MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University

Lecture 21.6:

Boundary conditions

Part 3a: Homogenous Dirichlet boundary conditions

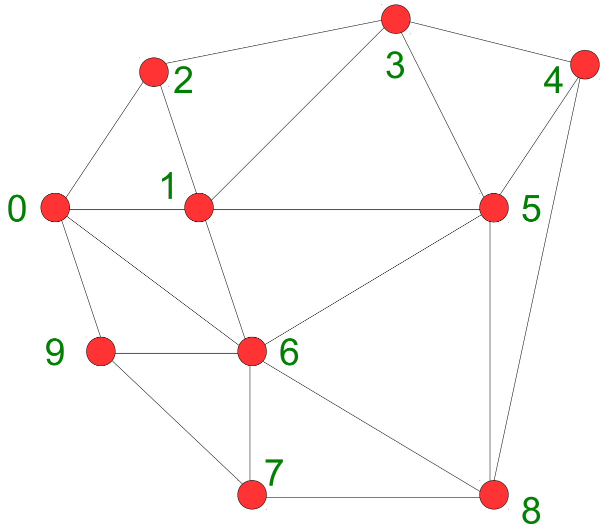
Zero Dirichlet conditions

Consider this simple example:

- Solve the Laplace equation with zero boundary values:

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \Omega \\ u &= 0 \qquad \text{on } \Gamma = \partial\Omega \end{aligned}$$

- Assume we use the following mesh:



{1,5,6}: "real" degrees of freedom

{0,2…4,7…9}: Constrained to zero

Zero Dirichlet conditions

There are three approaches:

- Consider only “free” degrees of freedom
- Consider all degrees of freedom
  – do local assembly without boundary conditions
  – copy to global system
  – eliminate boundary DoFs
- Consider all degrees of freedom
  – do local assembly without boundary conditions
  – eliminate boundary DoFs
  – copy to global system

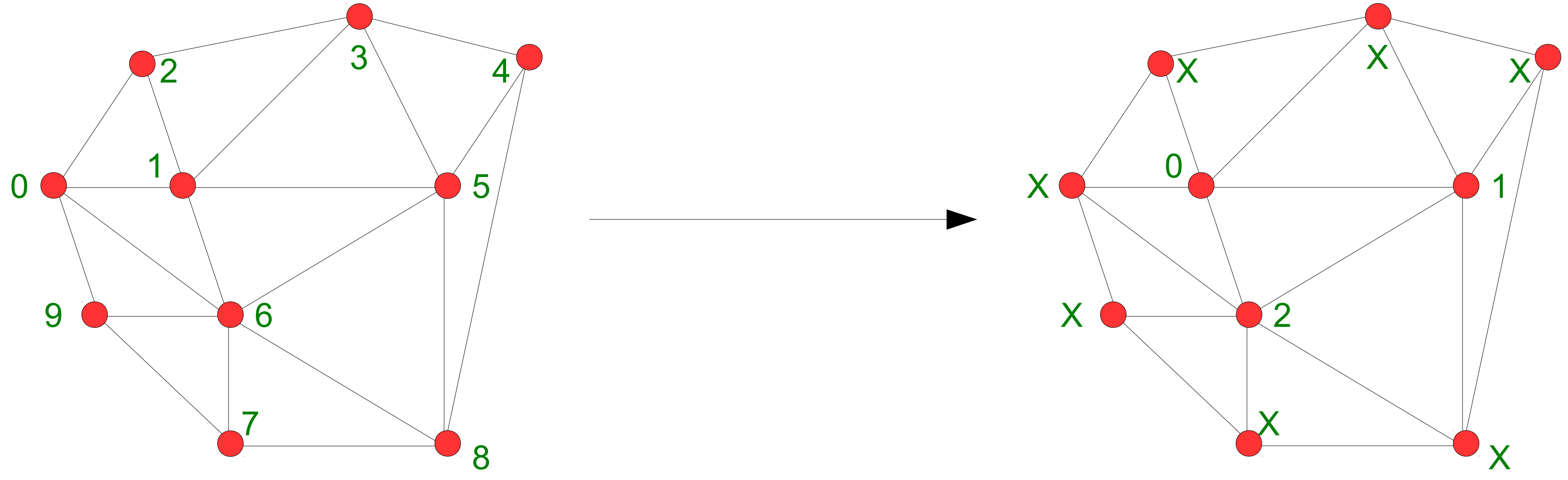
Criteria: All approaches are correct; which one we like best is determined by *considerations of algorithm design*!

Zero Dirichlet conditions

Approach 1:

- Don't enumerate constrained degrees of freedom
- Only enumerate the 3 unconstrained ones:

$$1 \rightarrow 0, \qquad 5 \rightarrow 1, \qquad 6 \rightarrow 2$$

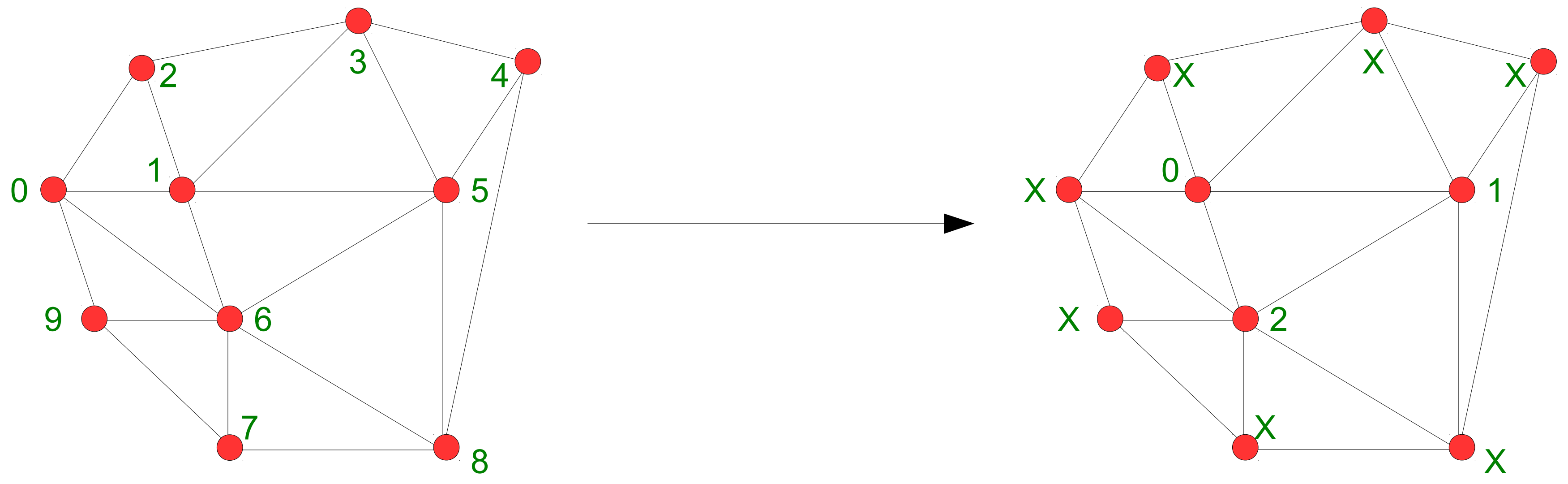


Zero Dirichlet conditions

Approach 1:

- Then assemble a 3x3 linear system as usual:

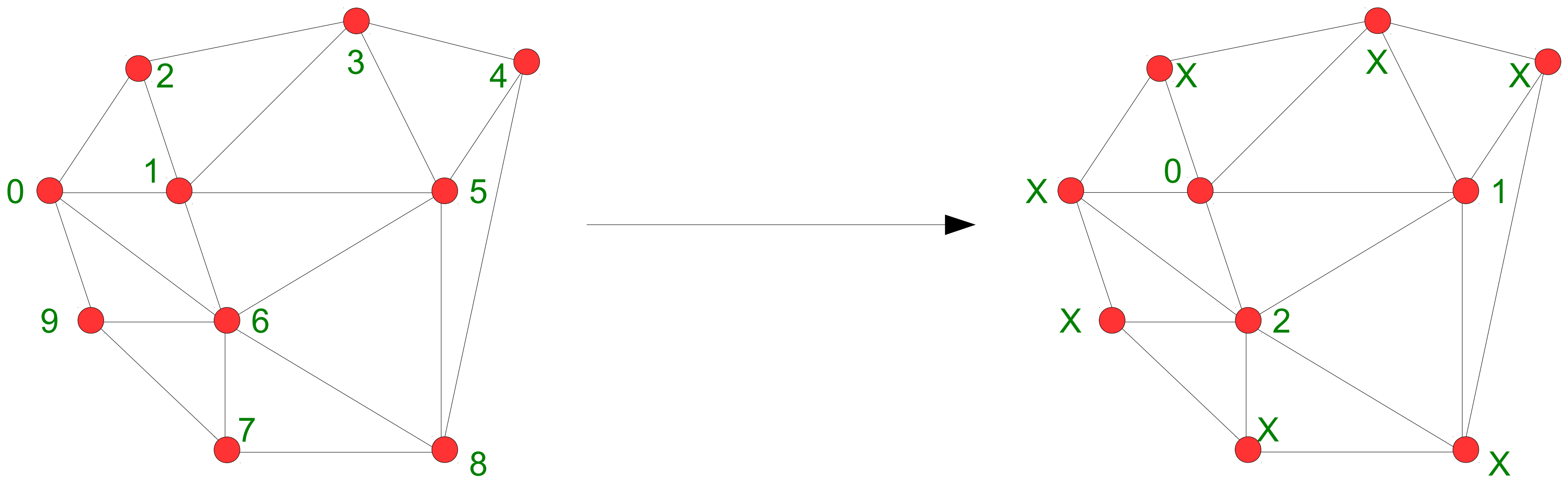
$$\sum_{j=0}^{2} \underbrace{\left[\sum_{K=0}^{11} (\nabla \phi_i, \nabla \phi_j)_K\right]}_{A_{ij}} U_j = \underbrace{\sum_{K=0}^{11} (\phi_i, f)_K}_{F_i} \qquad \forall i = 0 \ldots 2$$



Zero Dirichlet conditions

Advantages of approach 1:

- All degrees of freedom are really "free" (i.e., determined by a linear system)
- Linear system is *much* smaller (3x3 vs 10x10)

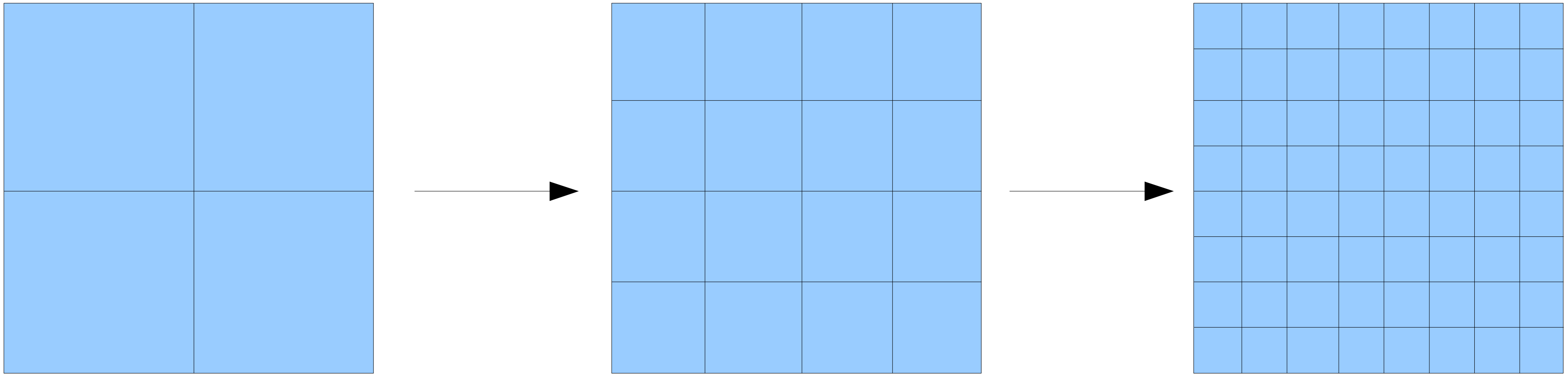


Zero Dirichlet conditions

Advantages of approach 1:

- All degrees of freedom are really "free" (i.e., determined by a linear system)
- Linear system is *much* smaller (3x3 vs 10x10)

But: This size reduction doesn't matter on finer meshes!



1 vs 9 DoFs

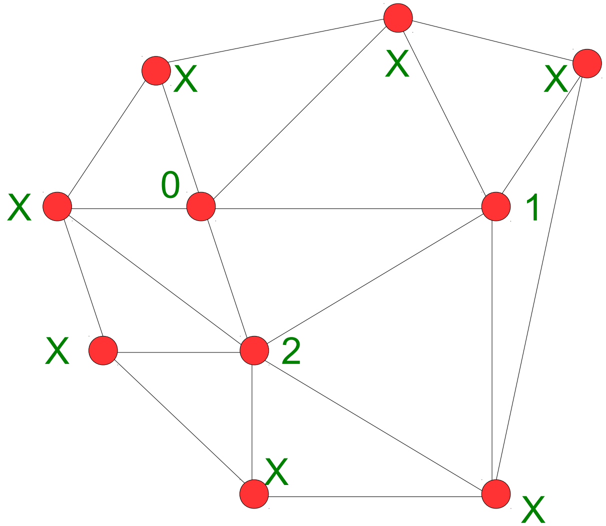
9 vs 25 DoFs

49 vs 81 DoFs

Zero Dirichlet conditions

Disdvantages of approach 1:

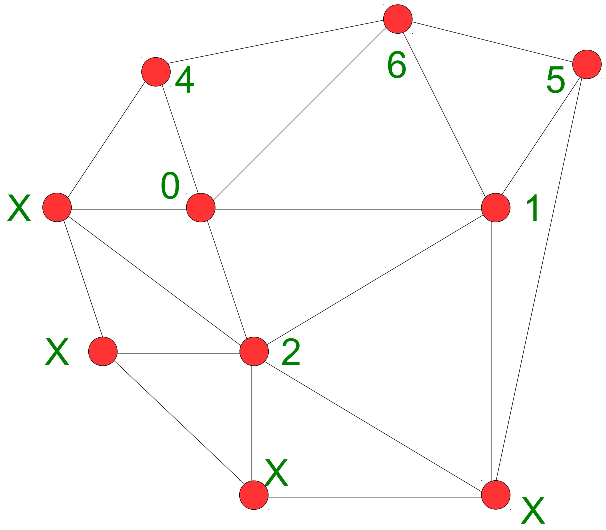
- The FEM is at its best if you can do the same on every cell:
  - when enumerating degrees of freedom
  - when assembling
  - when evaluating the solution at individual points
  - …



Zero Dirichlet conditions

Disdvantages of approach 1:

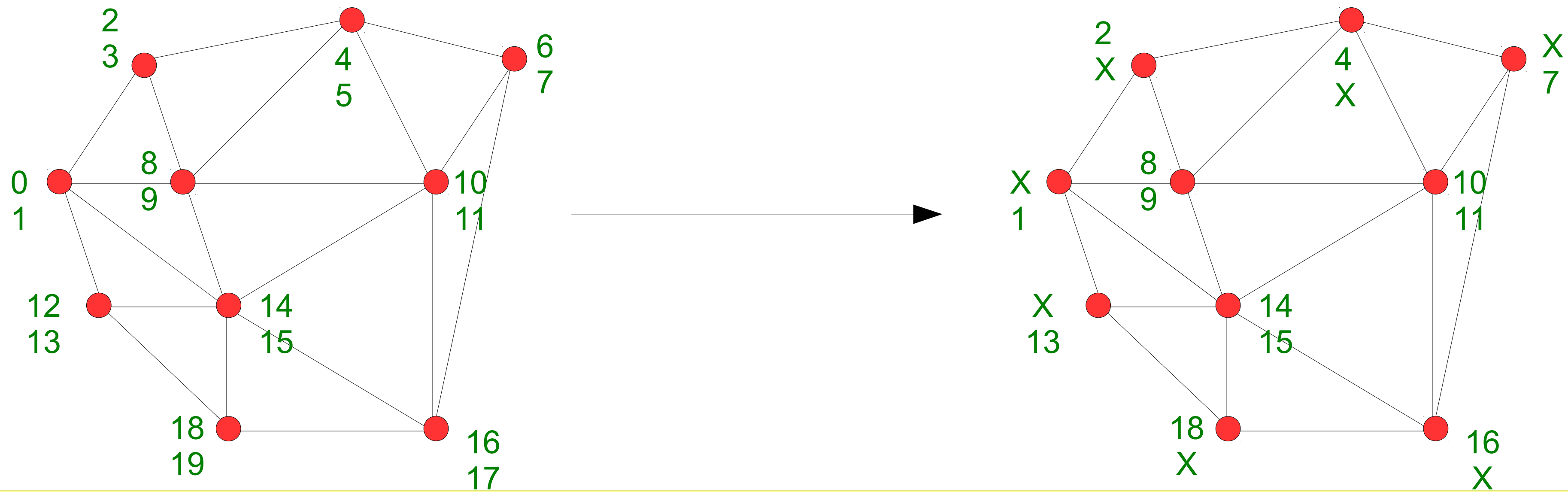
- If Dirichlet conditions only on parts of the boundary: Some boundary nodes would have to be treated differently than others.



Zero Dirichlet conditions

Disdvantages of approach 1:

- What to do in cases of vector-valued problems and conditions of the sort *u.n=0*?
- Example: Fluid flow with velocity components (u_x,u_y)
- Boundary condition here is $n_x u_x = -n_y u_y$



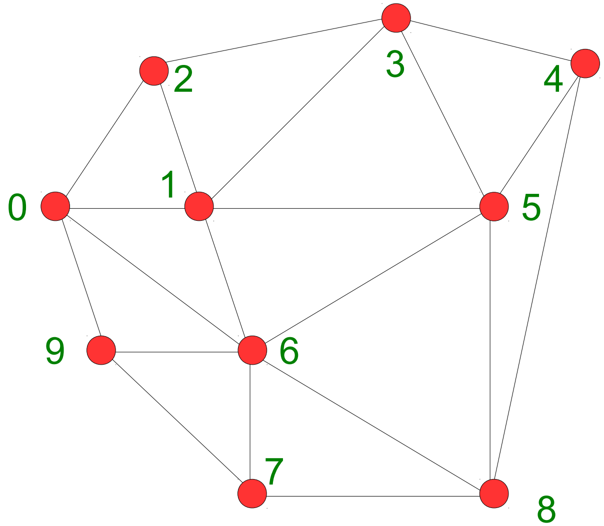
Zero Dirichlet conditions

Approach 2:

- Enumerate all degrees of freedom regardless of boundary conditions

Variation 2a:

- Compute local contributions regardless of boundary DoFs
- Assemble into one big linear system
- Zero out rows and columns for boundary DoFs {0,2,3,4,7,8,9}
- Fix up zero rows

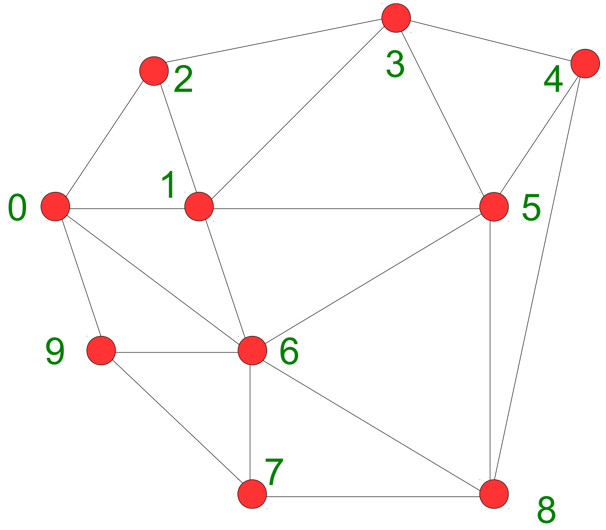


Zero Dirichlet conditions

Example for approach 2a:

- Compute local contributions regardless of boundary DoFs
- Assemble into one big linear system
(forget about sparsity for a moment)

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} & a_{06} & a_{07} & a_{08} & a_{09} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\ a_{60} & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\ a_{70} & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\ a_{80} & a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\ a_{90} & a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{pmatrix}$$

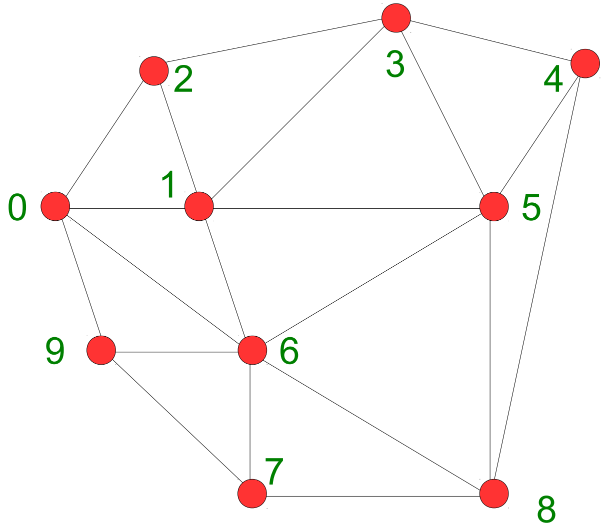


Zero Dirichlet conditions

Example for approach 2a:

- Compute local contributions regardless of boundary DoFs
- Assemble into one big linear system
- Zero out rows and columns for DoFs {0,2,3,4,7,8,9}

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} & a_{06} & a_{07} & a_{08} & a_{09} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\ a_{60} & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\ a_{70} & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\ a_{80} & a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\ a_{90} & a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{pmatrix}$$

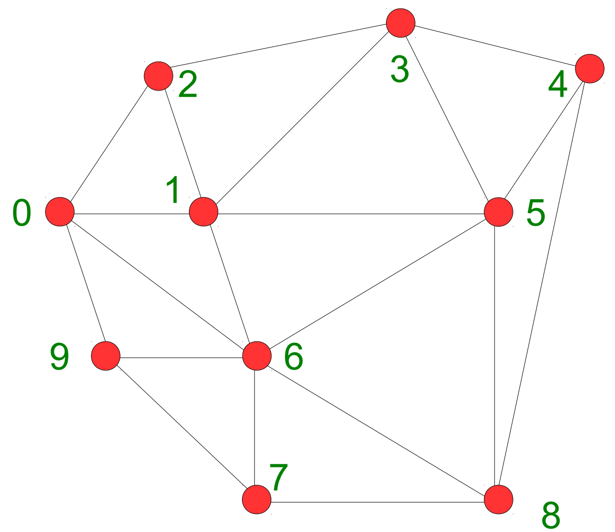


Zero Dirichlet conditions

Example for approach 2a:

- Compute local contributions regardless of boundary DoFs
- Assemble into one big linear system
- Zero out rows and columns for DoFs {0,2,3,4,7,8,9}
  The resulting system has a 3x3 sub-block for “free” DoFs

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{11} & 0 & 0 & 0 & a_{15} & a_{16} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{51} & 0 & 0 & 0 & a_{55} & a_{56} & 0 & 0 & 0 \\ 0 & a_{61} & 0 & 0 & 0 & a_{65} & a_{66} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix} = \begin{pmatrix} 0 \\ f_1 \\ 0 \\ 0 \\ 0 \\ f_5 \\ f_6 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

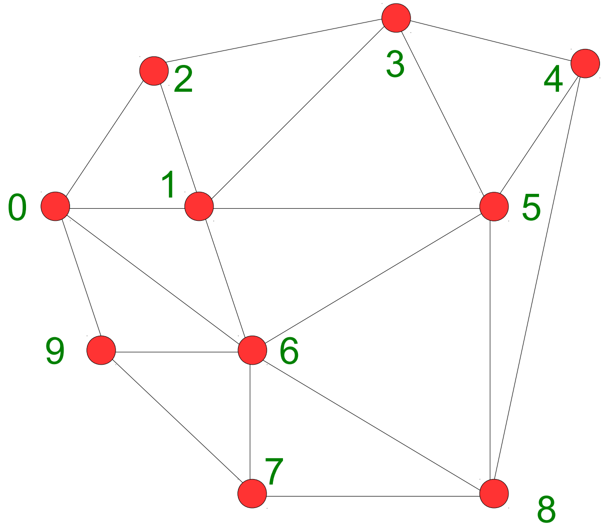


Zero Dirichlet conditions

Example for approach 2a:

- Compute local contributions regardless of boundary DoFs
- Assemble into one big linear system
- Zero out rows and columns for DoFs {0,2,3,4,7,8,9}
- Fix under-determination:

$$
\begin{pmatrix}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{11} & 0 & 0 & 0 & a_{15} & a_{16} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
0 & a_{51} & 0 & 0 & 0 & a_{55} & a_{56} & 0 & 0 & 0 \\
0 & a_{61} & 0 & 0 & 0 & a_{65} & a_{66} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1}
\end{pmatrix}
\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix}
=
\begin{pmatrix} 0 \\ f_1 \\ 0 \\ 0 \\ 0 \\ f_5 \\ f_6 \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$



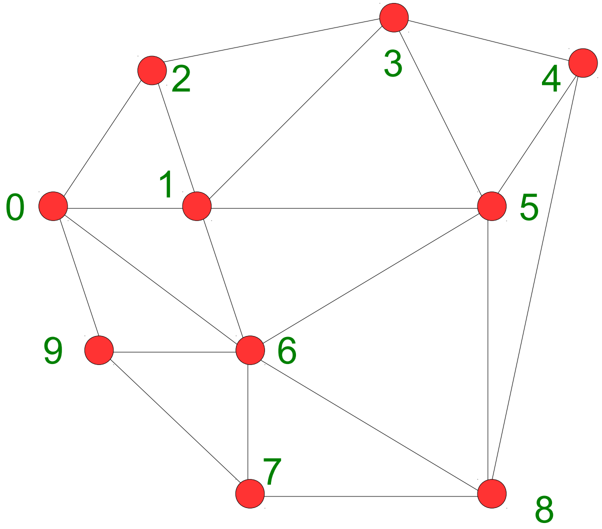
Result for approach 2a:

We get a linear system of the form

$$\begin{pmatrix} 1 & & & & & & & & & \\ & a_{11} & & & & a_{15} & a_{16} & & & \\ & & 1 & & & & & & & \\ & & & 1 & & & & & & \\ & & & & 1 & & & & & \\ & a_{51} & & & & a_{55} & a_{56} & & & \\ & a_{61} & & & & a_{65} & a_{66} & & & \\ & & & & & & & 1 & & \\ & & & & & & & & 1 & \\ & & & & & & & & & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{u_0} \\ u_1 \\ \boldsymbol{u_2} \\ \boldsymbol{u_3} \\ \boldsymbol{u_4} \\ u_5 \\ u_6 \\ \boldsymbol{u_7} \\ \boldsymbol{u_8} \\ \boldsymbol{u_9} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ f_1 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ f_5 \\ f_6 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

The solution satisfies

$$U_0 = U_2 = U_3 = U_4 = U_7 = U_8 = U_9 = 0$$

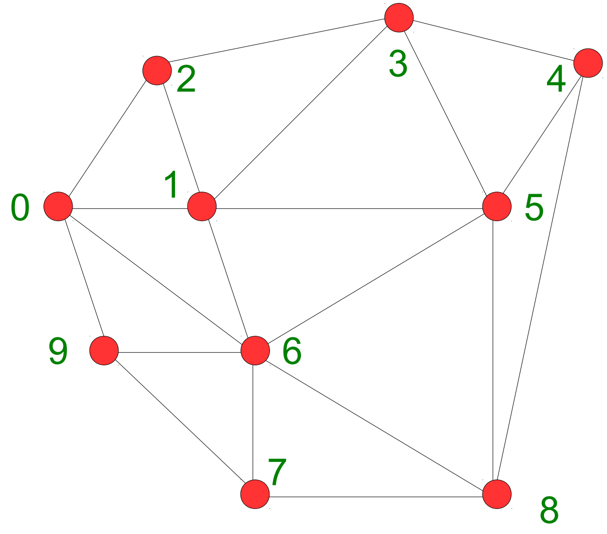


Zero Dirichlet conditions

Result for approach 2a:

We get a linear system of the form

$$\begin{pmatrix}
1 & & & & & & & & & \\
 & \boldsymbol{a_{11}} & & & & \boldsymbol{a_{15}} & \boldsymbol{a_{16}} & & & \\
 & & 1 & & & & & & & \\
 & & & 1 & & & & & & \\
 & & & & 1 & & & & & \\
 & \boldsymbol{a_{51}} & & & & \boldsymbol{a_{55}} & \boldsymbol{a_{56}} & & & \\
 & \boldsymbol{a_{61}} & & & & \boldsymbol{a_{65}} & \boldsymbol{a_{66}} & & & \\
 & & & & & & & 1 & & \\
 & & & & & & & & 1 & \\
 & & & & & & & & & 1
\end{pmatrix}
\begin{pmatrix} u_0 \\ \boldsymbol{u_1} \\ u_2 \\ u_3 \\ u_4 \\ \boldsymbol{u_5} \\ \boldsymbol{u_6} \\ u_7 \\ u_8 \\ u_9 \end{pmatrix}
=
\begin{pmatrix} 0 \\ \boldsymbol{f_1} \\ 0 \\ 0 \\ 0 \\ \boldsymbol{f_5} \\ \boldsymbol{f_6} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



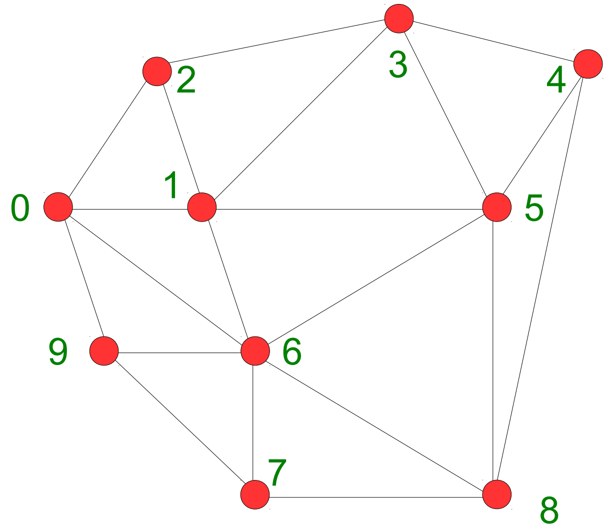
The solution satisfies

$$\begin{pmatrix} a_{11} & a_{15} & a_{16} \\ a_{51} & a_{55} & a_{56} \\ a_{61} & a_{65} & a_{66} \end{pmatrix}
\begin{pmatrix} u_1 \\ u_5 \\ u_6 \end{pmatrix}
=
\begin{pmatrix} f_1 \\ f_5 \\ f_6 \end{pmatrix}$$

Zero Dirichlet conditions

Summary approach 2a:

- Compute local contributions regardless of boundary DoFs
- Assemble into one big linear system
- Zero out rows and columns for boundary DoFs
- Fix up zero rows



This results in a linear system that:

- Has the desired solution
- Is invertible
- Has a few more zeros in it
- Is symmetric if the original matrix was symmetric
- Is SPD if the original matrix was SPD

Zero Dirichlet conditions

Implementation approach 2a (step-4):

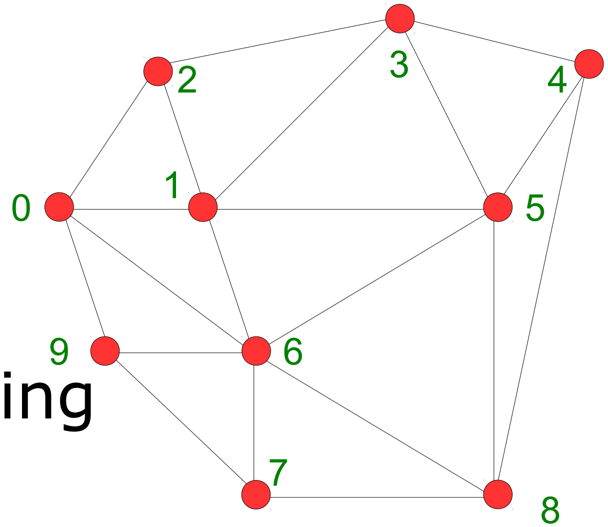
```
void Step4::assemble_system {
    …;
    for (cell=...) {
        ...assemble cell_matrix, cell_rhs...;
        ...add cell_matrix, cell_rhs to system_matrix, system_rhs...;
    }

    std::map<types::global_dof_index,double> boundary_values;
    VectorTools::interpolate_boundary_values (dof_handler, 0,
                                              ZeroFunction<dim>(),
                                              boundary_values);
    MatrixTools::apply_boundary_values (boundary_values,
                                        system_matrix, solution, system_rhs);
}
```

Zero Dirichlet conditions

Summary approach 2a:

- Compute local contributions regardless of boundary DoFs
- Assemble into one big linear system
- Zero out rows and columns for boundary DoFs
- Fix up zero rows



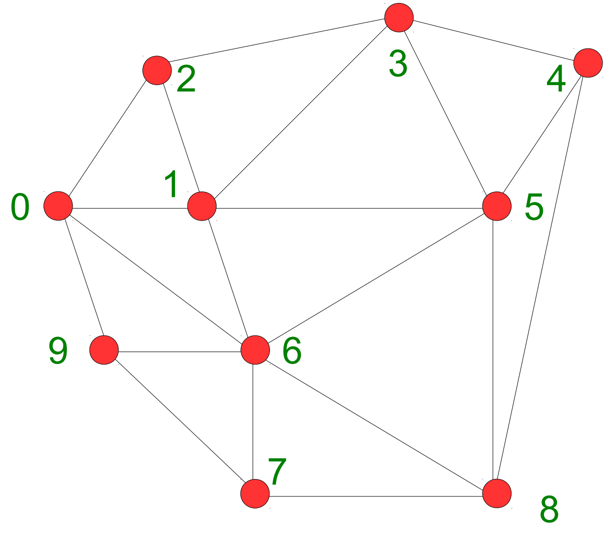
Advantages:

- Can do the exact same thing on every cell during assembly/postprocessing
- Relegate handling b.c. to a separate part of the code, independent of assembly
- Can use same solver regardless of boundary conditions

Zero Dirichlet conditions

Summary approach 2a:

- Compute local contributions regardless of boundary DoFs
- Assemble into one big linear system
- Zero out rows and columns for boundary DoFs
- Fix up zero rows



Disadvantages:

- We have a few more rows and columns
- We have to modify the linear system after global assembly

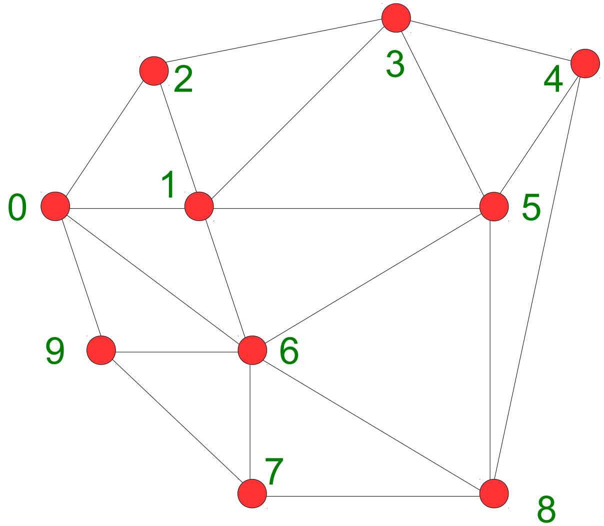
Zero Dirichlet conditions

Approach 2:

- Enumerate all degrees of freedom regardless of boundary conditions

Variation 2b:

- Compute local contributions regardless of boundary DoFs
- Zero out rows and columns for boundary DoFs {0,2,3,4,7,8,9}
- Fix up zero rows
- Assemble into one big linear system



Zero Dirichlet conditions

Example for approach 2b:

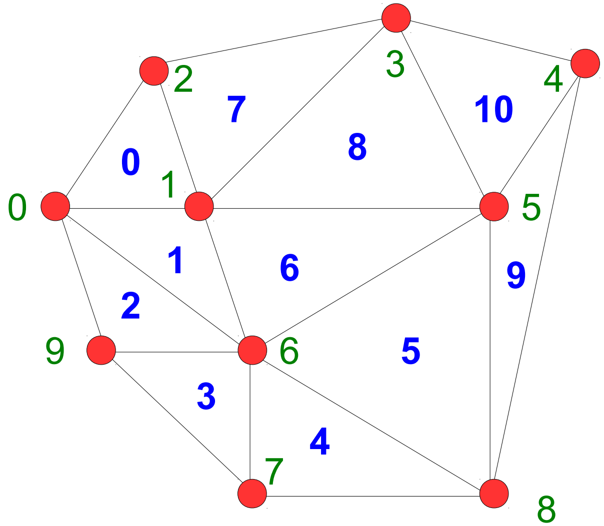
Recall that the linear system

$$AU = F$$

is computed from contributions of all cells:

$$A = \sum_{K=0}^{10} A^K$$

$$F = \sum_{K=0}^{10} F^K$$



Zero Dirichlet conditions

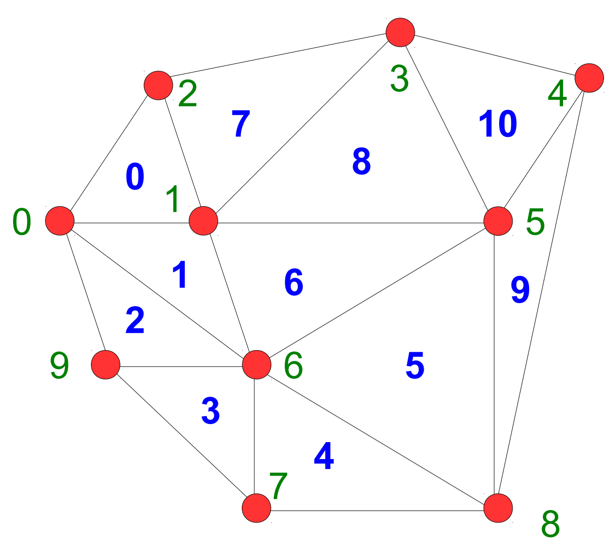
Example for approach 2b:

Recall that the linear system

$$AU = F$$

is computed from contributions of all cells. For example:

$$A^1 = \begin{pmatrix} a^1_{00} & a^1_{01} & 0 & 0 & 0 & 0 & a^1_{06} & 0 & 0 & 0 \\ a^1_{10} & a^1_{11} & 0 & 0 & 0 & 0 & a^1_{16} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a^1_{60} & a^1_{61} & 0 & 0 & 0 & 0 & a^1_{66} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad F^1 = \begin{pmatrix} f^1_0 \\ f^1_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ f^1_6 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



Zero Dirichlet conditions

Example for approach 2b:

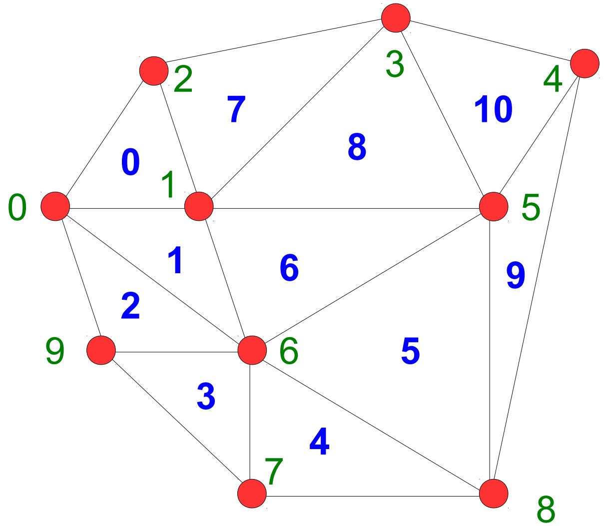
Recall that the linear system

$$AU = F$$

is computed from contributions of all cells.

In reality, of course, we only store the dense 3x3 sub-block:

$$A^{1,\text{loc}} = \begin{pmatrix} a_{00}^1 & a_{01}^1 & a_{06}^1 \\ a_{10}^1 & a_{11}^1 & a_{16}^1 \\ a_{60}^1 & a_{61}^1 & a_{66}^1 \end{pmatrix}, \quad F^{1,\text{loc}} = \begin{pmatrix} f_0^1 \\ f_1^1 \\ f_6^1 \end{pmatrix}$$



Zero Dirichlet conditions

Example for approach 2b:

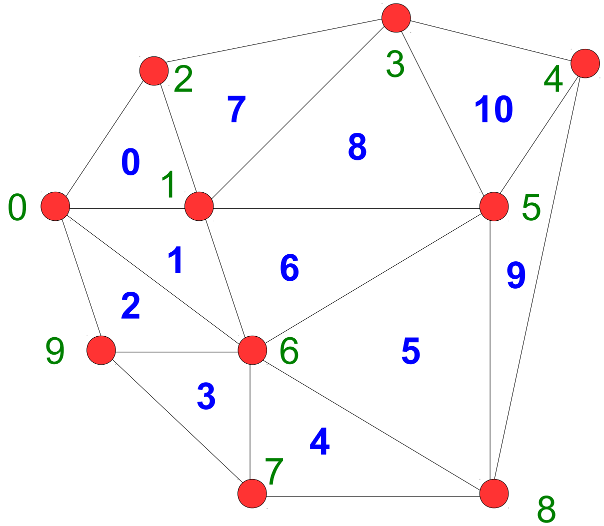
Recall that the linear system

$$AU = F$$

is computed from contributions of all cells.

Zero out rows and columns DoFs {0,2,3,4,7,8,9}:

$$A^{1,\mathrm{loc}} = \begin{pmatrix} a^1_{00} & a^1_{01} & a^1_{06} \\ a^1_{10} & a^1_{11} & a^1_{16} \\ a^1_{60} & a^1_{61} & a^1_{66} \end{pmatrix}, \qquad F^{1,\mathrm{loc}} = \begin{pmatrix} f^1_0 \\ f^1_1 \\ f^1_6 \end{pmatrix}$$



Zero Dirichlet conditions

Example for approach 2b:

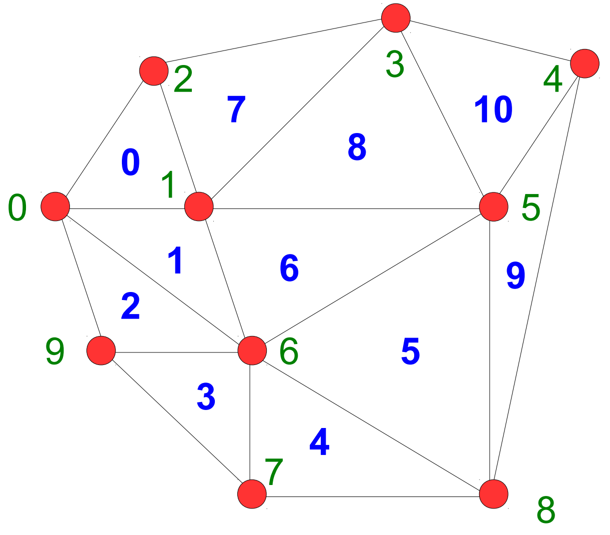
Recall that the linear system

$$AU = F$$

is computed from contributions of all cells.

Zero out rows and columns DoFs {0,2,3,4,7,8,9}:

$$A^{1,\mathrm{loc}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & a^1_{11} & a^1_{16} \\ 0 & a^1_{61} & a^1_{66} \end{pmatrix}, \quad F^{1,\mathrm{loc}} = \begin{pmatrix} 0 \\ f^1_1 \\ f^1_6 \end{pmatrix}$$



Zero Dirichlet conditions

Example for approach 2b:

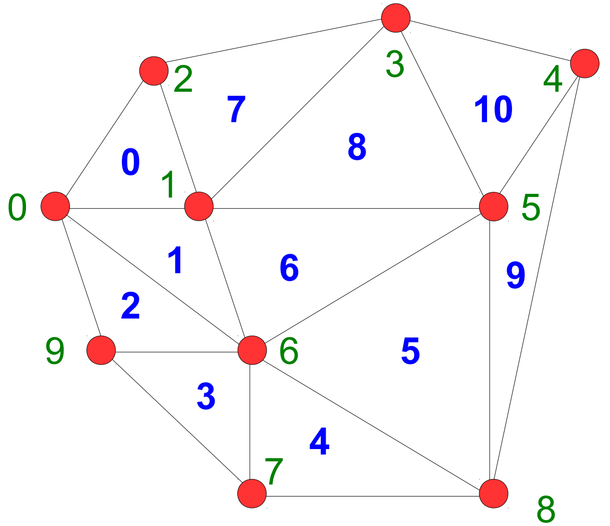
Recall that the linear system

$$AU = F$$

is computed from contributions of all cells.

Fix up under-determinedness:

$$A^{1,\mathrm{loc}} = \begin{pmatrix} \mathbf{1} & 0 & 0 \\ 0 & a^1_{11} & a^1_{16} \\ 0 & a^1_{61} & a^1_{66} \end{pmatrix}, \quad F^{1,\mathrm{loc}} = \begin{pmatrix} 0 \\ f^1_1 \\ f^1_6 \end{pmatrix}$$



Zero Dirichlet conditions

Example for approach 2b:

Recall that the linear system

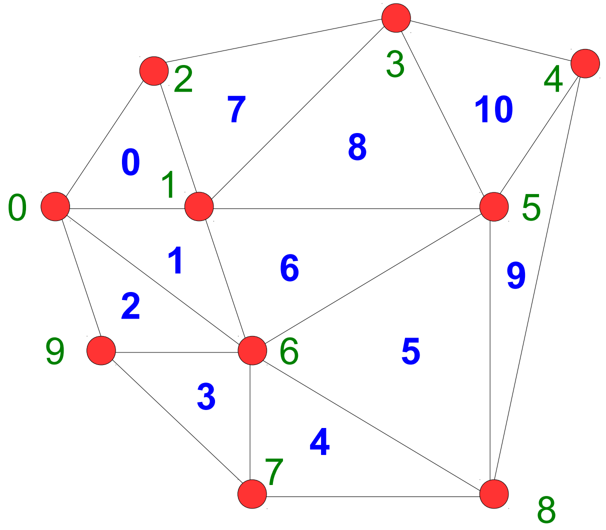
$$AU = F$$

is computed from contributions of all cells.

Then add to global linear system:

$$A := A + \text{expand}\left(A^{1,\text{loc}}\right)$$

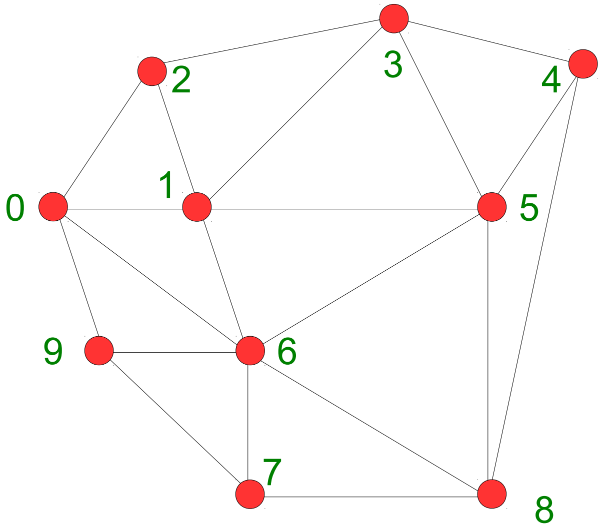
$$F := F + \text{expand}\left(F^{1,\text{loc}}\right)$$



Zero Dirichlet conditions

Summary approach 2b:

- Compute local contributions regardless of boundary DoFs
- Zero out rows and columns for boundary DoFs
- Fix up zero rows
- Assemble into one big linear system



Actual implementation:

- The last three operations are all done at the same time
- The local contributions are not modified

Zero Dirichlet conditions

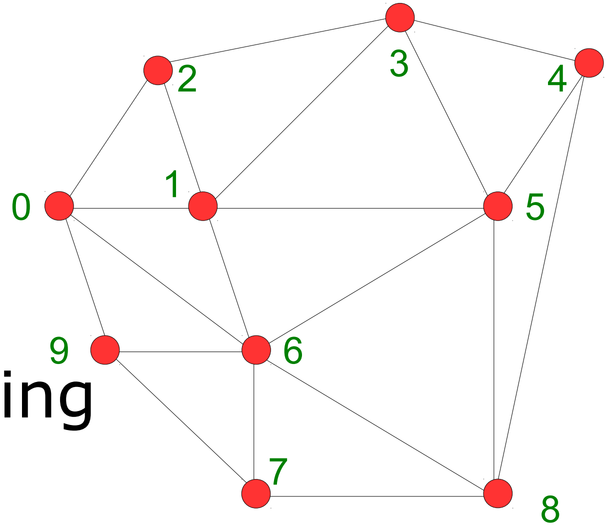
Implementation approach 2b (step-6):

```
void Step6::assemble_system {
   ConstraintMatrix constraints;
   VectorTools::interpolate_boundary_values (dof_handler, 0,
                                           ZeroFunction<dim>(),
                                           constraints);
   …;
   for (cell=...) {
      ...assemble cell_matrix, cell_rhs...;
      cell->get_dof_indices (local_dof_indices);
      constraints.distribute_local_to_global (cell_matrix, cell_rhs,
                                            local_dof_indices,
                                            system_matrix, system_rhs);
   }
}
```

Zero Dirichlet conditions

Summary approach 2b:

- Compute local contributions regardless of boundary DoFs
- Zero out rows and columns for boundary DoFs
- Fix up zero rows
- Assemble into one big linear system



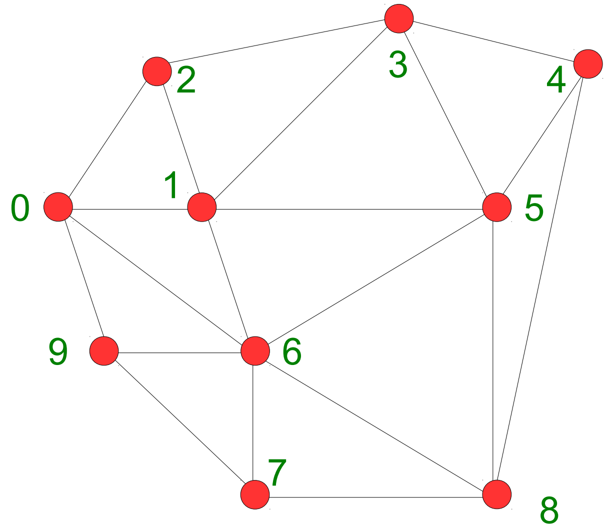
Advantages:

- Can do the exact same thing on every cell during assembly/postprocessing
- Relegate handling b.c. to a separate part of the code, independent of assembly
- Can use same solver regardless of boundary conditions

Zero Dirichlet conditions

Summary approach 2b:

- Compute local contributions regardless of boundary DoFs
- Zero out rows and columns for boundary DoFs
- Fix up zero rows
- Assemble into one big linear system



Disdvantages:

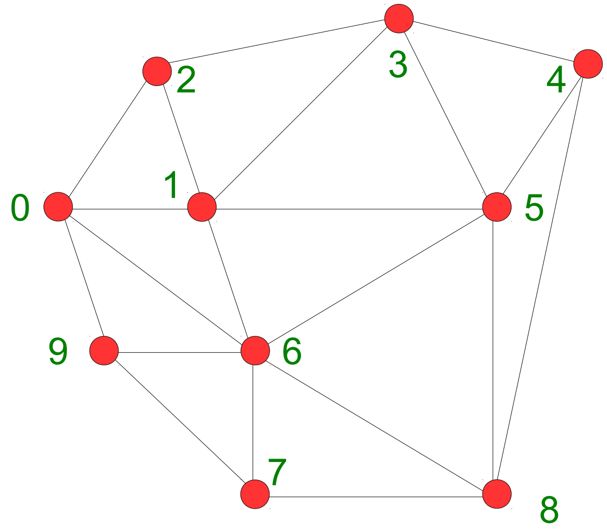
- We have a few more rows and columns
- We have to modify the linear system after local integration or during copy-local-to-global (both options are easy and cheap)

Summary

We are driven by algorithm design considerations:

- We want to do the exact same thing on every cell
- We want to isolate handling of boundary values to as few code pieces as possible

- Dealing with small, dense, local contributions is simpler than with global linear systems



Consequently:

- Enumerate all degrees of freedom
- Integrate locally regardless of boundary conditions
- Deal with boundary values during assembly or after

A final implementation detail

As a postscript:

Consider a linear system of the form

$$\begin{pmatrix} 1 &&&&&&&&& \\ & \boldsymbol{a_{11}} &&&& \boldsymbol{a_{15}} & \boldsymbol{a_{16}} &&& \\ && 1 &&&&&&& \\ &&& 1 &&&&&& \\ &&&& 1 &&&&& \\ & \boldsymbol{a_{51}} &&&& \boldsymbol{a_{55}} & \boldsymbol{a_{56}} &&& \\ & \boldsymbol{a_{61}} &&&& \boldsymbol{a_{65}} & \boldsymbol{a_{66}} &&& \\ &&&&&&& 1 && \\ &&&&&&&& 1 & \\ &&&&&&&&& 1 \end{pmatrix} \begin{pmatrix} u_0 \\ \boldsymbol{u_1} \\ u_2 \\ u_3 \\ u_4 \\ \boldsymbol{u_5} \\ \boldsymbol{u_6} \\ u_7 \\ u_8 \\ u_9 \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{f_1} \\ 0 \\ 0 \\ 0 \\ \boldsymbol{f_5} \\ \boldsymbol{f_6} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

This matrix is ill-conditioned if

$$|a_{ij}| \gg 1 \quad \text{or} \quad |a_{ij}| \ll 1$$

Consequence:

- We should not place 1.0 on the diagonal
- Algorithm instead use values “similar to other entries”

MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University