

Variable Grouping for IMASification

TSVV-5 VC

J. Gonzalez; 04-02-2022



Variable grouping



Why we want to group variables in Eirene

- **Clearer code:** when calling a procedure with multiple related variables, only the main variable will be passed.
- It helps to **identify variables** when reading and debugging code. Giving a structure to variables and “linking” them provides a helpful way to identify them.
- Better way to store and access arrays. For example:

```
X(1:nodes), Y(1:nodes), Z(1:nodes) => node(1:nodes)%X, node(1:nodes)%Y, node(1:nodes)%Z
```

- Code is **easier to modify**, specially with an OOP philosophy in mind. It is easier to expand a TYPE with a related variable than to add a new variable, modify interfaces, calls, modules...
- It could provide guidance to manage **Eirene input**. The way the input variables are structured could be rewritten in TYPES, helping with default values and also having an equivalence with new JSON format.



Previous experiences: Tallies for ASCII and HDF5 outputs



Simple OOP for HDF5 output in Eirene

- Currently, a simple implementation of OOP is used to deal with the output of tallies in ASCII and HDF5 formats.
- Eirene has different tallies: Input, Volume Averaged (Output) and Surface Averaged (Output).
- Each tally has different units and dimension and they are written in a different way.
- New abstract type for tallies, extended for each tally type.
- Each type has information about name, units, id and pointers to the data (same structure as before regarding data management).
- Each tally type has subroutines to write its own information in ASCII or HDF5 formats.
- Reduction of **IF** and **SELECT CASE** clauses.
- Much clearer code.
- Additional improvements could be done, but require a deeper modification of Eirene.
- **HOWEVER:** Current issue with ABSTRACT and algorithmic differentiation, not recommended.



Examples of Code

```
26 TYPE, ABSTRACT :: tally
27 !id: Unique identification for tally
28 INTEGER:: id = 0
29 !name: Description of tally
30 !units: Units of the tally
31 CHARACTER(60):: name='FREEXX', units=' ---'
32 !active: indicates if the tally is active
33 LOGICAL, POINTER:: active => NULL()
34 CONTAINS
35 !Initialize an tally
36 PROCEDURE(initialize_interface), DEFERRED, PASS:: initialize
37 !Write the Tally as ASCII format
38 PROCEDURE(writeASCII_interface), DEFERRED, PASS:: writeASCII
39 !Write the Tally as HDF5 format
40 PROCEDURE(writeHDF5_interface), DEFERRED, PASS:: writeHDF5
41
42 END TYPE tally
```

Fig 2. Generic type for tallies.

```
109 !Input tally with 1D data
110 TYPE, EXTENDS(tallyInput):: tallyInput1D
111 !Data of tally
112 REAL(DP), POINTER:: data(:)
113 CONTAINS
114 PROCEDURE, PASS :: integrate => integrate1D
115
116 END TYPE tallyInput1D
```

Fig 4. Input tally for 1D data.

```
77 !Extension for input tally
78 TYPE, ABSTRACT, EXTENDS(tally):: tallyInput
79 !Type of header
80 INTEGER:: type = 0
81 CONTAINS
82 PROCEDURE, PASS:: initialize => initInputTally
83 PROCEDURE, PASS:: writeASCII => writeInputASCII
84 PROCEDURE, PASS:: writeHDF5 => writeInputHDF5
85 !Weighting of the tally
86 PROCEDURE, PASS:: weighting => weightingInput
87 !Calculates the average value of the Tally
88 PROCEDURE, NOPASS:: average => averageInput
89 !Integrate tally. Each extension needs to define its own integration
90 PROCEDURE(integrate_interface), DEFERRED, PASS:: integrate
91
92 END TYPE tallyInput
```

Fig 3. Extension for input tallies.

```
118 !Input tally with 2D data
119 TYPE, EXTENDS(tallyInput):: tallyInput2D
120 !firstDimension: first dimension of the data array (firstdimension, number of cells)
121 INTEGER:: firstDimension=1
122 !Data of tally
123 REAL(DP), POINTER:: data(:, :)
124 CONTAINS
125 PROCEDURE, PASS :: integrate => integrate2D
126
127 END TYPE tallyInput2D
```

Fig 5. Input tally for 2D data.



Grouping for IMAS



Geometry

- Generic element to be used in the near future by Eirene.
- Structure could be organized in a similar way than GGD.



Output (tallies)

- Group velocity tallies (remember Eirene is 3D Cartesian, change to IMAS).
- Storing relevant metadata and data for a tally together.





**Thank you for your
attention**



J. Gonzalez | TSW 2022



DIFFER

Title

Title

- Text

