SOLPS-EIRENE IMAS interface status

Piotr Chmielewski, Dmitriy Yadykin, PSNC ACH

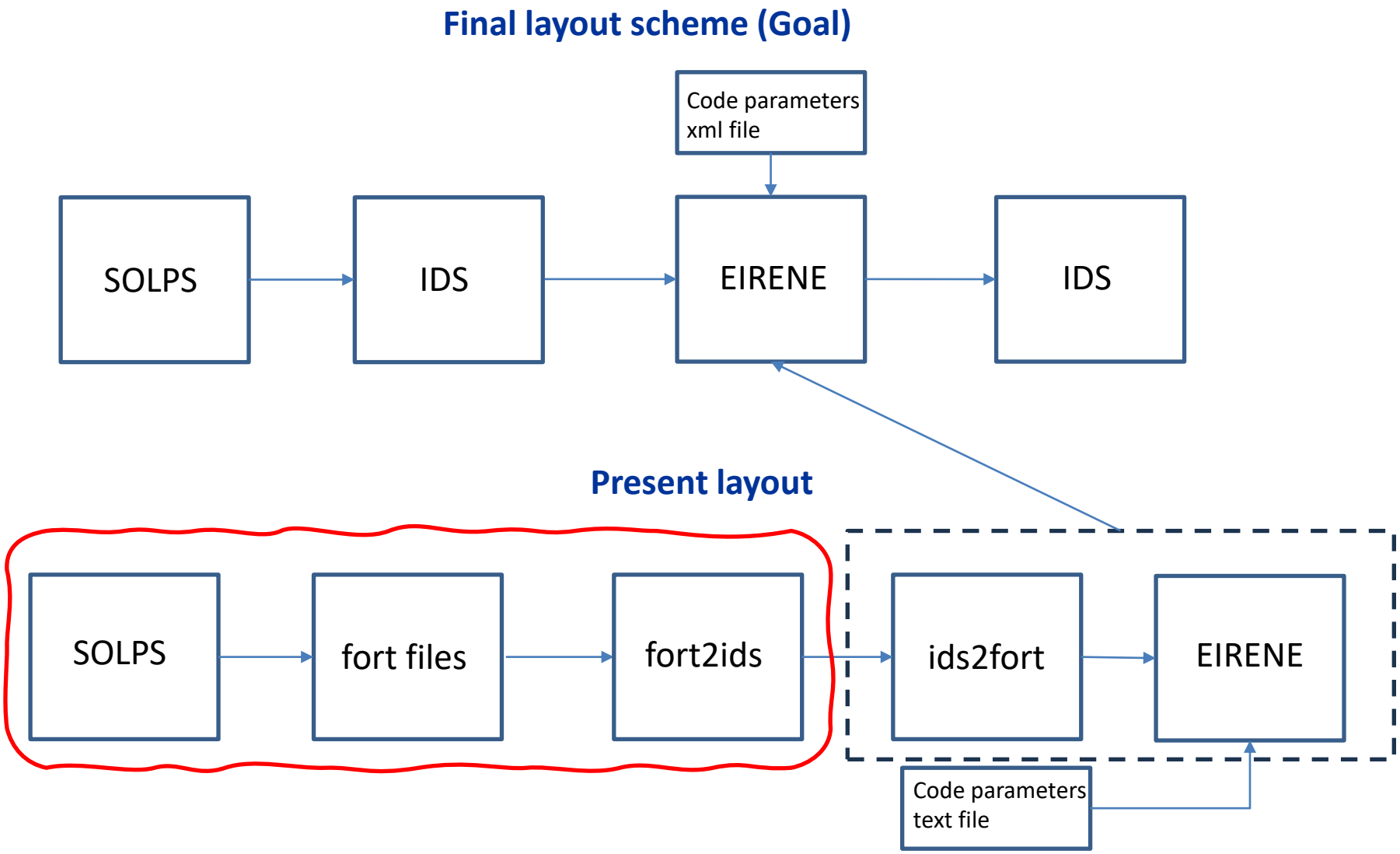
This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect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

SOLPS-EIRENE IMAS interface scheme

Outline

- Introduction
- SOLPS-ITER and EIRENE interaction
  - Output of SOLPS-ITER
- GGD structure
- Current interface structure
- Possibilities of development

Introduction



SOLPS-ITER and EIRENE interaction

SOLPS-ITER generates the following files

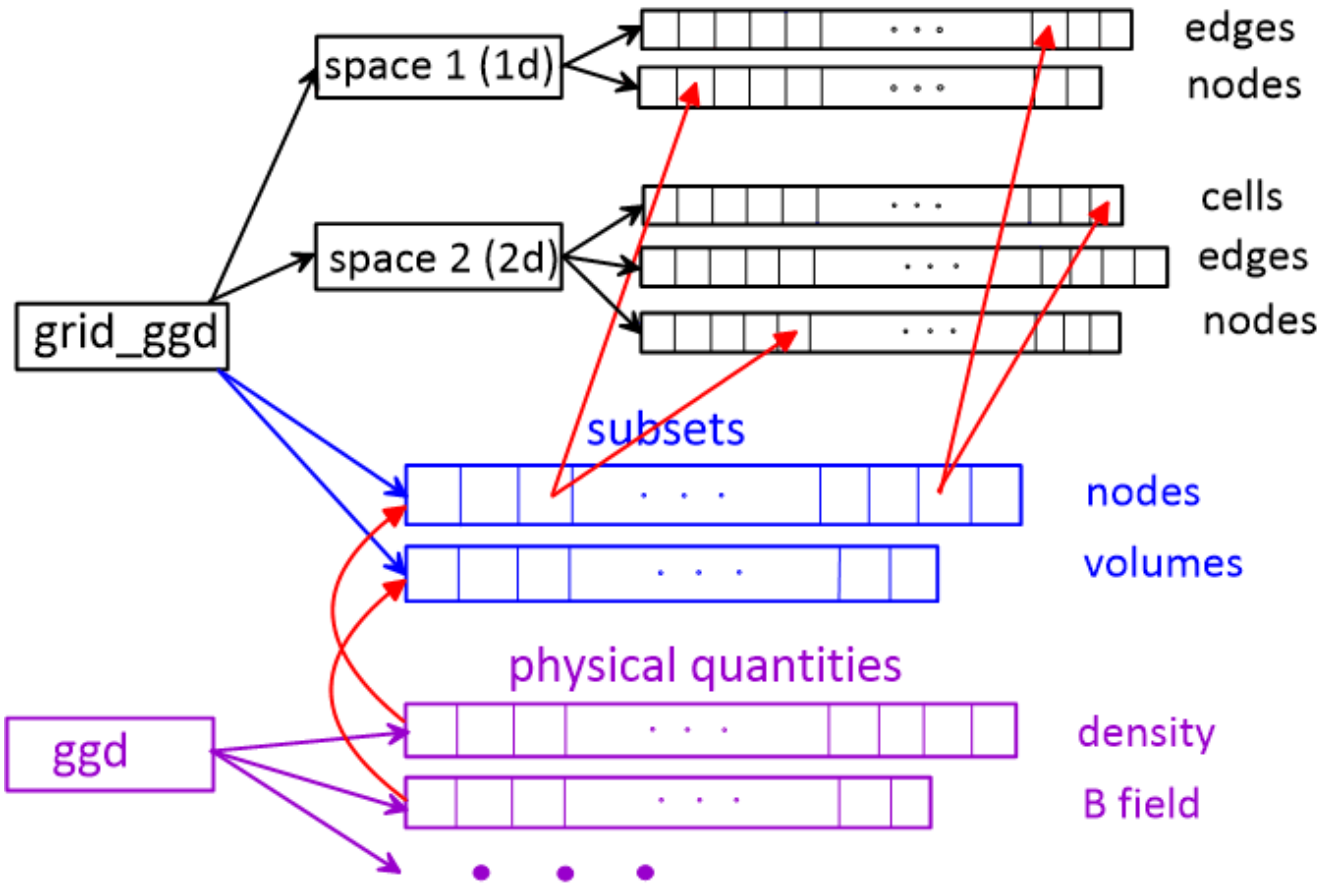
- Fort.30 – B2.5 grid data
  - Mesh cell size, numer of cuts and cuts index
  - R,Z coordinates of 4 vertexes of cel for each cell
- Fort.31 – B2.5 plasma background data
  - 23 variables of plasma background for each cell: ion density, poloidal, radial and toroidal velocities, electron and ion temperature, etc.
- Fort.33 – EIRENE grid nodes
- Fort.34 – EIRENE grid triangles
- Fort.35 – EIRENE grid triangle neigthbours
- Input.eir – Set of parameters for models and boundaries

General Grid Structure

GGD

- a data structure for storing grids of arbitrary structure, together with physical quantities given on these grids
- each IDS contains the structures *grid_ggd* and *ggd*
  - *grid_ggd* – holds description of grid elements (nodes, cells, etc.) and relations between them
  - *ggd* – quantities on this grid
- each physical quantity is “attached” to one or more grid subsets (for example, the subset of all grid points, of all grid cells, of SOL grid points, of separatrix edges etc.).
- Specification of GGD can be found in the Data Dictionary of most IMAS IDS (type *dd_doc* command).

SOLPS-ITER and EIRENE interaction

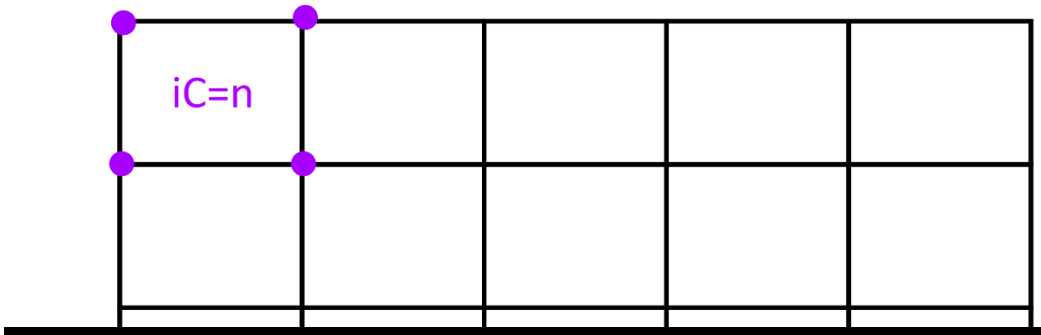


Current interface structure

- Interface read data from fort.33, fort.34, fort.35 files, put it to the GGD and save it in IMAS database

Current interface structure

- Interface read data from fort.33, fort.34, fort.35 files, put it to the GGD and save it in IMAS database
- Interface also read coordinated of vertex for each cell from fort.30 and save nodes (without dubplication of node coordinates)

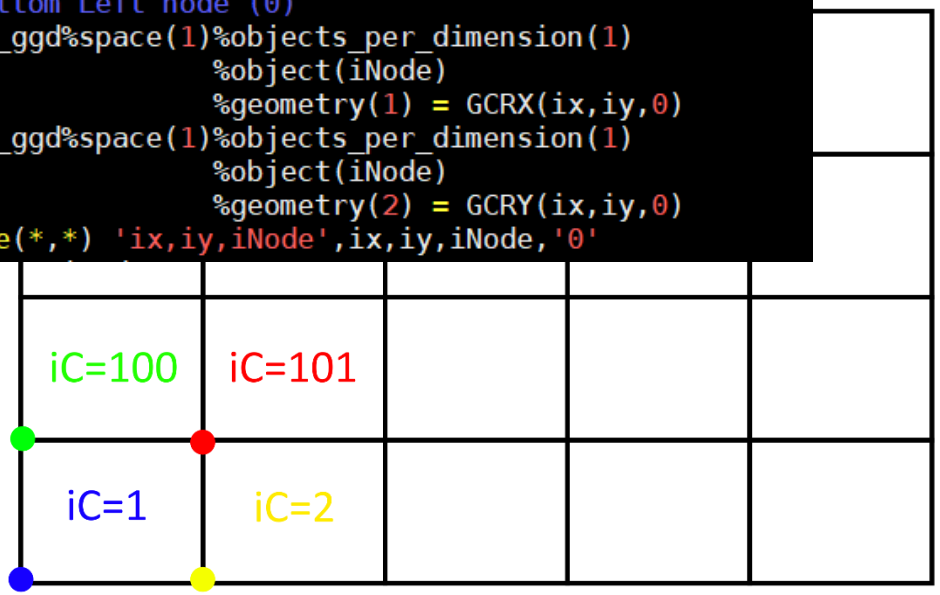



```
      INX=0
      DO ix=1,NX
       DO iy=1,NY
        INX=INX+1
        READ (30,*) CRX(ix,iy,1),CRX(ix,iy,3),CRX(ix,iy,2),CRX(ix,iy,0)
        READ (30,*) CRY(ix,iy,1),CRY(ix,iy,3),CRY(ix,iy,2),CRY(ix,iy,0)
!!        write(*,*) CRY(ix,iy,1),CRY(ix,iy,3),CRY(ix,iy,2),CRY(ix,iy,0)
       ENDDO
      ENDDO
```

Current interface structure

- Interface read data from fort.33, fort.34, fort.35 files, put it to the GGD and save it in IMAS database
- Interface also read coordinated of vertex for each cell from fort.30 and save nodes (without dubplication of node coordinates)

```
      allocate(grid_ggd%space(1)%objects_per_dimension(1)
     &                    %object(iNode)%geometry(2))
      ! Bottom Left node (0)
      grid_ggd%space(1)%objects_per_dimension(1)
     &                    %object(iNode)
     &                    %geometry(1) = GCRX(ix,iy,0)
      grid_ggd%space(1)%objects_per_dimension(1)
     &                    %object(iNode)
     &                    %geometry(2) = GCRY(ix,iy,0)
      write(*,*) 'ix,iy,iNode',ix,iy,iNode,'0'
```



Current interface structure

- Interface read data from fort.33, fort.34, fort.35 files, put it to the GGD and save it in IMAS database
- Interface also read coordinated of vertex for each cell from fort.30 and save nodes (without dubplication of node coordinates)
- Based on nodes object, cells have been created in the GGD:
  - *grid_ggd%space(1)%objects_per_dimension(1)%object(iNode)*
  - *grid_ggd%space(1)%objects_per_dimension(3)%object(iCell)*
- For now the single subset have been defined which contains whole simulation domain

Current interface structure

- Interface read data from fort.33, fort.34, fort.35 files, put it to the GGD and save it in IMAS database
- Interface also read coordinated of vertex for each cell from fort.30 and save nodes (without dubplication of node coordinates)
- Based on nodes object, cells have been created in the GGD:
  - grid_ggd%space(1)%objects_per_dimension(1)%object(iNode)
  - grid_ggd%space(1)%objects_per_dimension(3)%object(iCell)
- For now the single subset have been defined which contains whole simulation domain
- The goal is to devide on several subsets according to EIRENE needs (?)

Current interface structure

- Interface read data from fort.33, fort.34, fort.35 files, put it to the GGD and save it in IMAS database
- Interface also read coordinated of vertex for each cell from fort.30 and save nodes (without dubplication of node coordinates)
- Based on nodes object, cells have been created in the GGD:
  - grid_ggd%space(1)%objects_per_dimension(1)%object(iNode)
  - grid_ggd%space(1)%objects_per_dimension(3)%object(iCell)
- For now the single subset have been defined which contains whole simulation domain
- The goal is to devide on several subsets according to EIRENE needs (?)
- Interface reads also all variables of plasma background from the fort.31 file and as a test, put electron temperature data to defined subset

Possibilities of development

Ongoing work

- Adding all variables of plasma backgrouns to IDS
- Test script for puting and getting data from IMAS data base at Gateway Ef HPC

Questions?

- What are EIRENE requirements/needs for subset division of data?
- Which variables of plasma background from the fort.31 file are needed?
- Is there a need of recovering fort.30 file or fort.31 now? (in the future no)
- What kind of data from input.eir are needed?

Possibilities of development

Thank you for attension

Notes

Notes:

- What are EIRENE requirements/needs for subset division of data?
- Which variables of plasma background from the fort.31 file are needed
- Is there a need of recovering fort.30 file or fort.31 now? (in the future no)
- What kind of data from input.eir are needed?