ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

SPATIAL CLUSTERING STRATEGIES FOR HIERARCHICAL MULTI-SCALE MODELLING OF METAL PLASTICITY

Md Khairullah[1], Jerzy Gawad[1], Dirk Roose[1], and Albert Van Bael[2]

[1]Department of Computer Science, KU Leuven
3001 Leuven, Belgium
e-mail: {md.khairullah, jerzy.gawad, dirk.roose}@cs.kuleuven.be

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

Keywords: Multi-scale modeling, Polycrystalline materials, Spatial clustering, Dynamic adaptive clustering, Clustering criteria, Approximation error.

Abstract. *In multi-scale simulations of material forming processes, macroscopic zones of nearly homogeneous strain response occur. In such zones the evolution of plastic anisotropy at each finite element integration point can be approximated from the properties at a representative point. We show how these zones can be identified by a clustering algorithm and can be utilized to reduce the computational cost of the simulation.*

1 INTRODUCTION

The Hierarchical Multi-Scale (HMS) software, developed at KU Leuven, simulates the deformation of polycrystalline metallic alloys. HMS takes into account both the strain-driven evolution of the preferred orientation of crystallites (or texture) and the associated plastic anisotropy, as they significantly influence the mechanical and physical properties of the material [1]. At the macroscopic level, the deformation of the material is described by a Finite Element (FE) model and the anisotropy of the plastic properties is approximated by an analytical plastic potential function. On the other hand, the physics-based crystalline plasticity (CP) code provides the micro-scale stress response. The HMS incorporates two categories of material state: a) the direct material state (e.g. equivalent plastic strain, equivalent yield stress) that is directly needed by the FE code in each time step, and b) the extended material state that has to be updated regularly, but does not have to be accessed in each time step. The parameters of the plastic potential function, together with the texture data (represented by the orientation distribution function [2]) form the extended state variables at the FE integration points. The HMS model assumes that the texture and the coupled plastic anisotropy are initially identical in the whole volume of the material, but may evolve independently in every FE integration point with increasing plastic strain.

In principle, the texture can change at every time increment of the macroscopic FE model, and the parameters of the plastic potential function should be updated at every time step, resulting in a substantial computational cost. The HMS software partly resolves this issue by reconstructing the function not after every time increment, but only if a given deformation-based criterion is satisfied. Hence, the texture state variable remains constant in each time interval between the updating events. In order to decide whether such an update of the texture state variable should take place at a specific integration point, the HMS software tracks the recent history of the deformation tensor $\mathbf{d}(t) \in \mathbb{R}^{3\times3}$ in that point by

$$\mathbf{P}^t = \int_{t_i}^{t} \mathbf{d}(t)dt, \tag{1}$$

which is the plastic strain that has been accumulated (at time t) since the previous update (at time t_i). The texture related material properties are updated if

$$\|\mathbf{P}^t\| \geq P_{cr}, \tag{2}$$

with P_{cr} is a user-defined threshold. Then, the accumulated plastic strain tensor $\mathbf{P}^t$ is passed to the texture evolution model, which applies appropriate lattice rotations to the crystal orientations to obtain the new orientations [3].

Similarly, reduction of update operations in the spatial domain is also an option. Instead of updating the texture related material properties at each integration point inside a group of points, updating a single representative point is more feasible. In this paper, we present methods to reduce the simulation time of the HMS model by using spatial clustering of the finite element integration points w.r.t. some relevant feature of interest.

The paper is organized as follows. Section 2 presents the principles of the proposed enhancements to the existing hierarchical multi-scale software as well as the description of the test case used for the simulation experiments. Section 3 outlines estimations of the approximation error and the performance gain due to the enhancements. A detailed analysis and discussions of the simulation results are presented in Section 4. The last section contains a summary of the ongoing works, concluding remarks and future research.

2 SPATIAL CLUSTERING IN THE HMS SOFTWARE

Clustering is a technique to group data objects, based on information found in the data characterizing the objects and their relationships. The aim is that the objects within a group are similar to one another and different from the objects in other groups. The greater the similarity within a group and the greater the difference between groups, the better the clustering. Spatial clustering organizes objects based on spatial aspects such as distance, connectivity, relative density in space as well as other feature(s) of interest [4]. Due to its generic applicability, clustering is used in a broad range of applications, e.g. in materials modeling, material discovery, general finite element simulations and other fields of engineering [5, 6, 7, 8, 9]. The fundamental assumption in this paper is that similar micro-structural state variables in neighboring integration points subjected to a similar deformation history would evolve along nearly identical trajectories in state space. It can be expected that the derived macroscopic plastic anisotropy would be similar as well. Therefore, we perform the actual update of the plastic anisotropy 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 plastic anisotropy and subsequently the overall simulation time. Similar approaches to reduce computational cost of engineering applications based on clustering are reported in [8, 10, 11].

In this work we consistently employ the implementation of the agglomerative clustering algorithm provided by the scikit-learn library for machine learning [12]. This algorithm performs a hierarchical clustering using a bottom-up approach. Initially each object is a cluster on its own, and adjacent clusters are successively merged together. A variance-minimizing approach is used for the merge strategy. It merges the pair of clusters for which the sum of squared differences of the data points within the clusters is minimal. We developed an interface to this clustering algorithm, for which the inputs are a set of features of interest, information about the connectivity among the data points, and the number of clusters to construct. This interface computes the connectivity among the integration points. Note that clustering is carried out based on the data at the integration points and the connectivity among the integration points. This makes the proposed methods completely independent of chosen element/mesh types, which is an additional advantage.

However, the selection of input data, which represent the feature of interest, and the number of cluster is not trivial and are discussed below. We also discuss the cluster representative selection procedure.

Feature of interest: As stated in section 1, upon satisfaction of the deformation based update criteria, given in (2), the accumulated plastic strain $\mathbf{P}^t$ is passed to the texture evolution model. Thus, we assume that the accumulated plastic strain, i.e. the recent deformation history, determines the evolution of texture and the associated plastic anisotropy. However, as the updates at the integration points are independent and may occur out of sync, the accumulated plastic strain in a set of integration points can not be compared to measure the similarity in recent history. Rather, the total plastic strain (accumulated since the beginning) that represents the total plastic deformation history at an integration point, is a more logical feature to compare a set of integration points for our purpose. Furthermore, the plastic strain tensor having 9 components (6 independent components) may suffer from the limitations of clustering multidimensional data [13]. Dimensionality reduction (e.g. subspace clustering/ projected clustering) is the typical solution for this. Equivalently, state variable(s) with lower dimension which can correspond to the total plastic strain may be considered as an alternative feature for clustering. HMS incorpo-

rates the traditional approach of elastic predictor and plastic corrector in the elasto-plastic stress integration algorithm [14]. The plastic strain increment $\Delta\boldsymbol{\varepsilon}^p \in \mathbb{R}^5$ is assumed to be

$$\Delta\boldsymbol{\varepsilon}^p = \Delta\varepsilon^p \mathbf{A}, \tag{3}$$

where the scalar $\Delta\varepsilon^p$ represents the magnitude of the plastic strain increment, and $\mathbf{A}$ is the plastic strain rate mode that corresponds to the point on the yield locus in the direction of the trial stress [15]. The increment in equivalent plastic strain $\Delta\bar{\varepsilon} \in \mathbb{R}$ is calculated as

$$\Delta\bar{\varepsilon} = \Delta\varepsilon^p \psi^t(\mathbf{A}), \tag{4}$$

where ψ^t is the plastic potential function at time t. If $\boldsymbol{\varepsilon}^{p,t}$ and $\bar{\varepsilon}^t$ respectively are the plastic strain and the equivalent plastic strain at time t, at the end of time increment Δt they are respectively given by

$$\boldsymbol{\varepsilon}^{p,(t+\Delta t)} = \boldsymbol{\varepsilon}^{p,t} + \Delta\boldsymbol{\varepsilon}^p, \tag{5}$$

and

$$\bar{\varepsilon}^{t+\Delta t} = \bar{\varepsilon}^t + \Delta\bar{\varepsilon}. \tag{6}$$

As the calculation of the increment in plastic strain $\Delta\boldsymbol{\varepsilon}^p$ and in the equivalent plastic strain $\Delta\bar{\varepsilon}$ depends on the set of same variables $\Delta\varepsilon^p$, $\mathbf{A}$, the equivalent plastic strain, which is a scalar state variable at an integration point, can be considered as a scalar representation of the plastic strain, and also a feature of interest for clustering.

Number of clusters: Most of the clustering algorithm constructs some prescribed number of clusters instead of finding the optimal number of clusters. This limitation remains a fundamental and largely unsolved problem in cluster analysis. A simple rule of thumb suggests to construct $\sqrt{n/2}$ clusters out of n data points. The Elbow method recommends to choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. However, for several cases the stopping criteria can not be unambiguously evaluated. A number of similar but more concrete approaches, primarily based on cluster evaluation and assessment, have been proposed (e.g. [16, 17]) and reliably incorporated in popular software packages. Computational efficiency of these algorithms is poor as they construct a number of clusters, measure an evaluation metric (e.g. silhouette coefficient), and repeat with an incremented number of clusters. However, if the construction of clusters in the simulation is infrequent, one of these methods can be utilized to decide about the optimal number of clusters. As the cluster evaluation metric itself is computationally costly, in situations where we may frequently need to apply the clustering algorithm, we can utilize some simple heuristics instead. For example, we can measure the distance between the minimum and the maximum of the data points and decide to construct two clusters if the distance is larger than a threshold value. This can be recursively continued inside the generated clusters. However, the clusters generated in this way would lead to sub-optimal clusters, and still we need to choose a good threshold value.

Cluster representative: The selection of the cluster representative is quite intuitive. The mean (or, the centroid for multi dimensional data) is commonly considered as the most appropriate representation of the cluster data points, as this value minimizes the distance between the data points and the representative. However, we must select an integration point inside the cluster to represent the cluster for practical purposes (e.g. in our case the cluster representative would contain a number of extended state variables, which are common for all the members). We choose the integration point whose average dissimilarity to all the integration points in the cluster is minimal, known as the medoid, to serve this purpose in our case.

For a brief explanation of the proposed strategy, let us consider the two dimensional miniature model in Figure 1 with only 7 elements. Let each element has only one integration point located at the center. Integration points in each element are labeled with *A*–*G*. The borders of the clusters are marked by dashed lines, and the cluster representatives are encircled. Notice that although the points *C* and *E* show exactly the same field values, they are put in different clusters since point *D* with large difference in the field value is located between them, which violates the connectivity constraint for forming a spatial cluster. In the proposed method only the cluster representatives *B*, *D*, *E*, and *G* update the plastic anisotropy. Representative point *B* propagates its updated properties to *A* and *C*, whereas *G* propagates to *F*. As *D* and *E* are the single member of their own clusters, they only update the plastic anisotropy but need no propagation.

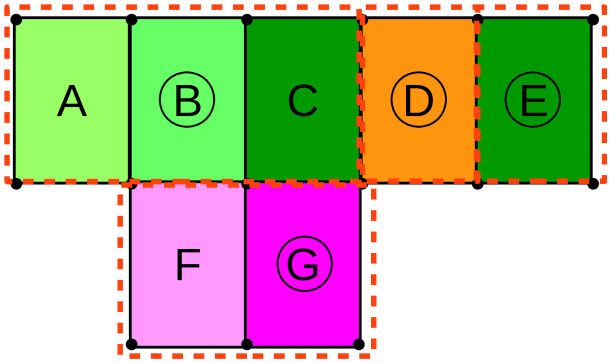


Figure 1: Subdivisions of the FE mesh into clusters with respect to the field value (color scale). The elements are labeled with letters, while the cluster representatives are distinguished by circles.

While the underlying micro-structure, texture and the associated plastic anisotropy are assumed to be the same for the whole cluster, other variables such as the strain and the corresponding stress at the integration points are independent of the clusters and evolve individually.

2.1 Selection of test case

In [18] we used a complex geometry specimen to simulate a tensile test. However, this test case can be considered as very simple in four aspects: (i) the homogeneity of plastic strain field across integration points at the beginning of plastic deformation is somehow maintained for the rest of the simulation, (ii) the magnitude of the accumulated plastic strain is sufficient as the clustering criterion because only one of the plastic strain tensor components is dominating, which is equivalent to the strain magnitude, (iii) in principle the specimen was two dimensional, and (iv) the model was comparatively small (2,226 integration points). For these reasons, a more complex deformation is considered in the test case of this paper.

We consider the specimen in Figure 2, which is used in the simulation of a two stage deformation. The FE mesh consists of 12,168 elements of type C3D8R (i.e. 8 node linear brick element with one integration point). The farthest Y-Z surface in the figure is always kept fixed. In the first stage a moment load is applied on the nearest Y-Z surface in the figure along the positive X direction and in the second stage tension is applied on the same surface in the same direction with small displacements along the positive Y and the negative Z directions. Figure 3a shows the equivalent plastic strain field at the integration points at some moment in the first stage during the simulation, where regions of almost homogeneous field values can be easily observed. If one groups the neighboring integration points with similar equivalent plastic strain

in a cluster, a single update is needed per cluster instead of updates at each integration point, based on the assumption of homogeneity of the underlying texture and the plastic anisotropy inside the cluster. The corresponding 9 clusters are represented by distinct colors in Figure 3b.

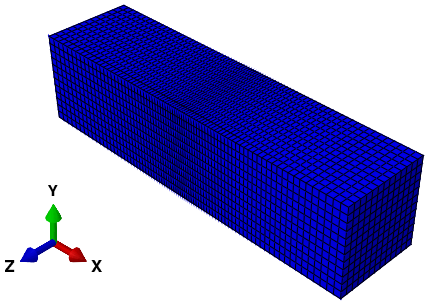
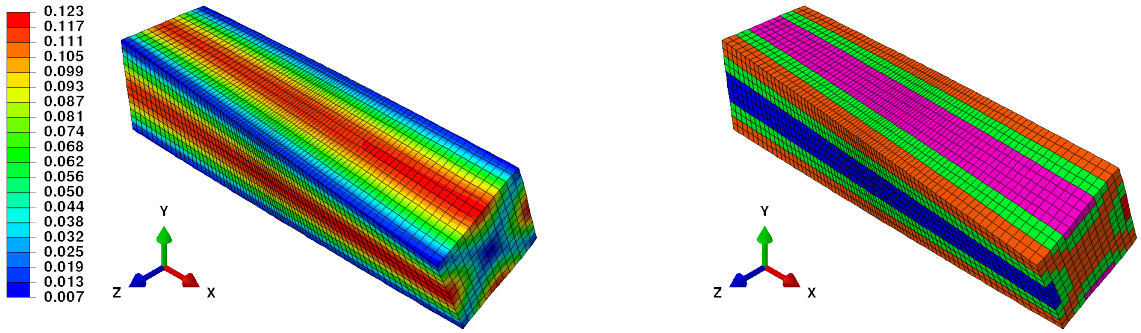


Figure 2: Specimen used in a two step deformation for the simulation experiments. The FE mesh consists of 12,168 C3D8R elements.



(a) (b)

Figure 3: a) The field of equivalent plastic strain in the specimen used in the two-stage deformation simulation; and b) clusters (represented by distinct colors) are constructed according to equivalent plastic strain in the highlighted area.

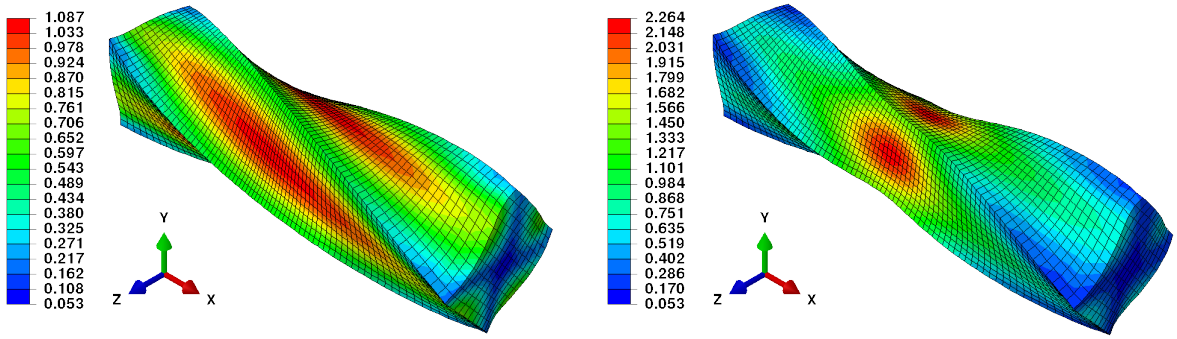
2.2 Static clustering

A static clustering strategy employs a fixed configuration of the clusters, which is constructed based on the field values at a particular moment in the simulation and subsequently kept constant. The advantage of this simple approach is that the overhead related to the construction of the clusters is minimal. However, the choice of the particular moment for constructing the clusters is crucial to obtain a good balance between the number of clusters (and thus the computational cost of the simulation) and the accuracy.

We can perform a number of time steps in the actual HMS simulation without clustering, and then construct the clusters based on the field of interest. If the clustering is performed in an early time step, it may be that insufficient deformation history is available, leading to suboptimal clusters. However, postponing the construction of the clusters leads to less savings in the computational cost. Therefore, we suggest to construct the clusters in the time step in which the update criterion is satisfied for the first time.

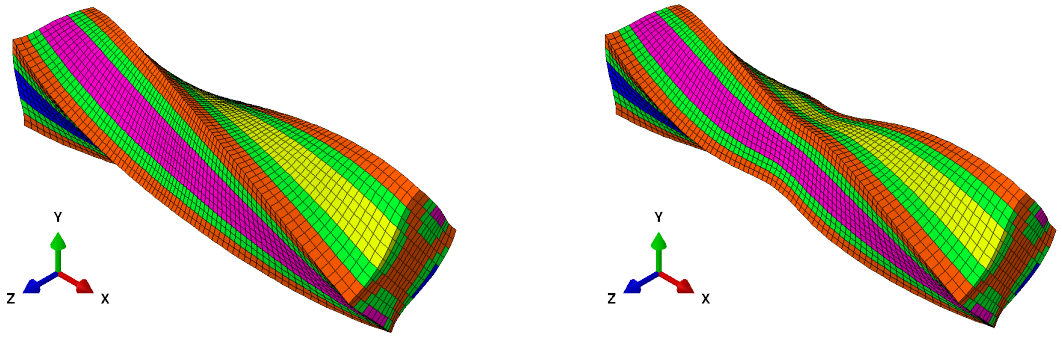
Figure 4a and 4b show the equivalent plastic strain field values at the middle and at the end of the second deformation stage respectively. These clearly indicate that the clusters constructed

by the static clustering approach do not reflect the homogeneous zones in the second stage. This is particularly true if we use a small number of clusters. For example, we can compare the nine clusters in Figure 5a and Figure 5b with the field values in Figure 4a and 4b. The incapability of the static clustering to reflect the variation in time of homogeneity of field values among the integration points is a fundamental drawback of it.



(a) (b)

Figure 4: The field of equivalent plastic strain at a) the middle, and b) at the end of the second deformation stage.



(a) (b)

Figure 5: Statically constructed clusters at a) the middle, and b) at the end of the second deformation stage.

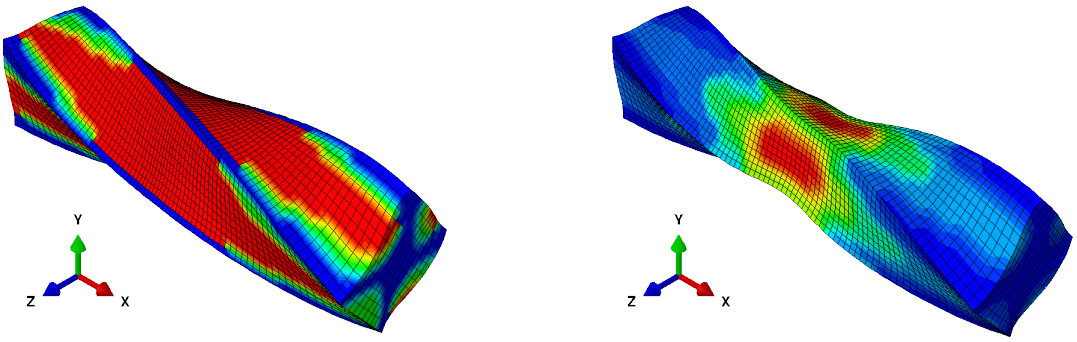
2.3 Dynamic adaptive clustering

To overcome the limitations of static clustering, a dynamic adaptive 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. the equivalent plastic strain among the cluster members. This dynamic approach is more realistic, since it is able to capture the evolution of strain and, more importantly, it does not rely on a single time step to determine the clusters. Hence, compared with static clustering, we expect improvements in the accuracy and, at the same time, performance gains, in particular if relatively large clusters can be retained for a longer time, leading to fewer updates of the plastic anisotropy.

At the beginning of the simulation, the amount of the plastic strain as well as the equivalent plastic strain is zero at each integration point and we assign all points to a single cluster. A

cluster is split into two clusters if the difference between the maximum and the minimum of the observed field values among the points inside the cluster exceeds a specified threshold value. Of course, this splitting criterion is only tested when the deformation-based updating criteria at an integration point is satisfied. As long as a cluster is not split, the cluster representative prevails the role. At present, for simplicity the adaptation of clustering is based on splitting only without the possibility of merging two clusters.

Clusters constructed with the dynamic adaptive approach are shown in Figure 6a and 6b. Obviously, these clusters better represent the actual homogeneity of field values in Figure 4a and 4b.



(a) (b)

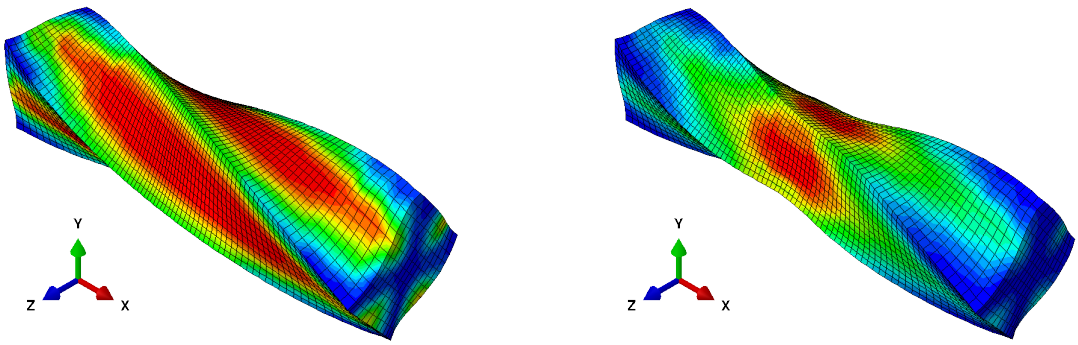
Figure 6: Adaptively constructed clusters at a) the middle, and b) at the end of the second deformation stage. In (a) and (b) the number of clusters are 3 and 62 respectively (only 24 colors are used).

A smaller threshold value for splitting (i.e. a more rigorous splitting condition) leads to an early split of the clusters and ultimately results in better clusters. Figure 7a and Figure 7b represent such a case. Comparison with the field values in Figure 4a and 4b makes it evident. However, such a small threshold value for splitting result in a large number of clusters, and consequently large number of updates and less computational gain.

Note that only 24 colors are used in Figure 6 and Figure 7 to visualize the clusters. In fact, formed clusters are assigned unique labels from 1 to n_c, where n_c is the number of clusters (e.g. $n_c = 3$ in Figure 6a and $n_c = 62$ in Figure 6b). These labels are assigned from low to high in an sorted order of the mean data in each cluster, i.e. the cluster with the lowest mean data is assigned label 1 and the cluster with the highest mean data is assigned label n_c.

3 ERROR ESTIMATION AND COMPUTATIONAL PERFORMANCE

We approximate the anisotropy of plastic properties at each integration point in a cluster by the properties at the cluster representative. This introduces an additional modeling error. More specifically, the plastic anisotropy model is utilized uniformly inside the cluster. The coefficients of the anisotropy model are periodically re-identified to follow the evolution of anisotropy solely at the cluster representative. This approximation also affects the field values (e.g. equivalent plastic strain, equivalent yield stress) that are calculated based on the plastic anisotropy. Hence, we will estimate the approximation error in both the plastic properties and the affected field values. In the rest of the paper the HMS software that is extended with spatial clustering to reduce the computational cost is referred as the *improved HMS* and the original HMS software that does not exploit spatial clustering is referred as the *reference HMS*. Note that



(a) (b)

Figure 7: Adaptively constructed clusters with a smaller threshold value at a) the middle, and b) at the end of the second deformation stage. In (a) and (b) the number of clusters are 2,119 and 6,013 respectively (only 24 colors are used).

the reference HMS is equivalent to the improved HMS using one integration point per cluster. Using this maximal number of clusters results in no approximation error but also no reduction of the computational cost.

3.1 Approximation error in terms of plastic properties

The plastic potential function can be utilized to estimate the error in plastic properties regardless of the loading direction. We calculate the Euclidean distance of the plastic potential function values at an integration point, computed with the reference HMS software, and the plastic potential function values at the representative point, computed with the improved HMS software. For an integration point, *m* almost evenly spaced strain rate modes in the 5D strain rate space are chosen [19] and the value of the potential function is calculated for each of them. We define the relative error in plastic properties as the relative difference in the plastic potential function using the formulas

$$e_i^{\psi} = \frac{1}{m}\sum_{j=1}^{m} \left|\frac{\psi(\mathbf{D}_j)_i - \psi'(\mathbf{D}_j)_{cr(i)}}{\psi(\mathbf{D}_j)_i}\right| \times 100\% \tag{7}$$

and

$$\bar{e^{\psi}} = \frac{1}{n}\sum_{i=1}^{n} e_i^{\psi}, \tag{8}$$

where $\psi(\mathbf{D}_j)_i$ and $\psi'(\mathbf{D}_j)_{cr(i)}$ are the values of the plastic potential function at an integration point i in the reference HMS and at the cluster representative $cr(i)$ in the improved HMS respectively, for a chosen strain rate mode $\mathbf{D}_j$. Hence, e_i^{ψ} and $\bar{e^{\psi}}$ represent the relative error respectively in integration point i and in the whole model. In principle we can compute these values at every time increment, but most often we are only interested in these errors at the end of the simulation.

3.2 Approximation error in terms of field values

The plastic strain during a time step is affected by the approximation of the plastic anisotropy using the clustering strategy. As the equivalent plastic strain field in the final state of the simulation contains the total plastic deformation history summarizing the total effect of the approximation, is selected for comparison with the reference HMS. In the improved HMS, this scalar

value is computed at each integration point using the plastic anisotropy computed at the cluster representative. Hence, we compute the relative approximation error in the equivalent plastic strain field at point *i* as

$$e_i^{\gamma} = \frac{|\gamma_i - \gamma_i'|}{\gamma_i} \times 100\%, \tag{9}$$

where γ_i and γ_i' are the equivalent plastic strain values at an integration point i calculated with the reference HMS and the improved HMS respectively. As in the previous case, we also compute the average of the error for the whole model

$$\bar{e^{\gamma}} = \frac{1}{n} \sum_{i=1}^{n} e_i^{\gamma}. \tag{10}$$

Note that (9) and (10) can be utilized to calculate the approximation error for any other relevant scalar variables, for example the equivalent yield stress.

3.3 Computational gain

We measure the relative performance improvement in terms of the calculation time by *gain in time* g_t, defined as

$$g_t = \frac{t_r}{t_c}, \tag{11}$$

where t_r is the simulation time with the reference HMS software and t_c is the simulation time with the improved HMS.

By using the methods proposed above, we only speed up the part of the software responsible for the evolution of texture and the associated plastic anisotropy. If that part represents a fraction f of the execution time, then the gain in time is limited by

$$g_t^{max} = \frac{1}{1-f}. \tag{12}$$

Hence, if $f \approx 94\%$, as in the simulations performed below, $g_t^{max} \approx 16$. The expected gain in time is

$$g_t^e = \frac{1}{(1-f) + \frac{n_c}{n} f} \leq g_t^{max}, \tag{13}$$

with n_c the number of clusters used in the improved HMS software, cf. Amdahl's law [20]. Note that g_t^e is an approximation of g_t, for two reasons. Primarily, the computational cost of all the integration points is assumed to be equal, which is not true in reality. In fact, different integration points in the model undergo a different number of updates based on the position of the point in the model and the applied loads and boundary conditions. For example, the undeformed integration points need no update but the integration points experiencing the largest deformation require the most updates. Additionally, there are some overheads to implement spatial clustering in HMS.

For the dynamic adaptive clustering approach, the gain in time may be additionally affected by the overhead due to construction and maintenance of the clustering information and may not reflect the actual savings in update operations. For this reason, we can measure the computational benefit of the proposed methods by the reduction in update operations. We define *gain in updates* g_u to be

$$g_u = \frac{u_r}{u_c}, \tag{14}$$

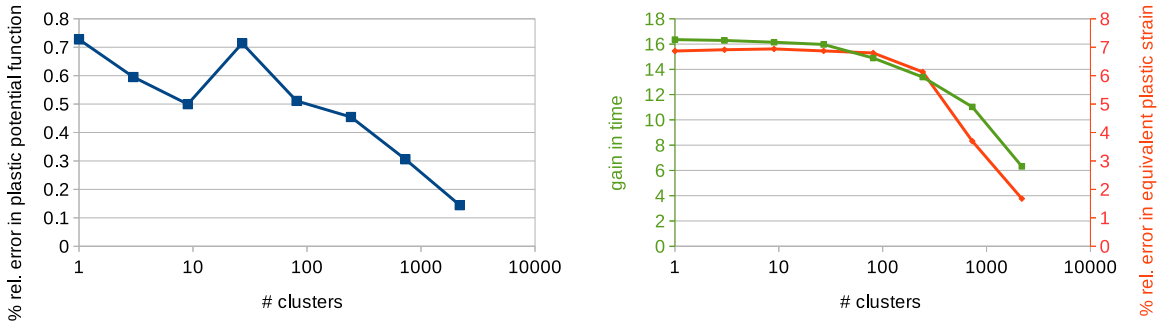
where u_r is the number of updates in the reference HMS and u_c is the number of updates in the HMS, which uses clustering.

4 RESULTS AND DISCUSSIONS

The improved HMS software is used to simulate the new test case described in 2.1. We compare the accuracy and the computational gain in the improved HMS and in the reference HMS. In all simulations the texture and plastic anisotropy at an integration point are updated if the norm of the accumulated plastic strain exceeds 0.05, i.e. $P_{cr} = 0.05$ in (2). The calculations were carried out on a twenty-core Intel Xeon compute node (2.8 GHz processor). Multiple cores were used for calculating the parameters of the plastic potential function. The simulation time with the reference HMS is around 47 h, of which 3 h is attributed to the macroscopic FE computations by Abaqus/Explicit 6.14-1 and required approximately 206,000 time increments in total. The remaining 44 h were consumed by 90,502 updates of the texture and the anisotropic plastic properties.

4.1 Static clustering

As described in section 2.2, the clusters are formed at the first update of texture and plastic anisotropy to obtain a balance between savings in computational cost and accuracy of the simulation. Figure 8(a) shows the measured approximation error for the plastic anisotropy (plastic potential function) $\bar{e^{\psi}}$, given by (8), for varying number of clusters. We observe a steady decrease in the approximation error with increasing of number of clusters, except for number of clusters around 30. Note that the approximation error is measured at the end of the simulation. So, it is possible that the statically constructed clusters, formed at the first update of texture and plastic anisotropy, are not representative for the spatial variation in plastic properties during the rest of the simulation. Figure 8b presents a comparison between the approximation error in the affected equivalent plastic strain field $\bar{e^{\gamma}}$ given by (10), and the corresponding gain in time g_t, given by (11). We see that varying the number of clusters up to 100 has no significant effect on the approximation error as well as on the gain in time. Hence, the error in equivalent plastic strain and the error in plastic potential function evolve differently as a function of the number of clusters. In fact, the equivalent plastic strain has a non-linear relationship with the plastic potential function. Moreover, in (8) we considered almost all directions in the strain rate space, whereas the equivalent plastic strain in (10) is affected only by the direction of corresponding strain rate.



(a) (b)

Figure 8: (a) Approximation error in plastic anisotropy (plastic potential function) $\bar{e^{\psi}}$. (b) Approximation error in equivalent plastic strain $\bar{e^{\gamma}}$ and gain in time g_t for varying number of clusters.

We measured that the evolution of texture and the associated plastic anisotropy consumes

94% of the simulation time (i.e. $f = 0.94$). Figure 9 compares the gain in time g_t obtained by the improved HMS software with the expected gain in time g_t^e for a varying number of clusters, see (13). We observe that the actual gain in time is higher than the expected gain. Possible causes are stated earlier.

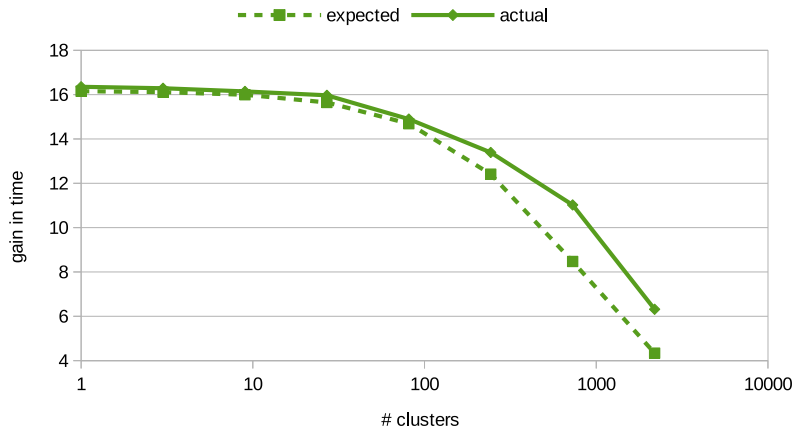


Figure 9: Actual gain in time g_t compared to the expected gain in time g_t^e for varying number of clusters.

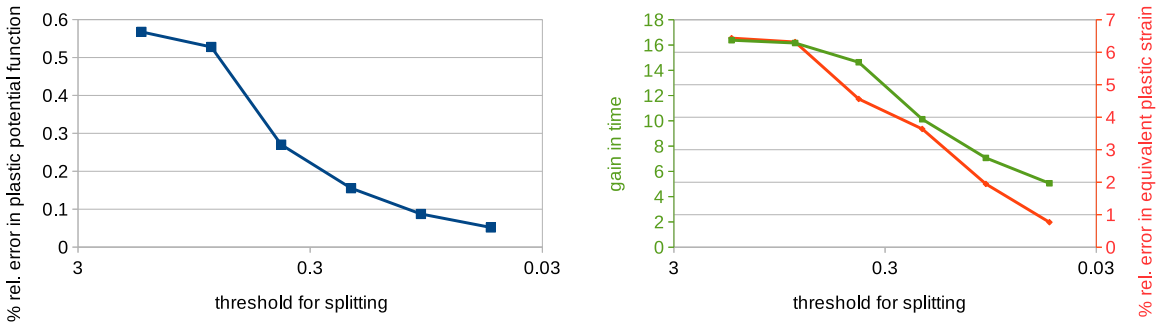
4.2 Dynamic adaptive clustering

A set of simulations with the same specimen and same settings were run using adaptive clustering. We mentioned above that a cluster is split if the difference between the maximum and the minimum of the equivalent plastic strain within a cluster exceeds the specified threshold value. If this threshold value increases, a cluster remains undivided for longer time, fewer clusters are constructed and the required number of update operations is reduced. Thus, a larger gain in time is achieved, whereby the approximation error increases. In Figure 10a we observe that the approximation error in plastic potential function $e^{\bar{\psi}}$ decreases rapidly with decreasing threshold values, except for very small threshold values where clustering reached the level of saturation. Figure 10b represents the effect of the adaptive clustering approach on the approximation error in the equivalent plastic strain $e^{\bar{\gamma}}$ and on the gain in time g_t for varying threshold value for splitting. Again, the general tendency of the approximation error $e^{\bar{\gamma}}$ is decreasing with decreasing threshold value (increasing number of effective clusters), but there is not a linear correspondence between the approximation error in plastic potential function $e^{\bar{\psi}}$ and in the equivalent plastic strain $e^{\bar{\gamma}}$.

For both static and dynamic adaptive clustering, we are interested in the evolution of the approximation error in the equivalent plastic strain $e^{\bar{\gamma}}$ and the gain in time g_t when the number of clusters increases. A clear relationship between approximation error and gain in computation time is observed in Figure 8b and Figure 10b. Moreover, the approximation error decreases faster than the gain in time, which is an important advantage.

4.3 Comparison between static and dynamic adaptive approach

Figure 11a and Figure 11b compare the clustering approaches w.r.t. obtained gain in time and accuracy. Clearly, the dynamic adaptive approach has a lower approximation error in plastic potential function $e^{\bar{\psi}}$, except when the approximation error is very high or very low. On the



(a) (b)

Figure 10: (a) Approximation error in plastic anisotropy (plastic potential function) $e^{\bar{\psi}}$, and (b) approximation error in equivalent plastic strain $e^{\bar{\gamma}}$ and gain in time g_t for varying threshold value for splitting a cluster.

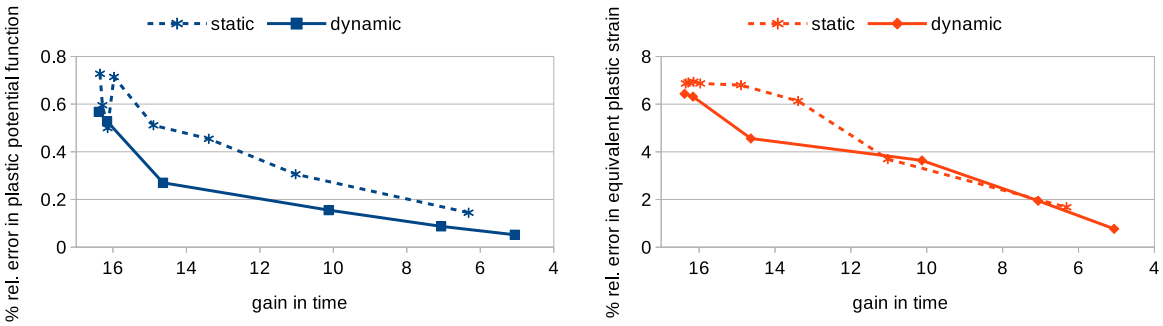
other hand, when we aim at a high gain in time, dynamic adaptive clustering is apparently superior to static clustering w.r.t. approximation error in equivalent plastic strain $e^{\bar{\gamma}}$. However, for low gain in time values both clustering strategies give nearly identical results.

For a large number (e.g. ≥ 729) of statically constructed clusters, each consisting of a few points, the clusters are not affected much by the evolution of homogeneous field values. On the other hand, in the dynamic adaptive clustering approach with a small threshold value (e.g. ≤ 0.2) we end up with a large number of clusters and the construction and maintenance of them require a significant amount of computation time. For example, with threshold value of 0.2 we have 1949 clusters at the end of the simulation, which requires at least 1948 split operations. Around 33 m of computation time is spent for this overhead, if a single split takes 1 s (roughly calculated), which is completely avoided in static clustering. This overhead increases exponentially as the number of generated clusters increases exponentially with the threshold value, also seen in Figure 12. However, if we ignore the computation time and pay attention to the savings in updates, the significance of the dynamic adaptive approach can be shown, see Figure 13. Again, the approximation error in plastic potential function is significantly less for the dynamic adaptive approach and the gap reduces for large number of static clusters. The dynamic adaptive approach leads to a lower approximation error also in equivalent plastic strain. Thus, we can infer that the dynamic adaptive approach can better capture the real dynamics and the deformation history.

5 CONCLUSIONS AND FUTURE WORK

The results presented above show that spatial clustering can considerably reduce the computational cost of the HMS model while retaining acceptable accuracy. Also, dynamic adaptive clustering is preferable to static clustering as the former has lower approximation error with the same number of updates. Additionally, 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.

At present, we only consider the equivalent plastic strain value at an integration point as the feature of interest for clustering. As described earlier, each of the strain tensor components influences the evolution of the texture and the associated plastic anisotropy. Thus clustering based on the strain tensor components is expected to generate more accurate clusters and sub-



(a) (b)

Figure 11: Approximation error in (a) plastic potential function, (b) equivalent plastic strain for varying gain in time.

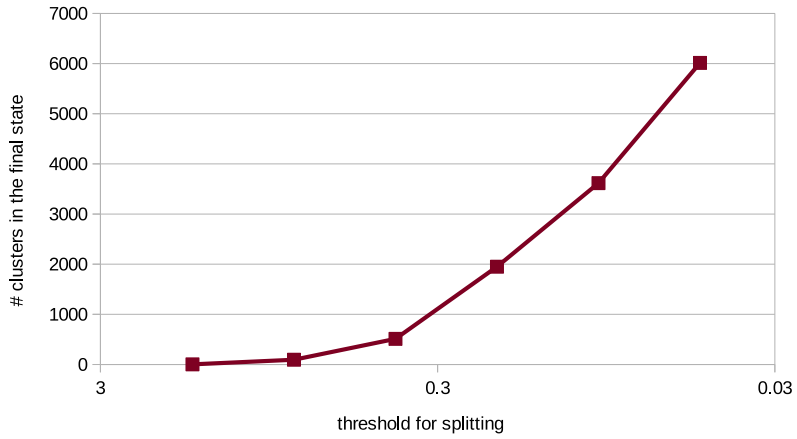
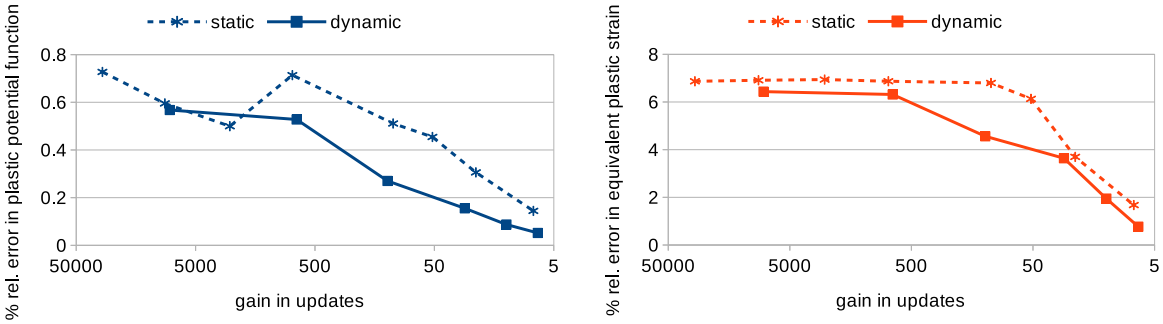


Figure 12: Resulting number of clusters in the dynamic adaptive approach for varying threshold value for splitting a cluster.



(a) (b)

Figure 13: Approximation error in (a) plastic potential function, (b) equivalent plastic strain for varying gain in updates (g_u), given in 14.

sequently a lower approximation error. We intend to implement a strain mode aware clustering and updating.

The overhead of the dynamic adaptive clustering approach is significant in the simulation experiments with the current test case. In the current approach of adaptive clustering, only

splitting of clusters is considered. We can also consider merging two adjacent clusters, which will effectively reduce the number of active clusters and update operations.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support from the Knowledge Platform M2Form, funded by IOF KU Leuven, and from the Belgian Federal Science Policy agency, contracts IAP7/19 and IAP7/21. The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Hercules Foundation and the Flemish Government department EWI.

REFERENCES

[1] J. Gawad, A. Van Bael, P. Eyckens, G. Samaey, P. Van Houtte, D. Roose, Hierarchical multi-scale modeling of texture induced plastic anisotropy in sheet forming. *Computational Material Science*, **66**:65–83, 2013.

[2] H.-J. Bunge, Texture analysis in materials science, Butterworth, London, 1982.

[3] P. Van Houtte, S. Li, M. Seefeldt, L. Delannay, Deformation texture prediction: from the Taylor model to the advanced Lamel model, *International Journal of Plasticity*, **21(3)**:589–624, 2005.

[4] J. Han, M. Kamber, A.K.H. Tung, Spatial clustering methods in data mining: a survey. *Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS*, 2001.

[5] C.A. Duartea, T.J. Liszka, W.W. Tworzydlob, Clustered generalized finite element methods for mesh unrefinement, non-matching and invalid meshes. *International Journal for Numerical Methods in Engineering*, 1–28, 2006.

[6] R.L. Bras, T. Damoulas, J.M. Gregoire, A. Sabharwal, C.P. Gomes, R.B.V. Dover, Computational thinking for material discovery: bridging constraint reasoning and learning. *CROCS 2010 - 2nd International Workshop on Constraint Reasoning and Optimization for Computational Sustainability*, 2010.

[7] A.S. Varde, E.A. Rundensteiner, C. Ruiz, D.C. Brown, M. Maniruzzaman, R.D. Sisson Jr., Integrating clustering and classification for estimating process variables in materials science. *Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, **13**:495–496 (Preface), 2006.

[8] A. Nakano, Fuzzy clustering approach to hierarchical molecular dynamics simulation of multiscale materials phenomena. *Computer Physics Communications*, **105(2–3)**:139–150, 1997.

[9] P.G. Ferreira, C.G. Silva, P.J. Azevedo, R.M.M. Brito, Spatial clustering of molecular dynamics trajectories in protein unfolding simulations. *Computational Intelligence Methods for Bioinformatics and Biostatistics, Lecture Notes in Computer Science*, **5488**:156–166, 2009.

[10] A. Alzraiee, L.A. Garcia, Using cluster analysis of hydraulic conductivity realizations to reduce computational time for Monte Carlo simulations. *Journal of Irrigation and Drainage Engineering*, **138(5)**:424–436, 2012.

[11] M. Vannucci, G.F. Porzio, V. Colla, B. Fornai, Use of clustering and interpolation techniques for the time-efficient simulation of complex models within optimization tasks *5th European Symposium on Computer Modeling and Simulation*, 6–11, 2011.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, **12**:2825–2830, 2011.

[13] H. Kriegel, P. Kröger, A. Zimek, Clustering High-dimensional Data: A Survey on Subspace Clustering, Pattern-based Clustering, and Correlation Clustering. *ACM Trans. Knowl. Discov. Data*, **3(1)**:1–58, 2009.

[14] J.C. Simo, T.J.R. Hughes, Computational Inelasticity. *Interdisciplinary Applied Mathematics*, **7**, Springer, New York, 1998.

[15] M.L. Wilkins, Calculation of elastic-plastic flow. *Methods in Computational Physics*, **3**:211–263, 1964.

[16] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B*, **63(2)**:411–423, 2001.

[17] R. Lletí, M.C. Ortiz, L.A. Sarabia, M.S. Sánchez, Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica Acta*, **515(1)**:87–100, 2004.

[18] M. Khairullah, J. Gawad, D. Roose, A. Van Bael, Accelerating the hierarchical multi-scale software by spatial clustering strategies. *Computational Plasticity XIII Fundamentals and Applications*, 932–943, 2015.

[19] P. Van Houtte, S.K. Yerra, A. Van Bael, The Facet method: A hierarchical multilevel modelling scheme for anisotropic convex plastic potentials, *International Journal of Plasticity*, **25(2)**:332–360, 2009.

[20] G.M. Amdahl, Validity of the single-processor approach to achieving large scale computing capabilities. *In AFIPS Conference Proceedings*, **30**:483–485, 1967.