Accelerating the Hierarchical Multi-Scale (HMS) Model by Spatial Clustering Strategies

Md Khairullah*^,1,2^, Jerzy Gawad[1], Dirk Roose[1], Albert Van Bael[3]

[1] KU Leuven, Department of Computer Science, 3001 Leuven, Belgium

[2] Department of Computer Science & Engg., Shahjalal University of Science & Tech., Bangladesh
e-mail: {md.khairullah, jerzy.gawad, dirk.roose}@cs.kuleuven.be

[3] KU Leuven, Department of Materials Engineering, 3001 Leuven, Belgium
e-mail: albert.vanbael@kuleuven.be

ABSTRACT

The Hierarchical Multi-Scale (HMS) software, developed at KU Leuven, is a computational plasticity tool that concurrently takes into account the strain evolution of the crystallographic texture and associated plastic anisotropy. To this end, the HMS adaptively reconstructs the macroscopic material properties, such as the plastic potential function, by exploiting appropriate physics-based micro-scale polycrystalline plasticity models [1]. The evolution of the plastic properties is calculated independently at each integration point in the macroscopic FE mesh, leading to a large computational cost. The HMS partly resolves this issue by reconstructing the plastic potential function not in every time increment, but only if a given deformation-based criterion is satisfied or if a sufficient deformation is achieved at a user-defined integration point.

In this paper we present a method to further reduce the simulation time by using spatial clustering [2] with respect to the accumulated plastic strain response, which determines the evolution of the material properties at integration points. The fundamental assumption is that similar state variables (e.g. crystallographic texture) subjected to a similar deformation history (i.e. the plastic strain) would evolve along nearly identical trajectory. It can be reasonably expected that the derived macroscopic plastic properties would be similar as well. Therefore, we perform the actual update of the material properties at a single representative integration point per cluster and propagate the updated properties to the other integration points belonging to the cluster. This significantly reduces the number of updates of the material properties. The approximation error and the computational benefit depend on the number of clusters used. A certain trade-off has to be made: by increasing the number of clusters the approximation error can be decreased, but at the same time the computational cost increases.

One may consider a static clustering strategy, which determines a fixed configuration of clusters at a particular moment in the simulation and subsequently keeps it invariant. The scheme poses several difficulties, such as sub-optimality of the clustering with respect to accuracy if the accumulated plastic strain evolves in time. To overcome these limitations, an adaptive dynamic clustering is proposed. In this strategy the integration points are re-assigned to the clusters using criteria based on minimization of the variance w.r.t. plastic strain among the cluster members. This dynamic approach appears more realistic, since it is able to capture the evolution of strain and, more importantly, it does not rely on a single state of simulation to determine the clusters. This results in improvements in the accuracy and, at the same time, performance gains are expected, in particular if relatively large clusters can be retained for a longer time, thus fewer updates of material properties are needed.

We present an assessment of the approximation error and performance gain for a set of test cases of varying complexity, including a complex geometry tensile test and a three-point bending test. The results obtained using both strategies are compared to the reference solution, computed by the HMS software that does not exploit the spatial clustering, which is equivalent to the finest clustering with one integration point per cluster. Finally, we elaborate on possible improvements to the dynamic clustering strategy.

REFERENCES

[1] J. Gawad et al., “Hierarchical multi-scale modeling of texture induced plastic anisotropy in sheet forming”, *Comp. Mater. Sci.*, **66**, 65-83 (2013).

[2] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python”, *J. Mach. Learn. Res.*, **12**, 2825-2830 (2011).