MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University

Lecture 21.65:

Boundary conditions

Part 3b: Inhomogenous Dirichlet boundary conditions

Nonzero Dirichlet conditions

Consider this simple example:

- Solve the Laplace equation with nonzero boundary values:

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

- For simplicity, assume that g can be exactly represented by (the trace of) finite element functions

- If this is not the case: we will instead solve a problem with an *interpolation* of g .

Nonzero Dirichlet conditions

General idea:

Decompose the solution of

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

into two parts,

$$u = u_0 + \tilde{g}$$

where

$$\begin{aligned} -\Delta u_0 &= f + \Delta\tilde{g} && \text{in } \Omega \\ u_0 &= 0 && \text{on } \Gamma = \partial\Omega \end{aligned} \qquad \begin{aligned} \tilde{g} &= \text{arbitrary} && \text{in } \Omega \\ \tilde{g} &= g && \text{on } \Gamma = \partial\Omega \end{aligned}$$

Note: There are many choices for $\tilde{g}$!

Nonzero Dirichlet conditions

At the discrete level:

Solve numerically the equations

$$\begin{aligned} -\Delta u_0 &= f + \Delta \tilde{g} && \text{in } \Omega \\ u_0 &= 0 && \text{on } \Gamma = \partial\Omega \end{aligned} \qquad \begin{aligned} \tilde{g} &= \text{arbitrary} && \text{in } \Omega \\ \tilde{g} &= g && \text{on } \Gamma = \partial\Omega \end{aligned}$$

for u_h with a particular

$$\tilde{g} \in V_h$$

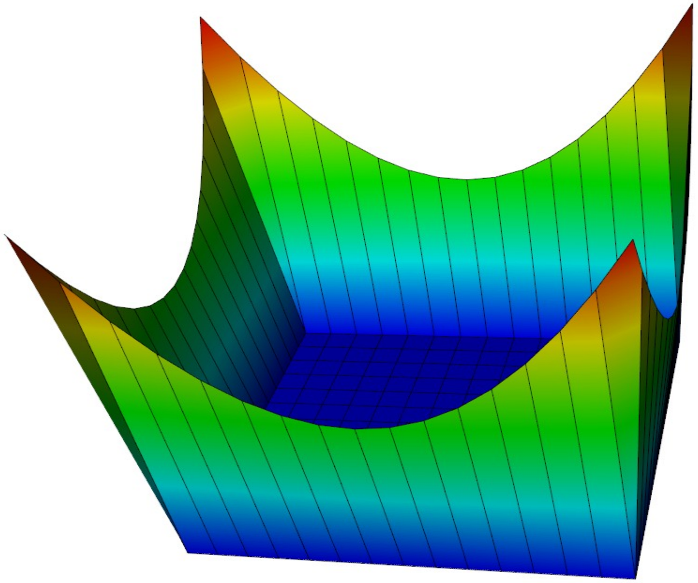
Then reconstruct the solution as

$$u_h = u_{h,0} + \tilde{g} \in V_h$$

Nonzero Dirichlet conditions

Concretely: Choose $\tilde{g} \in V_h$ so that
– it is zero on all interior nodes, and
– correct on the boundary

Example: In step-4, we use $g=(4x^4+y^4)$:



Approach

The strategy is then similar to approach 2 before:

- Assemble the system as if there were no boundary conditions. Obtain

$$AU = F$$

- Split the discrete solution into

$$U = U_0 + \tilde{G}$$

- Solve

$$AU_0 = F - A\tilde{G}$$

  with zero Dirichlet conditions, by doing the same modifications as in the previous lecture.

- Obtain the final solution as $U = U_0 + \tilde{G}$

Observations

Naive approach:

- Assemble the system as if there were no boundary conditions. Obtain

$$AU = F$$

- Next compute

$$H = F - A\tilde{G}$$

- Apply the boundary condition modifications to

$$AU_0 = H$$

- Solve the modified linear system
- Then compute $U = U_0 + \tilde{G}$

Observations

Naive approach:

- Assemble the system as if there were no boundary conditions. Obtain

$$AU = F$$

- Really compute

$$H = F - A\tilde{G}$$

- Apply the modifications to

$$AU_0 = H$$

- Solve the modified linear system
- The compute $U = U_0 + \tilde{G}$

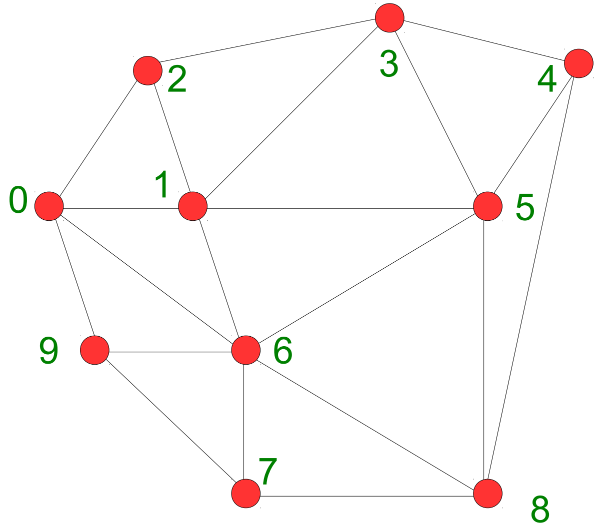
Can all be done in one operation!

Observations

A smarter approach:

- Recall that we chose $\tilde{g}$ (and consequently $\tilde{G}$) in a particular way
- Specifically, for the example from the previous lecture:

$$\tilde{G} = \begin{pmatrix} g(x_0) \\ 0 \\ g(x_2) \\ g(x_3) \\ g(x_4) \\ 0 \\ 0 \\ g(x_7) \\ g(x_8) \\ g(x_9) \end{pmatrix}$$



- On finer meshes, $\tilde{G}$ will be almost all zeros.

Observations

Next, consider the computing the product $A\tilde{G}$ **:**

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} & a_{06} & a_{07} & a_{08} & a_{09} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\ a_{60} & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\ a_{70} & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\ a_{80} & a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\ a_{90} & a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99} \end{pmatrix} \begin{pmatrix} g(x_0) \\ 0 \\ g(x_2) \\ g(x_3) \\ g(x_4) \\ 0 \\ 0 \\ g(x_7) \\ g(x_8) \\ g(x_9) \end{pmatrix} = \begin{pmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{40} \\ a_{50} \\ a_{60} \\ a_{70} \\ a_{80} \\ a_{90} \end{pmatrix} g(x_0) + \begin{pmatrix} a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{42} \\ a_{52} \\ a_{62} \\ a_{72} \\ a_{82} \\ a_{92} \end{pmatrix} g(x_2) + \ldots + \begin{pmatrix} a_{09} \\ a_{19} \\ a_{29} \\ a_{39} \\ a_{49} \\ a_{59} \\ a_{69} \\ a_{79} \\ a_{89} \\ a_{99} \end{pmatrix} g(x_9)$$

In other words: We need to touch exactly those columns of *A* that we will zero out next.

Algorithm

Algorithm: When modifying the linear system, do this:

- For each boundary DoF i,
  - subtract $g(x_i)$ times the ith column of A from the right hand side
  - zero out the ith column of A
  - zero out the ith row of A
  - set $A_{ii} = 1$
  - set the ith entry of the right hand side to $g(x_i)$
- Automatically yields the correct linear system
- Solution vector automatically satisfies $U_i=g(x_i)$

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- Start with:

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

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- *i=0*: Subtract multiple of row *i* from right hand side:

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} & a_{06} & a_{07} & a_{08} & a_{09} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\ a_{60} & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\ a_{70} & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\ a_{80} & a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\ a_{90} & a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix} = \begin{pmatrix} f_0 - a_{00} g(x_0) \\ f_1 - a_{10} g(x_0) \\ f_2 - a_{20} g(x_0) \\ f_3 - a_{30} g(x_0) \\ f_4 - a_{40} g(x_0) \\ f_5 - a_{50} g(x_0) \\ f_6 - a_{60} g(x_0) \\ f_7 - a_{70} g(x_0) \\ f_8 - a_{80} g(x_0) \\ f_9 - a_{90} g(x_0) \end{pmatrix}$$

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- *i=0*: Zero out column *i*:

$$
\begin{pmatrix}
0 & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} & a_{06} & a_{07} & a_{08} & a_{09} \\
0 & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\
0 & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\
0 & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\
0 & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\
0 & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\
0 & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\
0 & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\
0 & a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\
0 & a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99}
\end{pmatrix}
\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix}
=
\begin{pmatrix}
f_0 - a_{00} g(x_0) \\
f_1 - a_{10} g(x_0) \\
f_2 - a_{20} g(x_0) \\
f_3 - a_{30} g(x_0) \\
f_4 - a_{40} g(x_0) \\
f_5 - a_{50} g(x_0) \\
f_6 - a_{60} g(x_0) \\
f_7 - a_{70} g(x_0) \\
f_8 - a_{80} g(x_0) \\
f_9 - a_{90} g(x_0)
\end{pmatrix}
$$

Note: In practice, we zero out each matrix element immediately after using it for the right hand side.

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- *i=0*: Zero out row *i*:

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\
0 & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\
0 & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\
0 & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\
0 & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\
0 & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\
0 & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\
0 & a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\
0 & a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99}
\end{pmatrix}
\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix}
=
\begin{pmatrix}
0 \\
f_1 - a_{10} g(x_0) \\
f_2 - a_{20} g(x_0) \\
f_3 - a_{30} g(x_0) \\
f_4 - a_{40} g(x_0) \\
f_5 - a_{50} g(x_0) \\
f_6 - a_{60} g(x_0) \\
f_7 - a_{70} g(x_0) \\
f_8 - a_{80} g(x_0) \\
f_9 - a_{90} g(x_0)
\end{pmatrix}
$$

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- *i=0*: Set diagonal entry to one:

$$
\begin{pmatrix}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\
0 & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\
0 & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\
0 & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\
0 & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\
0 & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\
0 & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\
0 & a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\
0 & a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99}
\end{pmatrix}
\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix}
=
\begin{pmatrix}
0 \\
f_1 - a_{10} g(x_0) \\
f_2 - a_{20} g(x_0) \\
f_3 - a_{30} g(x_0) \\
f_4 - a_{40} g(x_0) \\
f_5 - a_{50} g(x_0) \\
f_6 - a_{60} g(x_0) \\
f_7 - a_{70} g(x_0) \\
f_8 - a_{80} g(x_0) \\
f_9 - a_{90} g(x_0)
\end{pmatrix}
$$

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- *i=0*: Set right hand side to correct value:

$$
\begin{pmatrix}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\
0 & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\
0 & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\
0 & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\
0 & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\
0 & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\
0 & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\
0 & a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\
0 & a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99}
\end{pmatrix}
\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix}
=
\begin{pmatrix}
g(x_0) \\
f_1 - a_{10} g(x_0) \\
f_2 - a_{20} g(x_0) \\
f_3 - a_{30} g(x_0) \\
f_4 - a_{40} g(x_0) \\
f_5 - a_{50} g(x_0) \\
f_6 - a_{60} g(x_0) \\
f_7 - a_{70} g(x_0) \\
f_8 - a_{80} g(x_0) \\
f_9 - a_{90} g(x_0)
\end{pmatrix}
$$

Note: This already guarantees $U_0 = g(x_0)$.

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- *i=2*: Subtract multiple of row *i* from right hand side:

$$
\begin{pmatrix}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\
0 & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\
0 & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\
0 & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\
0 & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\
0 & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\
0 & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\
0 & a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\
0 & a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99}
\end{pmatrix}
\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix}
=
\begin{pmatrix}
g(x_0) \\
f_1 - a_{10} g(x_0) - a_{12} g(x_2) \\
f_2 - a_{20} g(x_0) - a_{22} g(x_2) \\
f_3 - a_{30} g(x_0) - a_{32} g(x_2) \\
f_4 - a_{40} g(x_0) - a_{42} g(x_2) \\
f_5 - a_{50} g(x_0) - a_{52} g(x_2) \\
f_6 - a_{60} g(x_0) - a_{62} g(x_2) \\
f_7 - a_{70} g(x_0) - a_{72} g(x_2) \\
f_8 - a_{80} g(x_0) - a_{82} g(x_2) \\
f_9 - a_{90} g(x_0) - a_{92} g(x_2)
\end{pmatrix}
$$

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- *i=2*: Zero out column *i*:

$$
\begin{pmatrix}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{11} & 0 & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\
0 & a_{21} & 0 & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\
0 & a_{31} & 0 & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\
0 & a_{41} & 0 & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\
0 & a_{51} & 0 & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\
0 & a_{61} & 0 & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\
0 & a_{71} & 0 & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\
0 & a_{81} & 0 & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\
0 & a_{91} & 0 & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99}
\end{pmatrix}
\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix}
=
\begin{pmatrix}
g(x_0) \\
f_1 - a_{10} g(x_0) - a_{12} g(x_2) \\
f_2 - a_{20} g(x_0) - a_{22} g(x_2) \\
f_3 - a_{30} g(x_0) - a_{32} g(x_2) \\
f_4 - a_{40} g(x_0) - a_{42} g(x_2) \\
f_5 - a_{50} g(x_0) - a_{52} g(x_2) \\
f_6 - a_{60} g(x_0) - a_{62} g(x_2) \\
f_7 - a_{70} g(x_0) - a_{72} g(x_2) \\
f_8 - a_{80} g(x_0) - a_{82} g(x_2) \\
f_9 - a_{90} g(x_0) - a_{92} g(x_2)
\end{pmatrix}
$$

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- *i=2*: Zero out row *i*:

$$\begin{pmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{11} & 0 & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{31} & 0 & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ 0 & a_{41} & 0 & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\ 0 & a_{51} & 0 & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\ 0 & a_{61} & 0 & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\ 0 & a_{71} & 0 & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\ 0 & a_{81} & 0 & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\ 0 & a_{91} & 0 & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix} = \begin{pmatrix} g(x_0) \\ f_1 - a_{10} g(x_0) - a_{12} g(x_2) \\ f_2 - a_{20} g(x_0) - a_{22} g(x_2) \\ f_3 - a_{30} g(x_0) - a_{32} g(x_2) \\ f_4 - a_{40} g(x_0) - a_{42} g(x_2) \\ f_5 - a_{50} g(x_0) - a_{52} g(x_2) \\ f_6 - a_{60} g(x_0) - a_{62} g(x_2) \\ f_7 - a_{70} g(x_0) - a_{72} g(x_2) \\ f_8 - a_{80} g(x_0) - a_{82} g(x_2) \\ f_9 - a_{90} g(x_0) - a_{92} g(x_2) \end{pmatrix}$$

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- *i=2*: Set diagonal entry to one:

$$\begin{pmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{11} & 0 & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{31} & 0 & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ 0 & a_{41} & 0 & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\ 0 & a_{51} & 0 & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\ 0 & a_{61} & 0 & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\ 0 & a_{71} & 0 & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\ 0 & a_{81} & 0 & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\ 0 & a_{91} & 0 & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix} = \begin{pmatrix} g(x_0) \\ f_1 - a_{10} g(x_0) - a_{12} g(x_2) \\ f_2 - a_{20} g(x_0) - a_{22} g(x_2) \\ f_3 - a_{30} g(x_0) - a_{32} g(x_2) \\ f_4 - a_{40} g(x_0) - a_{42} g(x_2) \\ f_5 - a_{50} g(x_0) - a_{52} g(x_2) \\ f_6 - a_{60} g(x_0) - a_{62} g(x_2) \\ f_7 - a_{70} g(x_0) - a_{72} g(x_2) \\ f_8 - a_{80} g(x_0) - a_{82} g(x_2) \\ f_9 - a_{90} g(x_0) - a_{92} g(x_2) \end{pmatrix}$$

Algorithm

Algorithm: For boundary nodes *{0,2,3,4,7,8,9}*:

- *i=2*: Set right hand side to correct value:

$$
\begin{pmatrix}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{11} & 0 & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{31} & 0 & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\
0 & a_{41} & 0 & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\
0 & a_{51} & 0 & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\
0 & a_{61} & 0 & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\
0 & a_{71} & 0 & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\
0 & a_{81} & 0 & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\
0 & a_{91} & 0 & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99}
\end{pmatrix}
\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix}
=
\begin{pmatrix}
g(x_0) \\
f_1 - a_{10} g(x_0) - a_{12} g(x_2) \\
g(x_2) \\
f_3 - a_{30} g(x_0) - a_{32} g(x_2) \\
f_4 - a_{40} g(x_0) - a_{42} g(x_2) \\
f_5 - a_{50} g(x_0) - a_{52} g(x_2) \\
f_6 - a_{60} g(x_0) - a_{62} g(x_2) \\
f_7 - a_{70} g(x_0) - a_{72} g(x_2) \\
f_8 - a_{80} g(x_0) - a_{82} g(x_2) \\
f_9 - a_{90} g(x_0) - a_{92} g(x_2)
\end{pmatrix}
$$

Note: This now guarantees $U_0=g(x_0)$ and $U_2=g(x_2)$.

Algorithm

Algorithm: Do the same for *i=3,4,7,8,9* and obtain:

$$\begin{pmatrix} 1 &&&&&&&&& \\ & \boldsymbol{a_{11}} &&&& \boldsymbol{a_{15}} & \boldsymbol{a_{16}} &&& \\ && 1 &&&&&&& \\ &&& 1 &&&&&& \\ &&&& 1 &&&&& \\ & \boldsymbol{a_{51}} &&&& \boldsymbol{a_{55}} & \boldsymbol{a_{56}} &&& \\ & \boldsymbol{a_{61}} &&&& \boldsymbol{a_{65}} & \boldsymbol{a_{66}} &&& \\ &&&&&&& 1 && \\ &&&&&&&& 1 & \\ &&&&&&&&& 1 \end{pmatrix} \begin{pmatrix} u_0 \\ \boldsymbol{u_1} \\ u_2 \\ u_3 \\ u_4 \\ \boldsymbol{u_5} \\ \boldsymbol{u_6} \\ u_7 \\ u_8 \\ u_9 \end{pmatrix} = \begin{pmatrix} g(x_0) \\ \boldsymbol{f_1} - \sum_{i\in\{0,2,3,4,7,8,9\}} \boldsymbol{a_{1i} g}(x_i) \\ g(x_2) \\ g(x_3) \\ g(x_4) \\ \boldsymbol{f_5} - \sum_{i\in\{0,2,3,4,7,8,9\}} \boldsymbol{a_{5i} g}(x_i) \\ \boldsymbol{f_6} - \sum_{i\in\{0,2,3,4,7,8,9\}} \boldsymbol{a_{6i} g}(x_i) \\ g(x_7) \\ g(x_8) \\ g(x_9) \end{pmatrix}$$

The solution satisfies the boundary values and:

$$\begin{pmatrix} a_{11} & a_{15} & a_{16} \\ a_{51} & a_{55} & a_{56} \\ a_{61} & a_{65} & a_{66} \end{pmatrix} \begin{pmatrix} u_1 \\ u_5 \\ u_6 \end{pmatrix} = \begin{pmatrix} f_1 - \sum_{i\in\{0,2,3,4,7,8,9\}} a_{1i} g(x_i) \\ f_5 - \sum_{i\in\{0,2,3,4,7,8,9\}} a_{5i} g(x_i) \\ f_6 - \sum_{i\in\{0,2,3,4,7,8,9\}} a_{6i} g(x_i) \end{pmatrix}$$

Summary

Summary for dealing with nonzero Dirichlet boundary conditions:

- Conceptually, we decompose the solution into
  – a part that is zero at the boundary
  – a part that satisfies the boundary values

- Then use the algorithm from the previous lecture to solve for the first part

- In practice, all of this can be done in one step via algebraic manipulation
  – after building the matrix (approach 2a), or
  – during copy-local-to-global (approach 2b)

Implementation

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,
                                              BoundaryValues<dim>(),
                                              boundary_values);
    MatrixTools::apply_boundary_values (boundary_values,
                                        system_matrix, solution, system_rhs);
}
```

Implementation

Implementation approach 2b (step-6):

```
void Step6::assemble_system {
   ConstraintMatrix constraints;
   VectorTools::interpolate_boundary_values (dof_handler, 0,
                                             BoundaryValues<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);
   }
}
```

MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University