The hp-method within the deal.II framework

Oliver Kayser-Herold

Institut für Wissenschaftliches Rechnen
Technische Universität Braunschweig

The hp-method within the deal.II framework

1. Concept

2. Internals

3. Example (step-21)

4. Missing functions

5. hp-FEM for continuous elements

6. Summary and Outlook

1. Concept

- Standard DoFHandler: Same FiniteElement on all cells

- hpDoFHandler: Different FiniteElement on all cells!

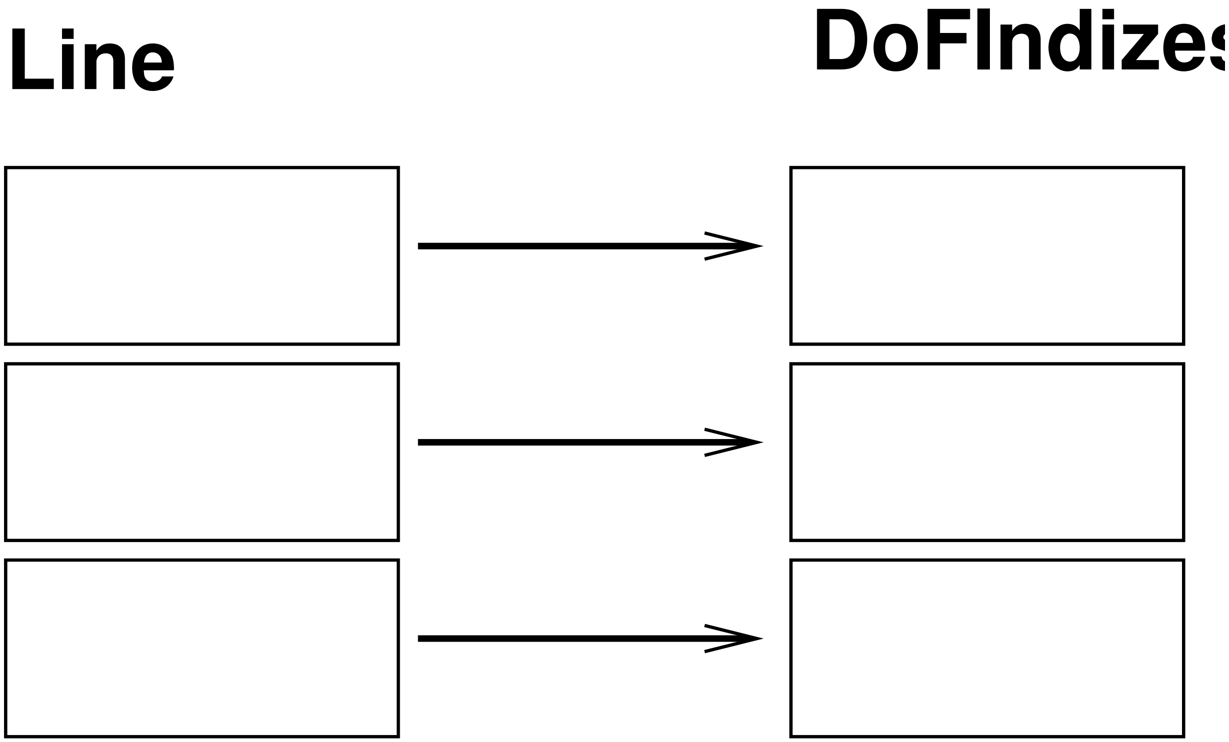
- DoFHandler as template parameter for Accessor classes.

- Instead of an unique FiniteElement, a collection of FiniteElements (FECollection)

- Collection of Mappings (MappingCollection) and Quadrature rules (QCollection).

2. Internals

hpLevelObject:



3. Example (step-21)

Class definition

```
template <int dim>
class DGMethod
{
[...]
    FECollection<dim>    fe_collection;
    hpDoFHandler<dim>    dof_handler;

    std::vector<Quadrature<dim> *> quad;
    QCollection<dim>   *quadrature;
    std::vector<Quadrature<dim-1> *> face_quad;
    QCollection<dim-1> *face_quadrature;

[...]
    const DGTransportEquation<dim> dg;
};
```

3. Example (step-21)

Constructor

```
template <int dim>
DGMethod<dim>::DGMethod () [...]
{
    quadrature = new QCollection<dim> ();
    Quadrature<dim> *quad_temp;

    const unsigned int hp_degree = 5;
    for (unsigned int i = 0; i < hp_degree; ++i)
    {
elm_no = fe_collection.add_fe (FE_DGQ<dim> (i+1));

quad_temp = new QGauss<dim> (i+2);
quad.push_back (quad_temp);
quadrature->add_quadrature (*quad_temp);
    }
}
```

3. Example (step-21)

Cell assembly

```
typename SelectFEValues<hpDoFHandler<dim> >::FEValues fe_v_x (
  mapping, fe_collection, *quadrature, update_flags);
  [...]
const unsigned int max_dofs_per_cell =
   fe_collection.max_dofs_per_cell ();
FullMatrix<double> u_v_matrix (max_dofs_per_cell,
                               max_dofs_per_cell);
fe_v_x.reinit (cell);
dg.assemble_cell_term(fe_v_x.get_present_fe_values (),
         u_v_matrix, cell_vector);
```

3. Example (step-21)

Face assembly

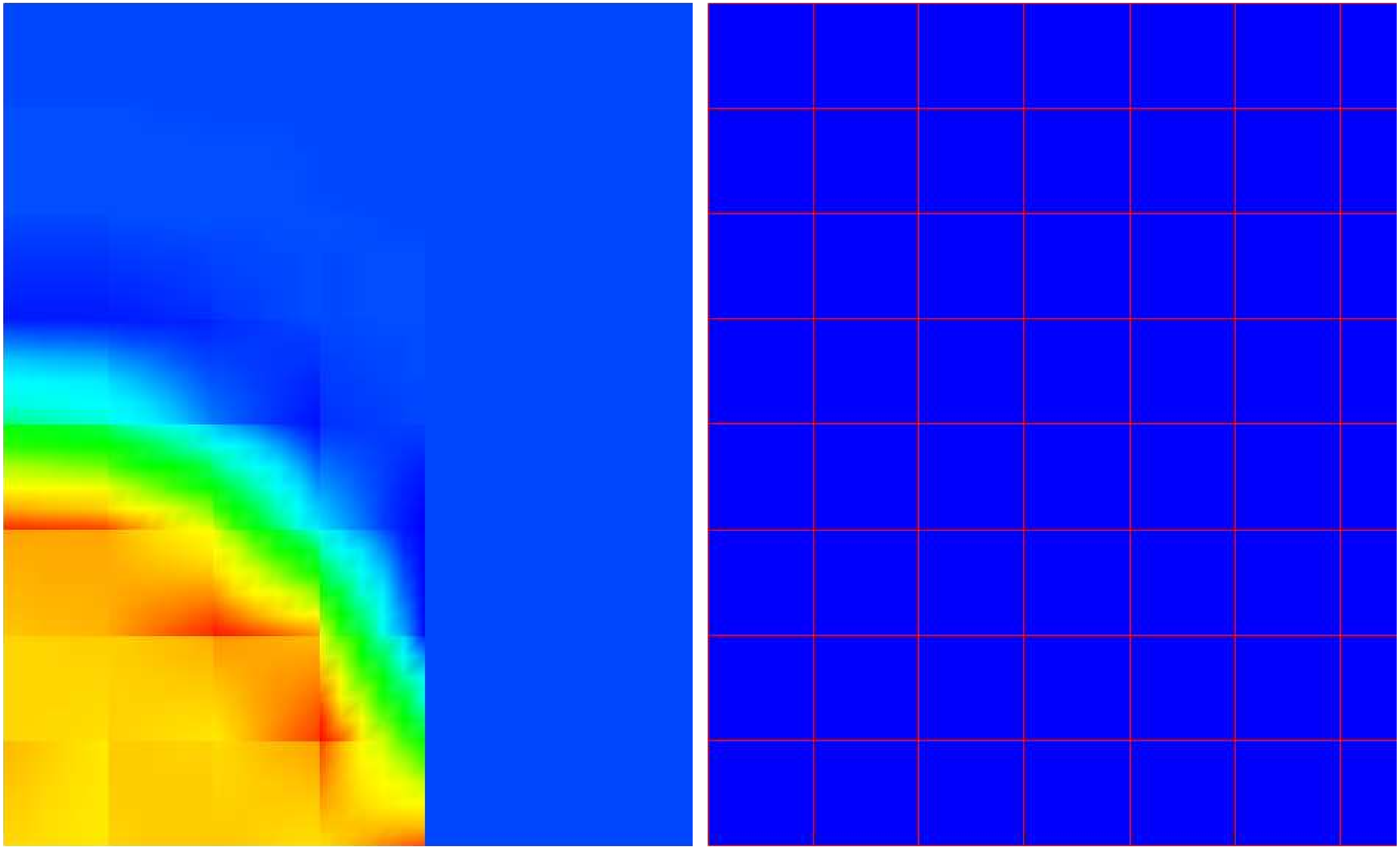
```
typename SelectFEValues<hpDoFHandler<dim> >::FEFaceValues
  fe_v_face_x (mapping, fe_collection,
               *face_quadrature, face_update_flags);

// Determine higher active_fe_index of both cells
const unsigned int use_fe_index =
   neighbor_child->active_fe_index () > cell->active_fe_index () ?
   neighbor_child->active_fe_index () : cell->active_fe_index ();

fe_v_subface_x.reinit (cell, face_no, subface_no, use_fe_index);
fe_v_face_neighbor_x.reinit (neighbor_child, neighbor2,
                             use_fe_index);
```

3. Example (step-21)

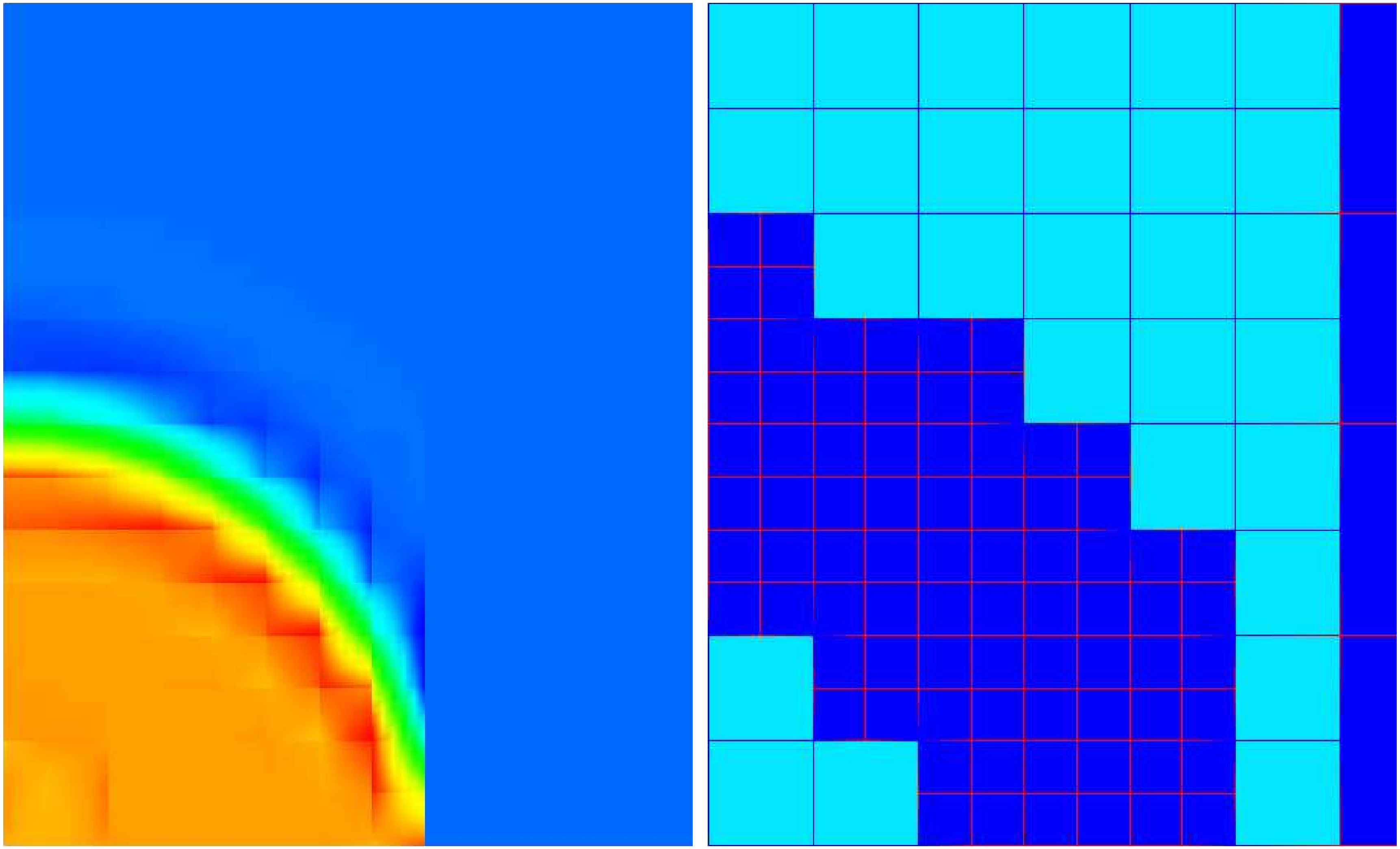
hp-Refinement

```
GridRefinement::refine_and_coarsen_fixed_number (
      triangulation, gradient_indicator, 0.3, 0.1);

                                 // Simple heuristic hp-Refinement.
cell = dof_handler.begin_active ();
for (; cell!=endc; ++cell)
  if (!cell->refine_flag_set ())
    cell->set_active_fe_index (cell->active_fe_index () + 1);

triangulation.execute_coarsening_and_refinement ();
```

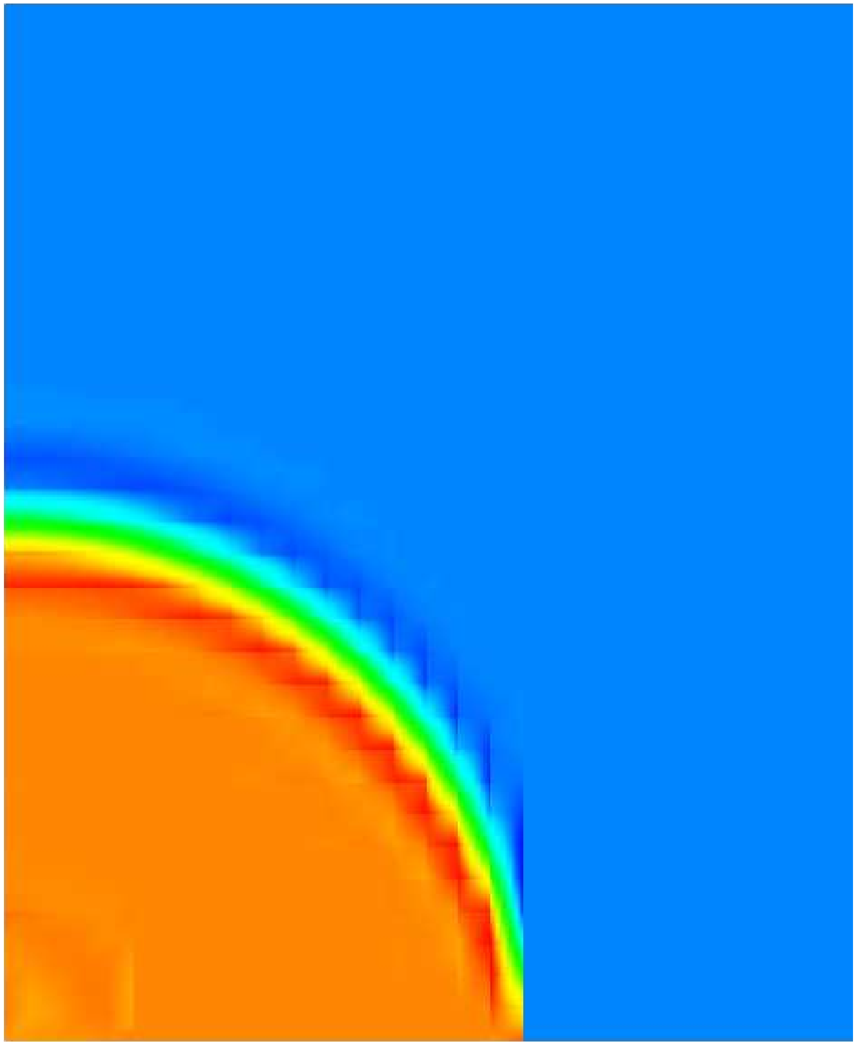
3. Results (step-21) I

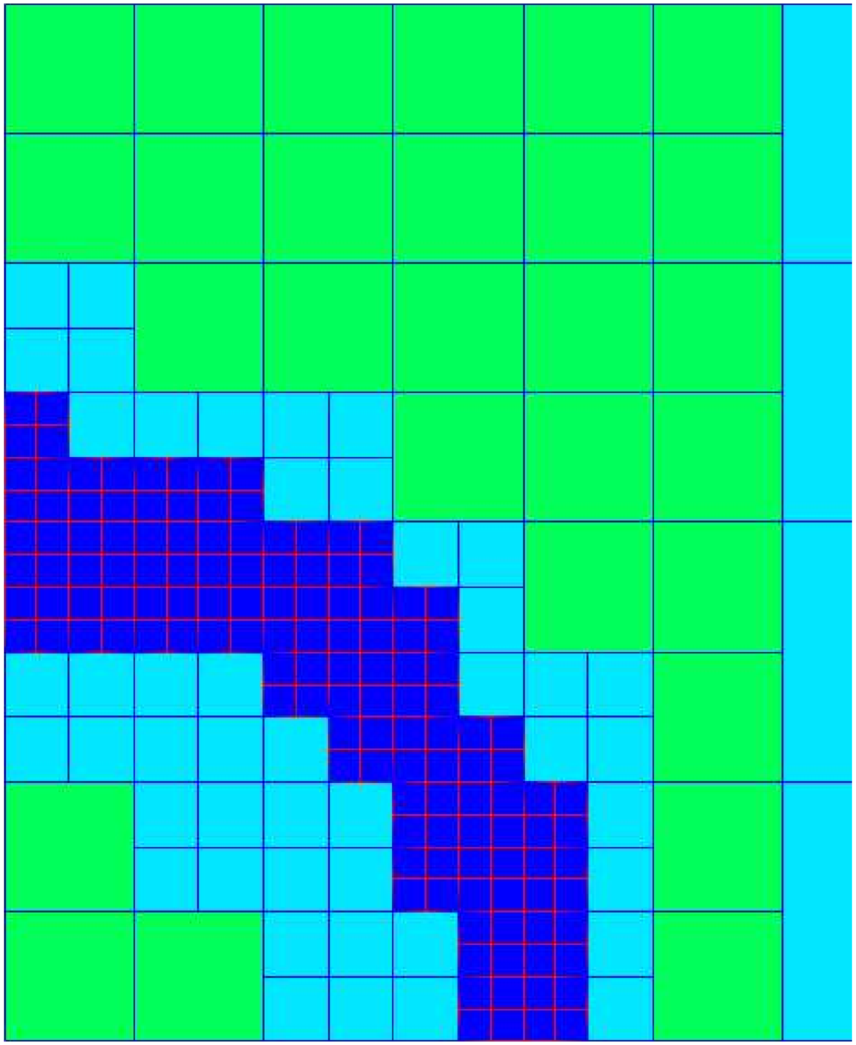


3. Results (step-21) II

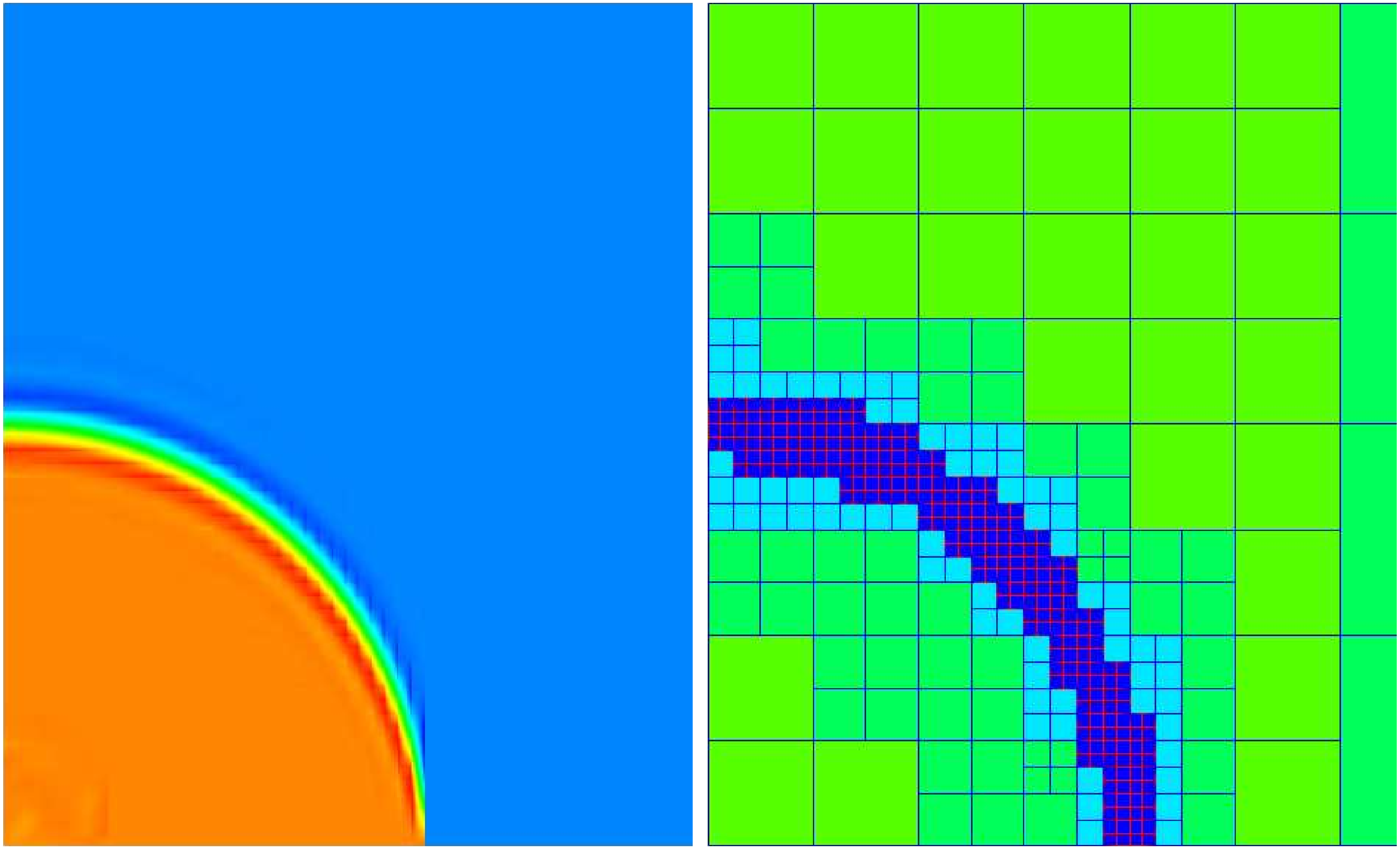


3. Results (step-21) III

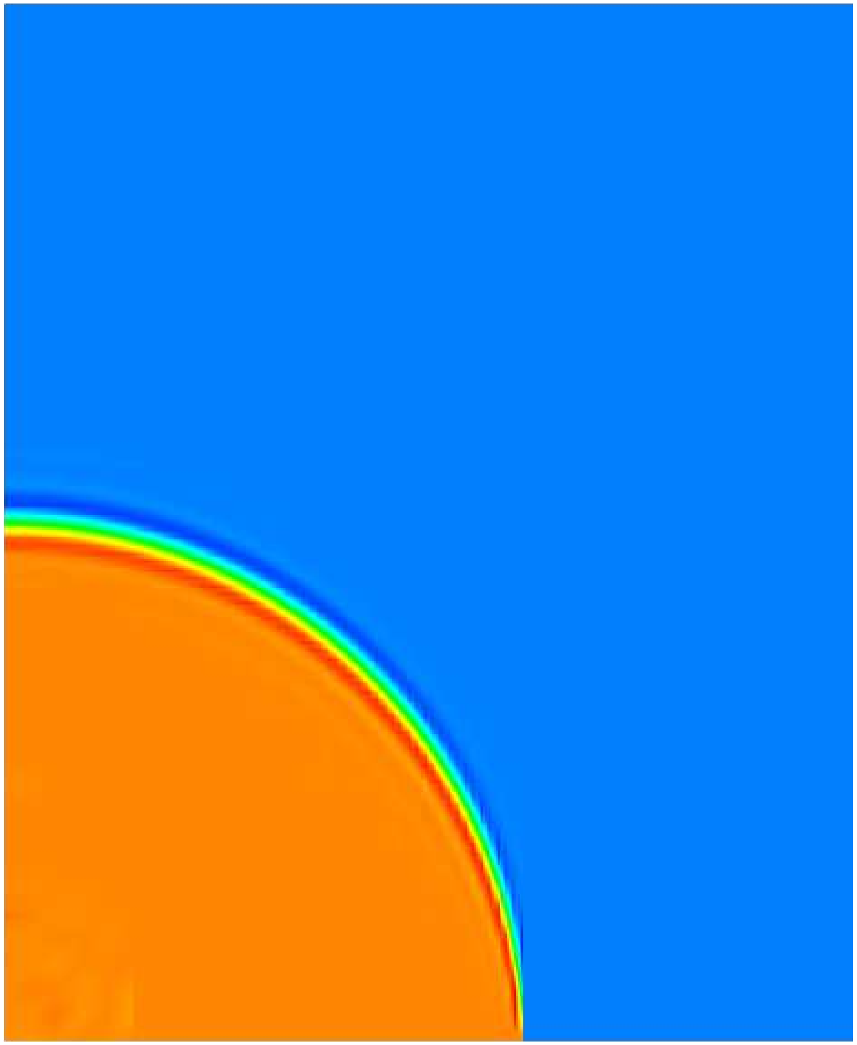


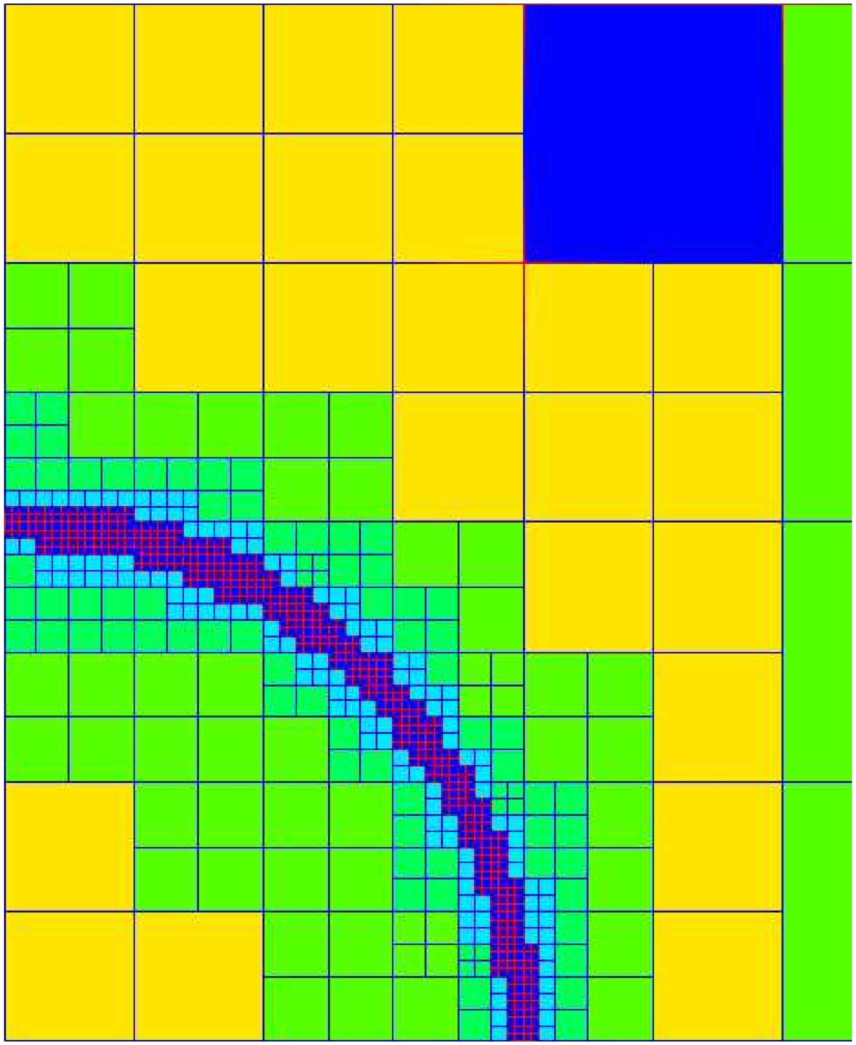


3. Results (step-21) IV



3. Results (step-21) V





4. Missing functions

- Several functions in DoFTools are not yet implemented for the hpDoFHandler

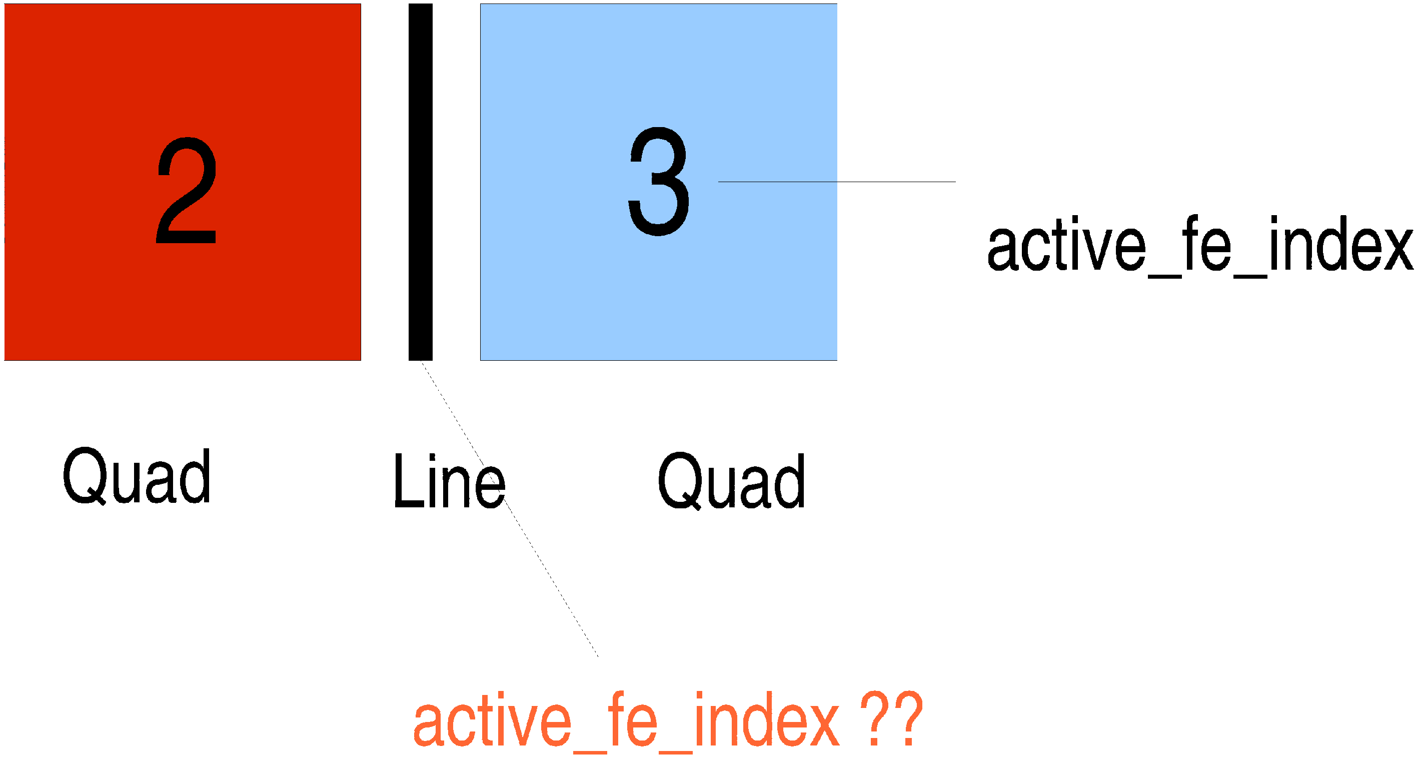
- Usually some changes are required to cope with a different number of `dofs_per_cell`.

5. DoFHandler algorithm

- Distribute DoFs on vertices (all D)
- Distribute DoFs on lines (all D)
- Distribute DoFs on quads (2D, 3D)
- Distribute DoFs on hexes (3D)

For DG the lower geometric entities are not equipped with DoFs, while they are for continuous elements!

5. hp-FEM for continuous elements



Key problem: **No** `active_fe_index` **on lower geometric entities (lines in 2D, lines and quads in 3D)!**

5. Proposal 1

Make the `active_fe_index` available on lower geometric entities:

```
class DoFObjectAccessor : public DoFAccessor<dim, DH>,
  public TriaObjectAccessor<celldim,dim>
```

Old:

```
    typedef DH<dim>  AccessorData;
```

New:

```
    typedef typename std::pair<DH<dim> const *,
                               unsigned int> AccessorData;
```

5. Proposal 1

Advantages:

- `active_fe_index` might be useful
- Distribution of DoFs similar to implemented procedure

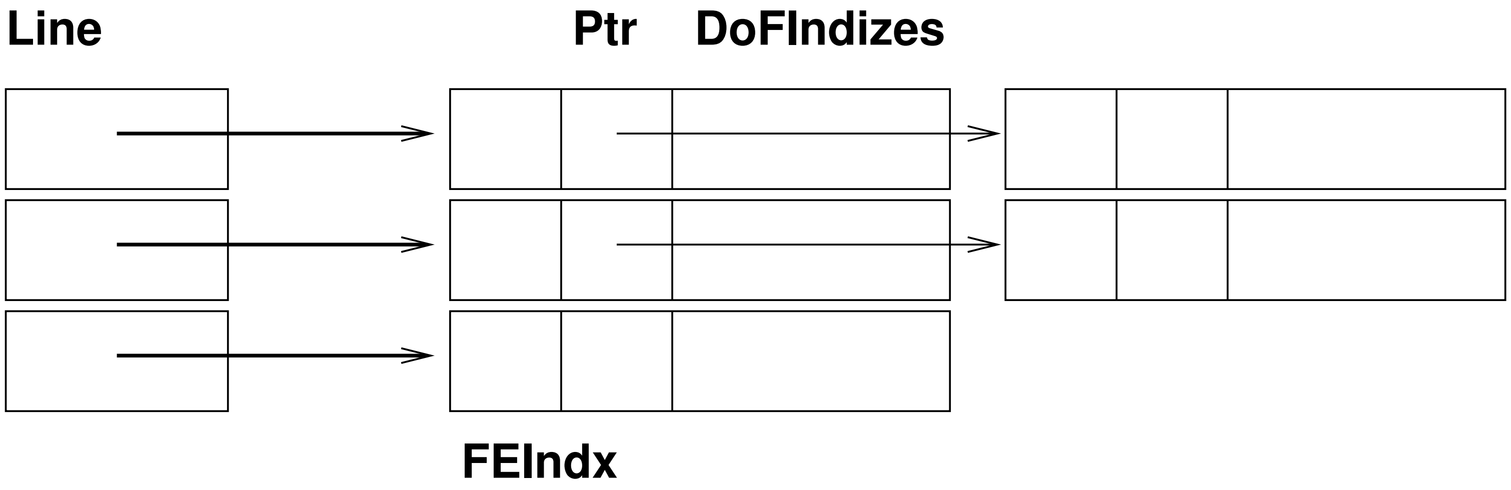
Disadvantages:

- Several parts of the deal.II lib will have to be modified

5. Proposal 2

- Change algorithm which distributes DoFs
- Instead of treating the geometric entities sequential, the DoFs will be distributed cell wise.
- Not completely clear if it is possible

5. Data structure



Line

Ptr

DoFIndizes

FEIndx

6. Summary and Outlook

- hp method works for for DG
- In some cases functions from DoFTools might not yet be available
- Adapting the methods to work with the hpDoFHandler pathes the way for future extensions (hybrid meshes)
- Some work has to be done for cont. elements
- hpMGDoFHandler ??