\[ R_u = T - \sigma^p \]

Changes relative to CFEMs

A \( R_u \) is specified ONLY on \( \partial \Omega_u \) (Domain Dirichlet boundary)

in FEM's strongly

\& Inside the domain: no al. PBS

\[ T = \sum_{i=1}^{n_P} a_i N_i^p + \sum_{i=1}^{n_P} \sigma_i N_i^p \]

- takes care of continuity
- \( R_u = 0 \) at nodes
- \((T^+ - T^- = 0)\) on \( \partial \Omega_u \) (prescribed inside the domain nodes)
So C FEM satisfies $\mathbf{R} \mathbf{u} = 0$ strongly

Side note 1

$\mathbf{R} \mathbf{u} = 0$ between prescribed dfs

Side note 2

Compare satisfaction of

(a) \[
\frac{[\mathbf{T}]}{\text{Jump in } T} = 0
\] (Dirichlet type jump)

(b) versus

$[\mathbf{q}]_{\text{Jump}} = 0$ (Neumann type jump)

in C FEMs.

Obviously (a) is satisfied strongly by construction.

But (b) is not explicitly enforced.

Idea: What if we penalize jumps
Going back to DG treatment of essential type jumps ...

\[ R_u = \mathcal{T} - T \]

This idea of interior penalty can be viewed as another way to formulate DG methods (as opposed to numerical flux idea)
\[ u = -1 \]

is enforced \textit{weakly}\n
\& so \( \gamma_p \) it's enforced across all element boundary

to do this \& extend \( \bar{T}(\partial \Delta u) \)
to all element boundaries we use \( T \) not \( \gamma_p \)

\( \bar{T} \): Numerical "flux" for \( T \)

\[
T^* = \begin{cases} 
T \text{ for } \partial \Delta u \\
T^{(\text{interior})} \text{ on } \partial \Delta u, \text{def} \\
T^{(\text{interface})} \left( T_1, T_2 \right) \text{ } \gamma_p \\
T^+ & \text{for } T^+ \end{cases}
\]
typically \( T^+ = \frac{\hat{T} + T^*}{2} \)

Now that we have all the residuals we can formulate the DG weak form:

Find the solution in element \( e \): the space of interpolation functions inside the element

\[
\int_{\Omega} \left( \nabla \cdot \mathbf{u} + \nabla q - \mathbf{Q} \right) \, dv + \int_{\Gamma_i} \mathbf{w} \cdot \mathbf{T} \, ds + \int_{\Gamma_D} \mathbf{w} \cdot \mathbf{N} \, ds = 0
\]

entirely new term in DG methods

linear operator on \( \mathbf{w} \):
Some comments on \( f(w) \):

1. If we use \( f(w) = w \) the dimensions of Ru-based and the rest of weighted residual are not the same, so as an element size \( \to 0 \) or increases some terms lose their effect (go to zero relatively for example) ... so we basically "don't enforce them":

\[
\text{Remedy 1 multiply by a factor that forces the physical dimension to be consistent (often depends on element size)}
\]

\[
\int_{\Omega} \nabla \cdot (T \nabla T) \, dx
\]

\[
(2) \text{ use a formulation that is dimensionally consistent.}
\]

2. \( f(w) = q(w) = -k \nabla T \)

I'll use this option.

Comparison:

2. Is more favorable for some energy norm type errors
   But if the element is interpolated with constant (0th order polynomial) function, this error is NOT enforced
We'll choose option two, and not use $p = 0$ elements

\[ \int w \left( C \mathbf{T} \mathbf{U} + \mathbf{V} \cdot \mathbf{q} \right) \mathbf{d} V + \int w ( \mathbf{Q} - \mathbf{q} \cdot \mathbf{n} ) \mathbf{d} s + \int w \left( \mathbf{Q} \cdot \mathbf{n} \right) (1 - T) \mathbf{d} s \]


\[ \int w \nabla \cdot \mathbf{q} \mathbf{d} V = - \int \nabla w \cdot \mathbf{q} \mathbf{d} V + \int w \mathbf{q} \cdot \mathbf{n} \mathbf{d} s \]

DG weak form

\[ T = \mathbf{N} \cdot \mathbf{u} \]
There are no problems with these monomial basis (other than as p increases first we need to use better basis and basis basis coordinate)

How do we solve (*)

\[ T = \sum N_i a_i \]

Unlike CFEM's don't need to have
the delta property \( N_i (n_j) = \delta_{ij} \) (not needed)
\[ T = N a = N(x) \alpha(t) \]
\[ B = \nabla N \]
\[ T = N(x) \alpha \]
\[ -\nabla N \times \nabla \theta \]
\[ \int (w c T - \nabla w \cdot \nabla Q) \, dv + \int w \, q \, ds + \int \nabla w \cdot (\nabla T - \nabla \theta) \, ds \]
\[ e \]
\[ R = \left\{ \begin{array}{l}
\int [N T (\nabla \alpha) + B^T \alpha (B \alpha) - N \theta] \, dv + \\
- \int N T \, q \, ds - \int \alpha B \cdot n (T - \nabla \theta) \, ds = 0
\end{array} \right. \]
\[ e \]
\[ R = \left\{ \begin{array}{l}
\int (N T \nabla w) \, dv \alpha + \int \left[ \int (B^T \times B) \, dv + \int \alpha B \cdot n w \, ds \right] \alpha \\
+ \int N T \, q \, ds - \int \alpha B \cdot \nabla T \, ds - \int \nabla \theta \, ds = 0
\end{array} \right. \]
To solve it we need to assemble this system for all elements.

Good design DON'T hard-code the choice of $T$, $q$

\[
\begin{bmatrix}
T^- & T^+ \\
\varphi^- & \varphi^+
\end{bmatrix} \rightarrow \frac{\partial c}{\partial T} \frac{\partial c}{\partial T^+} \frac{\partial c}{\partial \varphi^-} \frac{\partial c}{\partial \varphi^+} \\
\text{same with } q^\circ
\]

1D version and the same 3 element example

\[
T_L = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}
\]

\[
T = q + \alpha \chi
\]

\[
N = \begin{bmatrix} 1 & \chi \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}
\]

\[
T = \Gamma_{1x}^T T^q_1
\]
\[ T = [\begin{bmatrix} 1 & x \end{bmatrix} ] a_q \]

\[ R = [\begin{bmatrix} 0 & 1 \end{bmatrix} ] a_l \]

\[ q_l = -k a_l \]

\[ q_r = -k a_r \]

\[ N(t, x) = \frac{\partial}{\partial x} N \rightarrow [0 \ 1] \]

\[ \int (W C T - \bar{w} V g \cdot n a_q) dV + \int w g \cdot dJ + \int K V \cdot A (T-T_0) dV \]

\[ (\int N \cdot C \cdot N dV) \cdot \frac{\partial}{\partial t} + \left( \frac{\partial}{\partial x} \frac{\partial}{\partial x} \right) n a_q + \left( \int N \cdot Q \right) \cdot n a_q \]

\[ = 0 \]

\[ m_e \cdot a + k a = -F \]

\[ + [\begin{bmatrix} 1 \end{bmatrix} ] q^* + [\begin{bmatrix} 0 \end{bmatrix} ] q^*_L \]

\[ [1 \ 1 \ 1 \ A \ \sigma_0 (1)] \]
\[-K\begin{bmatrix} 0 \\ 1 \end{bmatrix}(-1) \left( \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \right) \]
\[+\kappa \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left( \begin{bmatrix} 0 \\ a_1 \end{bmatrix} \right) \left( \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ a_2 \end{bmatrix} \right) = 0 \]

\[m_0^e + K_0 \nu_0 + \begin{bmatrix} q_{\phi L}^R \\ q_{\phi R}^L \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \nu_0 \phi_R \\ (1 - \phi_L) \end{bmatrix} = 0 \]