Proposal for a generalized cell structure in the EIRENE code

Petra Börner

08/12/2022

General Considerations

- During the Code Camp in Nov. 2021 it was decided that a generalized mesh type should be implemented in the Eirene code which should in the long run replace all existing geometry types except for the user defined geometry.
- The aim is to simplify the handling of the geometry during the computation of trajectories and to avoid branching in this regard.
- The new mesh type should support easy access to GGDs.
- As triangle and tetrahedron meshes shall be covered by the new type it needs to be unstructured.
- The geometry needs to operate on cells rather than grid surfaces.
- Particles will travel through one cell only as is already done in FOLION.

Concept

How to define a cell?

A cell can be defined by

- Edges
- Faces
- Toroidal planes

Concept (continued)

Edges

- are straight lines defined by two vertices or
- curved lines defined by algebraic equations

Faces

- are plane surfaces limited by edges
- Triangle or quadrangle

Toroidal planes

- For toroidal approximation
- Defined by toroidal angle

Concept (continued)

- A mesh can have either 2D or 3D cells, no mixture
- Vertices and edges are specified counter-clock wise.
- Cells shall be convex.
- For the time being 3D cells should be limited to cubes and tetrahedra. The concept can be extended if needs arise.
- Each cell should know its neighbour cells (1 on 1 neighbouring) and reflection models
- Each cell needs to know its “center of mass”

Implementation

- How to proceed:
- introduce new option as LEVGEO=20
- set up geometry as has been done up to now
- convert geometry into mesh of general cells
- implement new subroutines for intersections with edges / faces
  - can be extracted from old routines,
  - define unit tests?
- Check the code for references to radial, poloidal, toroidal surfaces

Implementation (continued)

- check sampling of particle birth points (surface sources, step functions)
- Within a run after the conversion to the generalized mesh one should be able to remove the old grid information (deallocate the respective modules)
- extend plotting of geometry for new cell type

Data Types

```
type t_vertex
   real(dp) :: x, y, z
end type t_vertex
```

```
type t_direction
   real(dp) :: dx, dy, dz
end type t_direction
```

```
type t_edge
   Integer :: vertex_numbers(2)
   type(t_direction) :: direction
   type(t_direction) :: normal
   real(dp) :: equation(0:3)
   Integer :: surface_number
end type t_edge
```

```
type t_face
   Integer :: no_vertices, no_edges
   Integer, allocatable :: vertices
   Integer, allocatable :: edges
   type(t_direction) :: normal
   Integer :: surface_number
end type t_edge
```

Data Types

```
type t_cell
   Integer :: no_edges, no_linear_edges,
              no_curved_edges, no_vertices
   Integer :: no_tor_planes
   Integer :: no_faces
   Integer, allocatable :: vertices(:)
   Integer, allocatable :: linear_edges(:)
   Integer, allocatable :: curved_edges(:)
   Integer, allocatable :: faces(:)
   Integer, allocatable :: neighbors(:,:)
  Type(t_vertex) :: center_of_mass
end type t_cell
```

Comments please