PLOT_ORIENTEDMAP - A PROGRAM FOR PLOTTING ORIENTED MAPS

MATHIEU DUTOUR SIKIRIĆ

Abstract. This program allows to plot oriented maps on the sphere or torus and make adequate graphical representations. The map may have 1-gon or 2-gons.

1. Designs principles

(1) **Namelist**: Namelists are used by many other oceanographic and meteorological programs (WWM, COSMO, WAM, etc.) and so the same file format is used by the C++ programs.
(2) **Use SVG for representing the graphics**: .svg files are accessible on web browser and can be opened by the inkscape program.
(3) **Graphs are represented by oriented edges**: This is a general format that is the most practical for representing oriented maps

2. Namelist format

The input of the programs is done in namelist files that contains the information used. They are formed of blocks of the following kinds.

```
&PROC
 MODELNAME = "WWM"
 GridFile = "hgrid.gr3",
 HisPrefix = "WWM_output_",
/
```

The namelists are done according to following principles:

(1) The beginning of a block is `&PROC` and the ending is `/`.
(2) If a variable is not defined in a block then the default value is used.
(3) A variable cannot be defined two times.
(4) The blocks and variables can be ordered in anyway the users want. A block may be completely absent in which case all its variables are the default ones.

(5) Everything after a ! is considered as comment and not read.
(6) Strings can be delimited by ¨ or ’ or nothing, but endings cannot be mixed.
(7) When inconsistencies are detected, clear error messages are printed.

3. Graph representation

The graphs are represented implicitly by directed edges. For example if a graph is represented by

```
36
 3 6 9 0 12 15 1 18 21 2 24 27 4 22 20 5 28 26 7 30 14 8 13 33 10 31 17 11 16 3
 1 2 0 4 5 3 7 8 6 10 11 9 13 14 12 16 17 15 19 20 18 22 23 21 25 26 24 28 29 2
```

The number 36 is the number of directed edges.

The first line lists the inverses of the directed edge, the second line the next directed edge. For example directed edge numbered 0 has inverse 3 and next directed edge 1. We represent the corresponding permutations by σ_i and σ_n.

From the directed edges and the permutation we can reconstruct everything:

(1) The edges of the graph correspond to the orbits of the group generated by σ_i. For example $\{0, 3\}$ represent one edge of the graph.
(2) The vertices of the graph correspond to the orbits of the group generated by σ_n. For example $\sigma_n(0) = 1$, $\sigma_n(1) = 2$ and $\sigma_n(2) = 0$ so $\{0, 1, 2\}$ represent a vertex of the graph.
(3) The faces of the graph correspond to the orbits of the group generated by the product permutation $\sigma_i\sigma_n$.

This is a more complicated format than a list of clockwise adjacencies but also more powerful: we can represent 1-gons, 2-gons, without any ambiguity. And converting to that format is very easy.

4. Input file

The input file has the following sections in it: PLOT, EDGE, VERT, TORUS.

Let’s see the sections in turn:

```
&PLOT
 PlaneFile = "PLOT_1_PL_1_p31m_1_-_1_p3.plane",
 OutFile = "PLOT_1_PL_1_p31m_1_-_1_p3.svg",
 ViewFile = "PLOT_2_PL_6_Oh_2_-_1_S6.view",
 MAX_ITER_PrimalDual = -1,
```

```
 MAX_ITER_CaGe = 1000,
 CaGeProcessPolicy = 2,
 RoundMethod = 2,
 width = 600,
 height = 600,
 MethodInsert = 2,
 ListExportFormat = "eps",
/
```

Signification is following:

(1) `PlaneFile` (string): filename of the file containing the oriented map.
(2) `OutFile` (string): filename of the output file.
(3) `ViewFile` (string): file of viewfile precising the exterior face. It is needed for planar graph but not for toroidal maps.
(4) `MAX_ITER_PrimalDual` (integer): maximum number of iteration of primal dual scheme. Put -1 if leaving only after convergence.
(5) `MAX_ITER_CaGe` (integer): maximum number of iteration of the CaGe scheme.
(6) `CaGeProcessPolicy` (integer): policy when activating the CaGe process for toroidal maps. Best to leave it to default value of 2.
(7) `RoundMethod` (integer): rounding method used for representing coordinate in .svg file. Leave it at 2.
(8) `width/height` (double): width and height in the .svg file. Default value is 600.
(9) `MethodInsert` (integer): parameter for selecting the list of representing point. Technical. Leave it at 2.
(10) `ListExportFormat` (list of strings): list of selected export formats. Can be `eps`, `png` and `pdf`. This uses inkscape which of course needs to be installed.

Section `EDGE`. Example:

```
&EDGE
DoMethod1 = .T.,
DoMethod2 = .F.,
DoMethod3 = .F.,
MultTangent = 0.5,
NormalTraitSize = 1,
ListTraitIDE = 0,3,2,9,4,12,5,15,10,24,13,22,14,20,17,26,19,30,23,33,25,31,32,3
ListTraitGroup = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
ListTraitSize = 10,
DefaultRGB = 0,0,0,
SpecificRGB_iDE = ,
```

```
SpecificRGB_Group = ,
SpecificRGB_R = ,
SpecificRGB_G = ,
SpecificRGB_B = ,
/
```

Signification is following:

(1) `DoMethod1/DoMethod2/DoMethod3` (logical): best is to leave it at values of T/F/F.
(2) `MultTangent` (double): best is to leave at 0.5
(3) `NormalTraitSize` (double): trait size of normal edge
(4) `DefaultRGB` (list of integer): list of integer values specifying the default color.

Edges can have different colors and also different trait size. This is handled by groups. In the above example, `ListTraitSize` specifies that we have only one group for which the trait is 10. `ListTraitIDE` specifies the list of directed edge which belongs to specific edges. `ListTraitGroup` specifies the group to which belong those special directed edges.

Same system is done for the color except that in above example, nothing is specified.

Section `VERT`. It is same as edge except for vertices. Example of section:

```
&VERT
 NormalRadius = 0.010,
 ListRadiusIDE = ,
 ListRadiusGroup = ,
 ListRadius = ,
 DefaultRGB = 0,0,0,
 SpecificRGB_iDE = ,
 SpecificRGB_Group = ,
 SpecificRGB_R = ,
 SpecificRGB_G = ,
 SpecificRGB_B = ,
/
```

This section specifies the radius of the disk around the vertices and the color. Idea are similar as for edges, that is it works by directed edges.

Section `TORUS`. Example:

```
&TORUS
 minimal = 1e-11,
 tol = 1e-05,
```

```
 AngDeg = 0,
 scal = 0.5,
 shiftX = 0,
 shiftY = 0,
 FundamentalRGB = 255,0,0,
 FundamentalTraitSize = 2,
 DrawFundamentalDomain = .T.,
/
```

Those defaults are mostly not to be changed. Remarks:

(1) `minimal/tol` (double): the threshold values used in the primal dual process. If the program does not converge, increase tol and or decrease minimal. Basically, minimal is the square of tol.
(2) `scal` (double): the scale factor for representing the map. A factor of 0.5 means that two fundamental domains are represented which is most often adequate. If it becomes crowded, use a large value like 0.9.
(3) `DrawFundamentalDomain` (logical): sometimes, we want to represent the fundamental domain. If true then the trait size and color can be chosen accordingly.

Mathieu Dutour Sikirić, Rudjer Bosković Institute, Bijenicka 54, 10000 Zagreb, Croatia, Fax: +385-1-468-0245
E-mail address: mathieu.dutour@gmail.com