :history = "Created on 2010-03-08T13:23:31+0100, UNSTRUC" ;
        :Conventions = "CF-1.4:Deltares-0.1" ;

data:

```

```

NetNode_x = 1078.99694824, 1068.75195312, 1110.4510498, 1123.6159668,
1032.6619873, 1023.13500977, 984.9420166, 974.37200928, 936.1550293,
922.88897705, 886.71002197, 868.85601807, 837.2210083, 812.21697998,
// ...
time = 10.7142857142857 ;

FlowElem_xcc = 1800.29010953126, 1827.13454757167, 1892.73565627098,
// ...
FlowElemContour_x =
1799.14001465, 1843.04504395, 1796.97290039, 1758.60998535, _', _', _', _', _',
_', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _',
_', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _',
_', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _', _',
// ...
FlowLink =
1, 2,
2, 9,

```

D-Flow FM map file with z-layers (concept)

```

netcdf test_map {
dimensions:
    nNetNode = 706 ;
    nNetLink = 1313 ;
    nNetLinkPts = 2 ;
    nFlowElem = 665 ;
    nFlowElemMaxNode = 6 ;
    nFlowElemContourPts = 99 ;
    nFlowLink = 1143 ;
    nFlowLinkPts = 2 ;
    nLayer = 5;
    time = UNLIMITED ; // (1 currently)
variables:
    double NetNode_x(nNetNode) ;
        NetNode_x:units = "m" ;
        NetNode_x:standard_name = "projection_x_coordinate" ;
        NetNode_x:long_name = "netnodal x-coordinate" ;
    double NetNode_y(nNetNode) ;
        NetNode_y:units = "m" ;
        NetNode_y:standard_name = "projection_y_coordinate" ;
        NetNode_y:long_name = "netnodal y-coordinate" ;
    double NetNode_z(nNetNode) ;
    int NetLink(nNetLink, nNetLinkPts) ;
        NetLink:standard_name = "netlink" ;
        NetLink:long_name = "link between two netnodes" ;
    int NetLinkType(nNetLink) ;
        NetLinkType:long_name = "type of netlink" ;
        NetLinkType:valid_range = 0, 2 ;
        NetLinkType:flag_values = 0, 1, 2 ;
        NetLinkType:flag_meanings = "closed_link_between_2D_nodes link_between_1D_nodes
link_between_2D_nodes" ;
    double time(time) ;
        time:units = "seconds since 1992-08-31 00:00:00" ;
    double FlowElem_xcc(nFlowElem) ;
        FlowElem_xcc:units = "m" ;
        FlowElem_xcc:standard_name = "projection_x_coordinate" ;
        FlowElem_xcc:long_name = "Flow element circumcenter x" ;
        FlowElem_xcc:bounds = "FlowElemContour_x" ;
    double FlowElem_ycc(nFlowElem) ;
        FlowElem_ycc:units = "m" ;
        FlowElem_ycc:standard_name = "projection_y_coordinate" ;
        FlowElem_ycc:long_name = "Flow element circumcenter y" ;
        FlowElem_ycc:bounds = "FlowElemContour_y" ;
    double FlowElemContour_x(nFlowElem, nFlowElemContourPts) ;
        FlowElemContour_x:units = "m" ;
        FlowElemContour_x:standard_name = "projection_x_coordinate" ;

```

```

        FlowElemContour_x:long_name = "List of x-points forming flow element" ;
double FlowElemContour_y(nFlowElem, nFlowElemContourPts) ;
        FlowElemContour_y:units = "m" ;
        FlowElemContour_y:standard_name = "projection_y_coordinate" ;
        FlowElemContour_y:long_name = "List of y-points forming flow element" ;
int FlowLink(nFlowLink, nFlowLinkPts) ;
        FlowLink:long_name = "link/interface between two flow elements" ;
int FlowLinkType(nFlowLink) ;
        FlowLinkType:long_name = "type of flowlink" ;
        FlowLinkType:valid_range = 1, 2 ;
        FlowLinkType:flag_values = 1, 2 ;
        FlowLinkType:flag_meanings = "link_between_1D_flow_elements link_between_2D_flow_elements" ;
double FlowLink_xu(nFlowLink) ;
        FlowLink_xu:long_name = "Center coordinate of net link (velocity point)." ;
        FlowLink_xu:units = "m" ;
        FlowLink_xu:standard_name = "projection_x_coordinate" ;
double FlowLink_yu(nFlowLink) ;
        FlowLink_yu:long_name = "Center coordinate of net link (velocity point)." ;
        FlowLink_yu:units = "m" ;
        FlowLink_yu:standard_name = "projection_y_coordinate" ;
double FlowElem_zc(nFlowElem, nLayer) ;
        FlowElem_zc:long_name = "Vertical coordinates of layers underlying each FlowElement." ;
        FlowElem_zc:units = "m" ;
        FlowElem_zc:layer_type = "z" ; // see other example for sigma
        FlowElem_zc:standard_name = "???" ;
double FlowElem_bl(nFlowElem) ;
        FlowElem_bl:units = "m" ;
        FlowElem_bl:long_name = "bottom level" ;
        FlowElem_bl:standard_name = "sea_floor_depth" ; // wrt geoid, maar doen wij dat wel zo?
double sl(time, nFlowElem) ;
        sl:coordinates = "FlowElem_xcc FlowElem_ycc" ;
double ucx(time, nFlowElem, nLayer) ;
        ucx:coordinates = "FlowElem_xcc FlowElem_ycc" ;
double ucy(time, nFlowElem, nLayer) ;
        ucy:coordinates = "FlowElem_xcc FlowElem_ycc FlowElem_zcc" ; // AvD: Mogen de z-coordinaten
hier ook nog bij? (nFlowElem mapt naar x+y, nLayer naar z)

```

D-Flow FM map file with sigma coords (concept)

```

netcdf test_map {
dimensions:
    nNetNode = 706 ;
    nNetLink = 1313 ;
    nNetLinkPts = 2 ;
    nFlowElem = 665 ;
    nFlowElemMaxNode = 6 ;
    nFlowElemContourPts = 99 ;
    nFlowLink = 1143 ;
    nFlowLinkPts = 2 ;
    nLayer = 5;
    time = UNLIMITED ; // (1 currently)
variables:
    double FlowElem_bl(nFlowElem) ;
        FlowElem_bl:units = "m" ;
        FlowElem_bl:long_name = "bottom level" ;
        FlowElem_bl:standard_name = "sea_floor_depth" ; // wrt geoid, maar doen wij dat wel zo?
    double FlowElem_zc(nFlowElem, nLayer) ;
        FlowElem_zc:long_name = "Vertical coordinates of layers underlying each FlowElement." ;
        FlowElem_zc:units = "" ;
        FlowElem_zc:standard_name = "ocean_sigma_coordinate" ;
        FlowElem_zc:formula_terms = "sigma: FlowElem_zc eta: s1 depth: FlowElem_bl" ; // mooie
oplossing: bottomlevel (en sigma/zc) mag tijdsafhankelijk zijn
    double s1(time, nFlowElem) ;
        s1:coordinates = "FlowElem_xcc FlowElem_ycc" ;
    double ucx(time, nFlowElem, nLayer) ;
        ucx:coordinates = "FlowElem_xcc FlowElem_ycc" ;
// ...

```

D-Flow FM map file with subdomains for variable nr of layers (flexible layers, concept)


```

netcdf test_map {
dimensions:
    nNetNode1 = 106 ;
    nNetLink1 = 313 ;
    nNetLinkPts1 = 2 ;
    nFlowElem1 = 165 ;
    nFlowElemMaxNode1 = 6 ;
    nFlowElemContourPts1 = 99 ;
    nFlowLink1 = 343 ;
    nFlowLinkPts1 = 2 ;
    nLayer1 = 5;
    nNetNode2 = 486 ;
    nNetLink2 = 673 ;
    nNetLinkPts2 = 2 ;
    nFlowElem2 = 578 ;
    nFlowElemMaxNode2 = 6 ;
    nFlowElemContourPts2 = 99 ;
    nFlowLink2 = 840 ;
    nFlowLinkPts2 = 2 ;
    nLayer2 = 10 ;
    // ... idem for 3 and 4
    nSubdomain = 4 ;
    nSubdomainLink = 3 ; // for example: 4 subdoms in one row
    nSubdomainLinkPts = 2 ;
    time = UNLIMITED ; // (1 currently)
variables:
    double NetNode_x_1(nNetNode1) ;
        NetNode_x_1:units = "m" ;
        NetNode_x_1:standard_name = "projection_x_coordinate" ;
        NetNode_x_1:long_name = "netnodal x-coordinate" ;
    // ...
    double NetNode_x_2(nNetNode1) ;
        NetNode_x_2:units = "m" ;
        NetNode_x_2:standard_name = "projection_x_coordinate" ;
        NetNode_x_2:long_name = "netnodal x-coordinate" ;

    double s1_1(time, nFlowElem_1) ;
        s1_1:coordinates = "FlowElem_xcc_1 FlowElem_ycc_1" ;
    double s1_2(time, nFlowElem_2) ;
        s1_2:coordinates = "FlowElem_xcc_2 FlowElem_ycc_2" ;
    // ...
    integer SubdomainLink (nSubdomainLink, nSubdomainLinkPts) ;
        SubdomainLink:long_name = "Meta link between subdomain numbers."
    // .. randlinks tussen subdomains kunnen nuttig zijn, maar zijn ws niet nodig voor visualisatie?

    integer Subdomain ;
        Subdomain:interfaces = "SubdomainLink"

data:

SubdomainLink =
    1, 2,
    2, 3,
    3, 4 ;

Subdomain = 4; // weinig bijzonders.

```