NetCDF in MODFLOW 6

By the MODFLOW Development Team

MODFLOW 6 and FloPy: Take Your Modeling to the Next Level

Princeton, New Jersey, USA

Friday, May 31 – June 1, 2024

USGS science for a changing world

Deltares

GSI ENVIRONMENTAL

UCAR

NetCDF binary format

"NetCDF (Network Common Data Form) is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. It is also a community standard for sharing scientific data." (UCAR)

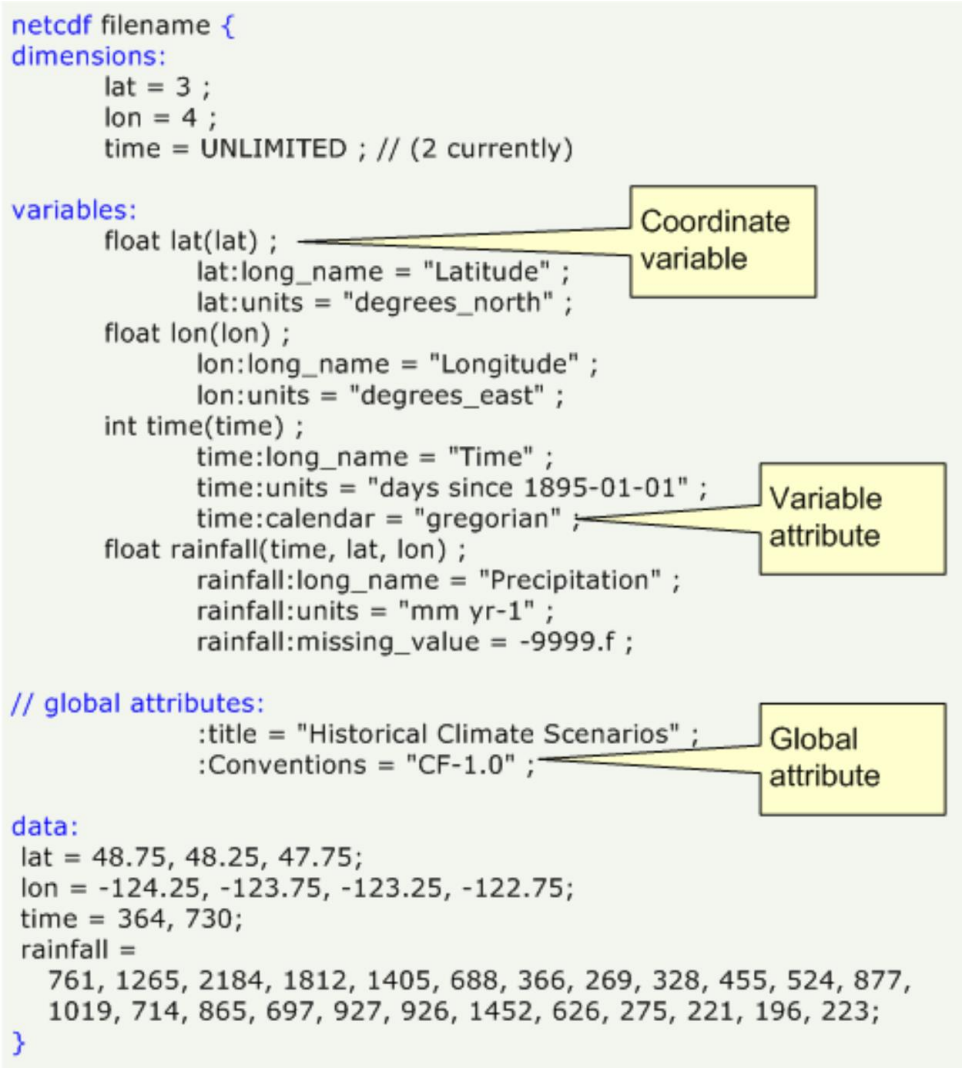
- The Unidata Program Center supports C, C++, Java, and Fortran NetCDF programming interfaces

- Programming interfaces are also available for Python, IDL, MATLAB, R, Ruby, and Perl.

NetCDF data model

- A NetCDF dataset contains dimensions, variables, and attributes

- Attributes associated with data describe the meaning of the data and relations to other data
  - These constitute the self-describing nature of NetCDF



Basic components of NetCDF file

NetCDF conventions in MODFLOW 6 exports

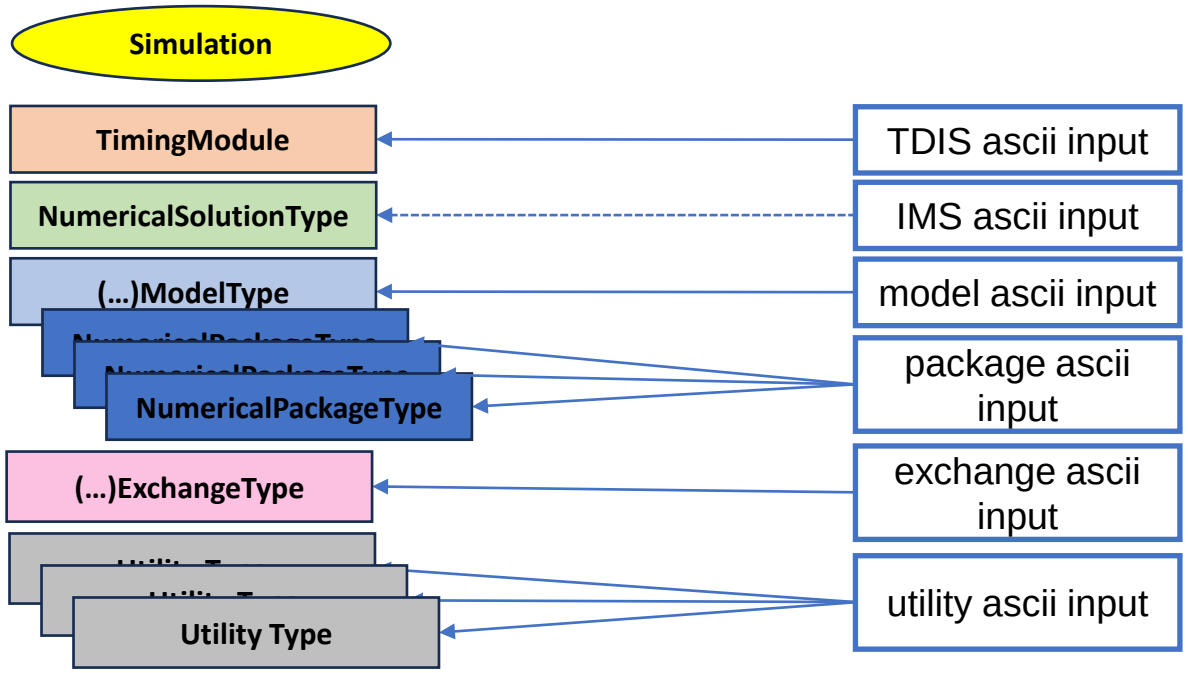
- Conventions are guidelines and recommendations
  - Which metadata and where to put it
  - Intended to promote the processing and sharing of files
- CF Conventions
  - Data provenance: e.g. `title`, `institution`, `contact`, `source`, `history`, `references`
  - Description of data: e.g. `units`, `standard_name`, `long_name`, `missing_value`, `valid_range`
  - Description of coordinates: e.g. `coordinates`, `bounds`, `grid_mapping`, `calendar`
  - Meaning of grid cells: e.g. `cell_methods`, `cell_measures`
- UGRID
  - Conventions for storing unstructured (or flexible mesh) grid topologies in NetCDF
  - Included in CF Conventions 1-11

MODFLOW 6 NetCDF integration

- NetCDF prototyping has been ongoing for both input and output (export) files
- NetCDF examples that follow are prototypes, not yet available in a release or nightly build
- The Input Data Processor (IDP) enables this effort

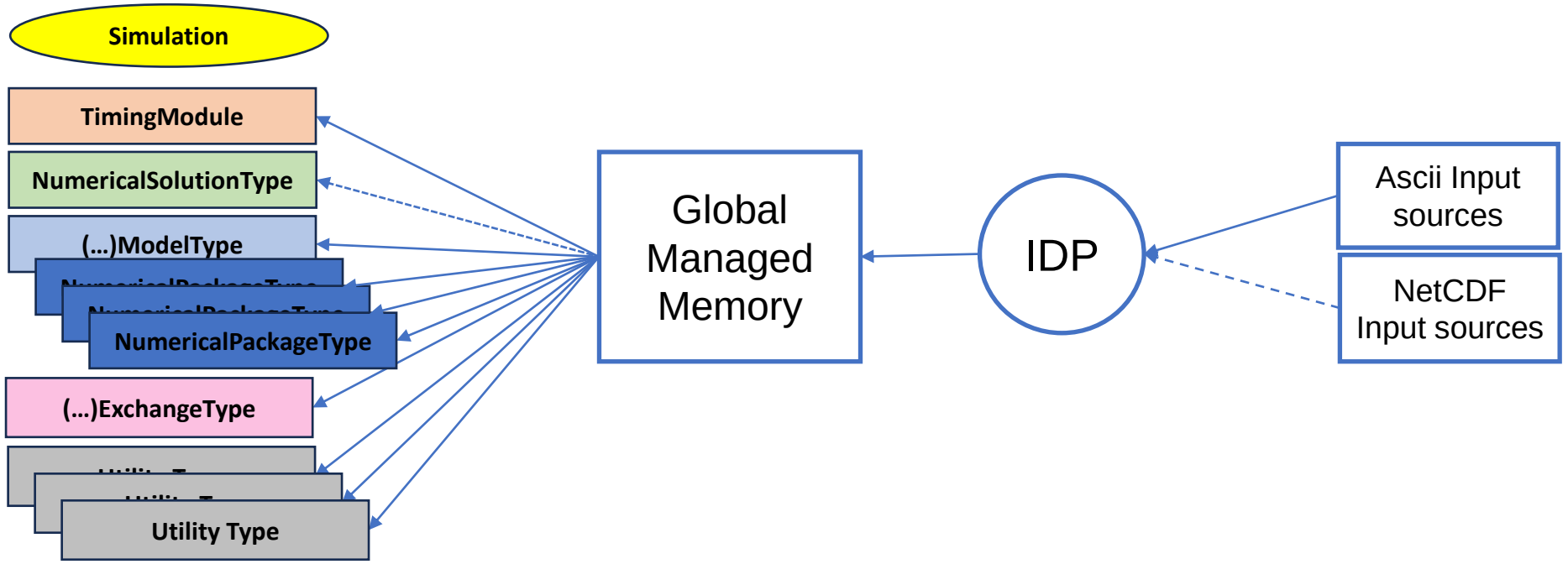
Input Data Processor (IDP) in MODFLOW 6

- Historically, inputs have been read by their associated components



Input Data Processor (IDP) in MODFLOW 6

- IDP isolates simulation components from input data sources
- Component by component conversion is ongoing (mf6io for table)



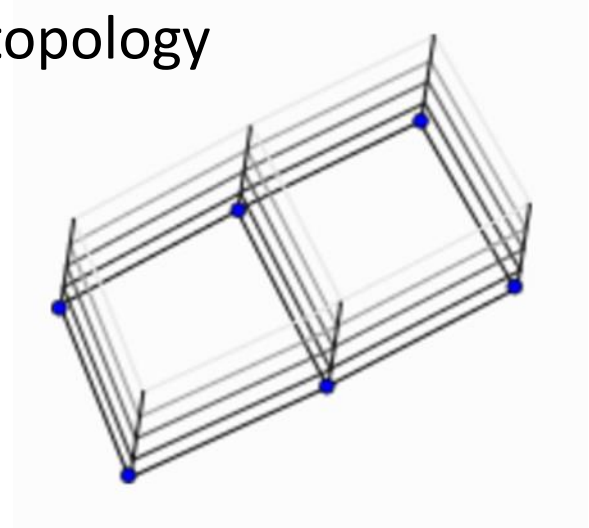
MODFLOW 6 NetCDF files

- A MODFLOW 6 NetCDF file is associated with a model:

```
// global attributes:
        :title = "GEO_MODEL hydraulic head" ;
        :source = "MODFLOW 6 6.5.0.dev2 (preliminary) 02/13/2024" ;
        :model = "GEO_MODEL: MODFLOW 6 Groundwater Flow (GWF) model" ;
        :history = "first created 2024/5/28 8:34:34.720" ;
        :Conventions = "CF-1.11 UGRID-1.0" ;
}
```

MODFLOW 6 NetCDF export integration

- MODFLOW 6 NetCDF outputs define a UGRID layered mesh topology
  - DIS and DISV packages are supported



```
int mesh ;
    mesh:cf_role = "mesh_topology" ;
    mesh:long_name = "2D mesh topology" ;
    mesh:topology_dimension = 2 ;
    mesh:face_dimension = "nmesh_face" ;
    mesh:node_coordinates = "mesh_node_x mesh_node_y" ;
    mesh:face_coordinates = "mesh_face_x mesh_face_y" ;
    mesh:face_node_connectivity = "mesh_face_nodes" ;
```

- All variables are layered:

```
double head_l1(time, nmesh_face) ;
    head_l1:units = "m" ;
    head_l1:standard_name = "soil_water__pressure_head" ;
    head_l1:long_name = "hydraulic head (layer 1)" ;
    head_l1:_FillValue = 1.e+30 ;
    head_l1:mesh = "mesh" ;
    head_l1:location = "face" ;
    head_l1:coordinates = "mesh_face_x mesh_face_y" ;
    head_l1:grid_mapping = "projection" ;
```

MODFLOW 6 NetCDF export integration

- MODFLOW 6 NetCDF export data
  - Timeseries data for the model dependent variable
  - Optional arrays of gridded input- assumes Input Data Processor (IDP) integration

```
double npf-k_l1(nmesh_face) ;
    npf-k_l1:_FillValue = 9.96920996838687e+36 ;
    npf-k_l1:long_name = "hydraulic conductivity (L/T) (layer 1)" ;
    npf-k_l1:coordinates = "mesh_face_x mesh_face_y" ;
    npf-k_l1:grid_mapping = "projection" ;
```

MODFLOW 6 NetCDF export integration

- MODFLOW 6 NetCDF configuration utility package
  - A utility package can be defined to provide projection (CRS) related information that is written directly to the NetCDF file in a grid mapping variable.
  - Expected to support other configuration options in time

```
int projection ;
    projection:wkt = "PROJCS[\"WGS_1984_UTM_ZONE_18S\",GEOGCS[\"GCS_WGS_1984\",DATUM[\"D_WGS_1984\",SPHEROID
    [\"WGS_1984\",6378137.0,298.257223563]],PRIMEM[\"GREENWICH\",0.0],UNIT[\"DEGREE\",0.0174532925199433]],
    PROJECTION[\"TRANSVERSE_MERCATOR\"],PARAMETER[\"FALSE_EASTING\",500000.0],PARAMETER[\"FALSE_NORTHING\",
    10000000.0],PARAMETER[\"CENTRAL_MERIDIAN\",-75.0],PARAMETER[\"SCALE_FACTOR\",0.9996],PARAMETER
    [\"LATITUDE_OF_ORIGIN\",0.0],UNIT[\"METER\",1.0]]" ;
```

Activate NetCDF export

- **Activate in model name file**
- Add package options to write gridded input data
- Define utility NCF6 package to add CRS string

```
gwf = flopy.mf6.ModflowGwf(
    sim,
    modelname=name,
    newtonoptions=newtonoptions,
    save_flows=True,
    export_netcdf="ugrid",
)
```

FloPy create GWF model

```
BEGIN options
  SAVE_FLOWS
  NEWTON
  EXPORT_NETCDF  ugrid
END options
```

MODFLOW 6 GWF name file options block

Write package gridded input to NetCDF file

- Activate in model name file
- **Add package options to write gridded input data**
- Define utility NCF6 package to add CRS string

```
dis = flopy.mf6.ModflowGwfdis(
    gwf,
    export_array_netcdf=True,
    nlay=nlay,
    nrow=nrow,
    ncol=ncol,
    delr=delr,
    delc=delc,
    top=top,
    botm=botm,
    filename=f"{name}.dis",
)
```

FloPy create GWF model discretization package

```
BEGIN options
  EXPORT_ARRAY_NETCDF
  NCF6  FILEIN  gwf_sto01_ncf.dis.ncf
END options
```

MODFLOW 6 DIS/DISV package options block

Write CRS projection string to NetCDF file

- Activate in model name file
- Add package options to write gridded input data
- **Define utility NCF6 package to add CRS string**

```
# netcdf configuration
flopy.mf6.ModflowUtlncf(
    dis,
    ogc_wkt=wkt,
    filename=f"{name}.dis.ncf",
)
```

FloPy create DIS NCF utility package

```
BEGIN options
  EXPORT_ARRAY_NETCDF
  NCF6  FILEIN  gwf_sto01_ncf.dis.ncf
END options
```

MODFLOW 6 DIS/DISV package options block

```
BEGIN options
  ogc_wkt  'PROJCS["NAD83 / UTM zone 18N", GEOGCS["NAD83", DATUM["North_American_Datum_1983", SPHEROID["GRS 1980",
  6378137,298.257222101], TOWGS84[0,0,0,0,0,0,0]], PRIMEM["Greenwich",0,AUTHORITY["EPSG","8901"]], UNIT["degree",0.
  0174532925199433,AUTHORITY["EPSG","9122"]], AUTHORITY["EPSG","4269"]], PROJECTION["Transverse_Mercator"],
  PARAMETER["latitude_of_origin",0], PARAMETER["central_meridian",-75], PARAMETER["scale_factor",0.9996], PARAMETER
  ["false_easting",500000], PARAMETER["false_northing",0], UNIT["metre",1,AUTHORITY["EPSG","9001"]], AXIS["Easting",
  EAST], AXIS["Northing",NORTH], AUTHORITY["EPSG","26918"]]'
END options
```

MODFLOW 6 NCF package options block

QGIS demo

- Hatari Labs teaching models
  - River GWF model
  - Regional GWF model

Next steps

- Tuesday poster session: more examples, stop by for discussion or to give feedback
- Additional prototypes
  - Explore UGRID 3D fully unstructured topology
  - Consider export not based on UGRID
- Evaluation of exports in other visualization tools, e.g. ParaView
- The NetCDF export capability will become available in a special "Extended" version of MODFLOW 6, which is planned for release later this year.