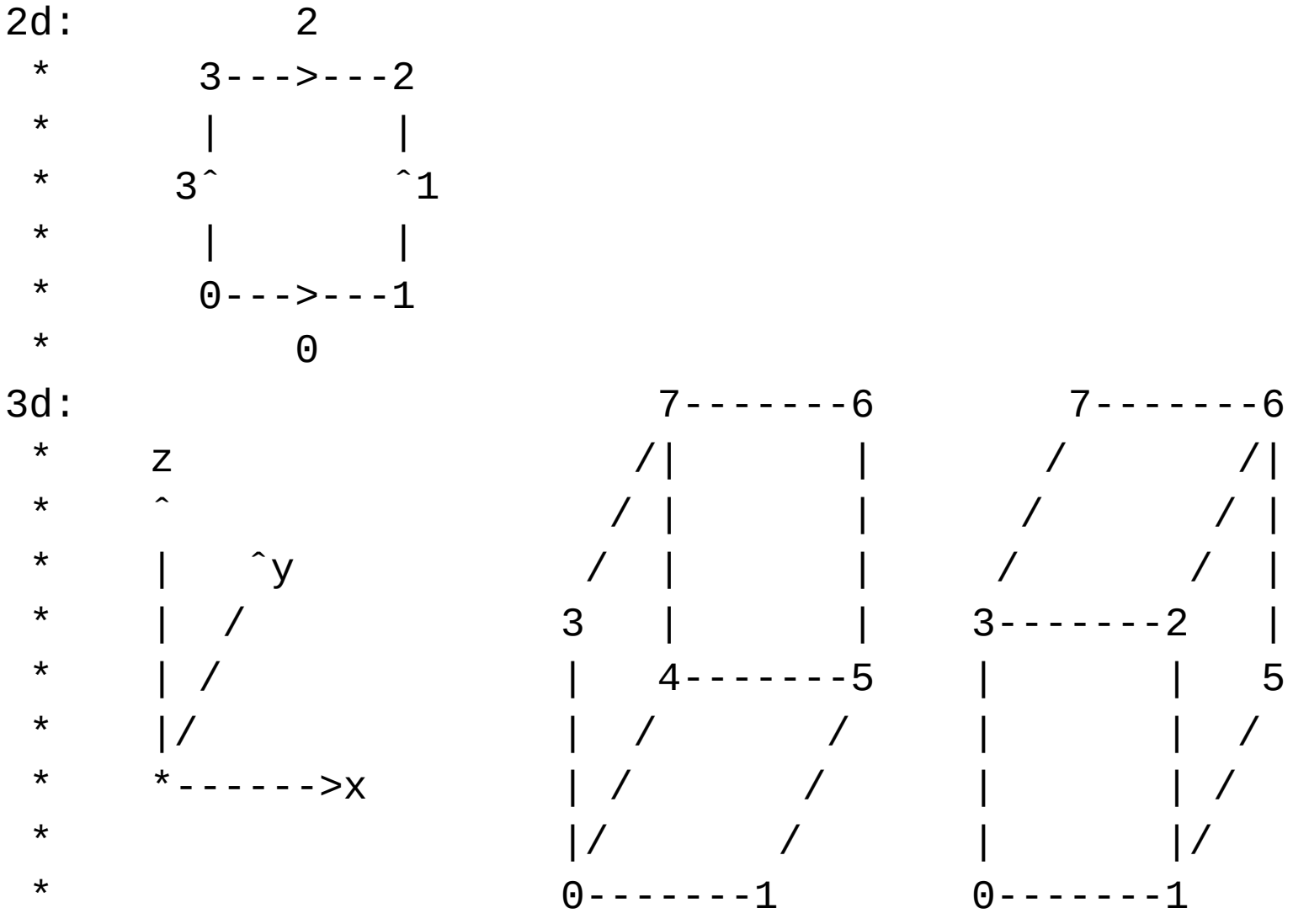
New numbering scheme in deal.II

Ralf Hartmann

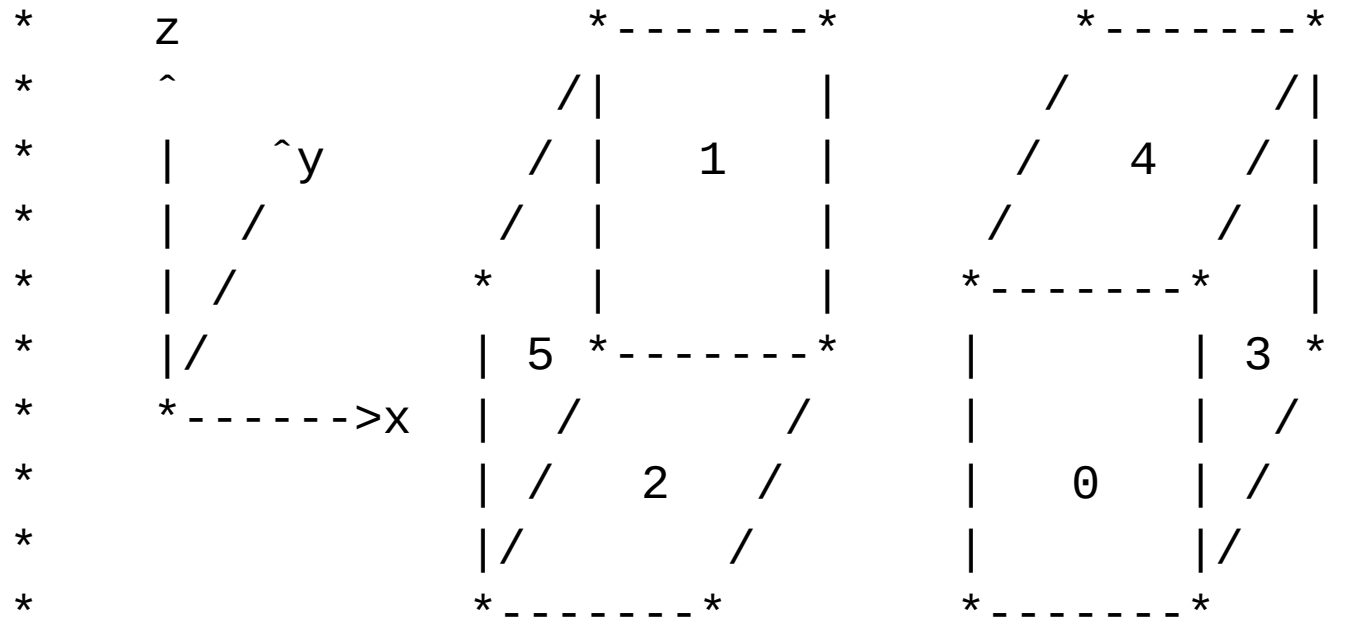
Numerical Methods
Institute of Aerodynamics and Flow Technology
German Aerospace Center (DLR), Braunschweig

The old numbering scheme in deal.II: Vertices



Ralf Hartmann, Institut für Aerodynamik und Strömungstechnik

The old numbering scheme in deal.II: Faces in 3d



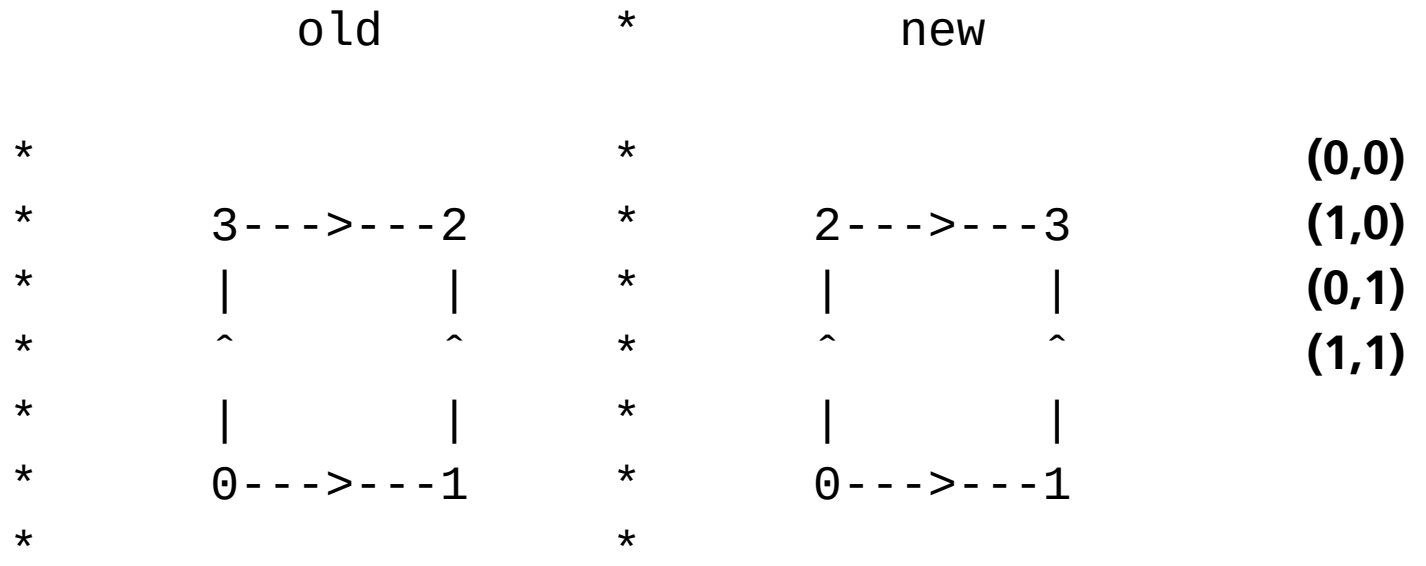

```
const unsigned int
GeometryInfo<3>::unit_normal_direction[6]={ 1, 1, 2, 0, 2, 0 };

const int
GeometryInfo<3>::unit_normal_orientation[6]={ -1, 1, -1, 1, 1, -1 };
```

The new numbering scheme: Vertices

- **vertices are numbered in lexicographic ordering (x running fastest)**

2d:

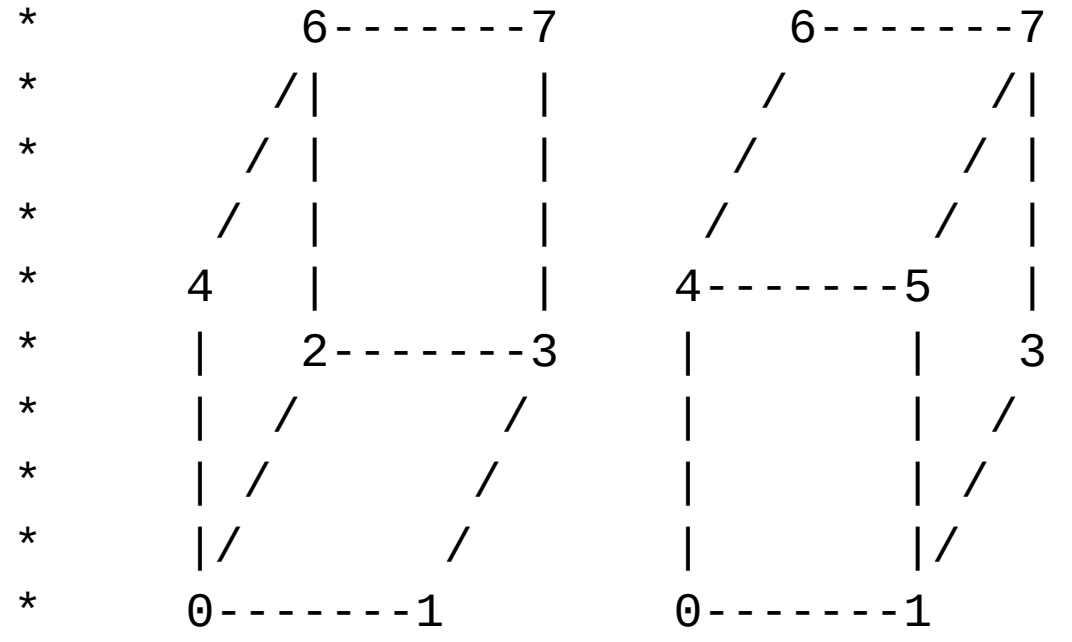


Ralf Hartmann, Institut für Aerodynamik und Strömungstechnik

The new numbering scheme: Vertices

- vertices are numbered in lexicographic ordering (x running fastest)

3d:



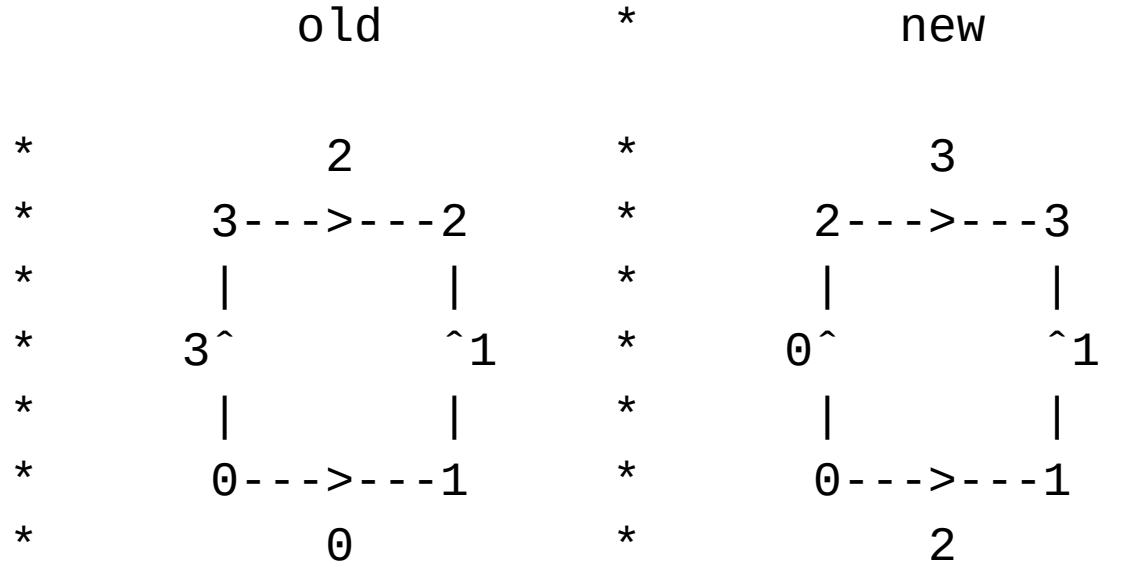
(0,0,0)
(1,0,0)
(0,1,0)
(1,1,0)
(0,0,1)
(1,0,1)
(0,1,1)
(1,1,1)

Ralf Hartmann, Institut für Aerodynamik und Strömungstechnik

The new numbering scheme: Faces

- first the two faces with normals in x-, then y- and z-direction
- for each two faces: first the face with normal in negative coordinate direction, then the one with normal in positive direction

2d:

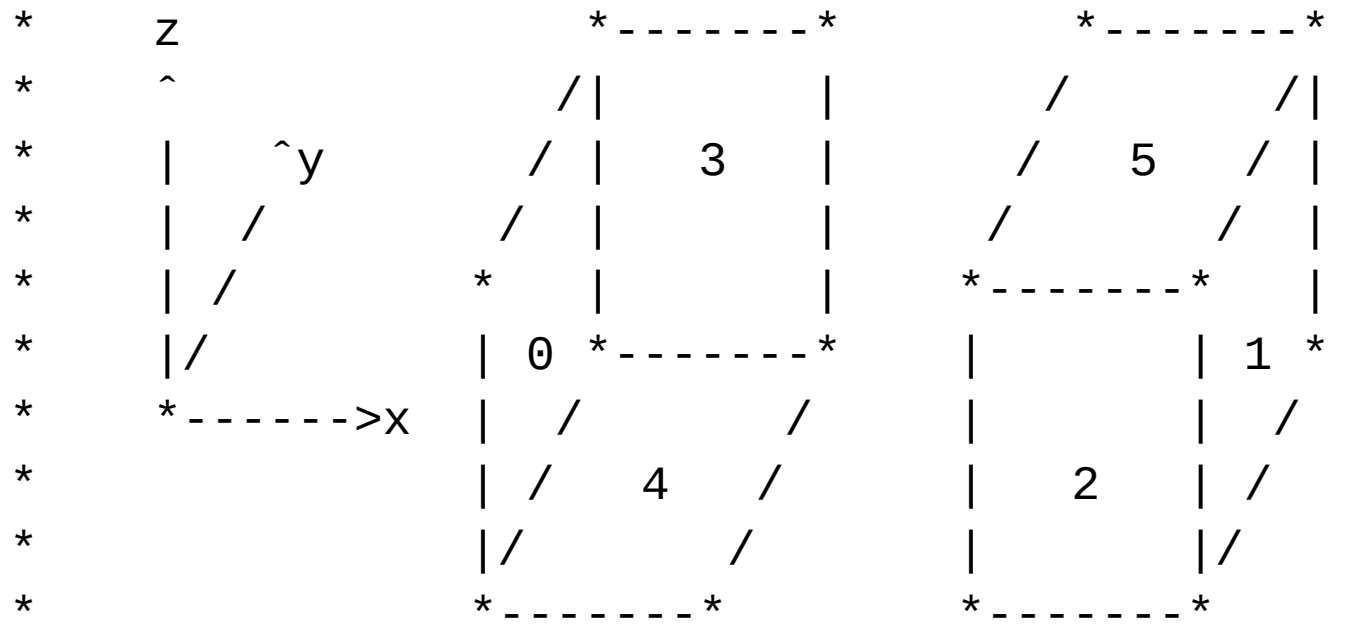


Ralf Hartmann, Institut für Aerodynamik und Strömungstechnik

The new numbering scheme: Faces

- first the two faces with normals in x-, then y- and z-direction
- for each two faces: first the face with normal in negative coordinate direction, then the one with normal in positive direction

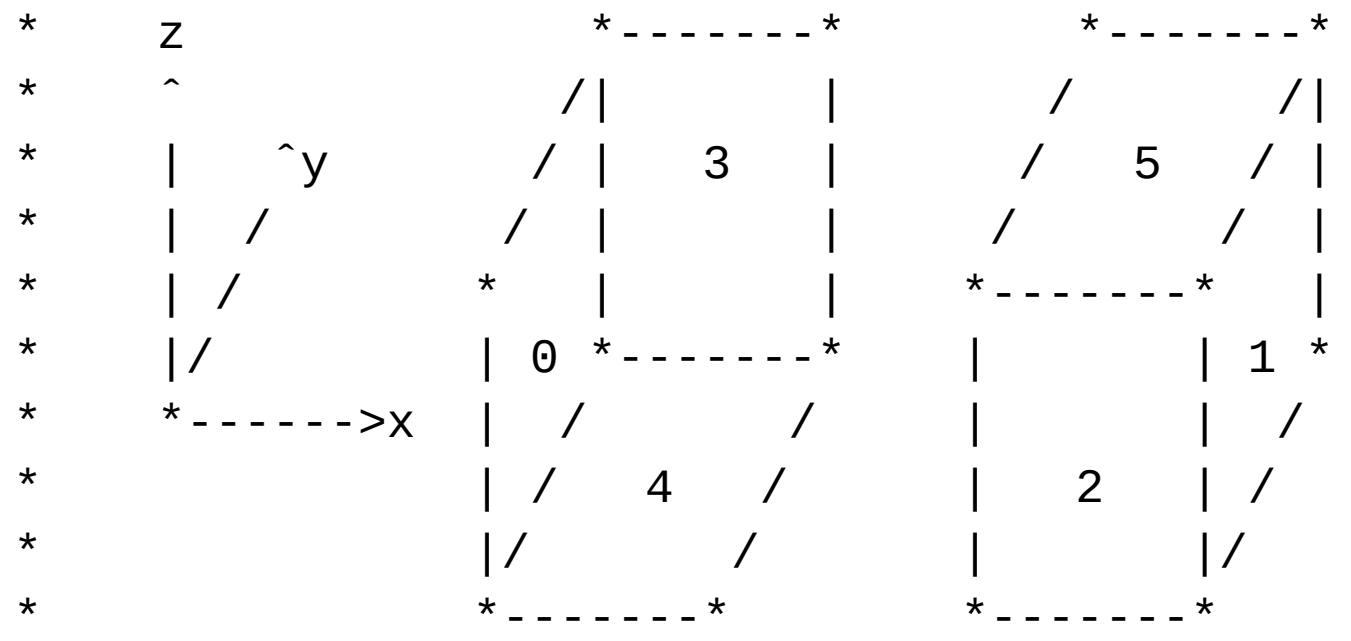
3d:



Ralf Hartmann, Institut für Aerodynamik und Strömungstechnik

The new numbering scheme: Faces

3d:




```
const unsigned int
GeometryInfoBase::unit_normal_direction[6]={ 0, 0, 1, 1, 2, 2 };

const int
GeometryInfoBase::unit_normal_orientation[6]={ -1, 1, -1, 1, -1, 1};
```

Simplification of code in deal.II: e.g. in data_out.cc

```
<            const unsigned int neighbor_patch_index
<              = this->patches[(*data.cell_to_patch_index_map)
<                              [neighbor->level()][neighbor->index()]].patch_index;
<
<          switch (dim)
<            {
<              case 1:
<                    patch->neighbors[f] = neighbor_patch_index;
<                    break;
<
<              case 2:
<                      switch (f)
<                        {
<                          case 0: patch->neighbors[2] = neighbor_patch_index; break;
<                          case 1: patch->neighbors[1] = neighbor_patch_index; break;
<                          case 2: patch->neighbors[3] = neighbor_patch_index; break;
<                          case 3: patch->neighbors[0] = neighbor_patch_index; break;
<                        }
<                      break;
<              case 3:
<                      switch (f)
<                        {
<                          case 0: patch->neighbors[2] = neighbor_patch_index; break;
<                          case 1: patch->neighbors[3] = neighbor_patch_index; break;
<                          case 2: patch->neighbors[4] = neighbor_patch_index; break;
<                          case 3: patch->neighbors[1] = neighbor_patch_index; break;
```


Ralf Hartmann, Institut für Aerodynamik und Strömungstechnik

Simplification of code in deal.II: e.g. in data_out.cc

```
<                         case 4: patch->neighbors[5] = neighbor_patch_index; break;
<                         case 5: patch->neighbors[0] = neighbor_patch_index; break;
<                       }
<                     break;
<
<             default:
<                     Assert(false, ExcNotImplemented());
<           }
---
>           patch->neighbors[f] = this->patches[(*data.cell_to_patch_index_map)
>                                 [neighbor->level()][neighbor->index()]].patch_index;
```

Changes required in user codes: e.g. step-14.cc

```
static const int cell_vertices[][GeometryInfo<dim>::vertices_per_cell]
  = {{0, 1, 6, 5},
     {1, 2, 7, 6},
     {2, 3, 8, 7},
     [....]
     {16, 17, 22, 21},
     {17, 18, 23, 22}};
[....]
tria.create_triangulation (vertices, cells, SubCellData());
```

replace by

```
[....]
tria.create_triangulation_compatibility (vertices, cells, SubCellData());
```

or replace by

```
static const int cell_vertices[][GeometryInfo<dim>::vertices_per_cell]
  = {{0, 1, 5, 6},
     {1, 2, 6, 7},
     [....]
     {17, 18, 22, 23}};
[....]
tria.create_triangulation (vertices, cells, SubCellData());
```


Ralf Hartmann, Institut für Aerodynamik und Strömungstechnik