# Accelerating Heterogeneous Multiscale Simulations of Advanced Materials Properties with Graph-Based Clustering

Maxime Vassaux, Krishnakumar Gopalakrishnan, Robert Sinclair, Robin Richardson, Peter Coveney

# Accelerating heterogeneous multiscale simulations of advanced materials properties with graph-based clustering

*Maxime Vassaux Krishnakumar Gopalakrishnan Robert C. Sinclair Robin. A. Richardson Peter V. Coveney**

Dr. M.V. (1), Dr. K.K. (1), Dr. R.C.S. (1), Dr. R.A.R. (1,2), Dr. P.V.C.(1)
(1) Centre for Computational Sciences - University College London, 20 Gordon Street, London, WC1H 0AJ, United Kingdom
(2) Netherlands eScience Center, Science Park 140, 1098 XG, Amsterdam
*p.v.coveney@ucl.ac.uk

Keywords: *multiscale modelling, model reduction, unsupervised learning, clustering, spline, graph theory*

Heterogeneous multiscale methods (HMM) capable of simulating asynchronously multiple scales concurrently are now tractable with the advent of exascale supercomputers. However, naive implementations display a large number of redundancies and are very costly. The macroscale model typically requires computations of a large number of very similar microscale simulations. In hierarchical methods, this is barely an issue as phenomenological constitutive models are inexpensive. However, when microscale simulations require for example high-dimensional molecular dynamics (MD) or finite element (FE) simulations, redundancy must be avoided. We propose a clustering algorithm suited for HMM workflows which automatically sorts and eliminates redundant microscale simulations. The algorithm features a combination of splines to render a low-dimension representation of the parameter configurations of microscale simulations and a graph network representation based on their similarity. The algorithm enables the clustering of similar parameter configurations into a single one in order to reduce to a minimum the number of microscale simulations required. We describe an implementation of the algorithm in the context of an HMM application coupling FE and MD to predict the chemically-specific mechanical behaviour of polymer-graphene nanocomposites. The algorithm furnishes a threefold reduction of the computational effort with limited loss of accuracy.

## 1 Introduction

Computer simulations of any system, whether inspired by experimental physics or industrial applications, that rely on a discretised description often present computational redundancies. Whether the system is simulated by solving a finite set of ordinary differential equations (ODEs) or partial differential equations (PDEs) it is frequently the case that computations may be performed which are very similar or even redundant.
The simulation of a material system typically implies finding the dynamics of a bulk under certain initial and boundary conditions. In doing so, the local thermodynamic state of the material needs to be fully determined, as well as their associated internal variables. Depending on the size of the system and its level of discretisation, the number of spatial locations where the local thermodynamic state must be computed will vary, often considerably. However, some spatial locations might be found concurrently in an identical or at least similar local thermodynamic state. Some spatial locations may also be found in a previously observed thermodynamic state. The state variables at given locations being similar, the full local thermodynamic state must also be. Based on this observation, one can choose to avoid computing the local thermodynamic state of all similar locations and only perform a few. The computational cost of the ensemble of evaluations of the local thermodynamic state is not necessarily significant with respect to the rest of the simulation. The cost is influenced by the complexity of the method chosen to express the thermodynamic state. However, in some cases, it is definitely worth exploiting these redundancies.

Let us consider the case of continuum mechanics applied to a volume of material as an illustrative example. The simulation of the mechanics of a solid requires one to solve the boundary value problem (BVP) comprising the balance of linear momentum PDE, initial and boundary conditions, and a local constitutive model. The constitutive model expresses the full local thermodynamic state of the material, most often relating the stress to the strain tensor. The resolution of the constitutive model determines the internal variable as a function of the state variable, which in turn enables one to solve the global BVP.

The resolution of the constitutive model for two configurations with identical state variables values results in identical internal variables. Now consider solving the BVP using the finite element (FE) method. The kinematic variables are computed at every spatial location, more specifically the displacement at the nodes of the mesh and the strain tensor at the quadrature points. The stress tensor can now also be evaluated at the quadrature points. In a single-scale problem, the constitutive model is described by a low-dimension set of equations whose only unknowns are the six independent components of the stress tensor. The limited computational effort required to evaluate the stress tensor in a given location usually does not justify an attempt to reduce the number of evaluations made.

In a multiscale problem, however, the computational effort to evaluate the stress is much more significant [1]. In heterogeneous multiscale methods [2–4], also called semi-concurrent methods [5], the evaluation of the stress tensor requires the simulation of a microscopic model constrained with the current thermodynamic state of the material (e.g. the current strain tensor in reversible mechanics). The finite element square (FE2) method [6, 7] or our computational workflow SCEMa [8], respectively rely on the FE or molecular dynamics (MD) simulations to predict the stress tensor. The computational effort associated with these microscopic FE or MD simulations is highly dependent on the amount of spatiotemporal detail embedded, and can rapidly amount to a few core hours. Bearing in mind that stresses need to be evaluated at each quadrature point at each time step to solve our BVP, it is essential to reduce the number of these evaluations. Otherwise, one's computational budget would be wasted before having solved even the slightest portion of the BVP. In this paper we will not focus on reducing details in the microscale model using surrogate modelling, for example ML-based models [9–11] which would revert to constitutive modelling. We will neither make use of model reduction methods such as coarse-graining [12, 13] for MD or proper othogonal decomposition [14, 15] and proper generalised decomposition [16, 17] for FE. Now we are particularly interested in techniques enabling the reduction of the number of evaluations of the macroscopic stresses. Reducing the dimensionality of the FE solution of our BVP is one way forward. Model reduction methods mentioned above are entirely applicable, but are highly dependent on the constitutive model used and therefore not compatible with replacing the constitutive model by a microscopic simulation.

The multiscale simulation methods we refer to are effective because they permit replacement of constitutive equations by fully detailed microscale simulations. When it comes to investigating the emergence of the materials properties of nanomaterials, the capability of these multiscale methods to preserve chemical specificity of the material is essential [18]. We have already applied the HMM to predict engineering properties of epoxy resins [8] and graphene-epoxy nanocomposites [19]. However, because the cost of such a multiscale approach remains unreasonably high we have not made use of the most versatile but expensive force fields (e.g. ReaxFF [20]) in order to capture complex mechanisms such as fracture.

We here propose to employ clustering techniques, such as those found in unsupervised learning approaches [21–23]. Unlike the aforementioned model reduction techniques, clustering does not require extensive calibration and validation, in other words *training*. The aim is not to reduce the dimensionality of the solution of our BVP but to directly identify redundant stress evaluations by clustering them and associate each cluster with a single microscale simulation. In continuum mechanics, the themodynamic state at a time $t$ and therefore the stress $\underline{\underline{\sigma}}^t$ depends on the atoms position at the previous time $\underline{u}^{t-1}$ and the applied strain $\underline{\underline{\epsilon}}^t$. As a result, $\underline{\underline{\sigma}}^t$ can be written as a function of the atoms initial position $\underline{u}^0$ and the strain trajectory $\{\underline{\underline{\epsilon}}\}_t$:

$$\begin{cases} \underline{\underline{\sigma}}^t = f\left(\underline{u}^{t-1}, \underline{\underline{\epsilon}}^t\right) \\ \underline{u}^{t-1} = h\left(\underline{u}^{t-2}, \underline{\underline{\epsilon}}^{t-2}\right) \end{cases} \implies \underline{\underline{\sigma}}^t = \bar{f}(\underline{u}^0, \underline{\underline{\epsilon}}^0, ..., \underline{\underline{\epsilon}}^t) = \bar{f}(\underline{u}^0, \{\underline{\underline{\epsilon}}\}_t) = \bar{f}(\{\underline{\underline{\epsilon}}\}_t) \tag{1}$$

In our multiscale simulation algorithm, SCEMa, the quadrature points in the macroscopic FE model are each associated with an independent ensemble of replicas of the microscale molecular model (see figure 1). The simulations are facilitated by the deal.II Finite Element library [24] for the macroscale, and

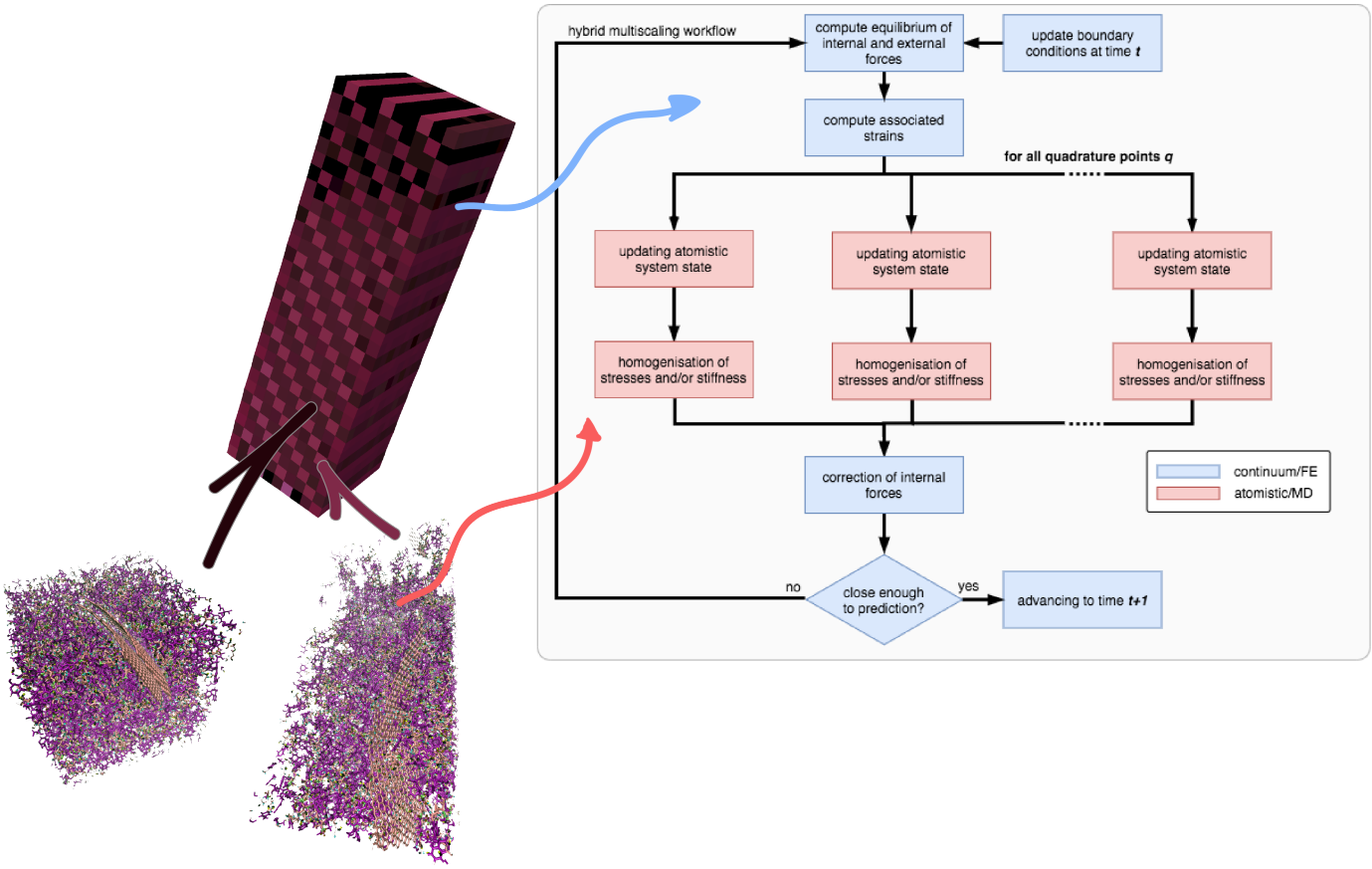



Figure 1: Workflow of a typical HMM application to simulate standard engineering test and predict materials properties of graphene-epoxy nanocomposites. Each quadrature point $q$ within the cells of the FE model (top left) queries an individual ensemble of simulations of replicas of the molecular model (bottom left). Replicas of the molecular model evolve independently, deforming differently according to the macroscopic strains or the microscale interatomic forces. However, some quadrature points may end up in a similar thermodynamic state, hence querying twice the same MD simulations.

LAMMPS [25] for the microscale. A complete description of the equations solved at the two scales, for both the continuum mechanics and the molecular dynamics problem, as well as a description of the scale separation scheme, can be found in Vassaux et al. [8]. Each of these ensembles of replicas departs from the same initial atomic configuration (i.e. position and velocities). Consequently, the thermodynamic state at each quadrature point only differs by its applied strain trajectory (see equation 1). In turn, we propose a clustering algorithm which focuses on gathering locations with identical strain trajectories. Building on our proposed clustering algorithm described in the section 2, we demonstrate significant speed-up of the multiscale simulation workflow with limited loss of accuracy with respect to global outputs at the macroscopic level in section 3. We also discuss the algorithm itself, its applicability, our results in section 4, and we present our conclusions in section 5.

## 2 Algorithm

The clustering algorithm which determines essential microscale simulations comprises three main stages:

1. Dimensionality reduction of the strain trajectory of each spatial location (quadrature point) of the macroscopic model.
2. Comparison of strain trajectories (all vs all).
3. Construction and coarse-graining of the resulting similarity graph.

The strain trajectory of any given point in the material consists of the six (unique) components of the strain tensor at that point ($\epsilon_{xx}$, $\epsilon_{yy}$, $\epsilon_{zz}$, $\epsilon_{xy}$, $\epsilon_{xz}$, $\epsilon_{yz}$), for each of the $N$ time steps the model has evolved

through. The final ($N^{th}$) entry is the deformation at the current step, for which the stress response is yet to be calculated (through execution of microscale simulations). This results in a $6N$ dimensional strain trajectory vector, where $N$ is the number of simulated macroscale timesteps. Even for short-duration simulations, inputs to the clustering algorithm rapidly become high-dimensional. A simple means of reducing trajectory vectors to a fixed and manageable length comes through the fitting of 6 splines, one for each component, along which $S_{cp}$ control points are placed evenly spaced in time. This results in strain trajectory vectors of fixed dimension $6S_{cp}$ for any given step. Such a reduced trajectory vector is calculated for every relevant location in the material – in the present case of our macroscopic model's FE mesh, this is at each quadrature point.

As performed in hierarchical clustering analysis, evaluation of the similarity of strain trajectories requires an all-vs-all pairwise comparison of each reduced trajectory vector $\tilde{\epsilon}$ and a similarity metric. In this work, we use a simple L2-norm as the measure of similarity. The similarity $s_{A,B}$ between the strain trajectories of quadrature points $A$ and $B$ (respectively $\tilde{\epsilon}_A$ and $\tilde{\epsilon}_B$) is computed as follows:

$$\frac{1}{s_{A,B}} \propto \sum_{n=1}^{S_{cp}} \sum_{i=1}^{6} \sqrt{\left(\tilde{\epsilon}_{A,n,i} - \tilde{\epsilon}_{B,n,i}\right)^2} \tag{2}$$

**Result:** list of similar trajectories to each quadrature point's strain trajectory
**for** *i in range(0, num_ranks)* **do**
  *target_rank* ← *this_rank* + *i* ; `// with ring periodic conditions`
  *sender_rank* ← *this_rank* − *i* ; `// with ring periodic conditions`
  **if** *target_rank != this_rank* **then**
    send all trajectories stored on *this_rank* to *target_rank*;
    receive list of trajectories sent to *this_rank* by *sender_rank*;
    **for** *each received trajectory A and each local trajectory B* **do**
      calculate distance $s_{A,B}$ between $A$ and $B$;
      **if** $s_{A,B} < \alpha$ **then**
        add $A$ to the list of similar trajectories to $B$;
      **end**
    **end**
  **else if** *target_rank == this_rank* **then**
    `/* no send/receive instructions */`
    **for** *each local trajectory A and each local trajectory B* **do**
      calculate distance $s_{A,B}$ between $A$ and $B$;
      **if** $s_{A,B} < \alpha$ **then**
        add $A$ to the list of similar trajectories to $B$;
      **end**
    **end**
  **end**
**end**

**Algorithm 1:** Description of the distributed algorithm computing the L2-norm distance between pairs of splines (associated with the strain trajectories of quadrature points).

Due to the distributed memory parallelism of the macroscopic model, different strain trajectory vectors may be held on each of the $P$ (MPI) ranks. As the goal of this work is to reduce the computational cost of the overall multiscale simulation, an efficient strategy for inter-rank communication must be chosen that minimises bottlenecks (see algorithm 1). We adopt a ring-like communication pattern in which any given rank, $p$, sends its trajectories to rank $p + i$ while receiving trajectories from rank $p - i$, repeated for all $i = 0..P-1$ and with periodic conditions on the rank number. Ranks compare incoming trajectory vectors with their own, storing the result locally if the similarity is within a certain (user-defined) threshold value $\alpha$. At $i = 0$ the rank compares its own trajectories with one another, without the need for communication. Once complete, each rank holds a record of all trajectories that are sufficiently similar (i.e. within the similarity threshold $\alpha$) to each trajectory stored on that rank.

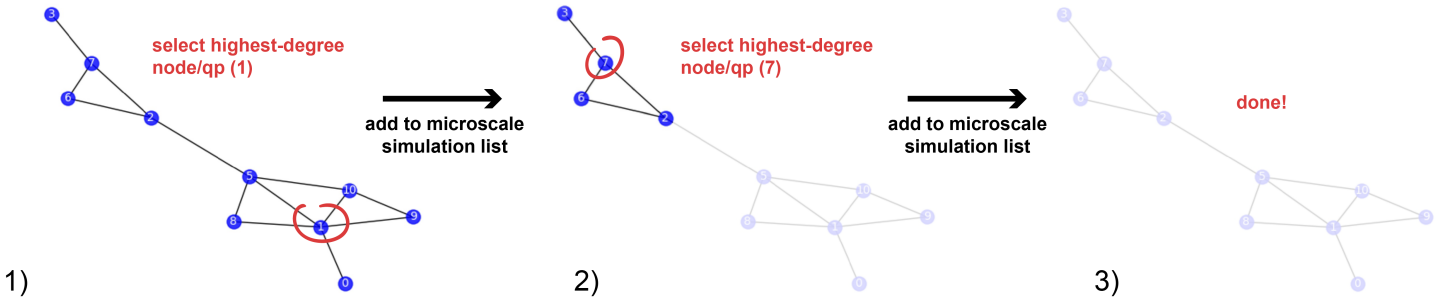



Figure 2: Schematic illustration of the clustering algorithm which decomposes the graph of similar trajectories into the smallest number of clusters of similar nodes.

This information can then be aggregated on a single rank and used to construct a graph in which each node represents a quadrature point of the macroscale mesh, and each edge links similar nodes. *Similarity* in this context means that quadrature points may be considered equivalent for the purposes of the current simulation step, and therefore that they may share the result of the same ensemble of microscale simulations rather than requiring a separate ensemble each. The goal is, therefore, to decompose this graph into the smallest possible number of clusters of similar nodes. This will result in the lowest number of distinct microscale simulation ensembles needing to be run for the step in question. While each pair of nodes sharing an edge is considered similar, nodes with more than two degrees of separation are not. The strategy we employ is to recursively choose the node with the highest degree (the highest number of edges), cluster it with all nodes in its direct entourage (its similar trajectories), and remove that cluster of nodes from the graph (see figure 2). The process is repeated until all nodes have been assigned to a cluster. Some resulting clusters may contain only a single node.

One ensemble of microscale simulations will be executed for each such cluster, and the resulting stress value used for every quadrature point whose corresponding trajectory node lies within that cluster. In practice, strain trajectories of points in the material that have diverged do not later converge, leading to a tree-like structure over time. Indeed, material regions that were previously regarded as similar begin branching off as their local strains begin to follow different paths. When this occurs, the atomistic state of the MD simulations (the microscale simulations of this work) must be duplicated and used as the starting structure for simulations in the new branch. This is important for conserving the non-linear, history-dependent effects of materials that have been strained beyond the elastic region.

## 3 Results

We now want to verify the implementation of the algorithm described in the previous section and to demonstrate its efficiency in reducing the number of microscale (expensive, and in our case, MD) simulations. The verification and demonstration are performed simultaneously by simulating two complementary test setups. The two setups consist of standard engineering tests: a pure tension test and a compact-tension test. Both generate a different type of mechanical state within the tested sample, namely on one side uniaxial (1D) tension, and on the other side multiaxial (3D) fully anisotropic and heterogeneous loading. The uniaxial setup is simulated with a coarse mesh involving only a couple of hundred quadrature points in the macroscale model in order to limit computational cost. The compact-tension setup is simulated with a more refined mesh involving tens of thousand quadrature points. The meshes and the boundary conditions applied in each setup are shown in figure 3.

The clustering algorithm features one main parameter, the so-called similarity threshold. It controls the trade-off between accuracy and computational reduction. For each setup, we perform a sensitivity analysis of the similarity threshold, demonstrating in the meantime how to choose an appropriate value. Accuracy is evaluated by computing the error on the output quantity of interest (that is the resulting force on the

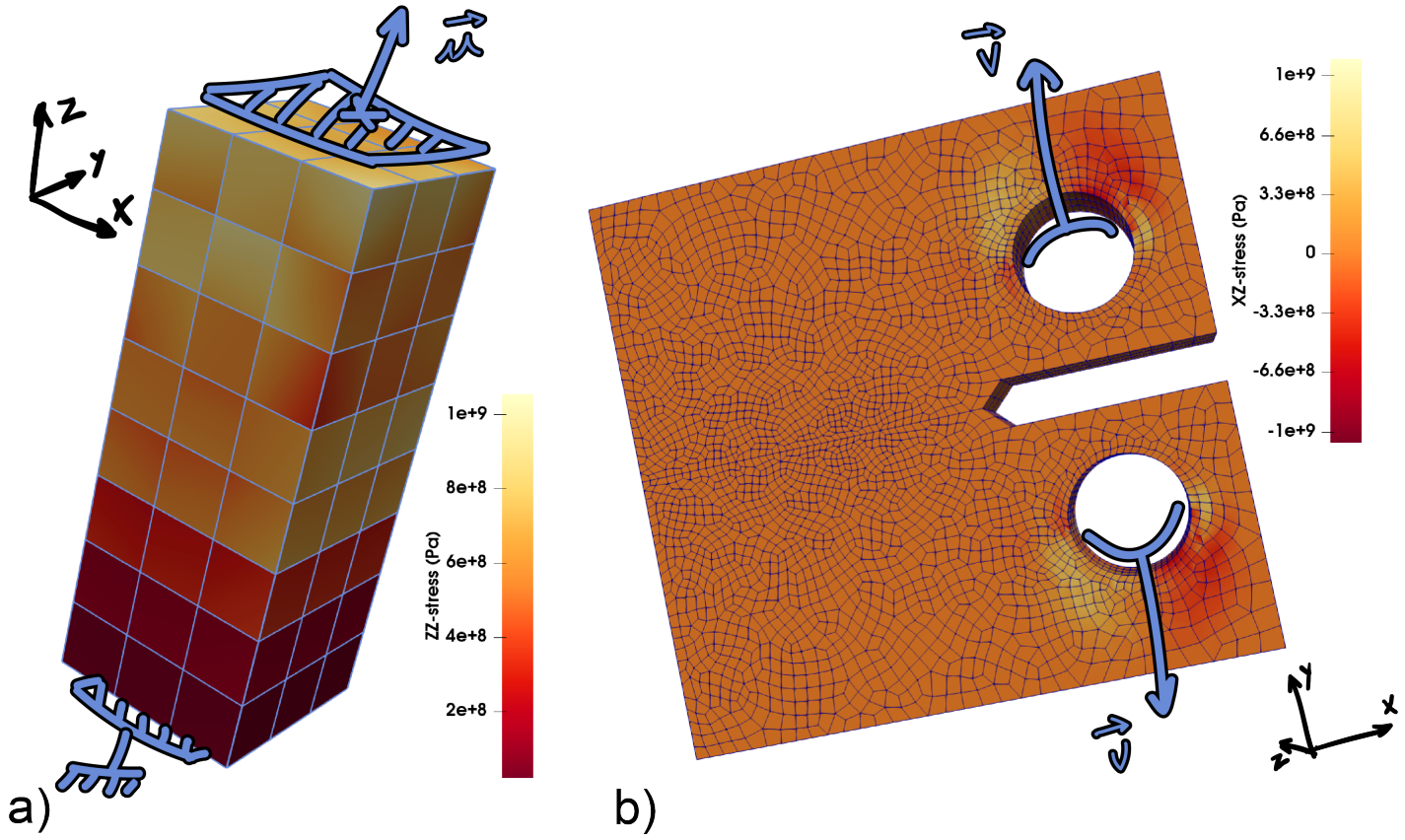



Figure 3: Visualisation of the two test setups simulation meshes as well as a schematic of the imposed displacement at the boundary: (a) uniaxial tension and (b) compact-tension. Boundary conditions are drawn schematically in blue, a displacement $\underline{u}$ and a velocity $\underline{v}$ are prescribed respectively for the uniaxial and the compact-tension setups. The colour legend correspond to the dominant component of the strain tensor within the system, namely the axial strain (ZZ) for the uniaxial tension setup and the shear strain (XZ) for the compact-tension setup.

Table 1: Clustering algorithm parameters

| Parameter | $\alpha$ | $S_{cp}$ | $t_a$ |
|---|---|---|---|
| Value | $10^{-14}$ through $10^{+1}$ | 10 | 5 |
| Role | similarity threshold | number of spline support points | number of steps before clustering activation |

boundary induced by imposed displacements) of the simulation with respect to a reference simulation for which the clustering algorithm is not applied. The computational reduction is defined as the number of MD simulations avoided by the use of the algorithm. In more detail, a simulation without clustering is run first to obtain the baseline reference results for comparison. Then, the parametric analysis of the clustering algorithm is performed by sweeping the similarity threshold parameter, $\alpha$ across multiple orders of magnitude from $10^{-14}$ through $10^{+1}$.

Furthermore: (i) constitutive relationships are only queried when the magnitude of the strain at a given quadrature point exceeds $10^{-10}$, otherwise no MD simulation is requested; (ii) splines fitting the strain trajectory of quadrature points comprise a constant number of support points ($S_{cp} = 10$), independently of the length of the trajectory; (iii) for sufficient precision of the strain trajectory comparison, the clustering algorithm is activated only after 5 timesteps have passed (vertical dashed line, see figures 4.a,b,c and 5.a,b,c). Table 1 summarises the complete set of model parameters and simulation conditions.

Each multiscale simulation is performed using the FE method with linear Lagrangian elements ($Q_1$). In turn, each FE cell features 8 quadrature points. In the absence of clustering, this implies that 8 MD simulations are required at each time-step for each cell to replace the constitutive relationship. In the following, we will consider that a MD simulation consists in evolving the dynamics of a single replica of the system. However, it is well known that ensembles of replicas are rather needed for accurate predictions (and often may contain tens to hundreds of replicas) [26]. When large ensembles are required, the benefits of the clustering algorithm are even more significant.

We will now work our way up in terms of increasing complexity and analyse successively the results of the simulations of the uniaxial and the compact-tension setups with and without the clustering algorithm.

### 3.1 Uniaxial tension setup

The setup consists of a 3x3x12cm$^3$ elastic cuboid (see figure 3.a) with properties equivalent to that of an graphene-epoxy nanocomposite. In this most simple case, the computational domain was constructed by meshing the cuboid into a 3x3x8 cartesian grid, that is a resolution of 0.01cm along each coordinate direction in three-dimensional space. A fixed time-stepping scheme is used to evolve the system's dynamics during 50 steps of 0.5$\mu$s each, that is 25$\mu$s in total. With this very coarse mesh, and assuming only 3 replicas simulated per quadrature point, the maximum number of MD simulations at each timestep is 1728. Vertical displacement is imposed on the upper surface at a constant strain rate of $2.10^{-3}$ s$^{-1}$, while the lower surface is fixed (see figure 3.a). In turn, the force resulting from the displacement is chosen as the quantity of interest.

The material is loaded dynamically, hence the oscillating nature of the resulting force (see figure 4.a). All simulations up to a similarity threshold value of $10^{-2}$ match perfectly the reference simulation (brown curve). This observation is confirmed by the computation of the evolution of the absolute error (force difference) for each simulation featuring the clustering algorithm and the reference simulation (see figure 4.b). The cumulative error (sum of the error at each timestep, see figure 4.d) remains below 1MN for simulations with a similarity threshold below $10^{-2}$ which, as we have seen, is equivalent to no discernible divergence of the global force evolution from the reference simulation.

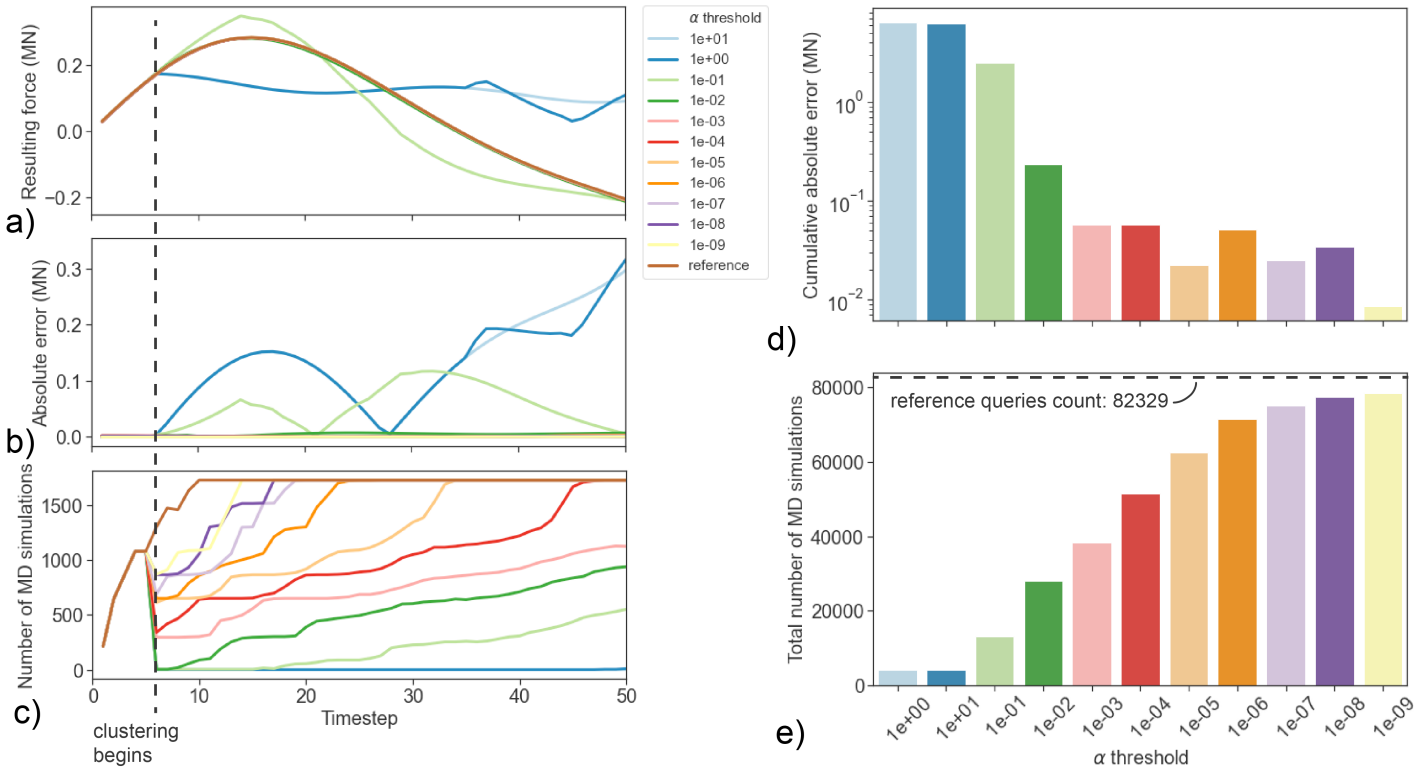



Figure 4: Influence of the similarity threshold of the clustering algorithm on the global accuracy and the computational cost of the simulation of the uniaxial tension test: evolution of (a) the resulting global force on the top surface of the system, (b) the error in the applied global force on the top surface with respect to the reference simulation (no clustering), (c) the number of MD simulations at each timestep; cumulative (d) error in the applied global force on the top surface; and (e) number of MD simulations throughout the complete simulations. In (a,b,c) the vertical dashed line indicates when the clustering algorithm starts to be applied, leading to the sudden drop in the number of MD queries.

The number of MD simulations at each timestep (see figure 4.c) increases steadily initially as the mechanical wave propagates through the material. Quadrature point strain becomes non-null, which causes the constitutive relationship to be probed (when the strain magnitude exceeds $10^{-10}$). The clustering algorithm is activated from timestep 6 (vertical dashed line, see figure 4.a,b,c), beyond which the count of MD simulations decreases abruptly for all similarity thresholds. Finally, the quadrature point strain trajectories start to diverge at a rate depending on the similarity threshold (see supplementary figure 1). The more tolerant (the higher threshold values), the slower the trajectories diverge and, the more the number of MD simulations increases. For low tolerance similarity thresholds, all quadrature point strain trajectories diverge before the end of the simulation, reaching a plateau at a count of 1728.

With such a benchmark, we are able to choose the right similarity threshold an optimal speed-up to accuracy compromise. The cumulative error only seems to increase significantly above a similarity threshold of $10^{-3}$. Assuming that we wish the cumulative error not to exceed 0.1MN, the clustering algorithm would generate a speed-up of 2.2 with respect to the reference simulation ($38,000$ against $82,000$ MD simulation queries). With a slightly increased tolerance, using a similarity threshold of $10^{-2}$, the clustering algorithm would generate a speed-up of 2.9.

### 3.2 Compact-tension setup

The compact-tension test consists of the setup prescribed in ASTM Standards [27] (ASTM/E1820) to assess fracture toughness. Making use of the symmetries, only a quarter (top, front part) of the structure shown in figure 3.b is simulated. The sample is heterogeneously meshed using GMSH [28] rendering 1318 tetrahedral cells, that is 10544 quadrature points. A fixed time-stepping scheme is used to evolve the macroscale model dynamics during 50 steps of $0.1\mu$s each, that is $5\mu$s in total. The holes in the structure are pulled apart at a speed of 0.1 mm.s$^{-1}$ (see figure 3.b).

The evolution of the resulting force on the inner part of the hole is non-steady (see figure 5.a). The force on the inner part of the hole is applied at a constant speed. The resulting force on the inner part of the hole increases non-linearly, reaching progressively 0.5MN in the reference simulation. Simulations featuring the clustering algorithm follow similar trends for threshold values up to 0.1. However, the reference trend is not followed as precisely as for the uniaxial test 3.1. Cumulative errors are systematically at least one order of magnitude higher (see figure 5.d). In order to limit the cumulative error to 0.1 MN, and replicate exactly force evolution trends from the reference simulations, similarity thresholds down to $10^{-14}$ must be chosen. At such threshold values, the computational gain is limited (see figure 5.e). However, if a reduced accuracy up to 1% can be tolerated, similarity threshold values up to $10^{-3}$ can be used as with the uniaxial test case. Once again, 3-fold computational speedups are attained.

Note that, between threshold values ranging from $10^{-3}$ to $10^{-10}$, the accuracy of the simulations remains almost identical, however it is assessed (evolution trends or cumulative error), but half of the MD simulation cost can be saved. For that level of accuracy, the benefits of the clustering are most obvious. For a threshold value of $10^{-1}$ the cumulative error appears to decrease, but this is more fortuitous than a systematic improvement. The force evolution diverges from the evolution of the reference simulation and follows a different trend, but fortunately remains closer than force evolution at lower threshold values.

Unlike the uniaxial tension test, the compact tension test generates highly spatially heterogeneous strain in the material sample. In turn, strain trajectories differ substantially from one quadrature point to another. The compact tension test certainly constitutes an extreme scenario in which to apply our clustering algorithm. Indeed, similar speed-ups of 3 are attained but cumulative errors are approximately one magnitude higher in the compact tension scenario. As might be expected, the algorithm proves to be somewhat less efficient for heterogeneous strain profiles.

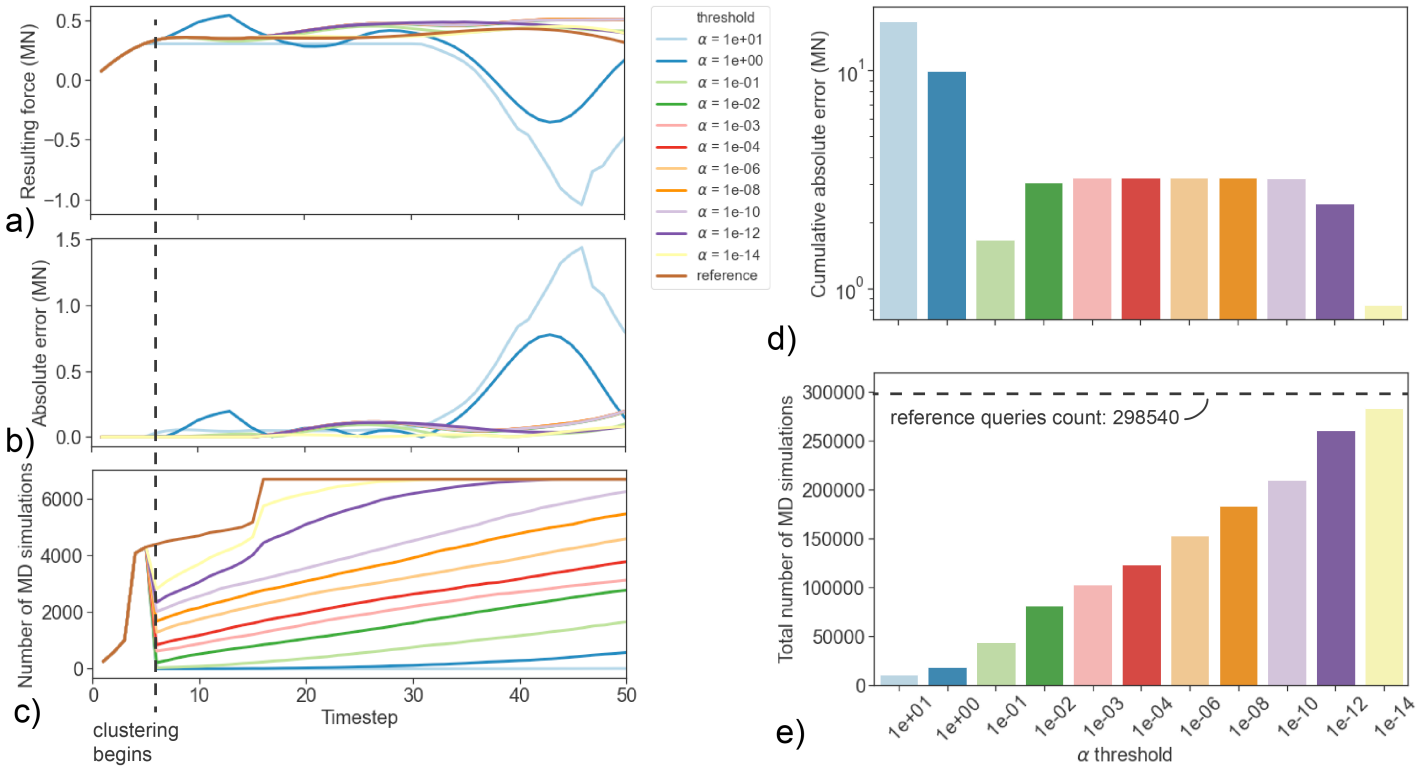



Figure 5: Influence of the similarity threshold of the clustering algorithm on the global accuracy and the computational cost of the simulation of the compact-tension test: evolution of (a) the resulting global force on the top surface, (b) the error in the applied global force on the top surface with respect to the reference simulation (no clustering), (c) the number of MD simulations at each timestep; cumulative (d) error in the applied global force on the top surface; and (e) number of MD simulations throughout the complete simulations. In (a,b,c) the vertical dashed line indicates when the clustering algorithm starts to be applied (identically to figure 4).

## 4 Discussions

The clustering algorithm introduced here has the capability to keep track of quadrature points of a FE mesh following similar evolutions of their mechanical state defined by their strain trajectories. It can be seen as a non-local (in space and time) coarsening of the mesh. This clustering algorithm is a typical example of *unsupervised learning* in the sense that it does not require any training data. It relies only on the current mechanical state of the quadrature points to achieve model reduction. Our approach makes use of the theory behind material deformation and exploits symmetries in the models to avoid running very similar (almost duplicate) simulations. As such it differs from traditional surrogate models, because it does not predict what the future stress response will be based on previous results. No arbitrary choice of training data has to be made, nor is there any need to ensure that the trained model generalises beyond this training dataset. Our approach is intended for simulations of materials for which a trivial algebraic description of the constitutive model is not easily available, for example, because constructing/training one would require too much data (and therefore be prohibitively expensive to acquire in the first place). However, by allowing the model to explore its (very high dimensional) phase space in the cheapest way possible, it could potentially also be used to aid in training of machine-learning based constitutive models.

The only preparatory work that is required to make use of our clustering algorithm is the selection of its two parameters: the number of control points $S_{cp}$ of the fitted splines and the similarity threshold $\alpha$. Strain trajectories being relatively smooth, we have set $S_{cp}$ to 10 and left this parameter fixed during the entire study within this paper. In other words, in practice the clustering algorithm we propose here depends only on a single parameter: the similarity threshold $\alpha$. From the two setups simulated in the results section, we have observed that the value of $10^{-3}$ for the similarity threshold offering a good compromise of computations reduction and accuracy. In turn, $10^{-3}$ is recommended as a default value, hence automating the model reduction enabled by the clustering algorithm.

Up to three-fold speedups were attained in exchange for what we considered a reasonable loss of accuracy during the simulation of the two tested setups (see section 3. During those simulations, each request for an evaluation of a quadrature point stress by the macroscopic model led to the MD simulation of an ensemble of replicas of the microscale model. Three replicas per ensemble were considered. Looking for accurate and reproducible results from MD simulations, ensembles generally contain a much larger number of replicas [29]. Because the microscale simulations do not depend on each other, the overall computational cost of the multiscale simulation grows linearly with the number of stress queries. In turn, since the speedup does not vary with the number of quadrature points or the number of replicas in an ensemble, no increase in the speedup is to be expected when simulating larger meshes or more realistic ensemble sizes. However, considering a fixed setup geometry, increasing the number of quadrature points in the mesh increases the likelihood of quadrature points experiencing a similar thermodynamic state. As a result, the expected speedup induced by the clustering algorithm increases. Furthermore, the absolute computational savings will increase linearly with these features of the simulation, so the speedup is expected to be maintained as the multiscale simulation grows. Furthermore, the present algorithm is a simple clustering method, but we expect speedups to increase making use of more refined unsupervised learning algorithms [30]. In particular, replacing spline-based dimension-reduction with more robust methods derived from principal component [31] analysis will enable more accurate comparison of the strain trajectories.

The relation between threshold values and speedup is not straightforward. Identical threshold values neither enable identical speedups nor accuracy reduction when comparing the results of the uniaxial and the compact tension case. At a given threshold value one cannot expect the same MD simulations to be spared independently of the test setup. Indeed, the similarity of the strain trajectories is assessed in terms of absolute amplitude values, and is therefore dependent on the materials properties and the test setup. To that extent, the clustering algorithm requires additional work to be calibrated and to find an optimum of accuracy and speedup, as is done in section 3. Then, it makes even more sense to use such an algorithm in the context of performing uncertainty quantification [32, 33], whereby the materials properties and the

test case parameters change by only limited amounts.

There is a non-negligible computational and storage cost associated with keeping track of strain trajectories, performing spline-fitting, measuring the distance between 6-dimensional splines, and finding the graph-informed optimal clustering of quadrature points. In classical single-scale applications of the FE method, the local computations associated with cells or quadrature points are small and of the same order of magnitude as the costs associated with the clustering algorithm. The cost of the clustering algorithm will certainly balance out the gain associated with reducing the number of these local computations. However, in multiscale use cases, where these local computations are expensive MD simulations, the gain is obvious. The cost of each individual MD simulation is of the order of 0.1 core hours, while the cost of the clustering algorithm at a given timestep of the compact-tension setup simulation is about 1 core hour. We have seen in the results section that we can spare up to two-thirds of the constitutive relationship queries per timestep while maintaining accuracy. The computational reduction is then clear and substantial.

In this work, we have chosen to quantify 'similarity' between two strain trajectories simply as the Euclidean distance (L2 norm), although this may not necessarily be the best choice in all contexts. Other measures of similarity are indeed possible, such as the L1 norm, and may assign importance to different characteristics (for example, by penalising momentary deviations more harshly). As the goal of the present work is to reduce overall simulation time in a heterogeneous multiscale model, the chosen similarity metric should be of low computational cost.

This work may also benefit in future from the use of more sophisticated graph clustering algorithms. The present use of thresholding implicitly assumes that all trajectories falling within the threshold similarity are equally similar. In contrast, a more precise selection of clusters may be possible by considering the individual values of similarity between each pair of nodes (and eschewing thresholding altogether). The graph reduction stage is currently serial (although very rapid in comparison to the other stages), more sophisticated clustering approaches would ideally permit some level of parallelism.

## 5 Conclusion

Concurrent multiscale simulations of physical processes are now becoming tractable with the advent of exascale computing [1], yet they remain expensive. We have proposed a novel algorithm which can be defined as a non-local model reduction technique that enables one to reduce the number of lower scale expensive simulations. We have implemented the algorithm in SCEMa, a FE-MD coupling library. The algorithm is based on a clustering technique to detect redundant stress queries, based on the similarity of the strain trajectories associated with the FE quadrature points. We have provided an in-depth description of the algorithm featuring spline-based dimension-reduction of the strain trajectories and the graph-based selection of essential MD simulations. We have also demonstrated the efficiency of the algorithm in speeding up around three-fold the simulations of a uniaxial tension test and a complex compact tension test, with limited loss of accuracy compared to a non-reduced simulation. For such expensive multiscale workflows the algorithm has the potential to save many millions of core hours. Finally we point out that the algorithm is of course not limited to multiscale workflows relying on the finite element method: it can be easily transposed to any type of asynchronous multiscale workflow.

## Acknowledgments

The authors acknowledge funding support from the European Union's Horizon 2020 Research and Innovation programme under grant agreement 800925 (VECMA project, `www.vecma.eu`), and from the EPSRC UK Consortium on Mesoscale Engineering Sciences (UKCOMES, `www.ukcomes.org`) EP/R029598/1. R.C.S was funded by an EPSRC studentship in association with the Molecular Modelling & Material Science Centre for Doctoral Training at UCL. The authors are grateful to the Leibniz Supercomputing Centre

for providing access to SuperMUC. The authors also made use of ARCHER, the UK's National High Performance Computing Service, funded by the Office of Science and Technology through the EPSRC's High-End Computing Programme.

## Authors contributions

M.V. contributed to the implementation of the clustering algorithm within SCEMa and to part of the validation, K.K. contributed to part of the validation, R.C.S. to the implementation of the clustering algorithm within SCEMa, R.A.R. conceived, designed and implemented the clustering algorithm, P.V.C. initiated the project and supervised the research; all authors contributed to the writing of the manuscript.

## Supporting information

The source code of the clustering algorithm, implemented within SCEMa and the data used in the section 3 to perform validation are available on GitHub: `https://github.com/UCL-CCS/SCEMa/releases/tag/0.1.1`.

## References

[1] A. G. Hoekstra, B. Chopard, D. Coster, S. Portegies Zwart, P. V. Coveney, *Phil Trans R Soc A* **2018**.

[2] W. E, B. Engquist, Z. Huang, *Physical Review B* **2003**, *67*, 9 092101.

[3] H. Talebi, M. Silani, S. P. A. Bordas, P. Kerfriden, T. Rabczuk, *Computational Mechanics* **2014**, *53*, 5 1047.

[4] T. D. Scheibe, E. M. Murphy, X. Chen, A. K. Rice, K. C. Carroll, B. J. Palmer, A. M. Tartakovsky, I. Battiato, B. D. Wood, *Groundwater* **2015**, *53*, 1 38.

[5] S. P. Xiao, T. Belytschko, *Computer Methods in Applied Mechanics and Engineering* **2004**, *193*, 17–20 1645.

[6] F. Feyel, J.-L. Chaboche, *Computer Methods in Applied Mechanics and Engineering* **2000**, *183*, 3–4 309.

[7] J. Oliver, M. Caicedo, E. Roubin, A. E. Huespe, J. A. Hernández, *Computer Methods in Applied Mechanics and Engineering* **2015**, *294* 384.

[8] M. Vassaux, R. A. Richardson, P. V. Coveney, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2019**, *377*, 2142 20180150.

[9] Z. Liu, C. T. Wu, M. Koishi, *Computer Methods in Applied Mechanics and Engineering* **2019**, *345* 1138.

[10] D. Z. Huang, K. Xu, C. Farhat, E. Darve, *arXiv:1905.12530 [physics]* **2020**, arXiv: 1905.12530.

[11] J. Kadupitiya, G. C. Fox, V. Jadhao, In J. M. F. Rodrigues, P. J. S. Cardoso, J. Monteiro, R. Lam, V. V. Krzhizhanovskaya, M. H. Lees, J. J. Dongarra, P. M. Sloot, editors, *Computational Science – ICCS 2019*, Lecture Notes in Computer Science. Springer International Publishing, Cham, ISBN 978-3-030-22741-8, **2019** 116–130.

[12] E. Brini, E. A. Algaer, P. Ganguly, C. Li, F. Rodríguez-Ropero, N. F. van der Vegt, *Soft Matter* **2013**, *9*, 7 2108.

[13] L. Zhang, J. Han, H. Wang, R. Car, W. E, *The Journal of Chemical Physics* **2018**, *149*, 3 034101.

[14] G. Berkooz, P. Holmes, J. L. Lumley, *Annual Review of Fluid Mechanics* **1993**, *25*, 1 539.

[15] P. Héas, C. Herzet, *Archives of Computational Methods in Engineering* **2018**, *25*, 1 87.

[16] F. Chinesta, R. Keunings, A. Leygue, *The Proper Generalized Decomposition for Advanced Numerical Simulations: A Primer*, Springer Science & Business Media, **2013**, google-Books-ID: CfS3BAAAQBAJ.

[17] M. Vitse, D. Néron, P. A. Boucard, *Finite Elements in Analysis and Design* **2019**, *153* 22.

[18] M. Vassaux, R. C. Sinclair, R. A. Richardson, J. L. Suter, P. V. Coveney, *Advanced Theory and Simulations* **2020**, *3*, 1 1900122.

[19] M. Vassaux, R. C. Sinclair, R. A. Richardson, J. L. Suter, P. V. Coveney, *Advanced Theory and Simulations* **2019**, *2*, 5 1800168.

[20] T. R. Mattsson, J. M. D. Lane, K. R. Cochrane, M. P. Desjarlais, A. P. Thompson, F. Pierce, G. S. Grest, *Physical Review B* **2010**, *81* 054103.

[21] J. A. Hartigan, M. A. Wong, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **1979**, *28*, 1 100, publisher: [Wiley, Royal Statistical Society].

[22] E. Hartuv, R. Shamir, *Information Processing Letters* **2000**, *76*, 4 175.

[23] Z. Liu, M. A. Bessa, W. K. Liu, *Computer Methods in Applied Mechanics and Engineering* **2016**, *306* 319.

[24] W. Bangerth, R. Hartmann, G. Kanschat, *ACM Trans. Math. Softw.* **2007**, *33*, 4.

[25] S. Plimpton, *Journal of Computational Physics* **1995**, *117*, 1 1.

[26] S. Wan, R. C. Sinclair, P. V. Coveney, *arXiv:2006.07104 [physics]* **2020**, arXiv: 2006.07104.

[27] ASTM, Standard Test Method for Short-Beam Strength of Polymer Matrix Composite Materials and Their Laminates, Technical report, ASTM International, **2016**.

[28] C. Geuzaine, J.-F. Remacle, *International Journal for Numerical Methods in Engineering* **2009**, *79*, 11 1309, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2579.

[29] D. W. Wright, B. A. Hall, O. A. Kenway, S. Jha, P. V. Coveney, *Journal of Chemical Theory and Computation* **2014**, *10*, 3 1228.

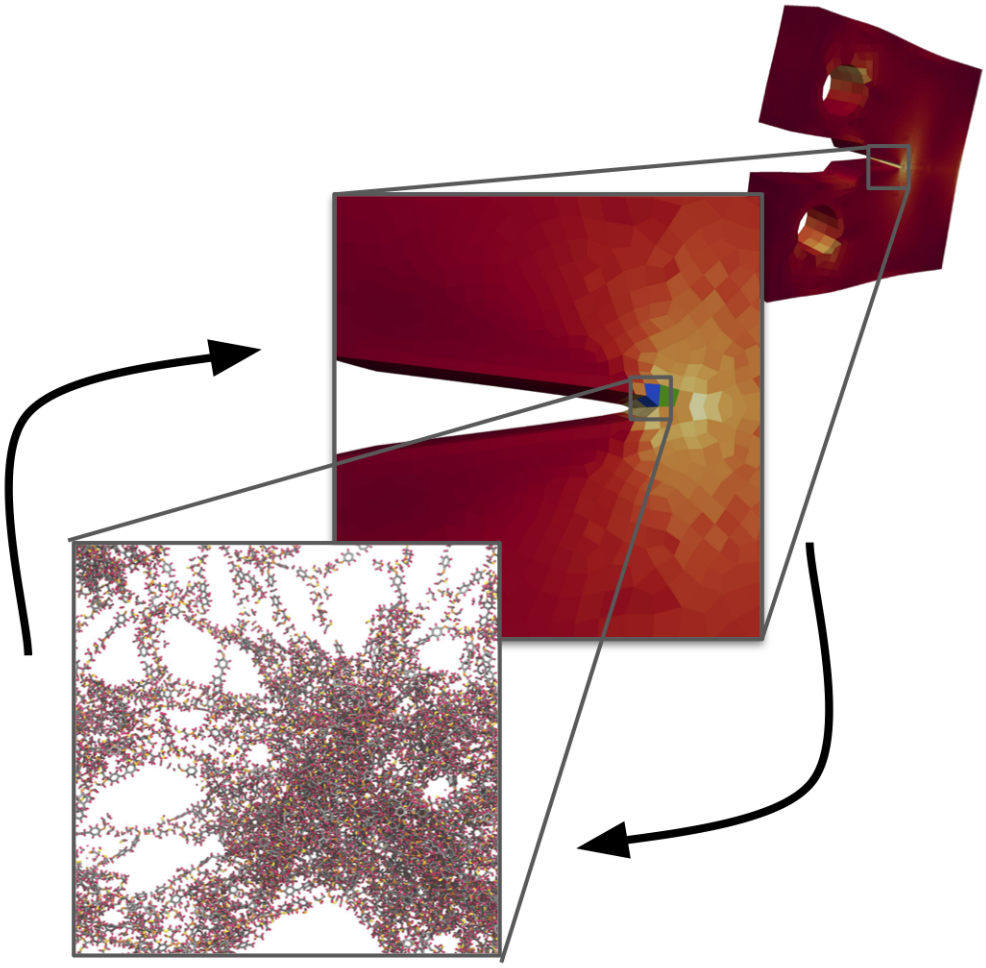
[30] L. Li, B. Guedj, S. Loustau, *Electronic Journal of Statistics* **2018**, *12*, 2 3071, publisher: The Institute of Mathematical Statistics and the Bernoulli Society.

[31] M. Ali, M. W. Jones, X. Xie, M. Williams, *The Visual Computer* **2019**, *35*, 6 1013.

[32] D. W. Wright, R. A. Richardson, W. Edeling, J. Lakhlili, R. C. Sinclair, V. Jancauskas, D. Suleimenova, B. Bosak, M. Kulczewski, T. Piontek, P. Kopta, I. Chirca, H. Arabnejad, O. O. Luk, O. Hoenen, J. Weglarz, D. Crommelin, D. Groen, P. V. Coveney, *Advanced Theory and Simulations* **2020**, *3*, 8 1900246, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adts.201900246.

[33] R. A. Richardson, D. W. Wright, W. Edeling, V. Jancauskas, J. Lakhlili, P. V. Coveney, *Journal of Open Research Software* **2020**, *8*, 1 11, number: 1 Publisher: Ubiquity Press.

## Table of Contents



Multiscale simulations methods have the potential to predict the properties of novel complex materials. Properties could be predicted from the chemical structure and used by engineers from the aeronautical or automotive industry. Hence avoiding expensive and extensive certification. However, multiscale simulations remain nowadays untractable, even for the largest supercomputers. Our novel reduction algorithm detects redundancies within multiscale simulations and enables substantial acceleration.