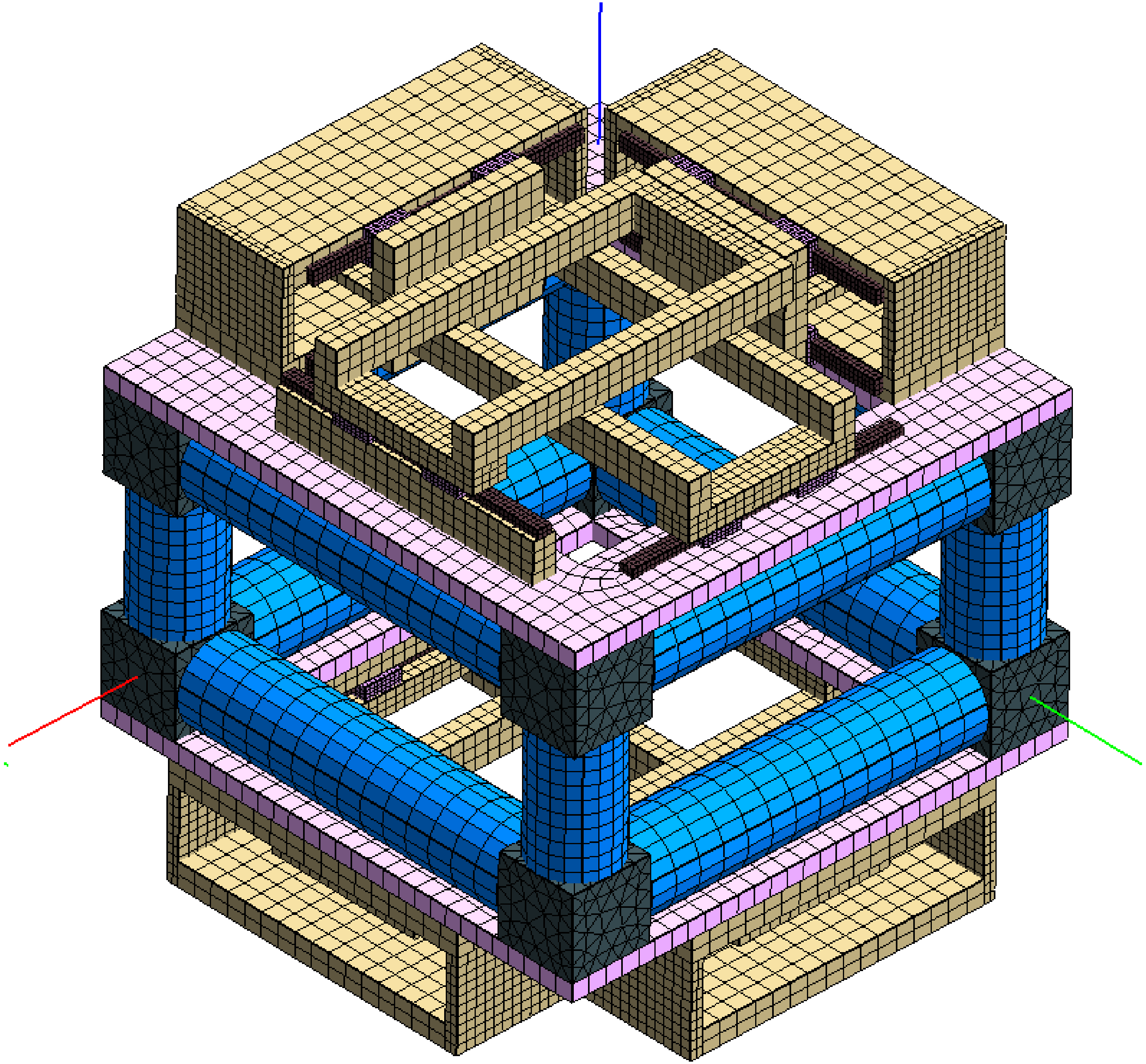
# Using dictionaries to store, manage and visualize 3D data

Alexandru Dadalau, EuroTcl 2015

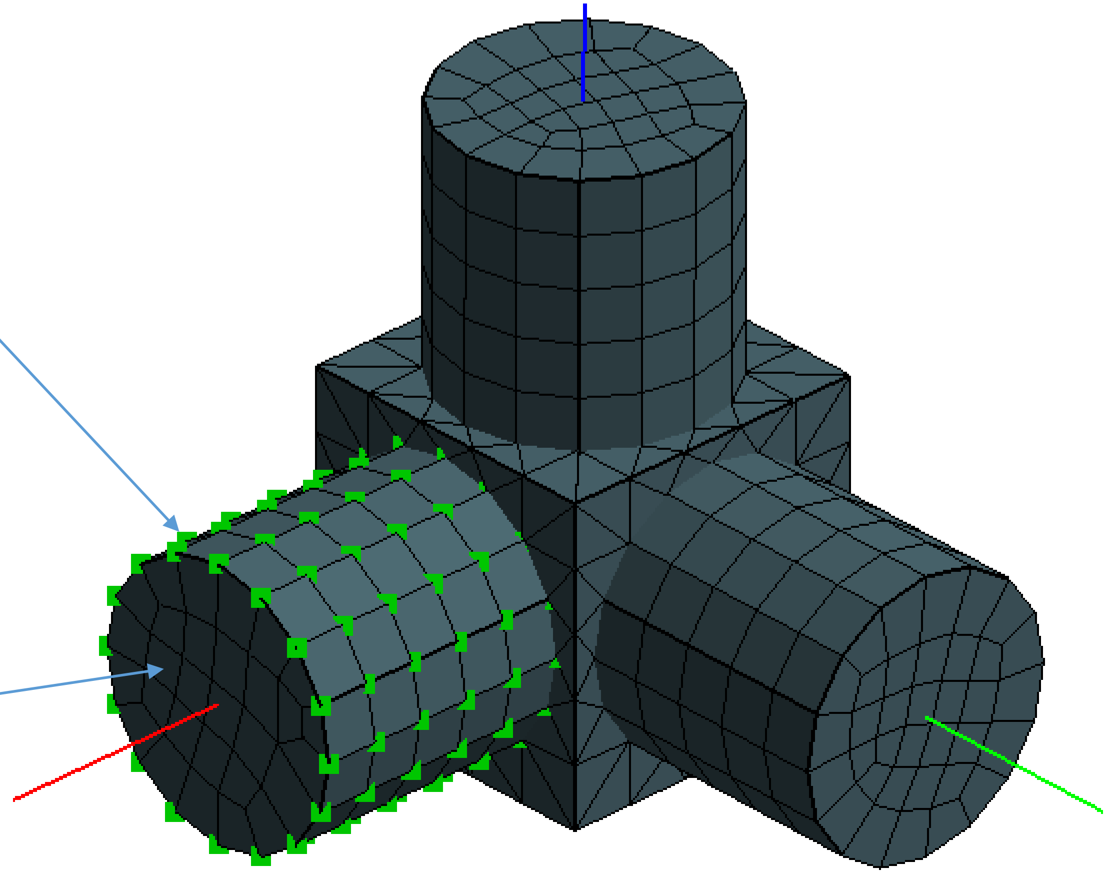


# **What** you will see:

- How I **read FE data** with **Tcl**
- How I use **Tcl** to extract the **surface mesh** from FE data
- How is the **performance** of Tcl **dict** with the 3D FE data

# Example of 3D FE mesh

Finite element model

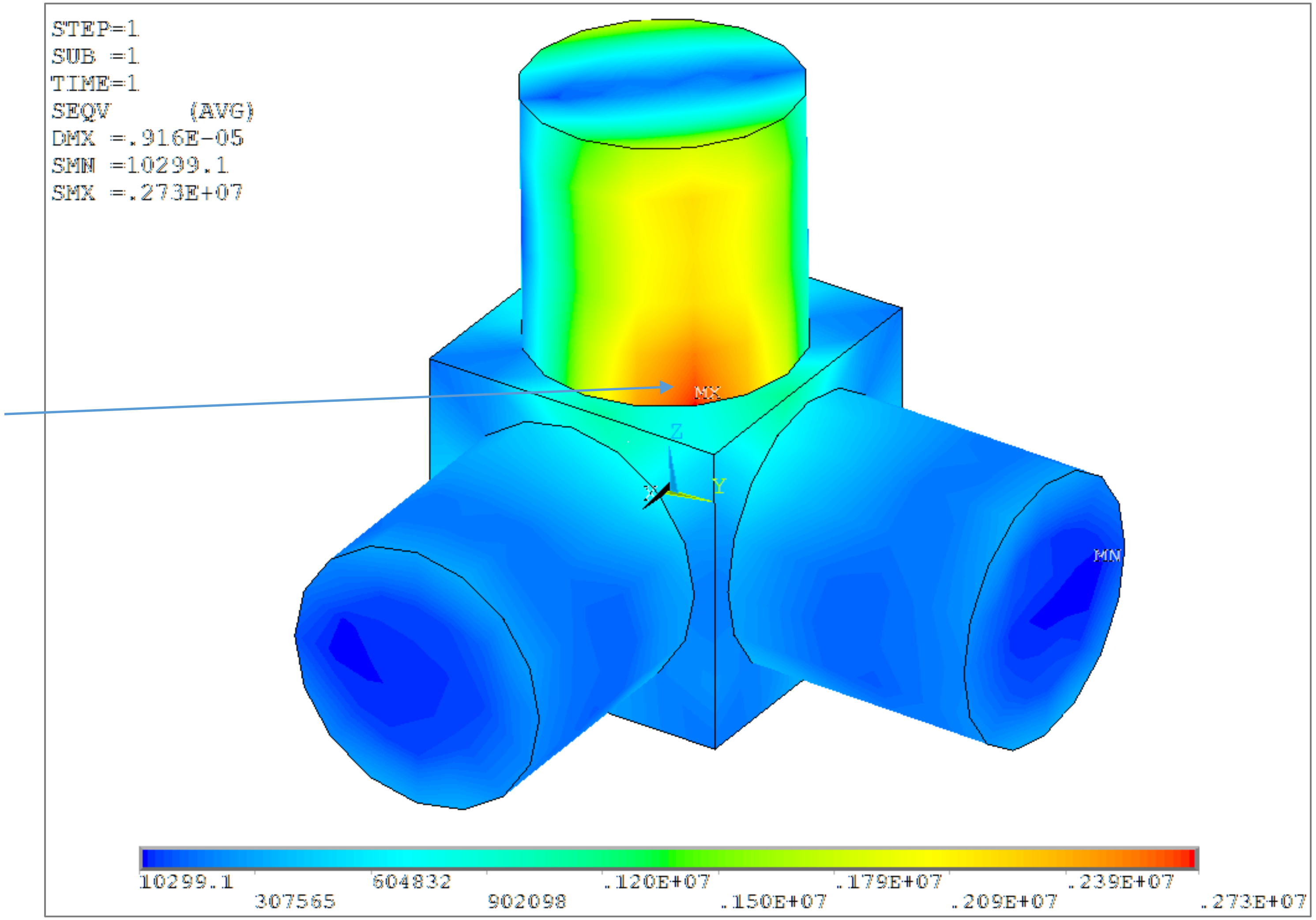


Nodes (x,y,z coordinates)

Elements (nodes + material properties)

# Example of 3D FE results

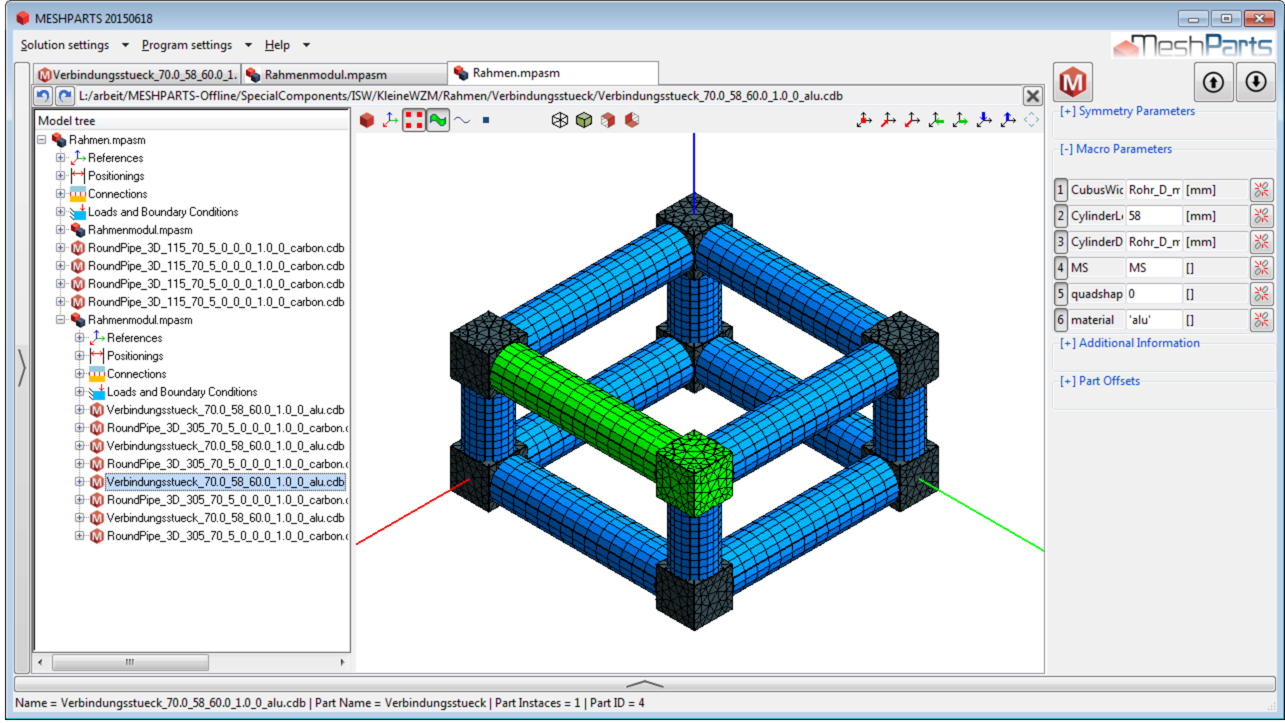
## Material stresses



```
STEP=1
SUB =1
TIME=1
SEQV      (AVG)
DMX =.916E-05
SMN =10299.1
SMX =.273E+07
```

Break point

# What is the role of Tcl/Tk in all this?



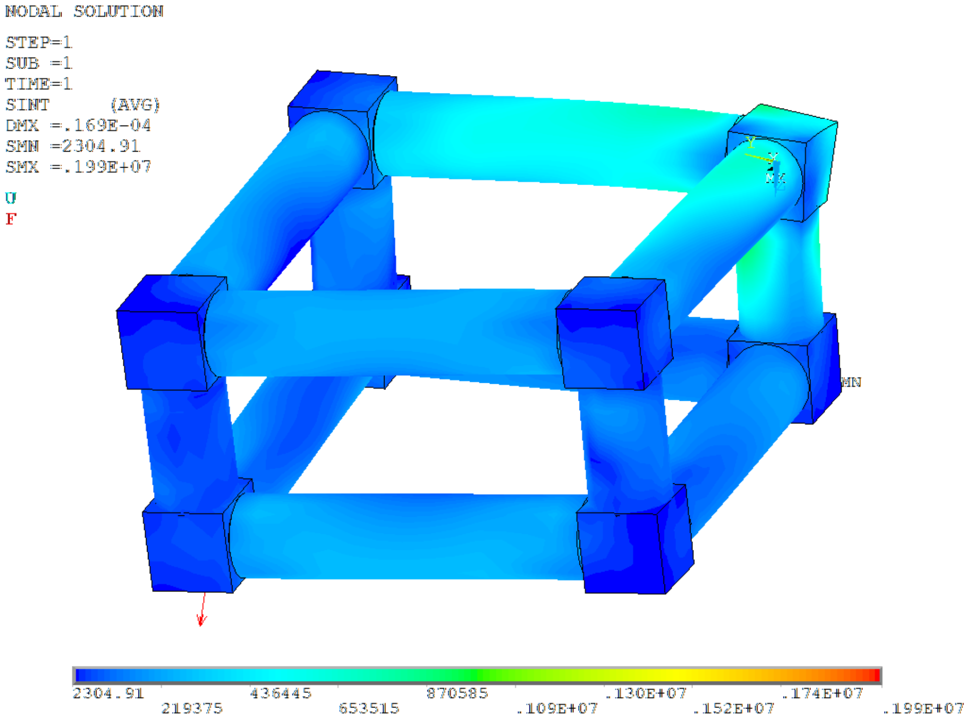
L:/arbeit/MESHPARTS-Offline/SpecialComponents/ISW/KleineWZM/Rahmen/Verbindungsstueck/Verbindungsstueck_70.0_58_60.0_1.0_0_alu.cdb

Model tree

- Rahmen.mpasm
  - References
  - Positionings
  - Connections
  - Loads and Boundary Conditions
  - Rahmenmodul.mpasm
  - RoundPipe_3D_115_70_5_0_0_0_1.0_0_carbon.cdb
  - RoundPipe_3D_115_70_5_0_0_0_1.0_0_carbon.cdb
  - RoundPipe_3D_115_70_5_0_0_0_1.0_0_carbon.cdb
  - RoundPipe_3D_115_70_5_0_0_0_1.0_0_carbon.cdb
  - Rahmenmodul.mpasm
    - References
    - Positionings
    - Connections
    - Loads and Boundary Conditions
    - Verbindungsstueck_70.0_58_60.0_1.0_0_alu.cdb
    - RoundPipe_3D_305_70_5_0_0_0_1.0_0_carbon.c
    - Verbindungsstueck_70.0_58_60.0_1.0_0_alu.cdb
    - RoundPipe_3D_305_70_5_0_0_0_1.0_0_carbon.c
    - Verbindungsstueck_70.0_58_60.0_1.0_0_alu.cdb
    - RoundPipe_3D_305_70_5_0_0_0_1.0_0_carbon.c
    - Verbindungsstueck_70.0_58_60.0_1.0_0_alu.cdb
    - RoundPipe_3D_305_70_5_0_0_0_1.0_0_carbon.c

| | | | |
|---|---|---|---|
| 1 | CubusWic | Rohr_D_m | [mm] |
| 2 | CylinderL | 58 | [mm] |
| 3 | CylinderD | Rohr_D_m | [mm] |
| 4 | MS | MS | [] |
| 5 | quadshap | 0 | [] |
| 6 | material | 'alu' | [] |

Name = Verbindungsstueck_70.0_58_60.0_1.0_0_alu.cdb | Part Name = Verbindungsstueck | Part Instaces = 1 | Part ID = 4



## Create a software that helps you to:

- **read** single FE models,
- **configure** FE assemblies
- **write** input files for the popular FE solvers

# Read FE data

Node IDs

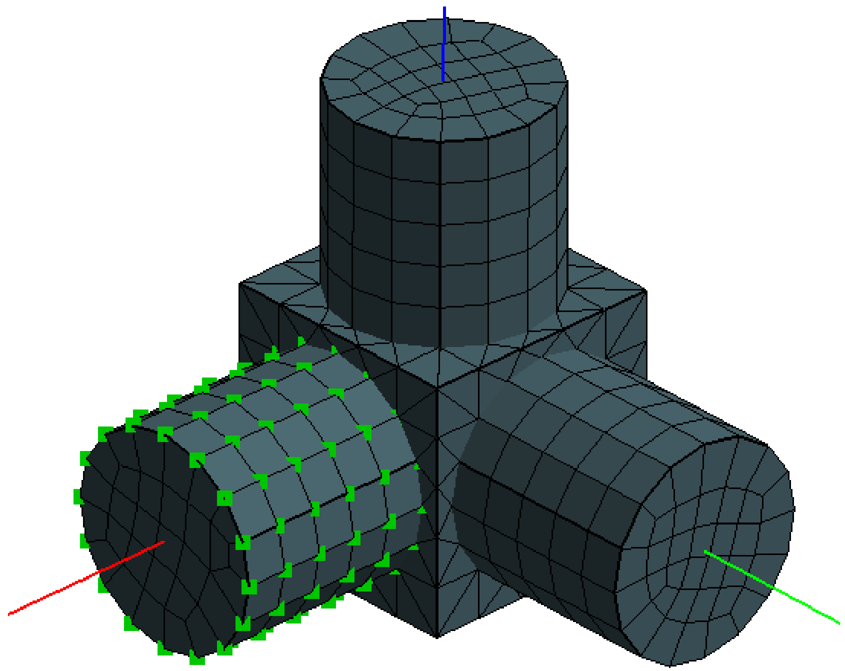
Node coordinates

```
NBLOCK,6,SOLID,      1030,      1030
(3i9,6e20.13)
        1      3.500000000000E-02
        2      0.000000000000E+00 3.500000000000E-02
        3      0.000000000000E+00 0.000000000000E+00 3.500000000000E-02
        4      3.500000000000E-02-9.4797148810905E-03 6.2958884784459E-03
        5      3.500000000000E-02 4.1394885988930E-03-7.7629911498991E-03
        6      3.500000000000E-02-1.5834484717831E-02 1.4929881199078E-02
        7      3.500000000000E-02-8.5376950800066E-03 1.6230790225320E-02
        8      3.500000000000E-02 2.2886530059457E-02 9.1420745366150E-03
        9      3.500000000000E-02 1.6340131003353E-02-1.3586974418736E-03
       10      3.500000000000E-02 7.5699484761175E-03-1.9193690274370E-02

EBLOCK,19,SOLID,      2065,      1950
(19i9)
    1      120      123      207      201      460      448      371      370
    2      460      448      371      370      461      449      378      377
    3      461      449      378      377      462      450      385      384
    4      462      450      385      384      463      451      392      391
    5      463      451      392      391       65       62      399      398
    6      123      124      204      207      448      492      372      371
    7      448      492      372      371      449      493      379      378
    8      449      493      379      378      450      494      386      385
    9      450      494      386      385      451      495      393      392
   10      451      495      393      392       62       61      400      399
   11      124      125      205      204      492      496      373      372
   12      492      496      373      372      493      497      380      379
```

Element IDs

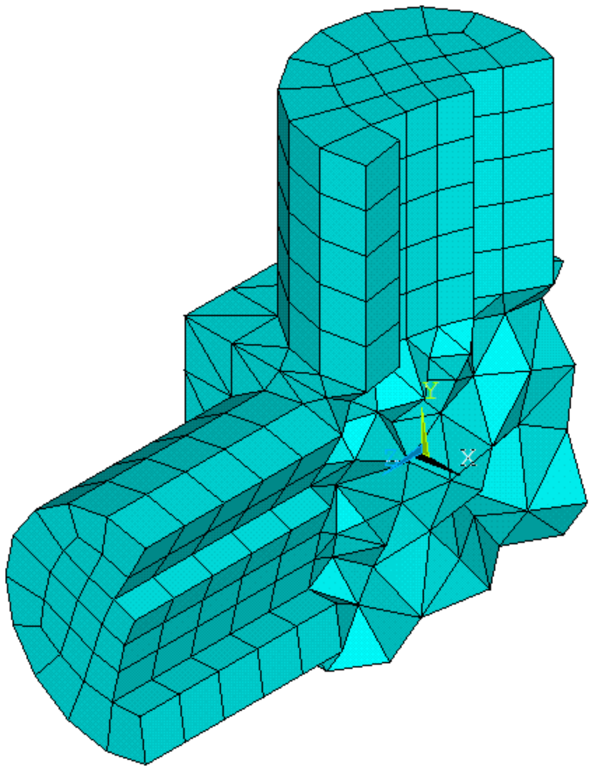
Node IDs

# Show FE data

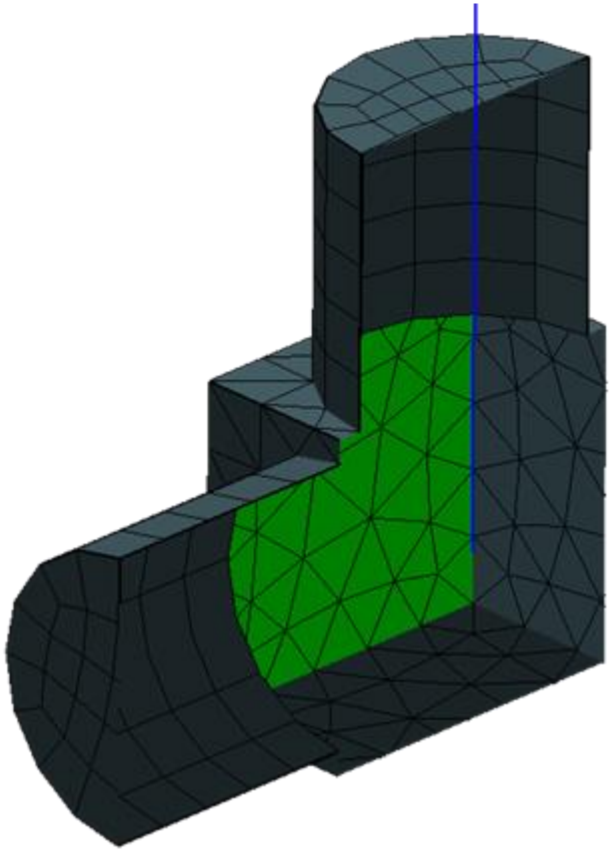


Paul‘s Obermeier Tcl3D package (http://www.tcl3d.org/)

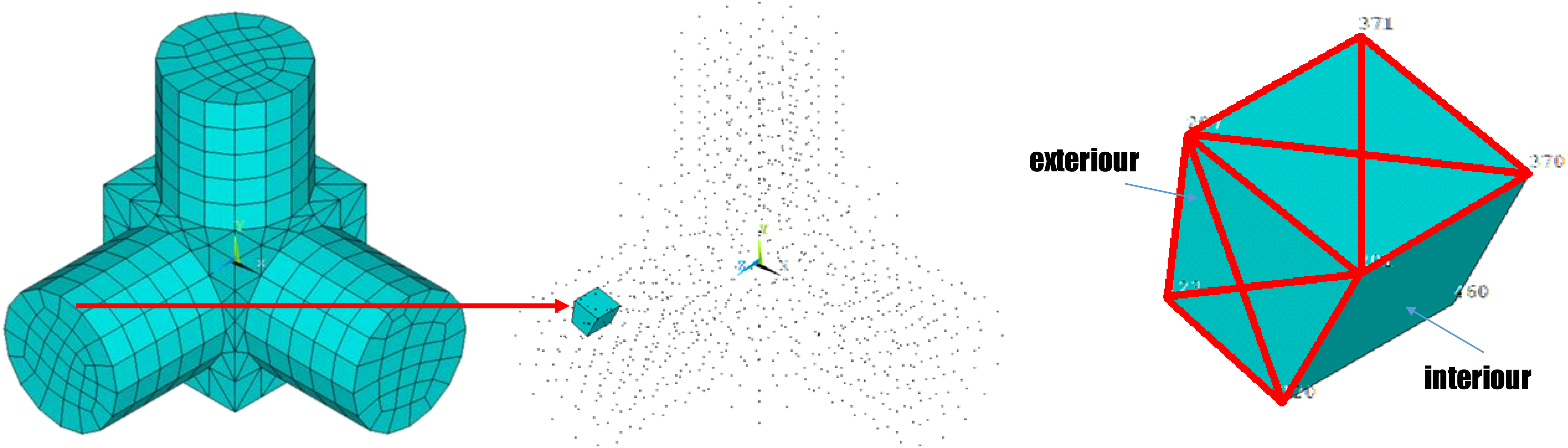
# A matter of **efficiency**



**Available volume mesh**



**Display only surface mesh**



Volume mesh

Isolated element

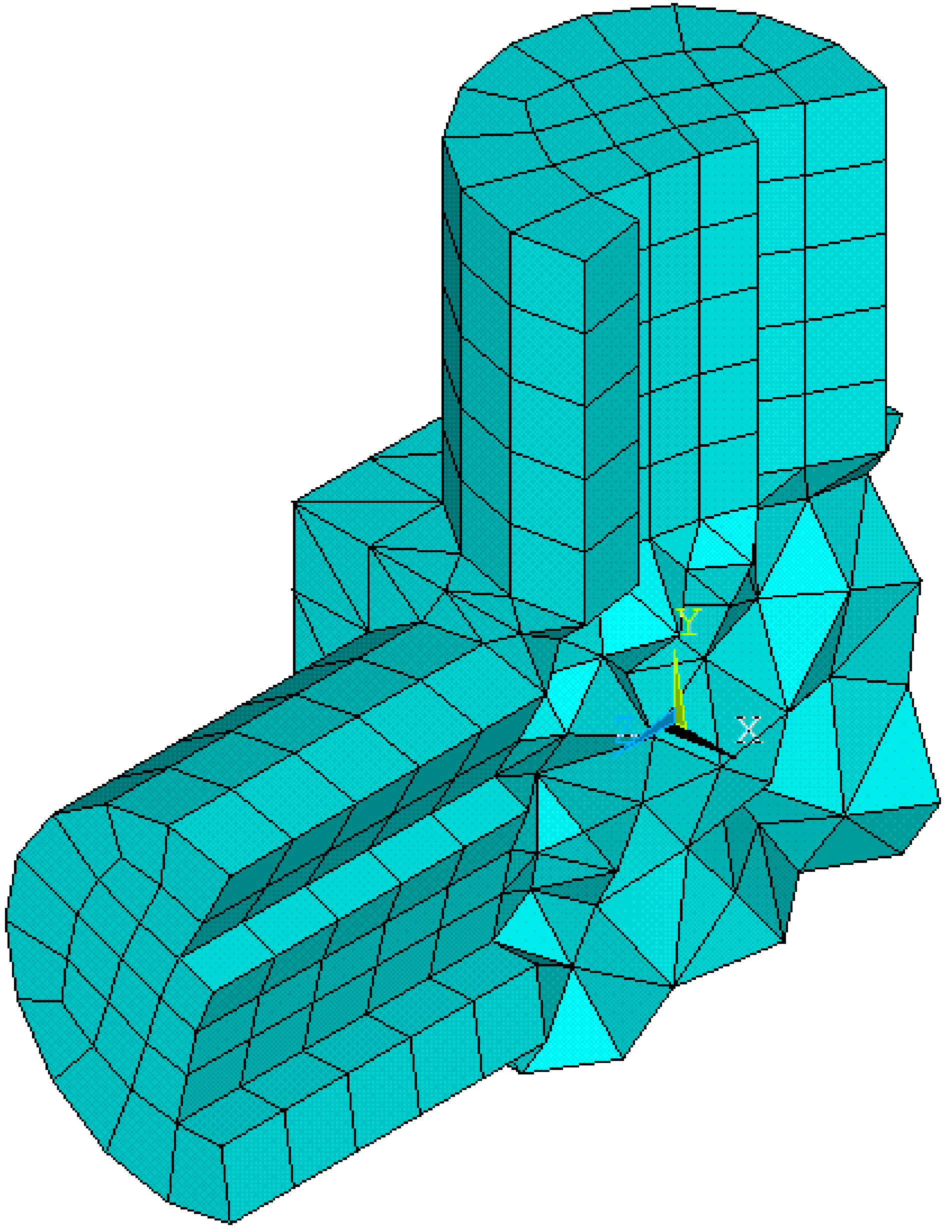
Element faces

**A surface mesh is a collection of exteriour element faces.**

**An element face is exteriour when it belongs to just one element.**

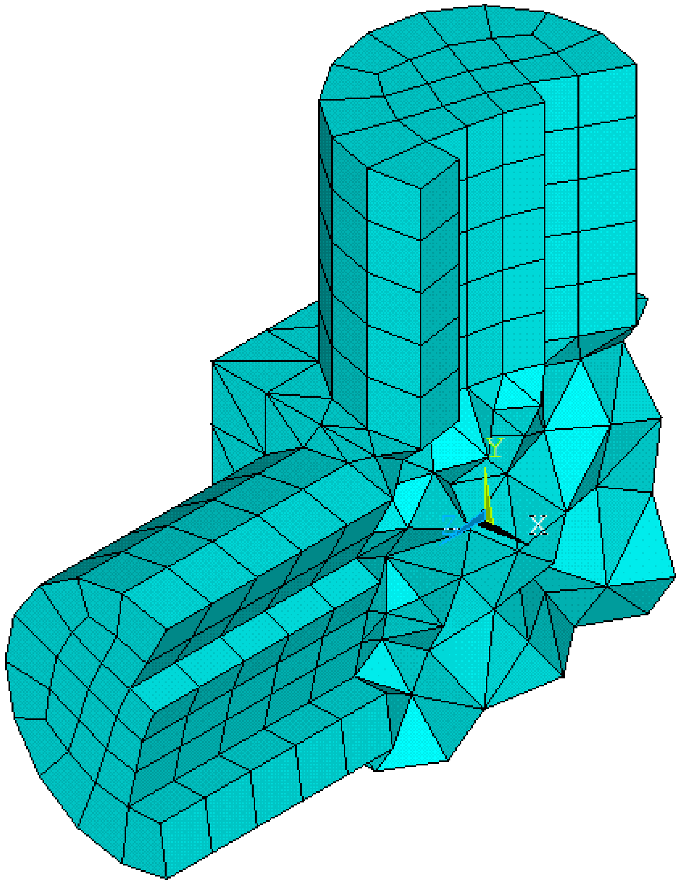
# Extract surface mesh

This is where Tcl **dictionaries** come into play.



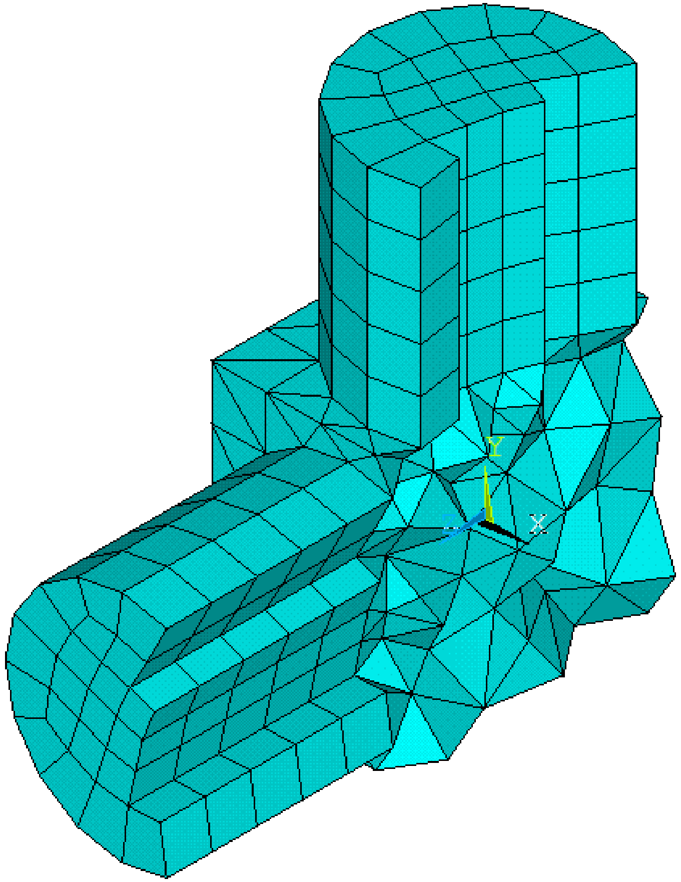
# Naive search of external faces:

```
Loop through all element IDs




End Loop
```



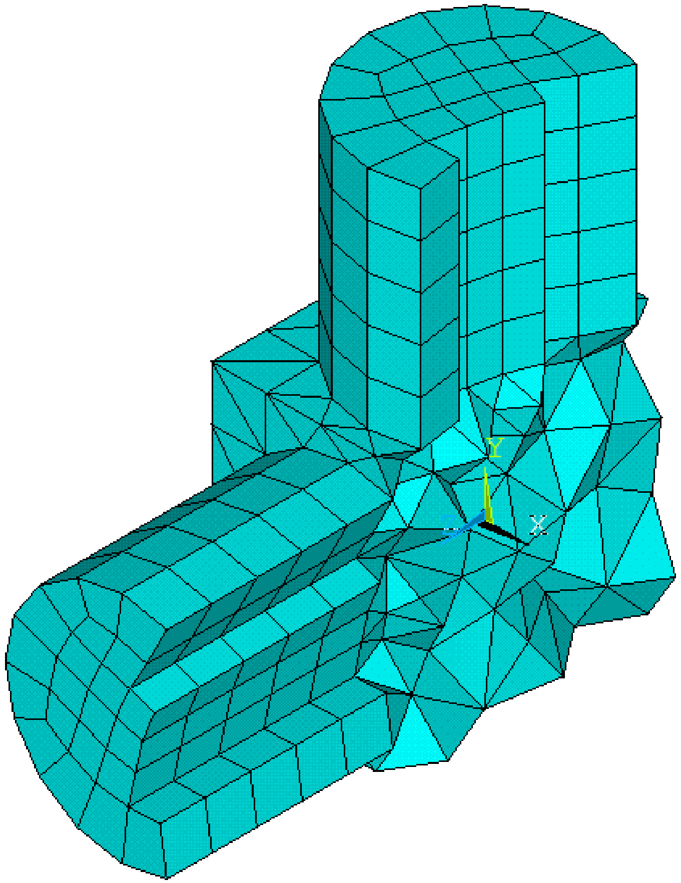
# Naive search of external faces:

```
Loop through all element IDs
   Loop through all element faces






   End Loop
End Loop
```



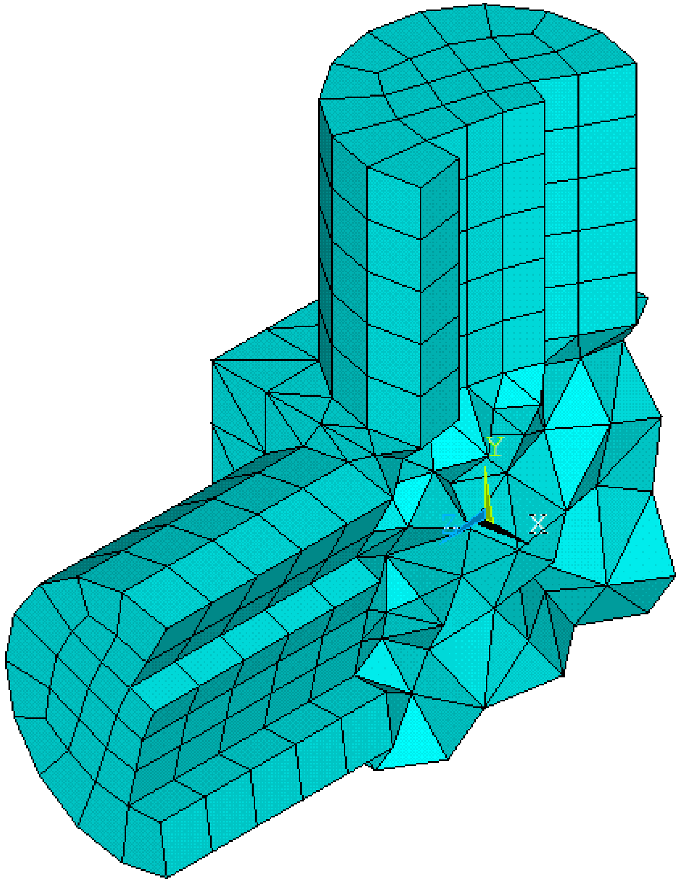
# Naive search of external faces:

```
Loop through all element IDs
   Loop through all element faces
      Get first list of face nodes






      If nothing found, this is an external face
   End Loop
End Loop
```



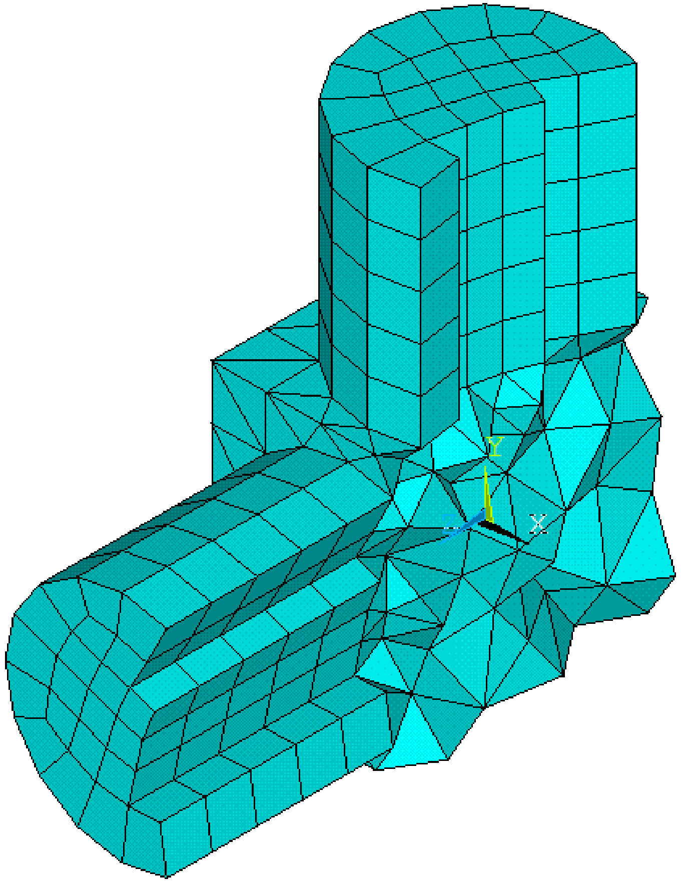
# Naive search of external faces:

```
Loop through all element IDs
   Loop through all element faces
      Get first list of face nodes
      Loop through all elements



      End Loop
      If nothing found, this is an external face
   End Loop
End Loop
```



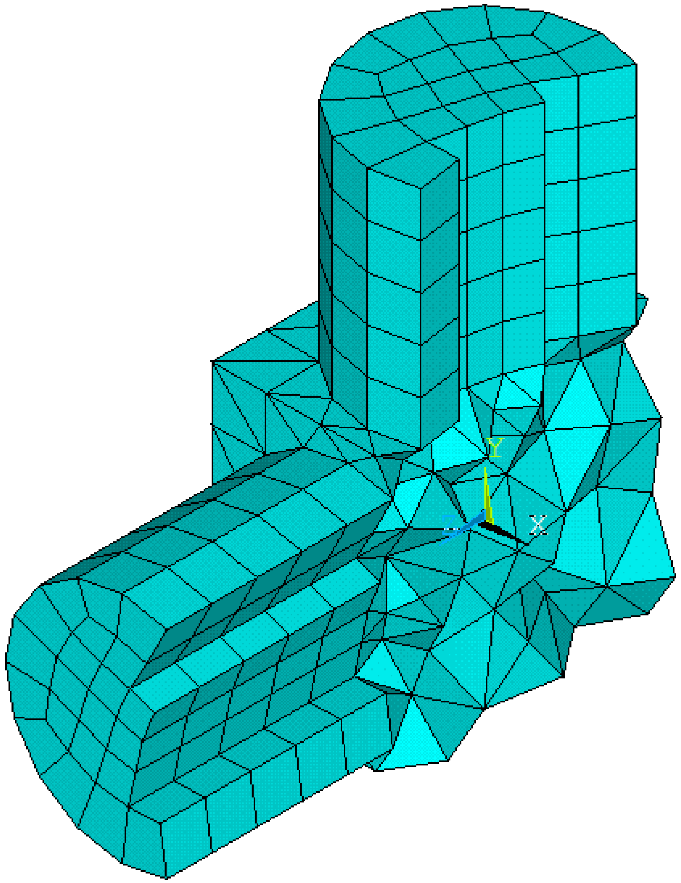
# Naive search of external faces:

```
Loop through all element IDs
   Loop through all element faces
      Get first list of face nodes
      Loop through all elements
         Loop through all element faces









         End Loop
      End Loop
      If nothing found, this is an external face
   End Loop
End Loop
```



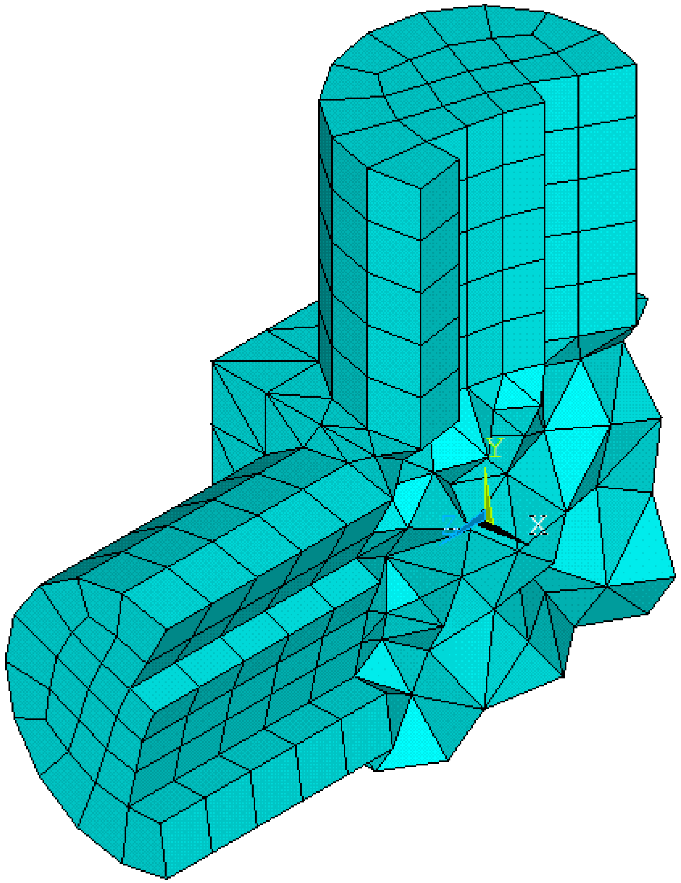
# Naive search of external faces:

```
Loop through all element IDs
   Loop through all element faces
      Get first list of face nodes
      Loop through all elements
         Loop through all element faces
            Get second list of face nodes




         End Loop
      End Loop
      If nothing found, this is an external face
   End Loop
End Loop
```



# Naive search of external faces:

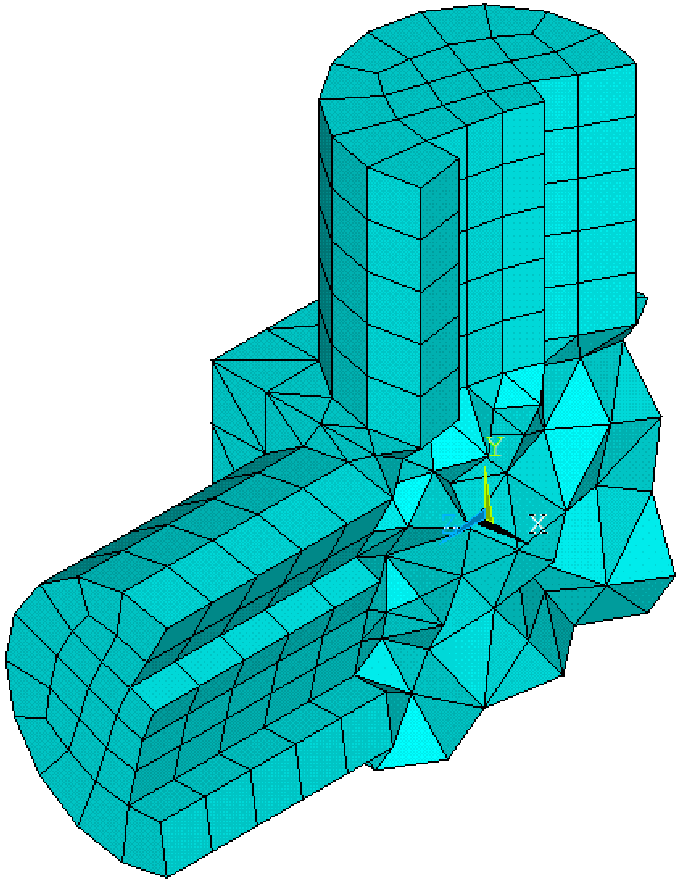
```
Loop through all element IDs
   Loop through all element faces
      Get first list of face nodes
      Loop through all elements
         Loop through all element faces
            Get second list of face nodes
            If first and second list have same nodes
               This is an internal face
               Exit loop
            End If
         End Loop
      End Loop
      If nothing found, this is an external face
   End Loop
End Loop
```



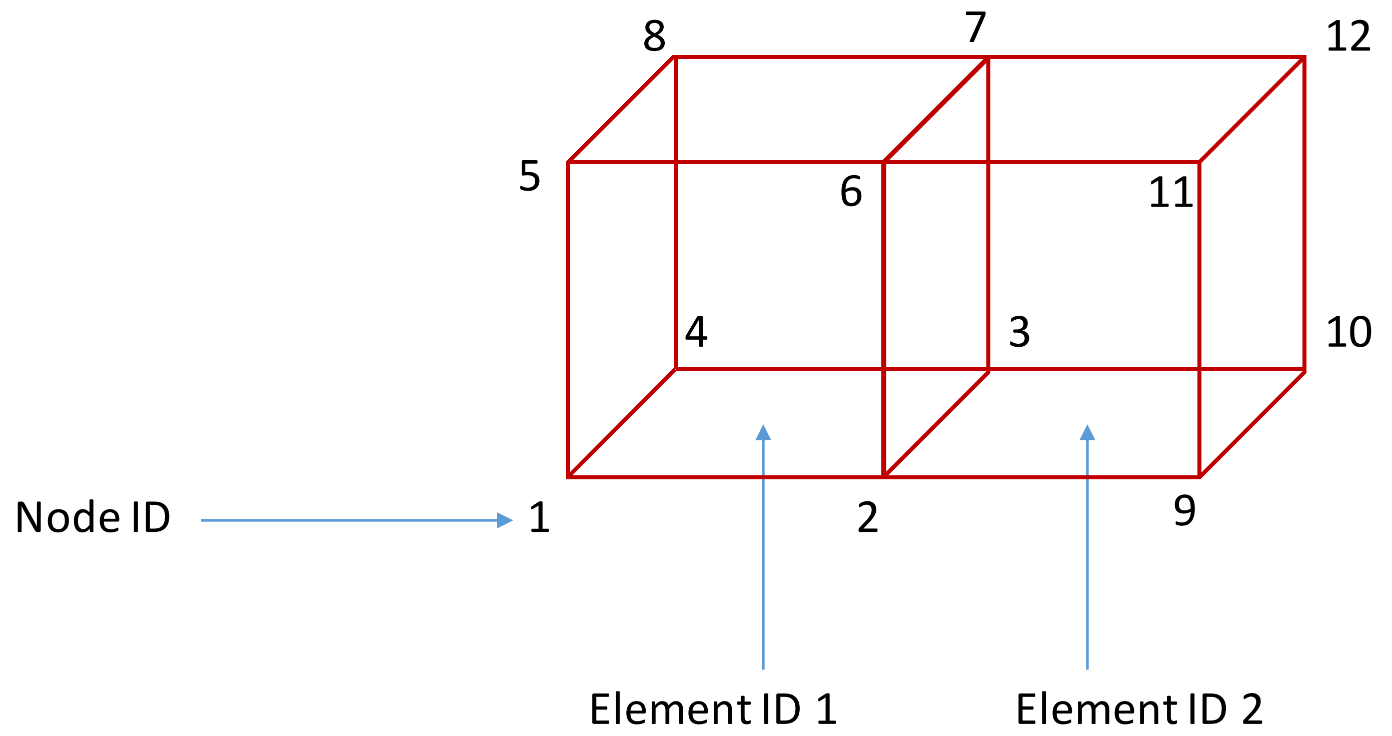
**Computing time has quadratic dependency on the element number.**

# Dictionary based search of external faces:

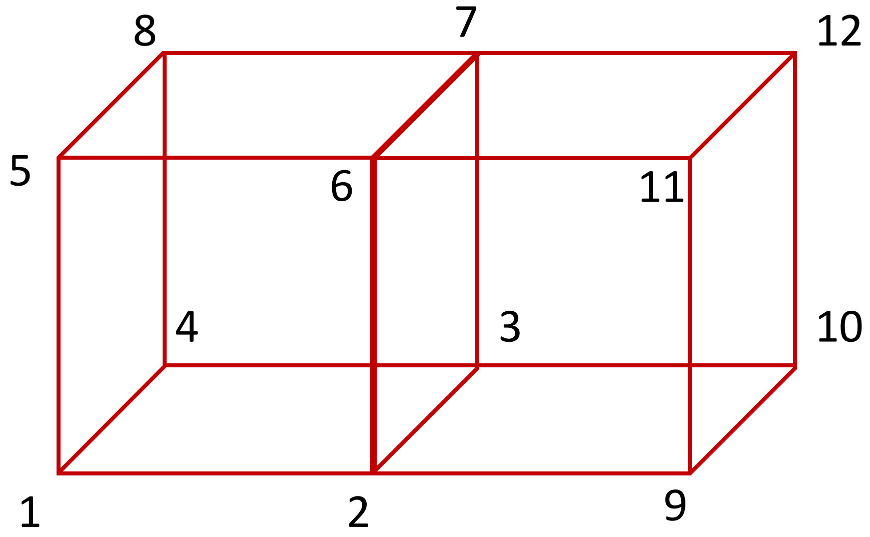
```
Loop through all element IDs
   Loop through all element faces
      Get list of face nodes
      Sort list of face nodes
      # This is the trick:
      dict set faces {*}$facenodes $elemid
   End Loop
End Loop
```



# Simple example:



# Simple example:



**The dictionary based search algorithm creates this dictionary structure:**

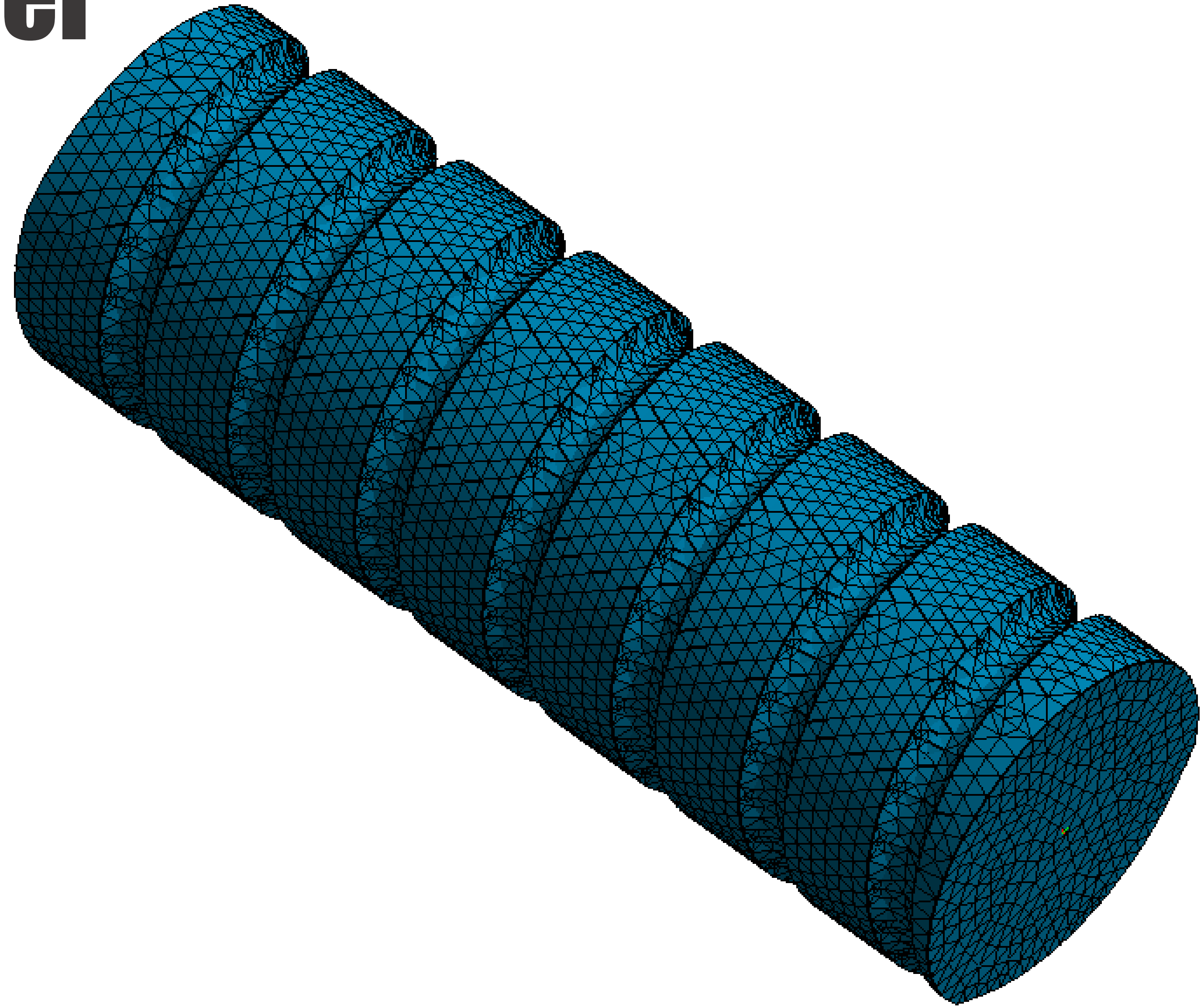
```
1 {2 {3 {4 {1}} 5 {6 {1}}} 4 {5 {8 {1}}}}
2 {3 {6 {7 {1 2}} 9 {10 {2}}} 6 {9 {11 {2}}}}
3 {4 {7 {8 {1}}} 7 {10 {12 {2}}}}
5 {6 {7 {8 {1}}}}
6 {7 {11 {12 {2}}}}
9 {10 {11 {12 {2}}}}
```

➡ **The face refresented by nodes 2,3,6,7 is identified as internal face, because it is shared by two elements (1 and 2)**

# Benchmark model

FE model of a ball screw, coarse mesh

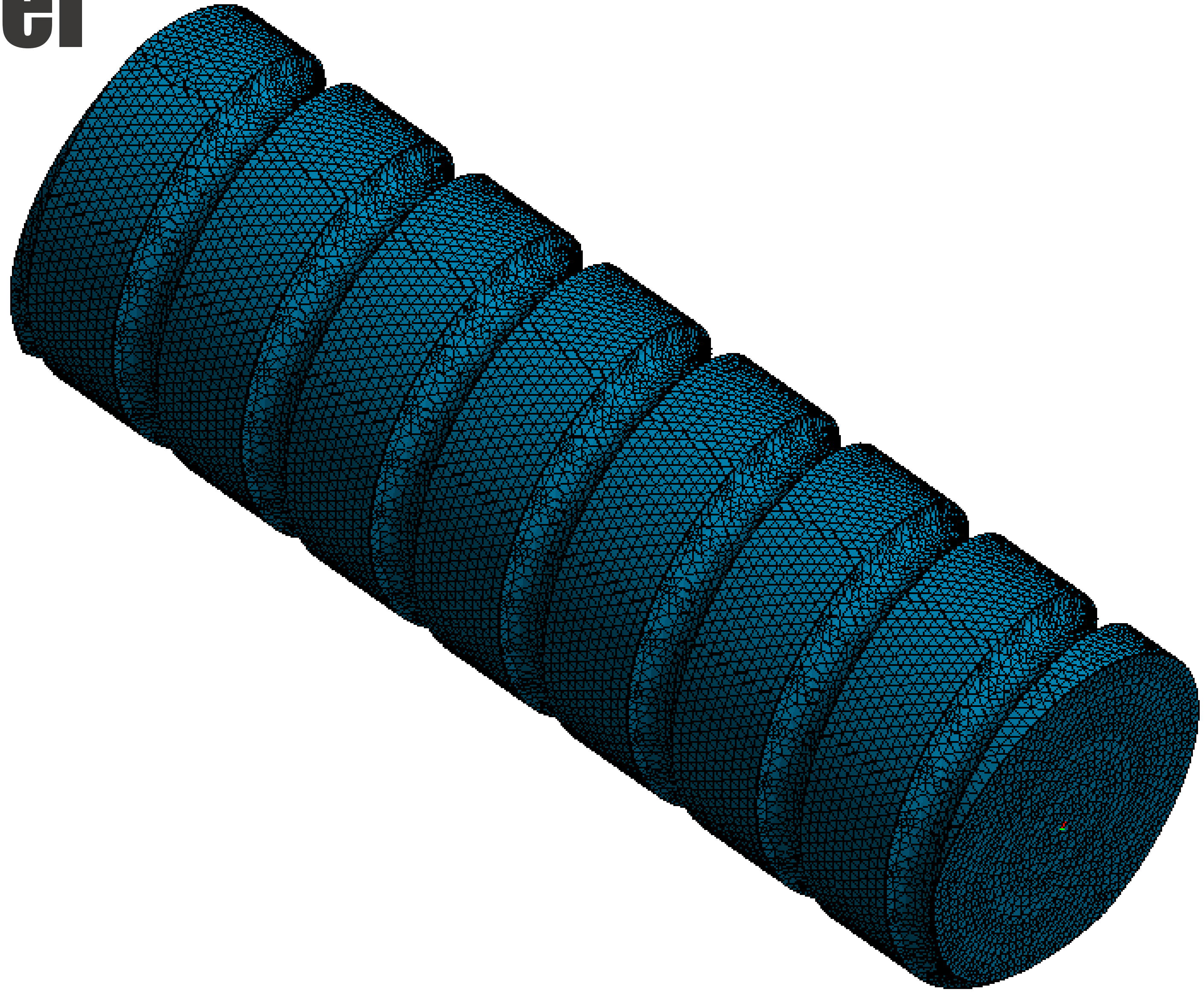
- NODES: 15,000
- ELEMENTS: 66,000
- Time to extract external surfaces: 4 seconds



# Benchmark model

FE model of a ball screw, medium mesh

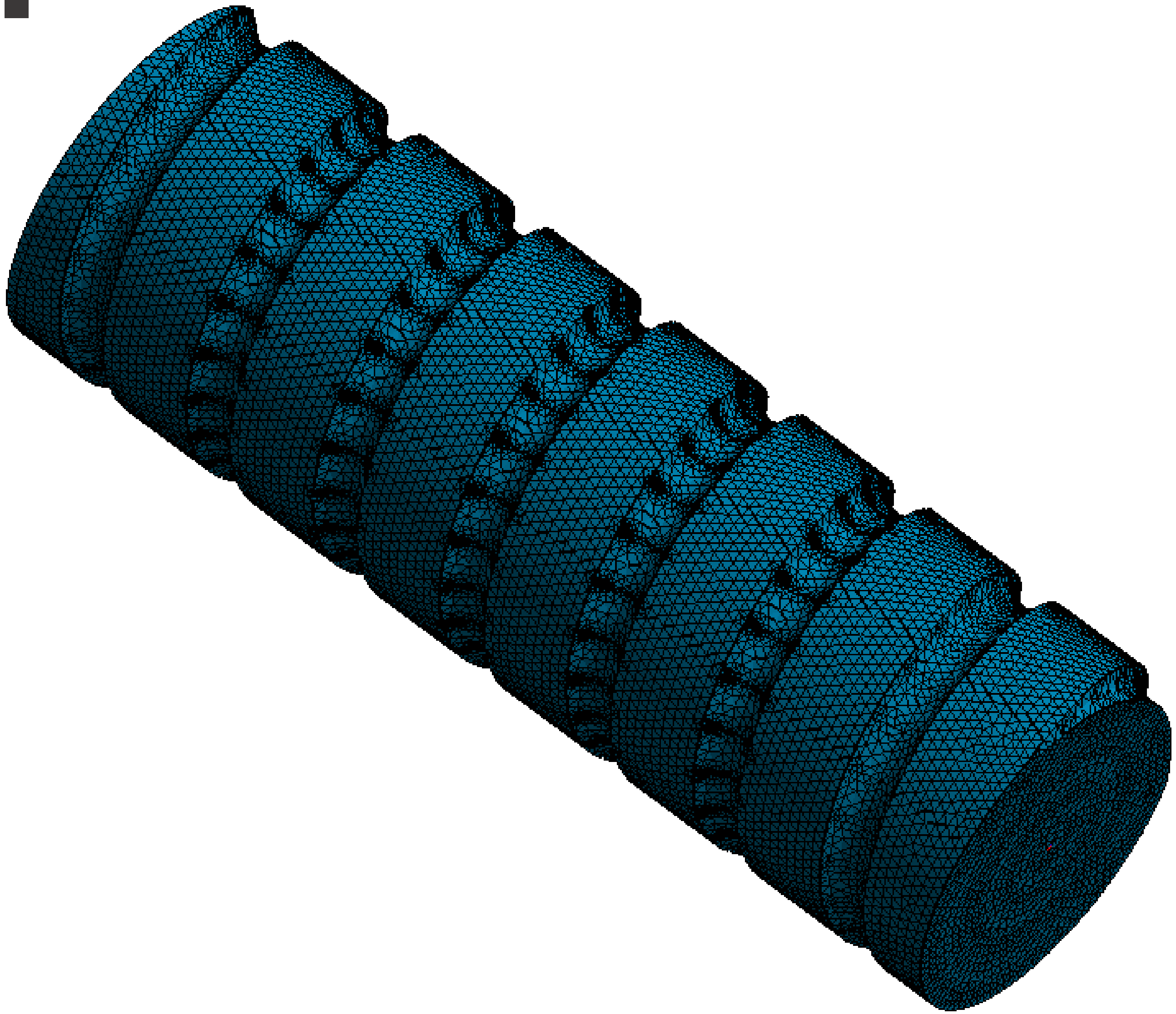
- NODES: 60,000
- ELEMENTS: 312,000
- Time to extract external surfaces: 15 seconds



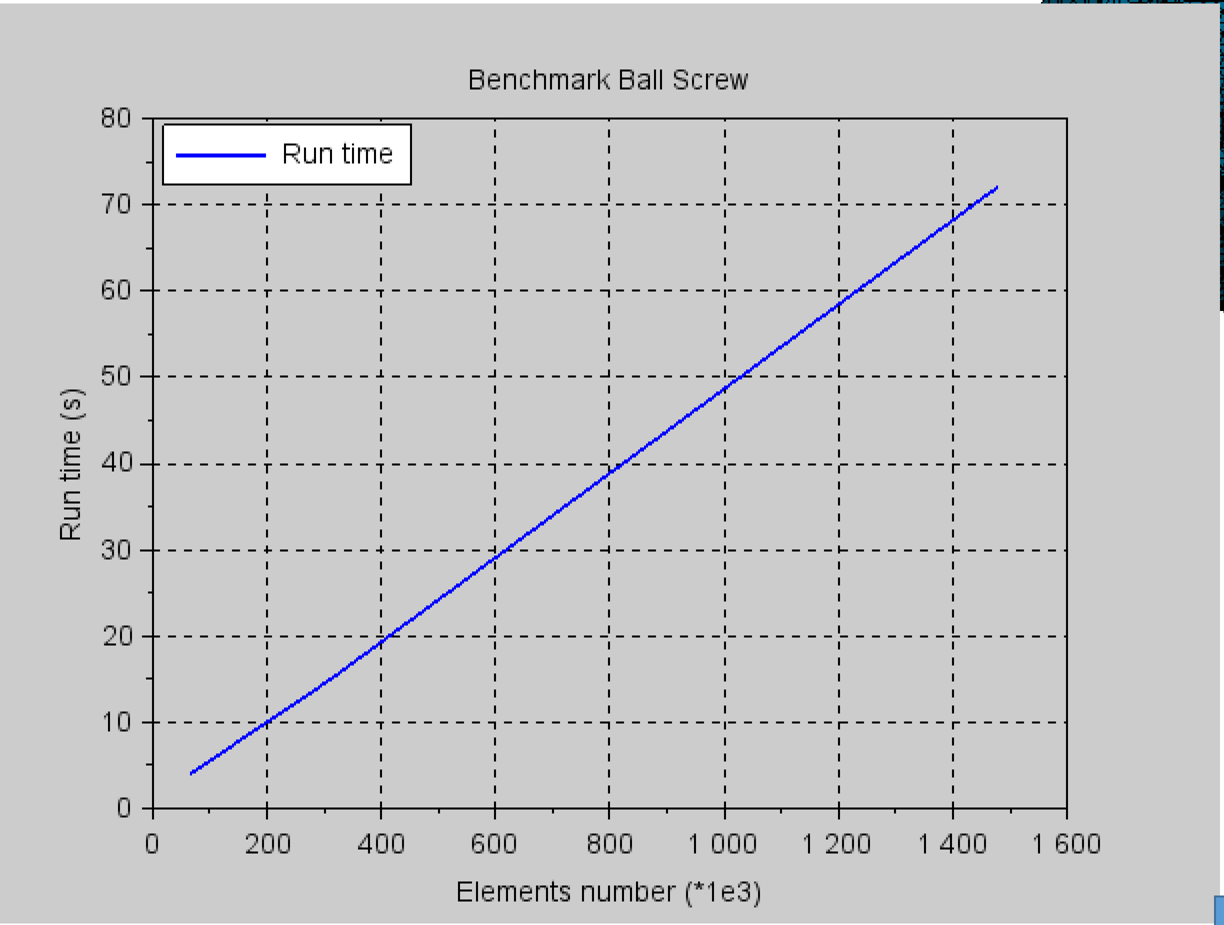
# Benchmark model

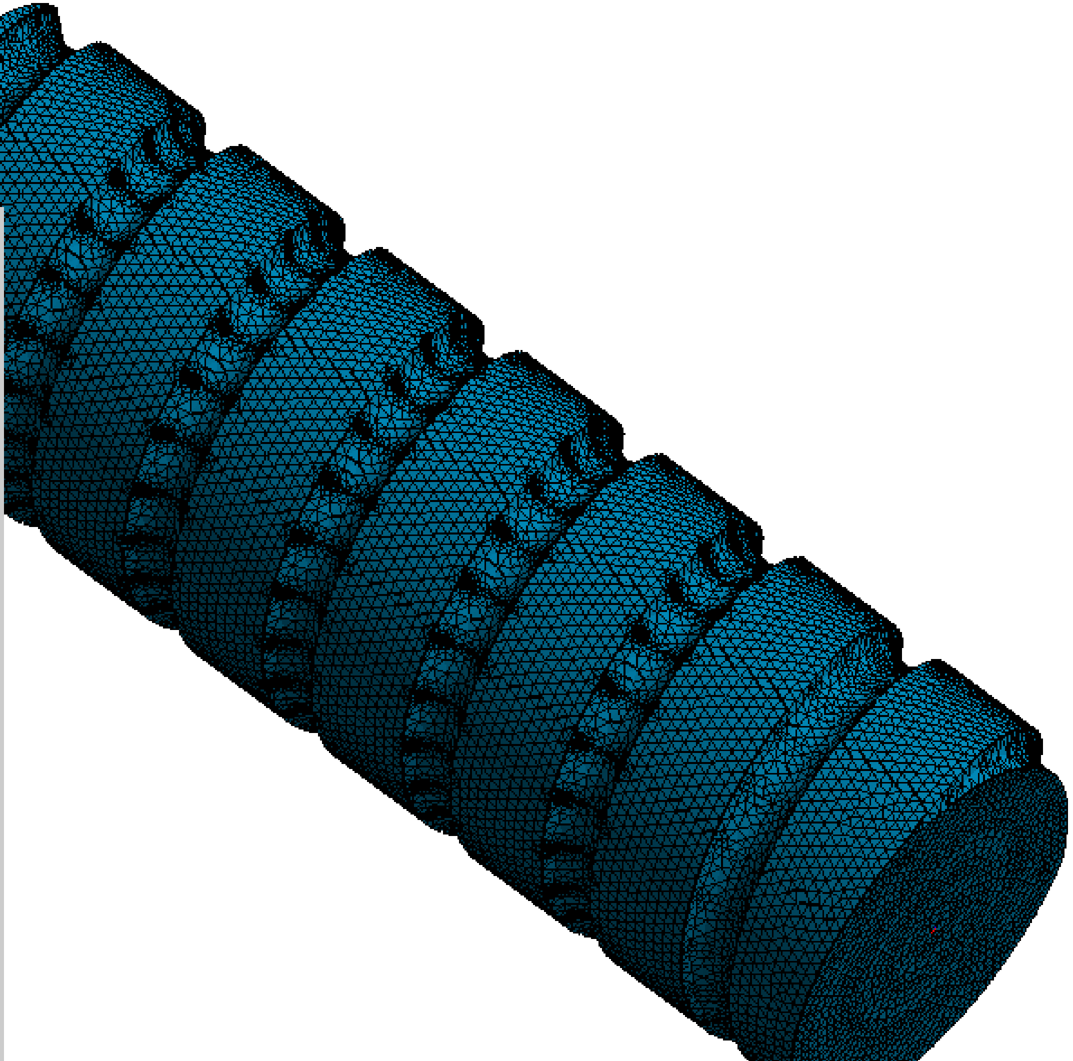
FE model of a ball screw, fine mesh

- NODES: 311,000
- ELEMENTS: 1,477,000
- Time to extract external surfaces: 72 seconds



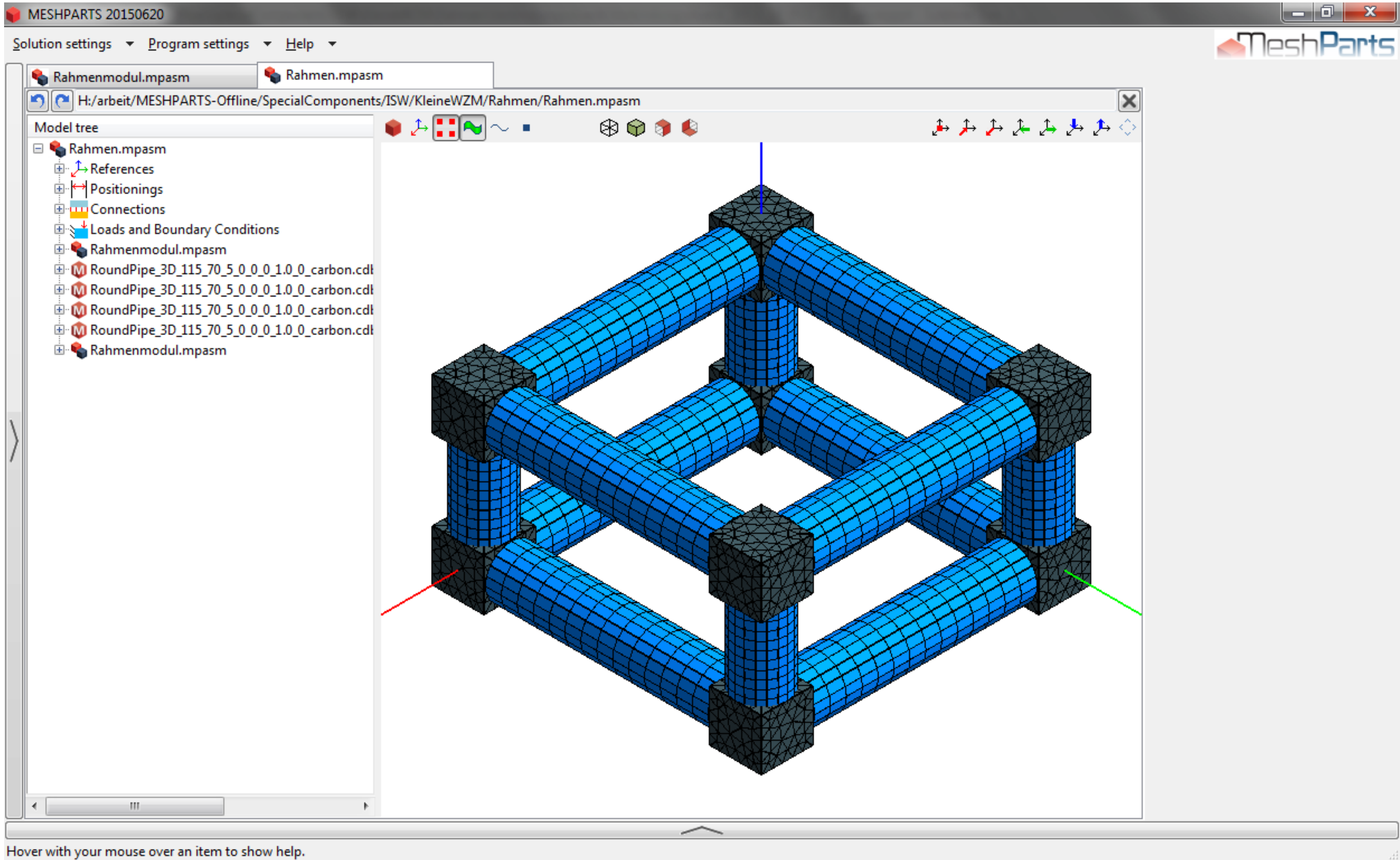
# Benchmark model





Computing time has linear dependency on the element number.

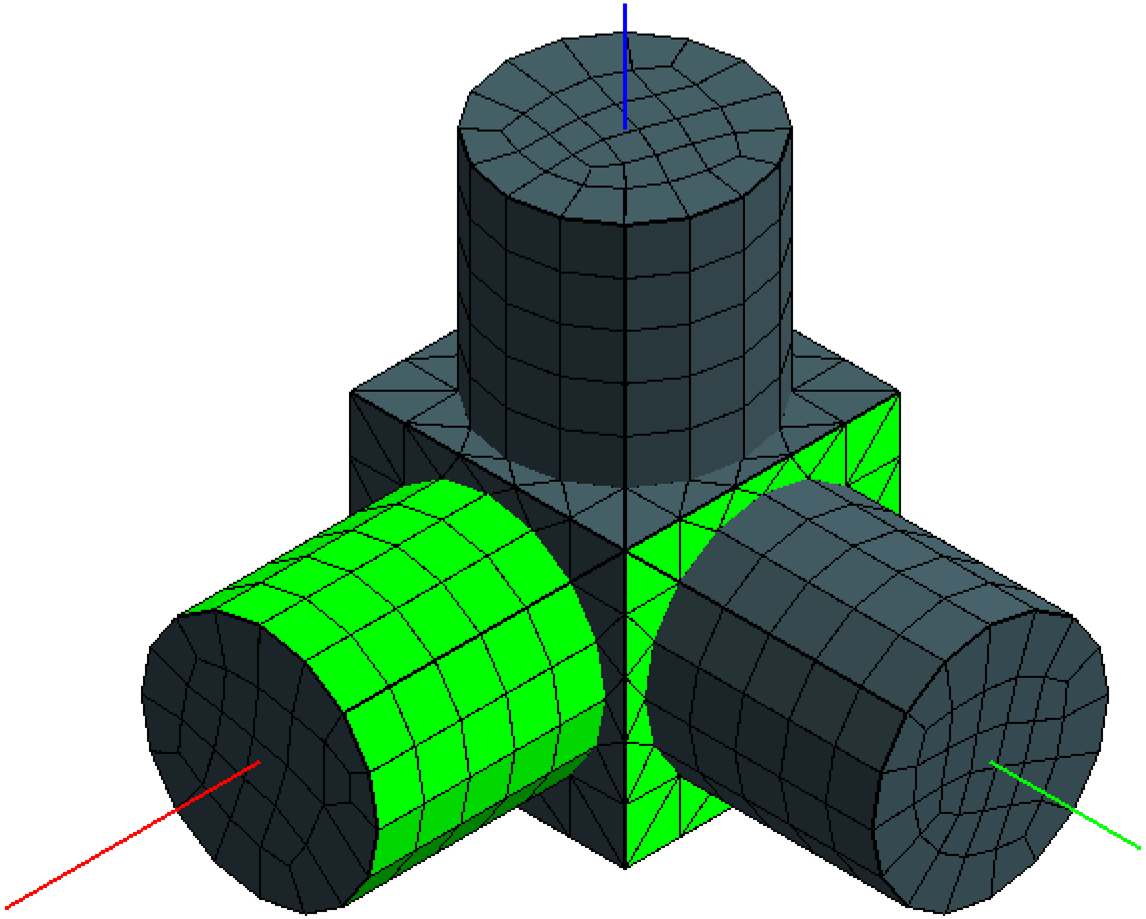
# Live software demo


MESHPARTS 20150620
Solution settings
Program settings
Help
MeshParts
Rahmenmodul.mpasm
Rahmen.mpasm
Model tree
Rahmen.mpasm
References
Positionings
Connections
Loads and Boundary Conditions
Rahmenmodul.mpasm
RoundPipe_3D_115_70_5_0_0_0_1.0_0_carbon.cdb
RoundPipe_3D_115_70_5_0_0_0_1.0_0_carbon.cdb
RoundPipe_3D_115_70_5_0_0_0_1.0_0_carbon.cdb
RoundPipe_3D_115_70_5_0_0_0_1.0_0_carbon.cdb
Rahmenmodul.mpasm

H:/arbeit/MESHPARTS-Offline/SpecialComponents/ISW/KleineWZM/Rahmen/Rahmen.mpasm

Hover with your mouse over an item to show help.

# Split the surface into geometry features



**Keywords:**

Mesh data structure,
Mesh Traversal

# Mesh data structure as a Tcl dictionary:

```
dict set modeldata nodes {1 {5.25E-02 0 0 0 0 0} 2 {0.0E+00
5.25E-02 0 0 0 0} 3 {...
dict set modeldata elems {1 {{98 101 174 170 340 334 310 309}
1 0} 2 {{340 334 310 309 341 335 317 316} 1 0} 3 {....
dict set modeldata face2normal {1.1 {0.0 0.0 1.0} 34.1 {0.0
0.0 1.0} 67.1 {...
dict set modeldata edge2faces {98.101 {1.1 58.1} 98.170 {1.1
34.1} 170.174 {...
dict set modeldata face2edges {1.1 {98.101 98.170 170.174
101.174} 34.1 {...
dict set modeldata node2faces {101 {1.1 4.1 37.1 58.1} 98 {1.1
34.1 58.1 67.1} 170 {...
dict set modeldata surfaces {1 {1.1 4.1 34.1 58.1 7.1 37.1
31.1 67.1 61.1 10.1 40.1 28.1 ...
```

# Thank you for your attention!

Alexandru Dadalau, EuroTcl 2015

https://www.meshparts.de