Considerations for implementing $H(\vec{\nabla}\times)$ Conforming Finite Elements

B.S. Cornille

University of Wisconsin-Madison

August 1, 2018
NIMROD Team Meeting

The FOSLS formulation with $\vec{B}$ and $\vec{E}$ leads to a simpler variational form.

- Ampère's Law is substituted into the Hall-MHD Ohm's Law

$$\frac{\partial \vec{B}}{\partial t} = -\vec{\nabla} \times \vec{E}$$

$$\vec{E} = \frac{\eta}{\mu_0}\vec{\nabla} \times \vec{B} - \left(\vec{V} - \frac{1}{\mu_0 N e}\vec{\nabla} \times \vec{B}\right) \times \vec{B} - \frac{\vec{\nabla} P_e}{Ne}$$

$$\vec{\nabla} \cdot \vec{B} = 0$$

- This formulation has more natural boundary conditions, e.g. $\vec{E} \times \hat{n} = 0$.
- There are fewer degrees of freedom to solve for than the formulation using $\vec{B}$ and $\vec{J}$.
- There is no divergence constraint on $\vec{E}$, as it would require higher than first-order derivatives, and this implies that $\vec{E} \in H(\vec{\nabla}\times)$.

In NIMROD the 3-D vector $H(\vec{\nabla}\times)$ expansion combines the 2-D H^1 and $H(\vec{\nabla}\times)$ conforming finite elements.

In NIMROD we will approximate $\vec{v} \in H(\vec{\nabla}\times)$ using

$$\vec{v} \approx \vec{v}_h = \sum_n \left[\sum_p (\tilde{v}_\xi)_{p,n} \beta_p^{(\xi)}(\xi,\zeta) \vec{\nabla}\xi + \sum_q (\tilde{v}_\zeta)_{q,n} \beta_q^{(\zeta)}(\xi,\zeta) \vec{\nabla}\zeta + \sum_r (\tilde{v}_\phi)_{r,n} \alpha_r(\xi,\zeta) \vec{\nabla}\phi \right] e^{in\phi}$$

with the highlighted parts being 2-D finite element approximations of $H(\vec{\nabla}\times)$ (green) and H^1 (yellow). Here, ξ and ζ refer to element coordinates of the 2-D meshed plane, whereas ϕ is the periodic (toroidal) coordinate.

Interpolation of $H(\vec{\nabla}\times)$ elements requires different Jacobian information than H^1 elements.

Given coordinate transformation $\vec{R}\left(\vec{\xi}\right)$ with Jacobian matrix $\mathbf{J}$ and determinant $\mathcal{J}$:

- Interpolation of H^1 elements and derivatives:

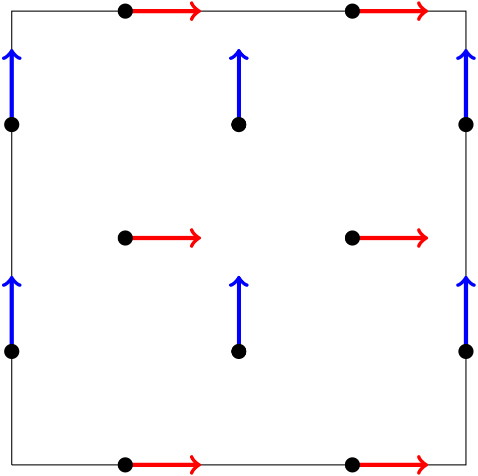
$$\vec{B}\left(\vec{R}\right) = \vec{B}\left(\vec{\xi}\right) \qquad \vec{\nabla}\vec{B}\left(\vec{R}\right) = \mathbf{J}^{-1}\vec{\nabla}\vec{B}\left(\vec{\xi}\right)$$

- Interpolation of $H(\vec{\nabla}\times)$ and derivatives:

$$\vec{E}\left(\vec{R}\right) = \mathbf{J}^{-1}\vec{E}\left(\vec{\xi}\right) \qquad \vec{\nabla}\times\vec{E}\left(\vec{R}\right) = \frac{1}{\mathcal{J}}\mathbf{J}\vec{\nabla}\times\vec{E}\left(\vec{\xi}\right)$$

Field types should apply transformations to return interpolated data.

Example of 2-D $H(\vec{\nabla}\times)$ conforming finite element.



Internal storage uses **1 value** for each DoF, but interpolation produces a **2-D vector**. Currently, there is no distinction between the dimensionality of the field and the amount of storage needed for each DoF.

Data structure and interface updates are required.

Old Implementation

- Jacobian information at quadrature points in arrays `dxdr`, `dxdz`, `dydr`, `dydz`, `jac2d`
- `qp_update` happens at block level and applies transformation
- Both field and quadrature point storage determined by `nqty`

New Implementation

- All jacobian information at quadrature points saved in `jacobian` data structure
- `qp_update` moved to field which applies transformation
- Field defines `nqty`, `fqty`, and `dfqty` for field or quadrature point storage