# An ordering package for 



- Jean-Paul Boufflet, Christophe Denis
  Laboratory Heudiasyc UMR CNRS
  University of Technology of Compiègne

- Piotr Breitkopf, Michel Vayssade
  Laboratory Roberval UMR CNRS
  University of Technology of Compiègne

# Outline

- Different criteria : fill-in, bandwidth, $F_{max}$
- User chooses a method according to the criterion
- Efficiency of an ordering method tightly related to :
  - type of solver used
  - matrix storage (skyline, sparse)
  - memory management
  - implementation
  - the problem

# Two approches :

1. An ordering scheme tightly coupled with the solver
2. a set of general purpose ordering tools

- The first approach is more efficient but should be developped independently for each solver
- The second approach is complementary and permits experimentation with various algoritms/solvers

# For example :

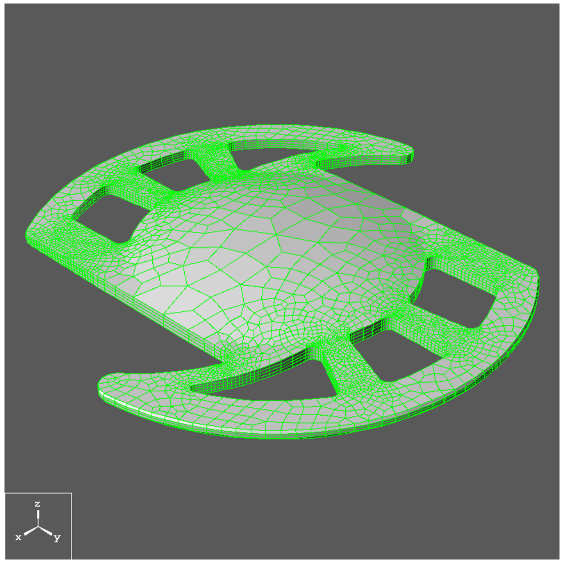


Figure 1: **The mesh of an ocular implant of 14648 finite elements and 19705 nodes.**

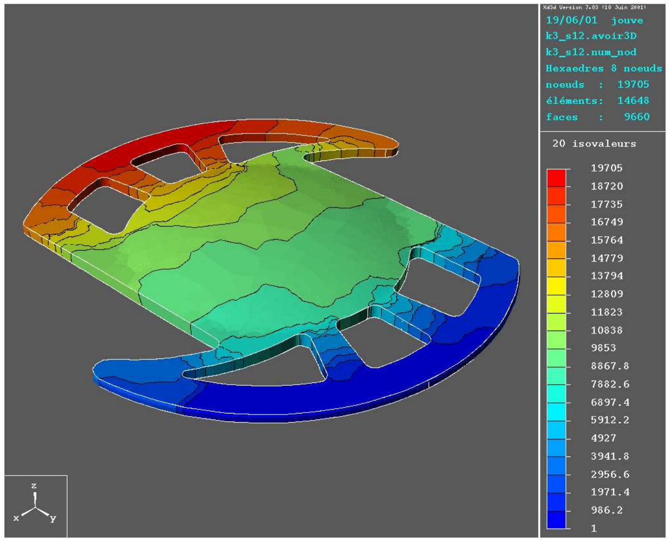


Figure 2: **Layer coloring of nodes of the mesh of figure 1.**

In this case, by applying a modified version of the Sloan method, the user gains 30% in CPU and 15% in storage as compared to the initial GPS ordering used.

## Ordering methods at our disposal (C code):

- E. Cuthill and J. McKee;
- reverse E. Cuthill and J. McKee;
- GPS of N. E. Gibbs, W.G. Poole Jr and P. K. Stockmeyer;
- generalized nested dissection of J. A. George and J. W. Liu;
- S. W. Sloan;
- S. W. Sloan with modifications (changing the ending and starting node choice strategies);
- I. S. Duff, J. K. Reid and J. A. Scott (for ordering finite element mesh);
- MD, minimum degree algorithm (with the supernodal approach);
- AMD of P. Amestoy, T. Davis and I. Duff (approximate minimum degree ordering);

## Interface of the Sloan method

$[Vna, Van] = sloan(< mesh\_name >, w1, w2);$

- $Vna[i] = j$, giving for the new number of node $i$ its former number $j$;
- $Van[j] = i$, giving for the former number $j$ its new number $i$;
- $< mesh\_name >$ is a scilab string, the filename where data are stored;
- $w1$ weight for the pseudo peripheral criterion;
- $w2$ weight for the current degree criterion.

The file format and the type of input data should be discussed.

- File ? Which format ?
- Matrix ? CSR (Compress Sparse Row) ? Skyline ?
- Filtering codes for Harwell-Boeing, IDEAS, and others ?

# Related works:

- *SIC* (Système Interactif de Conception) installed on a PC-cluster ( sequential and parallel solvers)
- Load balancing techniques based on an "energetic" (i.e. number of flops) criterion
- In progress : a model for estimating the computing time of our multifrontal solver
- Entries of our solver are subdomains issued from various partitioning codes (METIS, JOSTLE, SPLITMESH, ...)
- In progress : design of post-optimisation heuristics to balance the work load over the subdomains.
- In project : parallel version of a grid generator coupled with a solver (with A. Rassineux)