Spectral grids

Unknown macro: {hide-from}

Unknown macro: {show-to}

Please find on this page a **deprecated proposal** for netCDF output from the spectral wave model SWAN. In the mean time netCDF has been implemented in the SWAN trunk, in a collaboration between Deltares (on behalf or Rijkswaterstaat), BMT Argoss, and TU Delft using somewhat different names (see routine `agioncmd.f90`). We adopted this page a bit to match the final implementation. There is also a Matlab routine in OpenEarthTools to convert ASCII spectral files into netCDF. Elements from the SWAN input file are here refered to by `INPUT*`, whereas elements from the SWAN Fortran code swanmain.for are referred to by `OV*IVTYPE`.

SWAN netCDF-CF CDL scheme

```
 // A working draft proposal for CF compliance for netCDF output for SWAN
NetCDF SWAN.nc {

// proposal for CF and SWAN meta data in SWAN

dimensions:
        time      = 1; // time:                also put time in for single times
        mx        = 2; // yx:                (mxc+1) for 2D grids
        my        = 3; // xc:                (myc+1) for 2D grids
        direction = 4; // nd: (mdc+1) directions, now called nf90_def_dim
        frequency = 5; // ms: (msc+1) frequencies
        points    = 6; // points:        for 1D grids, for UNSTRUC and for squeezed [mx by my] grids

variables:
// dimensions consistent with ncBrowse, not with native MATLAB netcdf package.
// ----------------- COORDINATES -----------------
// Options:
// * like SWAN use same variable for CARTESIAN and SPHERICAL, but change attributes
// * use different variables for CARTESIAN and SPHERICAL, but change attributes
// if INPUT.CGRID.REGular
        double mx(mx) ;
                mx(mx):swan_xpc     = // INPUT value: optional for input, implemented in output
                mx(mx):swan_xlenc   = // INPUT value: optional for input, implemented in output
                mx(mx):swan_alpc    = // INPUT value: optional for input, implemented in output
                mx(mx):swan_comment = "mxc+1"
                mx(mx):_FillValue   = // INPUT.CGRID.xexc or OVEXCV{IVTYPE}
                mx:actual_range     = [~ ~]
                // if INPUT.CGRID.CARTESIAN
                mx:standard_name    = "projected_x_coordinate"
                mx:units            = "m"
                mx(mx):grid_mapping = "projected_coordinate_system"
                // elseif INPUT.CGRID.SPHERICAL
                mx:standard_name    = "longitude"
                mx:units            = "degrees_east"
                mx(mx):grid_mapping = "wgs84"
                // end
                ...
        double my(my) ;
                my(my):swan_ypc     = // INPUT value: optional for input, implemented in output
                my(my):swan_ylenc   = // INPUT value: optional for input, implemented in output
                my(my):swan_alpc    = // INPUT value: optional for input, implemented in output
                my(my):swan_comment = "myc+1"
                my(my):_FillValue   = // INPUT.CGRID.yexc or OVEXCV{IVTYPE}
                my:actual_range     = [~ ~]
                // if INPUT.CGRID.CARTESIAN
                my:standard_name    = "projected_y_coordinate"
                my:units            = "m"
```

```
                my(my):grid_mapping = "projected_coordinate_system"
                // elseif INPUT.CGRID.SPHERICAL
                my:standard_name    = "latitude"
                my:units            = "degrees_north"
                my(my):grid_mapping = "wgs84"
                // end
                ...
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
// elseif INPUT.CGRID.CURVilinear
        double XP(xc,yc) ;
                // idem
                ...
        double YP(xc,yc) ;
                // idem
                ...
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
// elseif INPUT.CGRID.UNSTRUCtured
        double x(points) ;
                // idem
                ...
        double YyP(points) ;
                // idem
                ...
// end
// ----------------- DIRECTIONS -----------------
        double spread_1d(md) ;
                spread_1d:standard_name    = ""
                spread_1d:units            = "degrees_true"
                spread_1d:standard_name    = "sea_surface_wave_from_direction" // sea_surface_wave_to_direction
                spread_1d:nautical         = 1 // or 0
        spread_1d.dir1             = // INPUT value: optional for input, implemented in output
                spread_1d.dir2             = // INPUT value: optional for input, implemented in output
                spread_1d:swan_comment     = "mdc+1"
                ...
// ----------------- FREQUENCY -----------------
        double frequency(ms) ;
                frequency:standard_name    = "wave_frequency"
                frequency:long_name        = "frequencies" // absolute vs relative, see attribute
'relative_to_current' of energy
                frequency:units            = "s-1"
                frequency.swan_flow        = // INPUT value: optional for input, implemented in output
                frequency.swan_fhigh       = // INPUT value: optional for input, implemented in output
                frequency:swan_comment     = "msc+1"
                ...
// ----------------- COORDINATE SYSTEMS -----------------
// The projected_coordinate_system information could optionally be added to
// either SWAN input or in a post-processing step.
// if INPUT.CGRID.CARTESIAN
//        int32 projected_coordinate_system() ;
//                projected_coordinate_system:name = "Amersfoort / RD New"
//                projected_coordinate_system:epsg = 28992
//                projected_coordinate_system:epsg_name = "Oblique Stereographic"
//                projected_coordinate_system:grid_mapping_name = " "
//                projected_coordinate_system:semi_major_axis = 6.3774e+006
//                projected_coordinate_system:semi_minor_axis = 6.35608e+006
//                projected_coordinate_system:inverse_flattening = 299.153
//                projected_coordinate_system:latitude_of_projection_origin = 52.0922
//                projected_coordinate_system:longitude_of_projection_origin = 5.23155
//                projected_coordinate_system:false_easting = 155000
//                projected_coordinate_system:false_northing = 463000
//                projected_coordinate_system:scale_factor_at_projection_origin = 0.999908
//                // optional PROJ4 REQUIRED FOR ADAGUC.KNMI.NL
//                projected_coordinate_system:proj4_params = "+proj=sterea +lat_0=52.15616055555555 +lon_0=5.
38763888888889 +k=0.999908 +x_0=155000 +y_0=463000 +ellps=bessel +units=m +towgs84=565.4174,50.3319,465.5542,
-0.398957388243134,0.343987817378283,-1.87740163998045,4.0725 +no_defs"
//                projected_coordinate_system:EPSG_code = "EPSG:28992"
//                projected_coordinate_system:projection_name = "Dutch rijksdriekhoek system"
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
// elseif INPUT.CGRID.SPHERICAL
// The (lat,lon) coordinates information could be hard-coded into SWAN
// with each of the options: CCM, QC or REPEATING
```

```
                int32 wgs84() ;
                wgs84:name = "WGS 84"
                wgs84:epsg = 4326
                wgs84:grid_mapping_name = "latitude_longitude"
                wgs84:semi_major_axis = 6.37814e+006
                wgs84:semi_minor_axis = 6.35675e+006
                wgs84:inverse_flattening = 298.257
                // optional PROJ4 REQUIRED FOR ADAGUC.KNMI.NL
                wgs84:proj4_params = "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs "
                wgs84:EPSG_code = "EPSG:4326"
                wgs84:projection_name = "Latitude Longitude"
                wgs84:wkt = "GEOGCS["WGS 84",...
// end
// ----------------- PRIMARY VARIABLES -----------------
        single density(), shape = [~ ~] // density_1d for 1D, density for 2D
                density:relative_to_current   = 0 // or 1, depending on respectively. AFREQ vs RFREQ
                density:swan_code             =            // OVKEYW{IVTYPE} = 'VaDens';
                density:swan_name             =            // OVSNAM{IVTYPE} = 'VaDens';
                density:long_name             =            // OVLNAM{IVTYPE} = 'spectral variance density';
                density:swan_units            =            // OVUNIT{IVTYPE} = 'm2/Hz';
                density:valid_range(1)        =            // OVLEXP{IVTYPE} = 0.;
                density:valid_range(2)        =            // OVHEXP{IVTYPE} = 100.;
                density:_FillValue            =            // OVEXCV{IVTYPE} = -99.;
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                density:units                 =            // OVCFUD{IVTYPE} = ''; NEW TABLE NEEDED INSIDE SWAN
                density:standard_name         =            // OVCFSN{IVTYPE} = ''; NEW TABLE NEEDED INSIDE SWAN
                // these do not need to be mapped         // OVSVTY{IVTYPE} = 5;     // means vector, scalar
etc.
                // these do not need to be mapped         // OVLLIM{IVTYPE} = 0.;    // print width for ascii
output
                // these do not need to be mapped         // OVULIM{IVTYPE} = 1000.; // print width for ascii
output
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                density:actual_range          = [~ ~]      // add optionally as extra service to user
                density:coordinates           = "XP YP"    //
                // if INPUT.CGRID.CARTESIAN
                density():grid_mapping        = "projected_coordinate_system"
                // elseif INPUT.CGRID.SPHERICAL
                density():grid_mapping        = "wgs84"
                // end
// ----------------- GLOBAL META_DATA -----------------
//global Attributes:
                :title = "INPUT.PROJECT.name,INPUT.PROJECT.nr"
                :institution = ""
                :source = ""
                :history = "Data produced by SWAN version 40.72AB"
                :references = ""
                :email = ""
                :comment = "INPUT.PROJECT.title1,INPUT.PROJECT.title2,INPUT.PROJECT.title3"
                :version = ""
                :Conventions = "CF-1.5"
                :CF:featureType = "Grid"
                :terms_of_use = "These data can be used freely for research purposes provided that the
following source is acknowledged: institution"
                :disclaimer = "This data is made available in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE."
}
```

Mapping of SWAN names to CF standard names

SWAN Standard names