Chair of Computation in Engineering
Department of Civil, Geo and Environmental Engineering
Technical University of Munich

Implementation of the Finite Cell Method (FCM) into a commercial Finite Element software

Team: Yousaf Muhammad Salman, Krishna Rahul, Wu Fong-Lin, Haider Irfan, Boulbrachene Khaled, Varadharajan Srikkanth

Supervisors: Philipp Kopp, Nina Korshunova, Lisa Hug, Davide D'Angella, Alexander Paolini

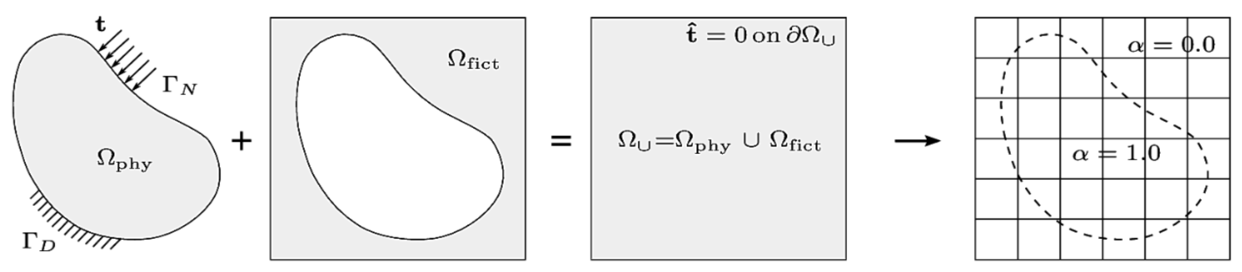
Abstract

Finite-cell method has been implemented in ABAQUS, a commercial FEM based software. The user defined element (UEL) feature of ABAQUS has been exploited for this implementation. Python scripts have been used to automate the modelling setup, as well as performing adaptive integration for evaluating the stiffness matrices. In this project, FCM is implemented only for 2D plane stress conditions.

Motivation

The finite element method is the standard tool for solving mechanical problems. Although, the computational time has significantly decreased due to better hardware performance, meshing the computational geometry still requires manual work. While automatic mesh generators today produce reasonable results even for complex geometries, problems arise especially if the input geometry is flawed. Healing such geometries is a tedious work and cannot be automated in many cases. This problem can be overcome by immersed methods, such as the finite cell method. However, there is not yet any commercial software that has implemented such a method.

Mathematical Formulation

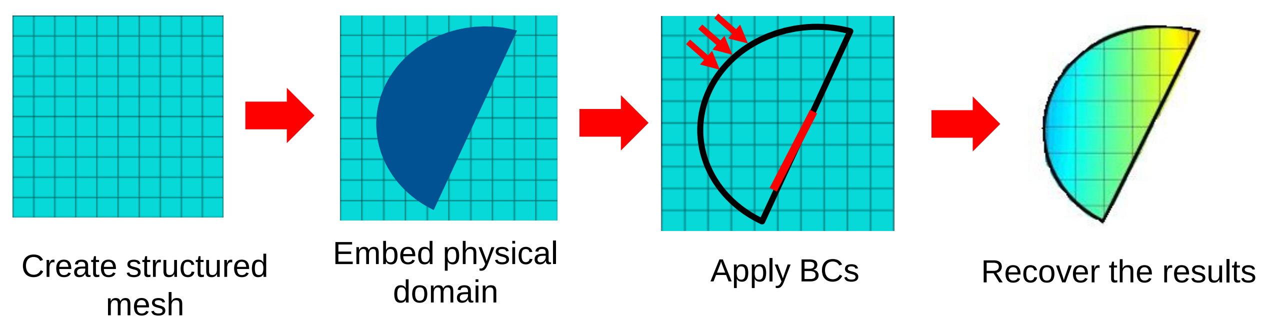


$B_U(u, v) = F(v)$ where

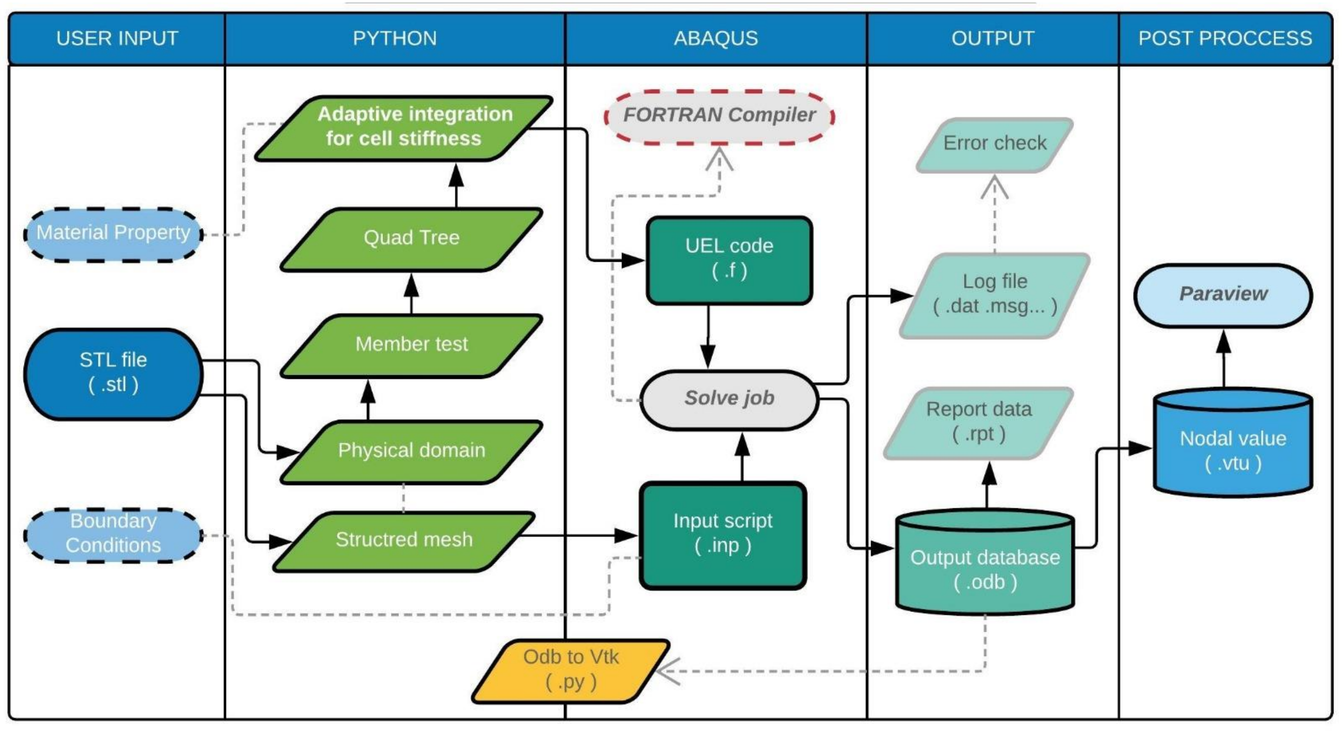
$$B_U(u,v) = \int_{\Omega_U} [Lv]^T \alpha C [Lu] d\Omega_U$$ [1]

The physical domain is embedded into a larger domain, which is discretized in a structured manner. Each discretized region is termed as a cell. Membership test is performed for every cell with respect to the physical domain, which specifies the contribution of the cell to the overall stiffness of the system using a parameter α.

Implementation overview

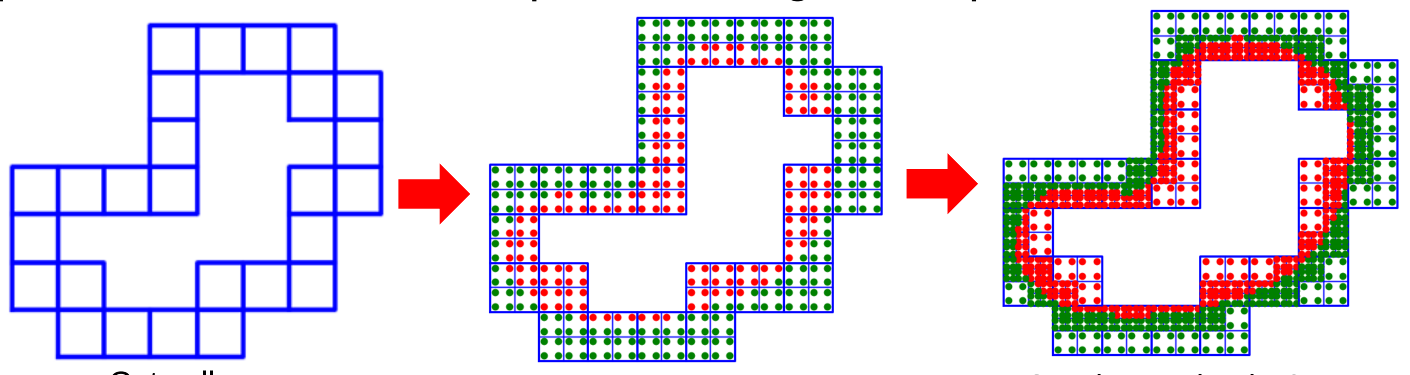


Overview of the developed workflow



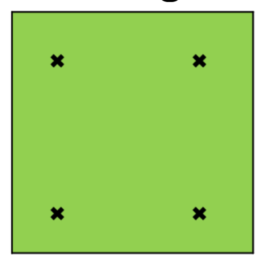
Quad-tree implementation

Quad-tree is used to recover the geometry and distribute integration points for each cut cell and inside/outside test is performed for each such point to assign the alpha values.

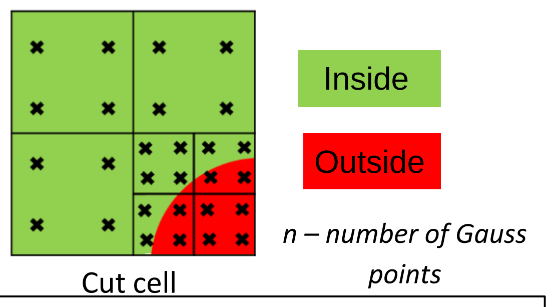


Cut cells Quad-tree: depth=1 Quad-tree: depth=2

Adaptive integration for the cut-cells



Uncut cell

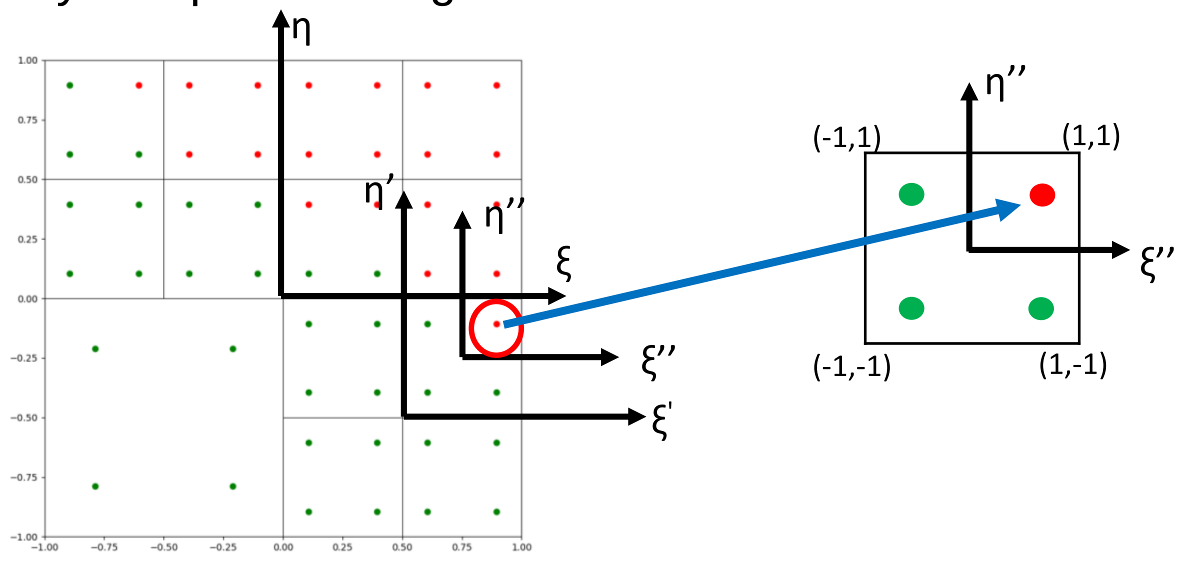


Cut cell

$$K_e = t \sum_{i=1}^{2} \sum_{j=1}^{2} B^T_{(\xi_i \eta_j)} C\, B_{(\xi_i \eta_j)}\, w_i\, w_j \det(J_{(\xi_i \eta_j)})$$

$$\boxed{K_e = t \sum_{i=1}^{n} B^T_{(\xi_i \eta_i)} C\, B_{(\xi_i \eta_i)} \alpha_i\, w_i \det(J_{(\xi_i \eta_i)})}$$

In traditional FEM the stiffness matrix of a plane stress element is evaluated using a 2x2 Guass quadrature rule. In FCM, the stiffness matrix is evaluated at relevant integration-points determined by the quad-tree algorithm.

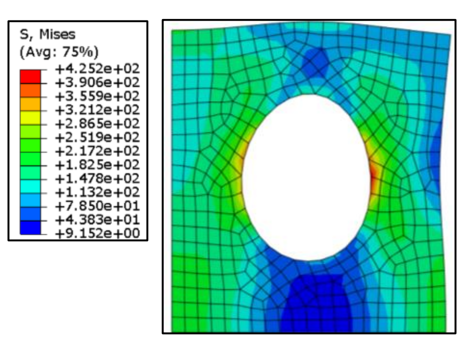


The integration points are well defined with respect to the local coordinate of the leaf. These points are mapped to the iso-parametric element's coordinates using mapping functions. The Jacobian of these mappings gives the weight for these integration points.

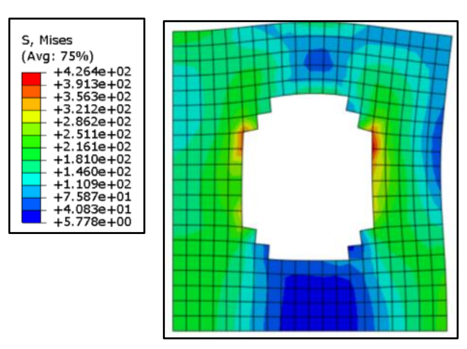
The stiffness matrices for these cut cells are fed to ABAQUS using the UEL subroutine feature.

Results

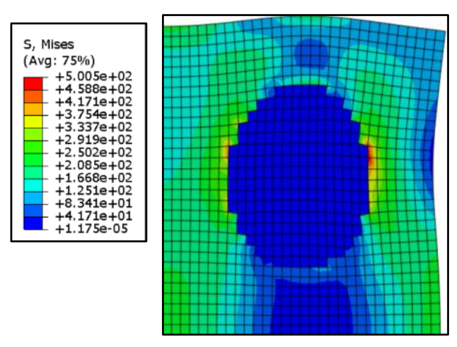
Von-mises stress



FEM



"FCM"



Zero stress on outside cells

Conclusions and outlook

- The simple workflow to implement FCM in ABAQUS has shown reasonable resemblance to FEM for a 2D plane stress problem.
- The boundary conditions are applied directly on the nodes, but FCM demands these be applied in weak form over the actual physical boundary.

References

[1] - Jamshid Parvizian, Alexander Düster, Ernst Rank, Finite Cell Method h- and p-extension for embedded domain problems in Solid Mechanics, Computational Mechanics manuscript