Three-dimensional cellular automata modelling of cleavage propagation across crystal boundaries in polycrystalline microstructures

A. Shterenlikht¹ and L. Margetts²,³

¹Mechanical Engineering Department, The University of Bristol, Bristol BS8 1TR, UK
²Directorate of IT Services, The University of Manchester, Manchester, UK
³Oxford e-Research Centre, The University of Oxford, Oxford, UK

A three-dimensional cellular automata (CA) with rectilinear layout is used in this work to create and cleave polycrystalline microstructures. Each crystal is defined by a unique randomly generated orientation tensor. Separate states for grains, grain boundaries, crack flanks and crack fronts are created. Algorithms for progressive cleavage propagation through crystals and across grain boundaries are detailed. The mesh independent cleavage criterion includes the critical cleavage stress and the length scale. Resolution of an arbitrary crystallographic plane within a 26-cell Moore neighbourhood is considered. The model is implemented in Fortran 2008 coarrays. The model gives realistic predictions of grain size and mis-orientation distributions, grain boundary topology and crack geometry. Finally, we show how the proposed CA model can be linked to a finite-element model to produce a multi-scale fracture framework.

1. Introduction

Cellular automata (CA) modelling of physical systems is a well-established field [1–3]. CA is a discrete time–discrete space framework. The model space is partitioned into identical cells with a finite number of states. A state of a cell at the next time increment is determined by
the state of this cell and the states of some neighbourhood at the previous time increment. Fixed or self-similar model boundaries can be used. This simple framework gives rise to a surprisingly rich range of behaviours, some of which are suitable for simulating physical processes such as lattice gas diffusion, phase transitions, wave propagation and multi-phase fluids [2]. In solid mechanics, it is popular to superimpose continuum fields, such as temperature, stress or strain, over the CA space. A combination of CA and finite elements, sometimes referred to as CAFE or CA-FE, has been used successfully for predicting ductile to brittle transitional fracture [4], oxide cracking in hot rolling [5], grain instability [6], solidification [7,8], friction stir welding [9], recrystallization [10–13] and dynamic strain induced transformation [14].

The vast majority of CA models explored over the years are two-dimensional. However, there are physical processes which cannot be accurately represented by two-dimensional models. Polycrystalline fracture is one example. Specifically, transgranular cleavage propagation across grain boundaries cannot be modelled adequately by a two-dimensional model, because grain boundary accommodation failure, due to mis-orientation of preferred cleavage planes in the neighbouring grains, cannot be taken into account [15–17].

It is important to highlight the major difference between the CA approach and Voronoi tessellation [18,19]. The crystals produced by the Voronoi method have easy geometrical description: faces, edges and vertices. By contrast, a CA-produced crystal is just a collection of connected cells. A CA grain has neither faces nor edges or vertices. This might appear to be a disadvantage. However, one must remember that Voronoi polyhedra is an idealization of crystal shapes in real polycrystalline materials. Indeed, one might argue that ‘blobs’ of irregular shape, produced by the CA approach, are closer to nature, as seen through the microscope, than nicely defined Voronoi polyhedra. However, the major advantage of the CA approach over the Voronoi tessellation is in the ability of the CA framework to model grain competition, recrystallization, grain boundary migration and other phenomena resulting in the evolution of the microstructure. This is easily achieved in the CA model, precisely because the crystals are not defined by the geometric means. Evolution of microstructure will require a lot more work if is to be implemented via the Voronoi tessellation approach. However, Voronoi tessellation can be useful for setting the initial cellular morphology [10].

This paper is concerned with the design of a three-dimensional CA model, feature rich and flexible enough to represent a wide range of polycrystalline microstructures and transgranular cleavage propagation in those.

The following notation is used in this work. Tensors of rank 2 are shown in bold: \( \mathbf{R} \). Vectors and scalars are in the upright math type: \( x \). Cell states are sans serif: \( c \).

## 2. The cellular automata model

A three-dimensional CA space with cubic cells and 26-cell Moore neighbourhood is created. The CA model is a rectilinear brick with \( d_1, d_2 \) and \( d_3 \) cells along dimensions 1, 2 and 3, respectively. The total number of cells in the model is \( D = d_1 \times d_2 \times d_3 \).

First, a polycrystalline grain microstructure is created by a simple solidification process in the following way. All cells are initially considered to be of liquid state, \( c_l = 0 \). \( N \) randomly chosen cells represent grain nuclei. These are assigned states \( c_G \in [1 \ldots N] \). Each grain (single crystal) is assigned a randomly chosen orientation tensor, \( \mathbf{R}^c \). For each iteration of the solidification process a liquid cell can acquire the state of one of the 26 randomly chosen neighbours. This is shown schematically in figure 1. This process is continuing until there are no liquid cells left in the model. Both fixed and self-similar boundary conditions can be used [6].

The model described above will produce equiaxed microstructure. By changing the initial distribution of grain nuclei, other popular distributions can be easily achieved, e.g. bimodal or columnar [6]. Our prior work suggested that, in order to achieve results independent of grain resolution, the CA model must be created with a sufficient resolution, \( N/D < 10^{-5} \), i.e. more than \( 10^5 \) cells must be used, on average, to represent each grain [6].
Figure 1. (a) A two-dimensional slice of the three-dimensional 26-cell Moore neighbourhood, showing a central liquid cell, \( c = 0 \), acquiring the state of one randomly chosen neighbour, indicated by an arrow, grain number 1, \( c = 1 \), in this example; (b,c) a two-dimensional illustration of a single iteration of three-dimensional solidification. Six liquid cells changed state to solid in this increment; three cells attached to grain 1, and the other three attached to grain 24. The arrows show the copying of cells states. (Online version in colour.)

A CA method itself has no concept of length or time scales. These scales are assigned to CA by the user, based on the exact physical process of interest. In relation to this work, this means that a polycrystal structure produced by the above algorithm can represent micro- or nano-crystal structures. One major difference between nano- and micro-crystalline materials is that in nano-materials, the volume occupied by grain boundary regions becomes comparable to that occupied by grain interiors. The fraction of grain boundary volume could be as high as 50% or even more [20,21]. This result emerges naturally from this CA model. If one considers all cells having neighbours belonging to other grains, as grain boundary cells, then their number (volume) will increase dramatically with decreasing mean grain size.

3. The quasi-cleavage algorithm

We use tensor, index-free, notation. All sub- and superscripts are not to be confused with lower and upper indices of index notation.

Orientation of each crystal with respect to the spatial (cellular) coordinate system (CS) is given by the rotation tensor, \( R^c \), with the usual meaning: a vector in the crystal CS, \( x^c \), is transformed into a vector in the spatial CS, \( x^s \), as \( x^s = R^c \cdot x^c \).

The stress tensor in the spatial CS, \( t^s \), is transformed into the stress tensor in the crystal CS as \( t^c = R^s \cdot t^s \cdot R^{-s} \), where \( R^s \equiv R^{-c} \equiv (R^c)^{-1} \equiv (R^c)^T \).

It is assumed that cleavage is controlled by the normal stress on a crystallographic plane, \( t^n \). For a plane \( (hkl) \), with normal \( n_{hkl} \), the normal stress is \( t_{hkl} = n_{hkl} \cdot t^n \cdot n_{hkl} \).

Each crystallographic plane is assumed to have a particular surface energy, \( \gamma_{hkl} \). We postulate that the work of cleavage is equal to the surface energy. It is further assumed that the work of cleavage is \( t_{hkl} \) times the distance necessary to break the atomic bonds. Following Gilman [22], this distance is taken equal to \( a_0 \), the relaxation distance, which is the atom diameter in the cleavage plane. The cleavage condition is \( t_{hkl}a_0 = \gamma_{hkl} \), from which the stress required to cleave the \( (hkl) \) plane can be calculated

\[
t_{hkl} = \frac{\gamma_{hkl}}{a_0}.
\] (3.1)

Example values for iron, from [22], are \( \gamma_{100} = 1440 \), \( \gamma_{110} = 1710 \) and \( \gamma_{111} = 5340 \text{ erg cm}^{-2} \) (1erg cm\(^{-2}\) = \( 10^{-7} \) J m\(^{-2}\)) and \( a_0 = 1.37 \times 10^{-10} \) m. This gives \( t_{100} = 1.05 \times 10^4 \) MPa, \( t_{110} = 1.25 \times 10^4 \) MPa and \( t_{111} = 4.90 \times 10^4 \) MPa.
In a CA model, these values must be scaled down, because the CA model is not applicable at the atomic scale but only at some intermediate, meso-scale, and because material imperfections and micro-plasticity elevate the stresses locally around the crack fronts [23]. Micro-plasticity has not yet been implemented in this model. Hence, in strict terms, the results are valid for some simplified polycrystalline material. This is the reason we use the term quasi-cleavage instead of simply cleavage.

When the cleavage condition of equation (3.1) is satisfied, the model crack advances for a characteristic length, which must be taken smaller than the corresponding characteristic length in the model, e.g. the mean grain size. Together, the cleavage criterion and the characteristic cleavage length form a cleavage model independent of the CA resolution.

More complex cleavage criteria have been proposed in the literature. These are based on detailed molecular dynamics and quantum mechanics analysis of inter-atomic bond potential, and formulation of cohesive zone type bond breaking models [24–26].

In bcc crystals, there are 24 symmetric rotation tensors, \( R_{sym}^{1,24} \), including the identity tensor. If \( n_{hkl} \) is a unit normal vector to some \( \{hkl\} \) plane, then \( n_{hkl}^{1,24} = R_{sym}^{1,24} \cdot n_{hkl} \) are 24 normal vectors describing planes of the same class. The weakest plane is that which maximizes \( t_{hkl} \):

\[
t^\text{max}_{hkl} = \max_{i=1...24} (n_{hkl}^i \cdot \hat{t}),
\]

where the only planes under consideration are \( \{100\}, \{1\bar{1}0\} \) and \( \{111\} \), although the surface energy of \( \{111\} \) planes is so high that it is practically impossible to cleave those. The normals to the planes of the maximum normal stress are \( n_{100}^{\text{max}}, n_{110}^{\text{max}} \) and \( n_{111}^{\text{max}} \). From equation (3.1), cleavage will occur when \( t^\text{max}_{hkl} \geq \gamma_{hkl}/\sigma_0 \), or when \( t^\text{max}_{hkl} = t^\text{max}_{hkl} / (\gamma_{hkl}/\sigma_0) \geq 1 \).

The first step is to find all \( t^\text{max}_{hkl} \) and the orientations of the corresponding planes, algorithm 1. The second step is choosing the weakest plane and setting the cleavage cell state accordingly, algorithm 2. The outputs are the unit vector, \( \mathbf{n}^c \), normal to the active cleavage plane, in the spatial CS, and the cleavage cell state, \( s \). Vector \( \mathbf{n}^c \) is first calculated in the crystal CS and then rotated to the spatial CS, \( \mathbf{n}^s \). Cell states \( c_{100}, c_{110}, c_{111} \) represent cleavage crack edges on \( \{100\}, \{110\}, \{111\} \) planes. The flag is true if cleavage condition is met, and false otherwise.

### Algorithm 1: Cleavage algorithm, calculating maximum normal stresses and their planes.

| input: \( \hat{t}, R, \gamma_{100}, \gamma_{110}, \gamma_{111}, \sigma_0 \) |
| output: \( p_{100}, p_{110}, p_{111}, p^\text{max}_{100}, p^\text{max}_{110}, p^\text{max}_{111} \) |
| \( \mathbf{t} = R^{-1} \cdot \hat{t} \cdot R \) |
| \( t^\text{max}_{100} = \max_{i=1...24} (n_{100}^i \cdot \hat{t}) \rightarrow n_{100}^\text{max} \) |
| \( t^\text{max}_{110} = \max_{i=1...24} (n_{110}^i \cdot \hat{t}) \rightarrow n_{110}^\text{max} \) |
| \( t^\text{max}_{111} = \max_{i=1...24} (n_{111}^i \cdot \hat{t}) \rightarrow n_{111}^\text{max} \) |
| \( p_{100} = t^\text{max}_{100} / (\gamma_{100}/\sigma_0) \); \( p_{110} = t^\text{max}_{110} / (\gamma_{110}/\sigma_0) \); \( p_{111} = t^\text{max}_{111} / (\gamma_{111}/\sigma_0) \) |
| \( p^\text{max} = \max(p_{100}, p_{110}, p_{111}) \) |

### 4. Cleavage representation in the cellular model

In the fracture cellular array, cells are initially intact. A number of crack nuclei, i.e. cells with cleavage crack edge states, \( c_{100} \) or \( c_{110} \), are positioned within the model. For example, the crack nuclei can be scattered at random, representing pre-existing micro- or nano-cracks in the material. In this manner, growth and/or interaction of a single or multiple cracks can be modelled.

Next, we scan over all intact cells. If an intact cell has a cleaved neighbour, such that the vector connecting the cleaved and the intact cells, \( \mathbf{e} \), is on or near the cleavage plane, then the state of the central cell is changed to the given cleavage state. Note that it is possible that the given cleavage state and the neighbour cleavage state will differ. The key decision in this approach is choosing a suitable threshold, \( t \), for deciding when \( \mathbf{e} \) belongs to the cleavage plane, defined by \( \mathbf{n}^s \). This
Algorithm 2: Cleavage algorithm, calculating the cleavage plane.

**input**: \( p_{100}, p_{110}, p_{111}, p_{\text{max}}, n_{\text{max}}^{100}, n_{\text{max}}^{110}, n_{\text{max}}^{111} \)

**output**: \( n^s, s, \text{flag} \)

\( n^s = 0; s = 0; \text{flag} = \text{false} \)

if \( p_{\text{max}} \geq 1 \) then

  cleavage on \{110\} plane \( \rightarrow \) \( \text{flag} = \text{true} \);
  \( n^c = n_{\text{max}}^{100}; s = \zeta_{100} \)

  if \( p_{110} > p_{100} \) then
    cleavage on \{110\} plane \( \rightarrow \) \( n^c = n_{\text{max}}^{110}; s = \zeta_{110} \)

  if \( p_{111} > p_{100} \) and \( p_{111} > p_{110} \) then
    cleavage on \{111\} plane \( \rightarrow \) \( n^c = n_{\text{max}}^{111}; s = \zeta_{111} \)

\( n^s = R^c \cdot n^c \)

The problem is analysed in appendix A. Assuming that such threshold can be chosen, the algorithm is summarized in algorithm 3.

Algorithm 3: Cleavage algorithm, propagating cleavage crack through the cellular model.

**input**: \( n^s, s, t \)

**output**: cell state change

for all cells do

  pick cell \( i \)

  for all 26 neighbours of \( i \) do

    pick neighbour \( j \)

    if cell \( j \) cleaved then

      \( e \) is a unit vector connecting cells \( i \) and \( j \)

      if \( e \cdot n^s < t \) then

        cell \( i \) state is changed to \( s \)

        exit

The cleavage criterion can be easily changed from a fully deterministic to a probabilistic, e.g. if \( e \cdot n^s < t \), then there is a probability that the state of cell \( i \) is changing to \( c \). This probability will be inversely related to \( e \cdot n^s \).

Algorithm 3 changes states only of the neighbouring cells. Thus, the speed of cleavage propagation in this algorithm is one cell per increment. With the use of the characteristic length scale, any crack propagation speed is achievable.

Algorithms 1, 2 and 3 are combined to simulate cleavage propagation across the whole cellular grain array \( G \). The algorithm grows cracks in a similar way to the grain growth algorithm: any intact cell of the fracture array \( F \) is allowed to join a cleavage crack if the following three conditions are met: (i) it has a neighbouring crack front cell, (ii) it lies on the cleavage plane, and (iii) the resolved stress is high enough. If \( t^c \) is changing very slowly, compared with the cleavage propagation speed, algorithms 1 and 2 need to be run only when the grain boundary is crossed, i.e. when the current cell, \( g_i \), in the grain array, \( G \), differs from the state of the neighbour, \( g_{\text{old}} \). The resulting algorithm is shown in algorithm 4. However, algorithm 4 does not take into account the full complexity of cleavage propagation across a grain boundary in three dimensions (e.g. [15,27,28]).

5. Crossing a grain boundary

The influence range in a CA model is one cell size. In this respect, CA is closer to a molecular dynamics approach, albeit on larger spatial and time scales, where the energy potentials are
Algorithm 4: Complete cleavage algorithm, top level view.

**input**: grain cellular array $G$, fracture cellular array $F$, $R^c$ array, $t^n$

**output**: possible cell state change to cleaved

$g_{old} = 0$

**for** all cells in $F$ **do**

pick cell $i$; read its grain number $g_i$; read its fracture state $f_i$

if $f_i = intact$ then

if $g_i \neq g_{old}$ then

run Algs. 1, 2 $\rightarrow n^s, s, flag$

$g_{old} = g_i$

if flag = true then

run Alg. 3 $\rightarrow$ change $f_i$ to $s$

invariably of a very close range, than to a weak formulation of continuum mechanics, where some form of global equilibrium is usually maintained. Creating and maintaining global entities, such as geometrical planes, edges or vertices, is very computationally expensive in CA formulation. In CA, a grain boundary is simply a cluster of cells of identical state $g_i$, each of which has a neighbour of a different type, $g_j$. Although it would be possible, in principle, to fit a plane over this cluster, e.g. via a linear minimization, this is not done in this work due to high computational costs. A grain boundary edge is a cluster of cells of identical state, each of which has neighbours of at least two different states.

Analysis of grain boundary fracture typically involves quantities such as crystallographic types of grain boundaries and grain boundary plane orientation. These quantities are not available in this CA formulation. Hence, simulating cleavage propagation across a grain boundary involves some extra considerations, compared with approaches where grains are modelled as polyhedra (e.g. [15]). In such geometrical (global) models, the process is simple: as soon as a cleavage crack reaches a grain boundary at some spatial point, a cleavage plane in the following, adjacent, grain is fully determined, thus allowing for the analysis of the fracture of the boundary fragment defined by the grain boundary plane and by the two cleavage planes in both grains.

By contrast, in a cellular (local) model, there is no global cleavage plane defined in a grain. Hence, each crack front cell at the grain boundary has a chance of starting a new cleavage crack in the adjacent grain. If left unchecked, this process quickly leads to a proliferation of cleavage cracks on multiple parallel crystallographic planes in the next grain. This situation is shown schematically in figure 2. Such a model is, of course, not physical. It must not be confused with river patterns which are sometimes seen in cleavage fracture surfaces (see e.g. [29] and references thereof).

Note that the geometrical model is not physical either. It simply looks at the final result of the cleavage propagation and tries to reproduce it. It is doubtful that the physical reality of cleavage propagation across the grain boundary is close to the global geometrical view.

To prevent the non-physical cleavage crack proliferation scenario illustrated in figure 2, only the first cleavage crack cell that touches the grain boundary is allowed to start a new cleavage crack in the adjacent grain. The grain boundary is marked as failed immediately, and after that no other cleavage crack is permitted to cross this grain boundary. To this end, the grain boundary connectivity array is created at the beginning of the cleavage simulation. The array contains an entry for each grain boundary, which is *inact* initially, and is updated to *failed* when crossed by a cleavage crack. A failed grain boundary cannot be crossed by another cleavage crack.

Another problem is that of simulating the accommodation failure of the grain boundary, to allow for a complete separation of the parts of the model. The region of accommodation failure is shown hatched in figure 3. Again, the local nature of the CA approach makes this
Figure 2. Schematic of cleavage crack proliferation at grain boundary crossing due to using only the local neighbourhood CA analysis. The arrows show the predominant cleavage propagation direction, from right (grain 1) to left (into grain 2). The dashed line represents the intersection of the cleavage crack in grain 1 with the grain boundary. Note that a single cleavage crack in grain 1 can initiate cleavage cracks in grain 2 at any point where it crosses the grain boundary.

Figure 3. Schematic of the accommodation failure of the grain boundary to allow a complete separation of the model due to a running cleavage crack. The region of the boundary that has to fail to allow for a complete separation is shown hatched. This is a conventional interpretation of grain boundary accommodations failure (e.g. [16]).

hard, because the knowledge of global geometrical quantities is required, specifically the relative spatial orientation of the regions of the grain boundary with respect to the cracks on both sides of the boundary. In the absence of this information, when a grain boundary cell is analysed, the immediate neighbourhood information is insufficient to decide whether the cell should fail or not.

6. Results

The model was implemented in modern Fortran 2008 [30]. The code is available under two-clause BSD licence from http://sourceforge.net/projects/cgpack/. The code uses coarrays for portability and performance. It is designed to be highly scalable so that it can be used on high-performance computers. The code has shown good scaling up to 32 000 processors on HECToR, the UK national supercomputer [31].
**Figure 4.** Two examples of simulated equiaxed polycrystalline microstructures, showing a cubic volume populated with (a) 40 grains and (b) 40,960 grains. The scale is arbitrary. Grey scale denotes a unique grain number that is linked with a randomly assigned rotation tensor, \( \mathbf{R} \). Hence, if the spatial volumes in (a,b) are considered identical, then the mean grain size in (b) is smaller than in (a). Alternatively, if the mean grain size is considered identical in (a,b), then example (b) represents a much bigger spatial volume of material than (a). (Online version in colour.)

**Figure 5.** Grain size histograms from simulated equiaxed microstructures, showing (a) data from the model with 5120 grains and (b) data from the model with 40,960 grains, shown in figure 4b. Note the increase in the relative size of the biggest grain with increasing the number of grains in the model. (Online version in colour.)

**Figure 4** shows two examples of simulated equiaxed microstructures. In both cases, a cubic volume was modelled. In the first case, it was populated with 40 grains, and in the second example 40,960 grains were grown.

**Figure 5** shows two predicted grain size histograms, with 5120 grains, figure 5a and with 40,960 grains, figure 5b. The value of these data is that they allow for a direct comparison with the experimental measurement. Another expected, but still very important, result is that the maximum grain size in the model is increasing with the model size. For the model with 5120 grains, the biggest grain is about 3.5 times bigger than the mean, figure 5a, while for the model with 40,960 grains, the biggest grain is four times the mean size, figure 5b. This observation is important because cleavage is often thought of as the weakest link model, and the largest grains...
Figure 6. Useful predictions from a 640-grain equiaxed microstructure model, showing (a) the histogram of predicted grain boundary mis-orientations in an equiaxed microstructure. The shape, the peak (45°) and the maximum angle (62.45°) match the theoretical predictions given, for example, in [33]; and (b) the histogram of the number of neighbouring grains, which roughly follows the normal distribution, with the mean around 15 neighbours. Note that this prediction is not easy to validate experimentally because most, if not all, experimental measurements of microstructures are done on two-dimensional slices, and three-dimensional information can be extrapolated only using some additional assumptions.

will have the lowest toughness [32]. Hence, the bigger the model, the higher the chances of representing an extremely low fracture energy event.

Figure 6 shows two important predictions obtained with a 640 grain model of an equiaxed microstructure with periodic (self-similar) boundary conditions. Figure 6a shows the predicted grain boundary mis-orientation distribution. The model prediction closely matches the theoretical results of Mackenzie [33]. The maximum calculated value is 62.5°, whereas the theoretical maximum is 62.8°. Figure 6b shows the predicted distribution of the number of neighbouring grains. The mean value is around 15 neighbours, whereas in three-dimensional geometric models, with the popular 14-hedra grain shape (the Kelvin polyhedron), each grain has 14 neighbours sharing a face [15,34]. This similarity is all the more striking given that a grain in a cellular model has no defined faces, edges or vertices. This comparison adds validity to the three-dimensional CA grain modelling results.

Figure 7 shows an example of a model with two grains, with fixed boundary conditions, in which a single cleavage micro-crack, i.e. a cleavage nuclei cell, was assigned a random location in grain 1. The results show the state of the model after 70 cleavage iterations. Both cracks are on {100} planes. Discretization of a randomly oriented plane within a 26-cell neighbourhood is very coarse (appendix A). This means that two planes of a different orientation, passing through the same cell, might be discretized identically, unless their orientations are substantially different. In some cases, the mis-orientation angle must be 45° for the planes to be discretized differently in a cellular model. Appendix A gives more details.

The grain boundary, and the intersections of both cracks with the grain boundary are irregular, i.e. hard to describe by global geometrical parameters (figure 7). We note that there is a school of thought that insists that fractal geometry, rather than differential geometry, is the only meaningful approach when dealing with polycrystalline fracture surfaces [35–39]. Also note that the grain boundary area between the two intersections with the cleavage cracks in figure 7b would ultimately fail by some other mechanism, possibly ductile shear [15,16,22,27–29]. This modelling result is consistent with the theoretical framework (see figure 3 and [15,16]).

Figure 8 shows an example of non-physical one-dimensional cracks. This is an artefact of allowing only two neighbouring cells to be associated with the cleavage plane passing through...
Figure 7. A model with two randomly oriented grains and a single cleavage micro-crack at a randomly chosen location in grain 1, showing (a) two \{100\} cleavage cracks in both grains, with grain cells removed for clarity, and (b) the same superimposed with grain 1. Note an irregular grain boundary and irregular intersections of both cracks with the grain boundary. (Online version in colour.)

Figure 8. An example of one-dimensional cracks produced when only two neighbouring cells are recognized as lying on the cleavage plane. This is a rare but possible modelling artefact when the threshold \(t = 0.1733\). See §4 and appendix A for more details. This fixed boundary model included two grains. (a) The \{100\} crack is planar in grain 1, but the cracks in grain 2 are discretized as one-dimensional linear cracks. The grains are not shown for clarity. (b) Grain 1 and the grain boundary are superimposed on the cracks. Note the irregularity of the grain boundary. (Online version in colour.)

As shown in appendix A, if \(t = 0.1733\), then for any \(n^s\), there will always be at least two neighbouring cells on the cleavage plane. As the orientation of these two cells with respect to the central cell is always the same, as long as \(n^s\) is constant, the only possible resulting crack shape is one-dimensional straight lines. An intuitive solution to this problem is to increase \(t\) ‘a little bit’, to make sure there are always at least four neighbouring cells on the cleavage plane. However, in this case, there are such \(n^s\) directions, for which the neighbouring cells considered to lie on the cleavage plane are not on the same geometrical plane. Algorithm 3 then results in three-dimensional cracks, which are equally non-physical (figure 9).

Figure 10 shows the grain boundary accommodation fracture for the two \{110\} cleavage cracks. As described in §4, at present, we have no good physically sound local criteria to decide which parts of the grain boundary should fail to achieve accommodation for the two cleavage cracks on either side of the boundary. Therefore, the area of accommodation fracture simply grows outward...
from the crack intersections with the boundary. The end product of this process is that the whole of the grain boundary is fractured, which is not physical.

Putting all the bits of the model together, figure 11 shows cleavage propagation through a model with 20 grains, after 300 cleavage iterations. Note that a significant fraction of grains exhibit multiple cracks on parallel planes. As the model explicitly forbids more than one crack to propagate across any grain boundary, this means that cracks propagate into these grains from different boundaries. For example, refer to figure 11c. A grain in the front bottom corner of the model has three cleavage cracks. It is likely that one of these cracks propagated into this grain from the grain to its top left, and the other two cracks propagated from the two grains to its right. Note that this example did not include the feedback from the microstructure to the structural model, i.e. there is no reduction of \( t \) in cracked grains. In practice, a cracked grain would receive a dramatically reduced \( t \), making further cracks on the same crystallographic planes very unlikely.

### 7. Combining with finite-elements for a multi-scale model

A complete CAFE (CA finite-element) multi-scale model was constructed by linking the CA model with the ParaFEM finite-element library [40,41]. Conventional localization and homogenization strategies [18,26,42] can then be used to exchange the information between the FE and the CA parts of the model. In this example, the FE stress tensor was redistributed over the CA according to the proximity of the cells to crack flanks, to replicate the stress intensity dominated stress fields. Cells closest to the crack flanks receive much higher stress tensors than the FE value. Cells sufficiently removed from crack flanks receive lower stress tensors than the FE value. The mean value of the CA stress tensors is equal to the FE stress tensor. The homogenization is done via scaling of the FE Young’s modulus according to the size of the cleavage crack. A representative example is given below.

A 10 \( \times \) 10 \( \times \) 10 mm\(^3\) cube of material is considered. The material model is linear elastic, with Young’s modulus of 200 GPa. The boundary conditions are \( u_1(x_1 = 0) = 0, u_2(x_2 = 0) = 0 \) and \( u_3(x_3 = 0) = 0 \). Distributed loading is applied along \( x_3 \) on a small element of surface \( x_3 = 10 \), with the total applied force of 1 kN. The mean grain diameter is taken as high as 1 mm, primarily to ease the visualization, because large grains result in large transgranular cracks. The complete model
Figure 10. A model with two grains, showing (a) the two \{110\} cleavage cracks in two grains. Neither the grains nor the grain boundary are shown for clarity; (b) same view as in (a) but with the fractured part of the grain boundary is also shown, and (c) a different view of the fractured grain boundary. Note that the fractured grain boundary area simply grows away from the intersection locations with each iteration. The end result of this process is that the whole of the grain boundary is fractured, which is not physically sound. (Online version in colour.)

thus implements 1000 randomly oriented grains. A single micro-crack is introduced at $x_1 = x_2 = 0$, $x_3 = 5$ mm. The results in figure 12 show the deformed finite-element mesh and the crack patterns produced in three runs of the model.

8. Concluding remarks and future work

The three-dimensional CA framework, designed and implemented as described, has been shown to be capable of simulating progressive quasi-cleavage propagation though a polycrystal. The model is based on the fracture stress criterion and is made mesh independent by including a length scale. The model takes into account random orientations of individual crystals. The immediate next step is to add orientation-dependent grain boundary energies [43,44]. Energies of the cleavage cracks might at first be taken equal to free surface energies, $\gamma_{hkl}$. Then, the total energies of fracture can be calculated simply by summing the number of cells of cleavage states, $c_{hkl}$, and of grain boundary fracture states, $c_{gb}$, multiplied by their energies. In this respect, one- and three-dimensional crack artefacts present a major obstacle. One-dimensional line crack
A cleavage propagation example with 20 grains, showing (a) grains and grain boundaries on the surface of the model, (b) three-dimensional view of grain boundaries in the interior of the model, (c) cleavage cracks superimposed on grain boundaries and (d) only cleavage cracks, with all other structures removed for clarity. (Online version in colour.)

Artefacts will result in artificially low energy, and three-dimensional solid body crack artefacts will dramatically increase the fracture energy. Hence, it is necessary to resolve the issue on non-physical one- and three-dimensional cracks if realistic fracture energy prediction is aimed for.

Data accessibility. All data are accessible from http://sourceforge.net/projects/cgpack/.

Acknowledgements. This work used the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk). This work also used HECToR, the UK’s national high-performance computing service, provided by UoE HPCx Ltd at the University of Edinburgh, Cray Inc and NAG Ltd. HECToR was funded by the Office of Science and Technology through EPSRC’s High End Computing Programme. This work also used the computational facilities of the Advanced Computing Research Centre, the University of Bristol (http://www.bris.ac.uk/acrc/) and the N8HPC Service, funded by the N8 consortium and EPSRC EP/K000225/1, coordinated by the Universities of Leeds and Manchester (http://n8hpc.org.uk).

Author contributions. A.S. contributed to all sections. L.M. contributed to the creation of the CAFE model, §7. Both authors gave final approval for publication.

Conflict of interests. We have no competing interests.

Appendix A. Resolving arbitrary plane within a three-dimensional cellular automata cell neighbourhood

In three-dimensional CA, a cell has 26 nearest neighbours: six share a common face, 12 share a common edge and eight share a common corner (vertex) (figure 13). This three-dimensional space
Figure 12. Three runs of a CAFE model in a steel-like material with large grains, showing (a) the deformed finite-element model with the distortion artificially magnified for clarity, (b–d) clusters of cracks on {100} planes (light grey or yellow (online)) and on {110} planes (dark grey or green (online)) forming a large main crack that is macroscopically normal to the maximum principal stress direction in that region. Each particular crack scenario is unique. Analysis of a statistically significant number of runs of this CAFE model can lead to the prediction of scatter in cleavage crack propagation results, such as the fracture energy. (Online version in colour.)

A Cartesian CS, with arbitrary origin, is aligned with cell axes. In single or poly-crystal models, a crystal is represented by a cluster of cells. A crystal can have arbitrary orientation with respect to the global (cellular) CS. By contrast, the cell neighbourhood can be discretized only into 26 directions, one for each neighbouring cell. Thus, a problem arises when crystallographic directions or planes have to be resolved within discrete cellular models.

A problem of particular importance is transgranular cleavage, where material is separated along a crystallographic plane, e.g. {100} or {110} plane in bcc crystals. To simulate cleavage in a cellular model, one must answer this question for every cell: which neighbouring cells are on the cleavage plane?

Let us illustrate the problem on a two-dimensional example (figure 14). Let \( \mathbf{n} \) be the unit vector normal to some cleavage plane, and \( \mathbf{e}_i, i = 1 \ldots 8 \), is a unit vector connecting a cell with
Figure 13. This $3 \times 3 \times 3$ cluster of 27 cubic cells represents the 26-cell nearest neighbourhood of the central cell. Six neighbours share a face ($'$), 12 neighbours share an edge ($'e'$) and eight neighbours share a corner ($'c'$). A Cartesian CS is a natural choice for this model.

Figure 14. Two-dimensional illustration of the problem of deciding which neighbouring cells are on the cleavage plane (cp, defined by its normal, $n$). When $n$ is aligned with one of the cell direction vectors, $e_i$, the answer is clear because the cleavage plane passes directly through the centres of two neighbouring cells ($'F'$ for failed), as in cases (a,b). In general, no $e_i$ vectors are normal to $n$, and it is unclear which, if any, cells (marked '?') should be considered to lie on the cleavage plane. Possible answers are zero, two or four.

One of its eight neighbours. If $n \cdot e_i = 0$, for any $i$ then the question of which neighbouring cells are on the cleavage plane is trivial (figure 14a,b). In the general case, the answer to this question is not clear. To help us answer it in the general case, we pose this problem: find $n$ that would maximize the minimum angle between the cleavage plane and each of $e_i$. This problem is solved by finding $\max_n(\min_{i=1...8} |n \cdot e_i|)$, which we denote $z$. Looking at figure 14c, this ‘MaxMin’ criterion corresponds to the cleavage plane at equal angles to two adjacent cell direction vectors. Therefore, the angle is $\theta = \pi/8 = 22.5^\circ$, $n = (\cos(\theta + m(\pi/4)), \sin(\theta + m(\pi/4)))$ for any integer $m$ and $z \approx 0.38$.

One can then construct a cleavage criterion based on $\theta$ or $z$. One possible criterion can be that a neighbouring cell $i$ is considered to lie on the cleavage plane if $|n \cdot e_i| \leq z$.

In the three-dimensional case, this analysis is more complex. If the centre of the central cell in figure 13 is given coordinates $(0,0,0)$, then the directions to centres of the neighbouring cells are given by the following 26 unit vectors, $e_i$, $i = 1...26$. There are six common face cells: $(\pm 1,0,0), (0,\pm 1,0), (0,0,\pm 1)$; 12 common edge cells: $(\pm 1/\sqrt{2},\pm 1/\sqrt{2},0), (0,\pm 1/\sqrt{2},\pm 1/\sqrt{2}), (\pm 1/\sqrt{2},0,\pm 1/\sqrt{2})$ and eight corner cells: $(\pm 1/\sqrt{3},\pm 1/\sqrt{3},\pm 1/\sqrt{3})$. Consider an arbitrary unit vector $n$. We formulate two geometric problems.
16

Figure 15. Schematic illustration of cell direction vectors, $e_i$, $i = 1 \ldots 26$. The central cell is at the origin. Unit vectors connect the central cell with the centres of its neighbours. Only vectors used in the solution of the problems are shown.

Problem 1: find $\max_n \left( \min_{i=1\ldots26} |n \cdot e_i| \right)$ for all possible $n$.
Problem 2: find $\min_n \left( \max_{i=1\ldots26} |n \cdot e_i| \right)$ for all possible $n$.

Although only problem 1 is related to the cleavage analysis, problem 2 is the opposite of problem 1, so we solve it as well.

(a) Problem 1

Figure 15 shows that the model has three planes of symmetry. It is therefore sufficient to choose $n$ only from one coordinate corner. For simplicity, we choose the corner with all coordinate axes positive. If $n = (n_1, n_2, n_3)$ solves problem 1, then so do the eight vectors $(\pm n_1, \pm n_2, \pm n_3)$. Furthermore, because the problem is symmetrical with any permutation of the coordinate axes, a further six vectors are obtained from each $n$ by permutation of its components. In total, there are 48 vectors $n$ which solve problem 1.

Refer to figure 15. The origin is at the central cell and the unit vectors show locations of the centres of some of the neighbouring cells. Owing to three symmetry planes, the space can be divided as left/right, top/bottom and front/back. In this terminology, $n$ is in the right top front corner. The problem is that of finding $n$ that makes the biggest angles (but $< \pi/2$) with three $e_i$ vectors. The first vector can be chosen arbitrarily. We choose $e_1 = (0, 0, 1)$. To maximize the angles between $n$, the other two vectors must lie in other corners. In the right bottom front corner, the vector making the biggest angle with $n$ is either $(0, 1/\sqrt{2}, -1/\sqrt{2})$ or $(1/\sqrt{2}, -1/\sqrt{2}, 0)$. We choose the first of these: $e_2 = (0, 1/\sqrt{2}, -1/\sqrt{2})$. This leaves the third vector from the left bottom front corner, $e_3 = (1/\sqrt{3}, -1/\sqrt{3}, -1/\sqrt{3})$. All other vectors in the front corners are either reflections of the chosen vectors about the symmetry planes, or make smaller angles with $n$, or angles $> \pi/2$ with $n$. The problem is then reduced to solving the following equations:

$$n \cdot e_1 = n \cdot e_2 \quad \text{and} \quad n \cdot e_2 = n \cdot e_3 \quad (A 1)$$

for $n_1$ and $n_2$; $n_3^2 = 1 - n_1^2 - n_2^2$. We omit the algebra and give the solution

$$n_1 = \rho n_2; \quad n_2 = \left( \frac{2(2 + \sqrt{2})}{3 + 2\sqrt{2}} + \rho^2 \right)^{-1/2}, \quad \text{where} \quad \rho = \frac{2 + \sqrt{3} + \sqrt{2}}{1 + \sqrt{2}}. \quad (A 2)$$

This gives $n \approx (0.8916, 0.4183, 0.1733)$ or any of its 47 equivalents constructed as discussed above. Solution to problem 1 is then

$$\max_n \left( \min_{i=1\ldots26} |n \cdot e_i| \right) \approx 0.1733. \quad (A 3)$$
(b) Problem 2

Again refer to figure 15. We are now looking for \( n \) such that it makes the biggest possible angles (but \(< \pi/2\)) with closest vectors \( e \). We choose \( e^1 = (1, 0, 0) \). Its two closest vectors in the same
corner are \((1/\sqrt{2}, 1/\sqrt{2}, 0)\) or \((1/\sqrt{2}, 0, 1/\sqrt{2})\). We choose the first of these: \( e^2 = (1/\sqrt{2}, 1/\sqrt{2}, 0) \).
The only choice for the third closest vector in the same corner is \( e^3 = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}) \). As in
problem 1, we then solve equations (A 1). The answer is
\[
n_1^2 = 9 - 2(\sqrt{6} + \sqrt{2}) \quad \text{and} \quad n_2 = n_1(\sqrt{2} - 1).
\] (A 4)

This gives \( n \approx (0.8865, 0.3672, 0.2817) \) or any of its 47 equivalents constructed as discussed
above. Solution to problem 2 is then
\[
\min_n \left( \max_{i=1\ldots26} |n \cdot e^i| \right) \approx 0.8865.
\] (A 5)

References


