TODO AND WISH LIST

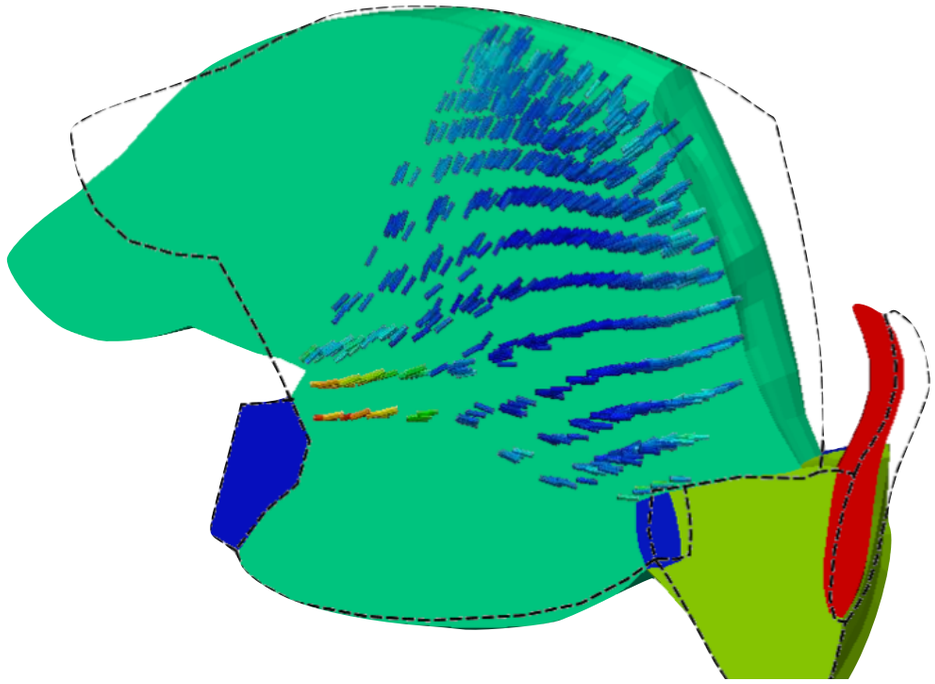
Shared triangulation

Metis-type distribution of triangulation

- hp-friendly

Work 99% completed by Denis Davydov

- Remove from limbo!



Continuum point (QP) history framework

Solid modelling with user defined

- Local parameters
- Internal history variables

Automatic update of internal data

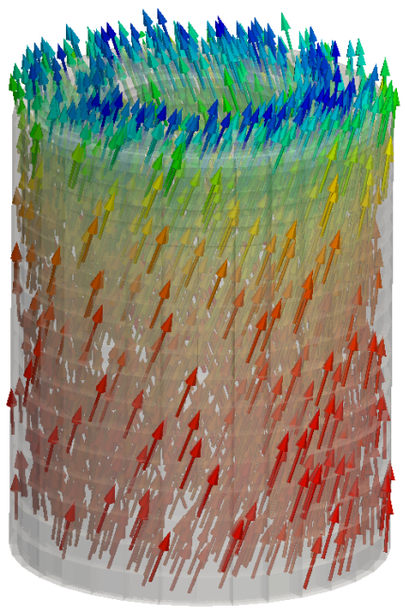
- Field values, gradients
- QP coordinate position

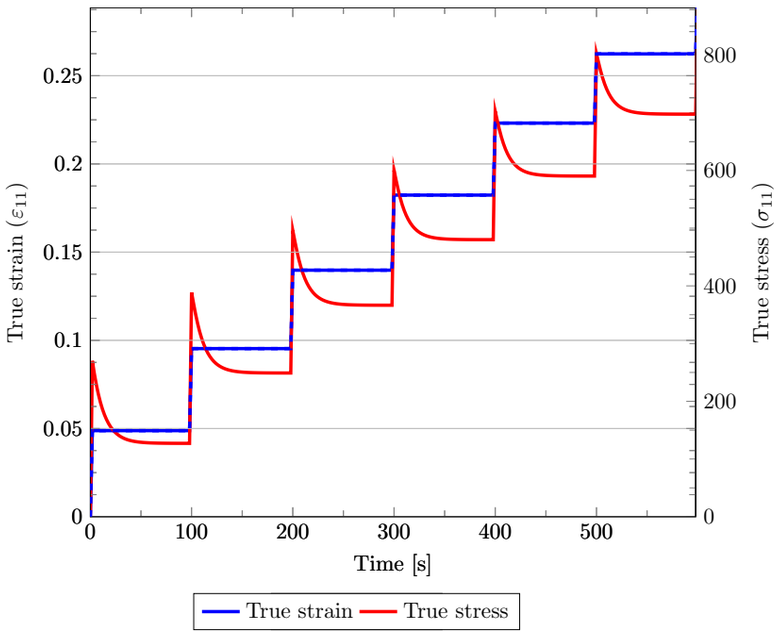
Interpolation / extrapolation on refinement

- hp compatible
- Tricky “corner-cases”

Parallel friendly

- Shared / Distributed (?) triangulation





Todo and Wish list

Update mesh conversion tool

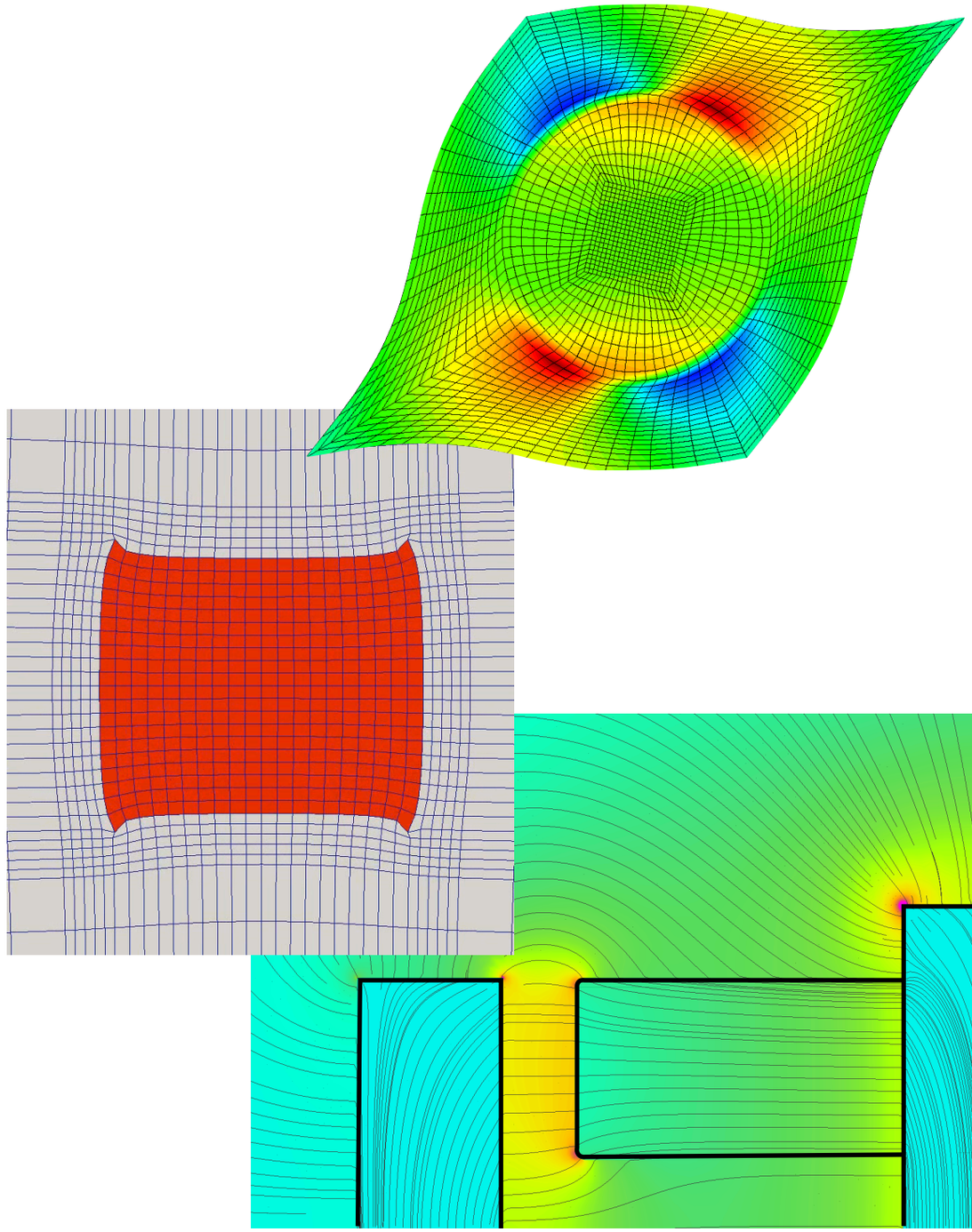
** Already / Partially implemented

Patches

- 2-d unstructured mesh issues **
- Krzysztof Bzowski
  - “Native” ABAQUS support

Testing

- Integrated testing within deal.II test suite



Boundary conditions

Periodic BC’s

- Allow translational periodicity

General bending of the rules

- Point/line/interface constraints
- “Internal” BC’s
- Engineering necessity > Mathematical rigour
- Difficulties:
  - Continuous + discontinuous / primitive FE’s
  - Refer to VectorTools::interpolate_boundary_values

Todo and Wish list

** Already / Partially implemented

Schur complement

More general implementation

- Arbitrary A,B,C,D block from block matrix/vector **
- Nested operators
  - Schur Inception :-)

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \quad \begin{bmatrix} \mathbf{D} & \mathbf{C} \\ \mathbf{B} & \mathbf{A} \end{bmatrix}$$

Copy block to non-block systems **

Enable switch between direct, iterative solvers

- Particularly for Trilinos

$$\begin{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{C} \\ \mathbf{B} & \mathbf{A} \end{bmatrix} & \begin{bmatrix} \mathbf{E} \\ \mathbf{F} \end{bmatrix} \\ \begin{bmatrix} \mathbf{G} & \mathbf{H} \end{bmatrix} & \begin{bmatrix} \mathbf{J} \end{bmatrix} \end{bmatrix}$$

General talking points

Trilinos + Sacado

- Needs an interface update?

Field-based interaction

- Simplification of interaction with blocks, FEExtractors, Component masks, DoF indices
  - matrix.block("u","p")
  - fe_values.get_gradients("u")
  - VectorTools::interpolate_boundary_values(… , XYZ.componentMask("psi"))
  - VectorTools::interpolate_boundary_values(… , "psi", localPsiComponentMask)