Current status of GSL

Yu. Yakovenko, P. Chmielewski

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Outline

- Introduction
- How GGD is organized
- What is available
- What is missing at the moment

Terminology

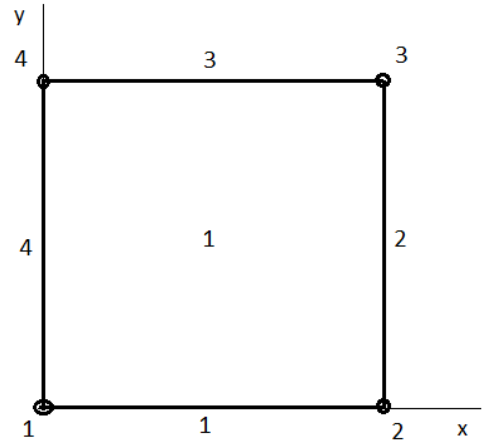
- GSL is GGD Standard Library – a Fortran-90 library intended for work with GGD data.
- GGD is a data structure for storing grids of arbitrary structure, together with physical quantities given on these grids.
- GGD is found in the Data Dictionary of each IMAS IDS.
- GSL can work with both structured and unstructured grids.
  - ‘Structured grids’ are grids with quadrangular (hexahedra) cells organized in rows and columns.
  - ‘Unstructured grids’ are all other grids (e.g., triangular-cell grids).
- The aim of the presentation is to review what is available in the library.
- The review is based on the documentation to version 1.4, 03 Apr 2018 (available at iter.org).

GGD basics

- Each IDS contains the data structures `grid_ggd` and `ggd`. `grid_ggd` holds grid description; `ggd`, quantities on this grid.
- The grid may vary with time (separate grid for each IDS time slice) or be constant (just make sure that the reader knows it!).
- The AOS (array of structures) `grid_ggd` describes the grid elements (nodes, edges, cells etc.), their geometry, and relations between them.
- Physical quantities are “attached” to grid subsets (for example, all grid points, all grid cells, SOL grid points, separatrix grid edges etc.).
- If you want to store, say, ion flux at separatrix edges, you put a list of the flux values to `ggd` and refer to the separatrix-edges subset (defined in `grid_gdd`).

How the grid structure is described

- To prepare a GGD grid, we need
  - To allocate all arrays of grid elements
  - To provide to node coordinates
  - To provide all relations between objects :
- Example: one cell in 2D-space




```
...objects_per_dimension(1)%object(1)%geometry(1) = 0.0  ! x-coord, node 1
...objects_per_dimension(1)%object(1)%geometry(2) = 0.0  ! y-coord, node 1
!...................   The coordinates of all other nodes
...objects_per_dimension(2)%object(2)%node(2)=2  ! Nodes of edge 2
...objects_per_dimension(2)%object(2)%node(2)=3  !
!...................   Nodes of all other edges
...objects_per_dimension(3)%object(1)%node(1)=1  ! Nodes of cell 1
...objects_per_dimension(3)%object(1)%node(2)=2  !
...objects_per_dimension(3)%object(1)%node(3)=3  !
...objects_per_dimension(3)%object(1)%node(4)=4  !
!...................   Nodes of all other cells
```

GSL content: edge plasma IDS

- Module `ids_helper` provides routines for initializing and writing several IDS's (`edge_profiles`, `edge_sources`, `edge_transport`, `transport_solver_numerics`).
- Module `ids_grid_common` defines symbolic notations for standard IDS constants (say, `coordtype_rho_tor=107`).

Generic (unstructured) grids in GSL

- Module `ids_grid_access` contains routines for accessing basic grid information (number of objects etc.).
- In addition, it contains function `ggdtimeslice`, reading a time slice from IDS `edge_profiles`.
- Modules `ids_grid_object` and `ids_grid_objectlist` provide service routines to deal with grid objects (say, to get a list of nodes constituting a cell).
- Module `ids_grid_subgrid` contains routines for organizing grid subsets.
- Module ids_grid_data contains subroutines `gridwritedatascalar`, `gridwritedatavector`, and `gridwritedatamatrix`, which set a scalar data field given in a scalar / vector/ matrix representation to a generic IDS data field; the input array is of dimension 1 for a scalar, …, 3 for a matrix.
  - It is not quite clear to me what is “a scalar data field given in a vector / matrix data representation”.

Generic grids in GSL, cont.

- Service routines:
  - Module `ids_grid` contains subroutines for grid sanity check.
  - Module `ids_assert` provides methods to deal with abnormal situations.

Structured grids in GSL

- Module `ids_grid_structured` provide tools for setting up a structured grid and working with them.
  - One simply gives meshes and types for all coordinates; the objects and relations between them are set up automatically.
  - Tools for getting properties of a grid and its objects
    - For example, measure (length, volume) of an object
  - Interface to modules `gridstructurereaddata` and `gridstructurewritedata`
- Module `gridstructurereaddata` contains subroutines which read data from a `ggd` data field and return a “usual” array (1D, 2D, 3D, …, 6D).
- Module `gridstructurewritedata` contains subroutines which put a “usual” array (1D - 6D) into a `ggd` data field.

Summary

Available service

	Generic grids	Structured grids
Grid setup	No	Yes
Conversion from non-GGD format	No	Yes
Info on grid	Yes	Yes
Info on grid objects	Yes (to a lesser extent)	Yes
Work with grid subsets	Yes	Yes
Reading and writing data	Partly	Yes

How to start

- Information on GGD: “GGD Guide” https://confluence.iter.org/download/attachments/178133297/GGDguide.pdf?version=1&modificationDate=1648825366807&api=v2
- The “GGD Guide” contains examples of the grid setup and how the GSL library can help with it (especially for structured grids). This is the best way to start.
- The Fortran text of the examples is a part of the libraty; it is available at https://git.iter.org/projects/IMEX/repos/ggd/browse/examples/f90
- Documentation of IMAS GGD Grid Service Library: Fortran 90 https://user.iter.org/?uid=VSF9KY&action=get_document is a reference document.

Conclusions

- The service provided for work with structured grids is much better than for unstructured ones (EIRENE).
- This presentation was intended to revise what is currently available.
- For detailed information on work with GSL see
  - GGD guide https://confluence.iter.org/download/attachments/178133297/GGDguide.pdf?version=1&modificationDate=1648825366807&api=v2
  - Fortran examples provided with GSL (they are described in GGD Guide) https://git.iter.org/projects/IMEX/repos/ggd/browse/examples/f90
  - Documentation of IMAS GGD Grid Service Library: Fortran 90 https://user.iter.org/?uid=VSF9KY&action=get_document

Thank you for attention!