Dirichlet BC's and constraints

Boundary conditions are required so that the PDE problem becomes well posed. Dirichlet BC's prescribe the unknown on the boundary.
This provides constraints that reduce the number of unknowns in the discrete problem.
$\Rightarrow$ need to understand how to eliminate constraints!

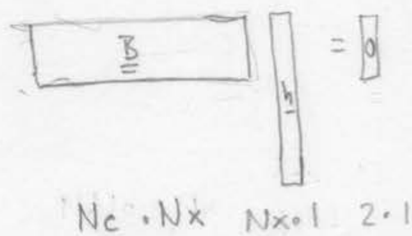
Example 1: Homogeneous Dirichlet BC's

PDE: $\quad -\frac{d}{dx}\left(\kappa \frac{dh}{dx}\right) = 1 \qquad x \in [0, L]$

BC: $\quad h(0) = h(L) = 0$

Need to write the BC's as as a linear system.

$\underline{\underline{B}}\, \underline{h} = \underline{0}$



$\underline{\underline{B}}$ is a Nc by Nx matrix, where Nc is the # constraints
$N_c \ll N$

Full statement of discrete problem:

$$\boxed{\begin{array}{ll} \text{PDE:} & \underline{\underline{L}}\, \underline{h} = \underline{f}_s \\ \text{BC's:} & \underline{\underline{B}}\, \underline{h} = \underline{0} \end{array}}$$

where $\underline{\underline{L}}$ is Nx by Nx "system matrix"
$\underline{\underline{B}}$ is Nc by Nx "constraint matrix"

Need to combine these into a single reduced linear system by eliminating the constraints $\underline{\underline{B}}$ from $\underline{\underline{L}}$.

Reduced Linear System

Constraints reduce number of unknown dofs
$\Rightarrow$ expect to solve a smaller/reduced Linear system.

Reduced system: $\boxed{\underline{\underline{L}}_r \, \underline{h}_r = \underline{f}_{s,r}}$

if N_c is the number of constraints

$\underline{h}_r$ is $(N_x - N_c)$ by 1 reduced solution vector

$\underline{f}_{s,r}$ is $(N_x - N_c)$ by 1 reduced rhs vector

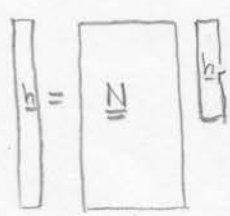
$\underline{\underline{L}}_r$ is $(N_x - N_c)$ by $(N_x - N_c)$ reduced system

"Projection" matrix

What is the relation between $\underline{h}_r$ and $\underline{h}$. ?
$\underline{f}_{sr}$ and f_s ?
$\underline{\underline{L}}_r$ and $\underline{\underline{L}}$?

Remember everything is linear!
$\Rightarrow$ Two vectors of different length are related by rectangular matrix

so that
$$\underset{N_x \cdot 1}{\underline{h}} = \underset{N_x \cdot (N_x - N_c)}{\underline{\underline{N}}} \; \underset{(N_x - N_c)\cdot 1}{\underline{h}_r}$$



What is $\underline{\underline{N}}$?

For now we just require that $\underline{\underline{N}}$ is orthonormal.

If $\underline{n}_i$ is the i-th column of $\underline{\underline{N}} = \begin{bmatrix} | & | & | & \\ \underline{n}_1 & \underline{n}_2 & \underline{n}_3 & \dots \\ | & | & | & \end{bmatrix}$ then $\underline{n}_i^T \cdot \underline{n}_i = 1 \;\; \forall i$
$\underline{n}_j^T \cdot \underline{n}_i = 0 \quad j \neq i$

Then it follows that

a) $$\underset{(N_x - N_c)\cdot N_x}{\underline{\underline{N}}^T} \cdot \underset{N_x \cdot (N_x - N_c)}{\underline{\underline{N}}} = \underset{(N_x - N_c)\cdot(N_x - N_c)}{\underline{\underline{I}}_r}$$ identity matrix in reduced space

b) $$\underset{N_x \cdot (N_x - N_c)}{\underline{\underline{N}}} \; \underset{(N_x - N_c)\cdot N_x}{\underline{\underline{N}}^T} = \underset{N_x \cdot N_x}{\underline{\underline{I}}'}$$ "identity" matrix in full space but with N_c zeros on the diagonal!

If this is the case and $\underline{h} = \underline{\underline{N}}\,\underline{h}_r$ then

$$\underline{\underline{N}}^T h = \underline{\underline{N}}^T\underline{\underline{N}}\,\underline{h}_r = \underline{\underline{I}}_r \underline{h}_r = \underline{h}_r$$

So that $\boxed{\begin{array}{l}\underline{h} = \underline{\underline{N}}\,\underline{h}_r \\ \underline{h}_r = \underline{\underline{N}}^T\underline{h}\end{array}}$ where $\underline{\underline{N}}$ is a matrix that allows us to go back & forth between full & reduced solution

We say that $\underline{\underline{N}}^T$ projects vector of unknowns the reduced solution space.
(Note: $\underline{\underline{N}}^T$ is not a proper projection matrix - not square)

Similarly $\quad \underline{f}_s = \underline{\underline{N}}\,\underline{f}_{s,r} \qquad \underline{f}_{s,r} = \underline{\underline{N}}^T\underline{f}_s$

How is the system matrix projected into reduced space?

$$\underline{\underline{L}}\,\underline{h} = \underline{f}_s \qquad \text{left multiply by } \underline{\underline{N}}^T$$

$$\underline{\underline{N}}^T\underline{\underline{L}}\,\underline{h} = \underline{\underline{N}}^T\underline{f}_s = \underline{f}_{s,r} \qquad \text{insert } \underline{\underline{I}}' = \underline{\underline{N}}\,\underline{\underline{N}}^T \text{ on left}$$

$$\underbrace{\underline{\underline{N}}^T\underline{\underline{L}}\,\underline{\underline{N}}}_{\underline{\underline{L}}_r}\;\underbrace{\underline{\underline{N}}^T\underline{h}}_{\underline{h}_r} = \underline{f}_{s,r} \qquad \Rightarrow \qquad \boxed{\underline{\underline{L}}_r = \underline{\underline{N}}^T\underline{\underline{L}}\,\underline{\underline{N}}}$$

Reduced linear system: $\quad \underline{\underline{L}}_r\underline{h}_r = \underline{f}_{s,r} \quad$ where $\quad \underline{L}_r = \underline{\underline{N}}^T L\,\underline{\underline{N}}$
$\underline{h}_r = \underline{\underline{N}}^T\underline{h}$
$\underline{f}_{s,r} = \underline{\underline{N}}^T\underline{f}_s$

Now we just need to find $\underline{\underline{N}}$!

$\underline{\underline{N}}$ needs to contain information about the boundary conditions, i.e., $\underline{\underline{B}}$!

Null space of the constraint matrix

In which space should we look for solution?

$\Rightarrow$ Any solution that satisfies the BC's; ie the constraints.

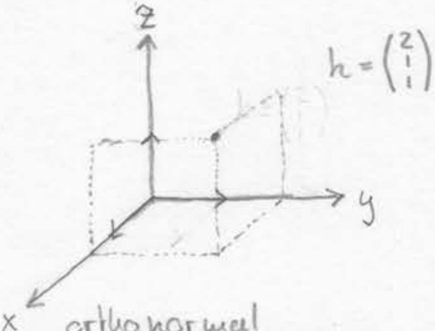
All $\underline{h}$ that satisfy $\underline{\underline{B}}\,\underline{h} = \underline{0}$, i.e. all vectors that are zero at boundary.

This is the nullspace $\mathcal{N}(\underline{\underline{B}})$ of the constraint matrix.

The matrix $\underline{\underline{N}}$ can be <u>any</u> orthonormal basis for $\mathcal{N}(\underline{\underline{B}})$.

Note there are always many such bases!

The basis is a collection of vectors that allow you to "access" any point within the vectorspace via linear combination.



orthonormal basis $\underline{\underline{N}} = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$

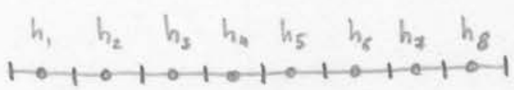
but any linearly independent set of vectors is a basis.

In Matlab we can find nullspace with the commands: $\underline{\underline{N}} = \text{null}(\underline{\underline{B}})$

or $\underline{\underline{N}} = \text{spnull}(\underline{\underline{B}})$ (download)

However, this is too slow for very large systems

It turns out we can find basis for null space easily.



BC's set $h_1 = h_8 = 0$ so that constr. matrix is first & last row of $\underline{\underline{I}}$

$$\underline{\underline{B}} = \begin{bmatrix}1&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&1\end{bmatrix}$$

$$\underline{\underline{I}} = \begin{bmatrix}1&&&&&&&\\&1&&&&&&\\&&1&&&&&\\&&&1&&&&\\&&&&1&&&\\&&&&&1&&\\&&&&&&1&\\&&&&&&&1\end{bmatrix}$$

The remaining unknown dof's are $h_2 \ldots h_7$

$\Rightarrow$ basis for reduced solution space are 2^{nd} to 7^{th} columns of $\underline{\underline{I}}$

$$\underline{\underline{N}} = \begin{bmatrix}0&0&0&0&0&0\\1&&&&&\\&1&&&&\\&&1&&&\\&&&1&&\\&&&&1&\\&&&&&1\\0&0&0&0&0&0\end{bmatrix}$$

Columns of $\underline{\underline{I}}$ are a basis for full solution space

i.e. $\underline{h} = \underline{\underline{I}}\,\underline{h}$

Notes on implementation:

Create a vector for Dirichlet BCs

```
dof_dir = [Grid.dof_xmin; Grid.dof_xmax];
% Build B from I by selecting rows corresponding to dof_dir
B = I(dof_dir,:);
% Build N from I by deleting colums corresponding to dof_dir
N = I;
N(:,dof_dir) = [];
```

Essentially we are splitting $\underline{\underline{I}}$ into $\underline{\underline{B}}$ and $\underline{\underline{N}}$!