Nested multiresolution hierarchical simulated annealing algorithm for porous media reconstruction

Laurent Lemmens,[1] Bart Rogiers,[1] Diederik Jacques,[1] Marijke Huysmans,[2] Rudy Swennen,[2] Janos L. Urai,[3] Guillaume Desbois,[3] and Eric Laloy[1]

[1]*Belgian Nuclear Research Center SCK•CEN, Mol, Belgium*

[2]*Departement of Earth and Environmental Sciences, KU Leuven, Leuven, Belgium*

[3]*Institute of Structural Geology, Tectonics and Geomechanics, RWTH Aachen University, Germany*

(Received 16 May 2019; published 25 November 2019)

Microstructure strongly influences flow and transport properties of porous media. Flow and transport simulations within porous media, therefore, requires accurate three-dimensional (3D) models of the pore and solid phase structure. To date, no imaging method can resolve all relevant heterogeneities from the nano- to the centimeter scale within complex heterogeneous materials such as clay, reservoir rocks (e.g., travertine, chalk, ...), hardened cement paste, and concrete. To reconstruct these porous materials it is thus necessary to merge information from different 2D and potentially 3D imaging methods. One porous media reconstruction methodology that has been around for at least two decades is simulated annealing (SA). However, realizations with SA typically suffer an artificially reduced long-range connectivity, while multiphase reconstructions are not feasible in most cases because of a prohibitive computational burden. To solve these problems we propose a hierarchical multiresolution and multiphase simulated annealing algorithm. To decrease the computational cost of multiphase simulation, our algorithm sequentially simulates one phase after another, in a hierarchical way, which enables handling multimodal distributions and topological relations. Building upon recent work, our algorithm improves long-range connectivity and CPU efficiency by simulating larger particles using a coarser resolution that is subsequently refined compared to standard SA; our proposed extension not only offers the possibility to perform multiphase reconstruction but also allows us (i) to improve binary reconstruction quality, as quantified, e.g., by multiple-point histograms by up to one order of magnitude and (ii) to achieve an overall speed-up. The proposed algorithm is also shown to outperform the direct sampling multiple-point statistics method for the generation of cement paste microstructure with respect to both generation time and quality.

DOI: 10.1103/PhysRevE.100.053316

I. INTRODUCTION

To develop a detailed process understanding of flow and transport in heterogeneous materials such as natural or engineered porous media like clay and concrete, an in-depth description of the pore and solid phase structure is needed [1–4]. This requires a high resolution three-dimensional (3D) representation of the materials. The first step in obtaining this information in a nondestructive way is the use of a 3D imaging technique [5]. The most popular 3D imaging method since the 1980s is computed tomography (CT) [6,7]. While CT and μCT provide information on the bigger structure of the material in three dimensions, they cannot resolve structures below half a μm. Hence, they have insufficient resolution to image the pores and their throats in low permeable materials such as clay [8]. To reveal these pore throats, it is necessary to use higher resolution imaging techniques such as scanning electron microscopy (SEM) [9], which has a resolution of a few nm. Another advantage of SEM imaging is that it also provides the capability to distinguish between different mineralogical phases through the use of EDS spectroscopy [9]. The shortcoming of SEM, however, is that it only offers 2D information and thus does not allow for a direct calculation of the 3D transport properties. This is especially problematic for anisotropic materials. Recently focused ion beam scanning electron microscopy (FIB-SEM) gained popularity for 3D imaging at the nanoscale [10]. Nevertheless, FIB-SEM has a small field of view with an edge length of a few μm, as well as its high cost of acquiring the images in terms of equipment and operating costs. In general, it can be stated that for clay and cement paste none of the existing imaging methods on its own can derive a sufficiently representative representation of the material and that a combination of the different imaging methods is required [8].

One popular alternative to obtain the 3D structure representation of a porous medium is stochastic reconstruction. Overall, it consists of processing the available information and to generate stochastic structure realizations that are in agreement with the derived medium characteristics. There are basically two types of reconstruction methodologies [3].

The first type relies on process-based modeling. This method attempts to mimic the formation of natural materials in their original geological environment, or the process of creating engineered materials. A commonly used process-based algorithm to reconstruct sandstones was, for example, introduced by Øren [11]. Methods for producing cement paste structures are HYMOSTRUC [12] and CEMHYD3D [13]. This class of methods is good at modeling the temporal evolution of the considered material during diagenesis or for example hardening. Diagenesis is thereby defined as the ensemble of material transformation processes after sedimentation and before final lithification. However, currently available process-based methods only consider certain aspects and have

 ©2019 American Physical Society

difficulties for describing the material structure accurately even in terms of plain visual inspection. An approach which tries to tackle those issues is the phase field method as described by Ankit *et al.* and Prajapati *et al.* [14,15].

The second type of reconstruction methodologies takes a more data-driven approach, and makes use of statistics [3]. Those statistics are mainly derived from segmented images obtained at various resolutions. This reconstruction methodology itself can be further subdivided into two subclasses.

The first subclass uses two-point statistics information such as the autocorrelation or probability [2], lineal path [16], and cluster functions [2] (see Sec. II A for more details). The most frequently used two-point statistics reconstruction methodology is simulated annealing (SA) [4,17–19]. As opposed to multiple point statistics (MPS) based approaches (see next paragraph) that can be quite complex and may require a lot of hyperparameter tuning, the simplicity of SA is conceptually appealing. Yet SA classically suffers from a large computational demand, which is caused by the pixel or voxel switching process (see Sec. II B 1 for details). This problem becomes more prominent when more than two phases are considered. It is therefore often not possible to do multiphase reconstructions with SA. Furthermore, standard SA also typically shows a decreased long-range connectivity [20]. To reduce computational burden and to improve long-range connectivity Pant *et al.* [20] recently proposed a multigrid approach. The underlying idea is to first generate a coarse representation and to refine the grid resolution afterward. Doing so, the larger clasts are generated faster and hence fewer iterations are needed. More details about the approach can be found in Sec. II B.

The second subclass relies on MPS simulations. MPS techniques were introduced at the beginning of the 1990s [21] and the field has been flourishing ever since. Nowadays, some widely used MPS algorithms are the single normal equation simulation algorithm (SNESIM) [22], the direct sampling method (DS) [23], and the cross-correlation simulation (CCSIM) [24]. In addition, MPS simulations using deep generative models have recently been proposed [25–29].

We propose a SA reconstruction algorithm, which performs nested, multiresolution, hierarchical reconstruction. The computational efficiency of our approach makes it possible to perform multiphase simulations. Besides decreasing computation time compared to standard SA, the proposed approach improves long-range connectivity of the realizations. Our algorithm extends the multigrid hierarchical simulated annealing approach by Pant *et al.* [20] to the multiphase case. We use the term multiresolution instead of multigrid because multigrid has a different meaning in the MPS field. In the MPS framework, multigrid means that a given realization is generated using a number n of increasingly finer grids, whereas the spatial distribution of the pixel remains constant. Each of the $i = 1, \ldots, n$ grids thus features $2i - 1$ pixels in the final simulation grid [22]. In the approach by Pant *et al.* [20] each pixel of a coarser grid is replaced by 2×2 pixels on the refined grid, hence we find the term "multiresolution" more appropriate.

The remainder of this paper is structured as follows: In Sec. II A, we describe the structural descriptors that are used to judge the reconstruction quality throughout the SA process, and the other descriptors used to validate our methodology. Section II B details the functioning of our proposed algorithm before its performance is compared against that of both standard SA and the DS code in Sec. III. This comparison includes two case studies involving hardened cement paste and natural clay together with 2D and 2D–3D reconstructions. In Sec. IV, we discuss the advantages and drawbacks of our approach before concluding with a summary of our most important findings in Sec. V.

II. METHODS

This section details all the building blocks of our approach.

A. Structural descriptors

To define the different descriptors, we follow the notation of [2], and use the characteristic function I_r^j:

$$I_r^j = \begin{cases} 1, & \text{if r } \epsilon \text{j} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The first structural descriptor used within our SA algorithm is the two-point probability function $S_2^j(\mathbf{r}_1, \mathbf{r}_2)$, short S_2, which gives the probability that two pixels at positions $\mathbf{r}_1$ and $\mathbf{r}_2$ belong to the same phase. $S_2^j(\mathbf{r}_1, \mathbf{r}_2)$ is defined as

$$S_2^j(\mathbf{r_1}, \mathbf{r_2}) = \langle I^j \mathbf{r_1} I^j \mathbf{r_2} \rangle. \tag{2}$$

The second descriptor considered by our SA algorithm is the lineal-path function, $L_2^j(\mathbf{r}_1, \mathbf{r}_2)$, in short, L_2. This function calculates the probability to find a line segment spanning from $\mathbf{r}_1$ to $\mathbf{r}_2$ that lies entirely in phase j [2].

Note that to perform a reconstruction, our SA algorithm solely uses S_2 and L_2.

To assess reconstruction quality, we use the two-point cluster function $C_2^j(\mathbf{r}_1, \mathbf{r}_2)$, in short, C_2. C_2 computes the probability to find two pixels at positions $\mathbf{r}_1$ and $\mathbf{r}_2$ that belong to the same cluster j [2]. In this context, a cluster is defined as the ensemble of pixels which belong to the same phase and are connected to each other. Note that in practice, two-point structural descriptors such as S_2, L_2, and C_2 are computed for each phase along spatial directions: typically along the x and y axes, and the main diagonals d_{xy} and d_{yx} for the 2D case, together with the z axes and main diagonals d_{xz}, d_{zx}, d_{yz}, and d_{zy} for the 3D case. Furthermore, S_2, L_2, and C_2 are evaluated in a function of the distance between pairs of pixels (cf. classic two-point variogram-based geostatistics [31]). For instance, for the two-point probability function along the x direction one calculates $S_{2,x}^j(\mathbf{h})$ where the $\mathbf{h}$ ("lag") vector contains the $i = 1, \ldots, n_x$ possible number of pixels separating two points along the x direction of the considered 2D (or 3D) image, and $S_{2,x}^j(h_i)$ is the average over all the $S_{2,x}^j(r_1^i, r_2^i)$ values belonging to h_i.

The three probability curves are shown for each phase of the the hardened cement paste image in Figs. 1(b)–1(e).

For the the quality assessment, we used three additional reconstruction quality criteria. The first one is a multiple-point histogram (MPH) [32] using a seven-point template (Fig. 2). To calculate the MPH, the frequency of all possible pattern combinations within the selected seven-point template is determined. The second is the particle or pore area

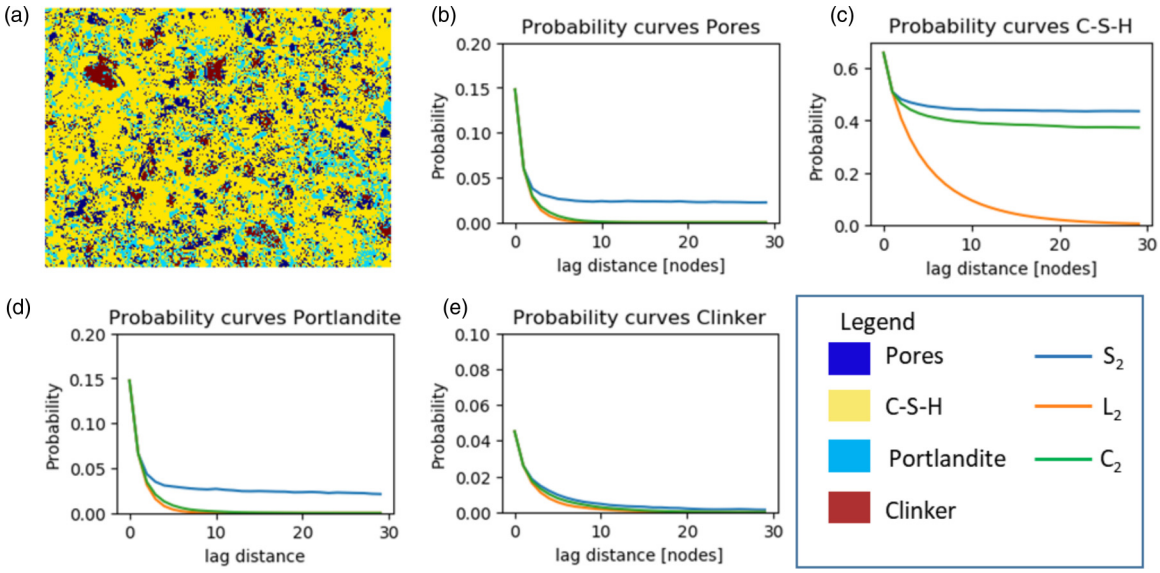

FIG. 1. Training image of cement paste from Phung *et al.* [30] with the probability curves of the different cases. The length of the TI is 192 × 256 pixels.

distribution function (PPA), calculated by counting the number of pixels within a single component. The third is the perimeter distribution function (P), calculated by counting the number of edges on the outer surface of a particle.

Because SA tries to optimize S_2 and L_2, to such a degree that they cannot be distinguished anymore from the training image (TI), the Euclidian distance is used to compare the different structural descriptor functions between the TI and the produced reconstruction. PPA and P require specific attention owing to their logarithmic behavior and the need to work with classes rather than with continuous values. The derived log-transformed continuous PPA and P values were therefore sorted into deciles from which the individual mean values were calculated. To account for the low probability events of observing either a very small or a very large particle or pore, the first and the last decile were not taken into account for the PPA and P calculations.

For the assessment of reconstruction quality for Boom clay in Sec. III B, we use an additional descriptor called the mean relative histogram difference (HIS). The descriptor calculates per subphase the absolute difference in the frequency for this subphase between a given realization and the TI normalized by the frequency of this phase in the TI. In this context a subphase is defined as a small subpart of the phase; more details are found in Sec. II B 4. This descriptor is only used for Boom Clay (see Sec. III B), as we resegmented the realizations after the simulation. Hence, only for this material, the HIS misfit will not be zero for SA.

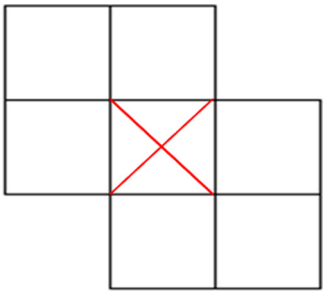

FIG. 2. Template used for the multiple point histogram. The cross, always marks the central cell element, which is put on top of each pixel in the structure that is analyzed, for the MPH calculation.

B. Reconstruction methodology

This section is subdivided into five different parts. The first part describes the basic principles of SA for the statistical reconstruction of a binary image using a single resolution. In the second part, the hierarchical approach is detailed. The third part details the multiresolution approach and is followed by a part about the nested approach of the algorithm. The final part details the full principle of the proposed algorithm.

1. Simulated annealing (SA)

The basic SA reconstruction scheme introduced by Yeong and Torquato [17] is given in Algorithm 1. The approach uses a simulated annealing process to minimize the energy E of a system. In the context of porous media reconstruction, E encodes the deviations of the generated image from the TI in one or more structural descriptor values. For a given two-point descriptor calculated for phase j along spatial direction s and vector distances, $\mathbf{h}$, $D_2^j(\mathbf{h})$, E is typically expressed as a sum of squared residuals (SSRs),

$$E = \sum_{i=1}^{n_s} \left[D_{2,s}^{j,TI}(\mathbf{h}_i) - D_{2,s}^{j,R}(\mathbf{h}_i) \right]^2. \tag{3}$$

In this formula n_s is the considered maximum number of pixels along a spatial direction s (e.g., x, y, z, d_{xy}, d_{xz}, or d_{yz}), the $i = 1, \ldots, n_s$ h_i values are the considered distances along s, TI signifies the training image and R denotes the considered realization generated by the SA.

Algorithm 1. Standard simulated annealing.

```
procedure SG_new = SA(TI, SG_old, λ, N_ite, N_sto, E_thr)
  ▷ SG_new ≡ Final representation of simulation grid
  ▷ TI ≡ Training image
  ▷ SG_old ≡ Previous representation of simulation grid
  ▷ N_ite ≡ Max number of iterations
  ▷ N_sto ≡ Max number of consecutively denied pixel swaps
  ▷ E_thr ≡ Energy threshold
  ▷ λ ≡ Cooling tuning factor
    if SG_new == empty then
        Initialize SG_new
    else
        SG_new= SG_old
    end if
    Calculate descriptors for TI
    Calculate descriptors for SG
    Calculate energy E of the system
    Determine initial temperature T_0
    Set number of consecutive denied swaps N_con to 0
    for i in 1 to N_ite do
        Swap two pixels
        Calculate energy E' of current state
        Determine ΔE =E − E'
        if ΔE < 0 then
            Set E to E'
            Set N_con to 0
            if E < E_thr then
                end procedure
            end if
        else
            Calculate Metropolis probability M_P
            if M_P > random number between 0 and 1 then
                Set E to E'
                Set N_con to 0
            else
                Reswap the pixels
                N_con = N_con + 1
                if N_con == N_sto then
                    end procedure
                end if
            end if
        end if
    end for
end procedure
```

For the 2D case, our algorithm calculates E for four different spatial directions: x, y, d_{xy}, and d_{yx}. For the 3D case, nine spatial directions are considered: x, y, z, d_{xy}, d_{yx}, d_{xz}, d_{zx}, d_{zy}, and d_{yz}. While for all directions an individual E is calculated, eventually only the total sum is optimized and analyzed during the quality assessment.

In principle, any structural descriptor or combination of descriptors can be used to compute E. Overall, the more descriptors considered to define E, the larger the computational demand. Although our proposed algorithm is capable of using any combination of the descriptors described in Sec. II A, as written earlier we only used herein a combination of S_2 and L_2. We decided to use the $S_2 - L_2$ combination for the following reasons: First, the $S_2 - L_2$ set is commonly used in literature [17–19]. Second, evaluation of the S_2 and L_2 incurs a smaller computational cost than that of MPH, PPA or P. Restricting the calculations to the $S_2 - L_2$ set is therefore CPU efficient. Third, prior extensive testing revealed that the incorporation of additional descriptors does not significantly improve the reconstruction quality for the considered case studies (not shown herein). Last, using the $S_2 - L_2$ set leaves us with C_2, MPH, PPA, and P for quantitative validation of our approach, in addition to visual inspection.

The SA reconstruction process starts with a random structure that honors the different phase fractions found in the TI. After the energy E of this 2D or 3D field is determined, new fields (or states) are sequentially proposed by swapping the values of two randomly chosen pixels belonging to different phases. After each swap, the energy E' of the resulting state is calculated, which leads to the energy difference $\Delta E = E' - E$ between the current and proposed state. The new state is then accepted with the Metropolis probability $p(\Delta E)$,

$$p(\Delta E)\begin{cases}1, & \text{if } \Delta E \leqslant 0\\ \exp\left(-\frac{\Delta E}{T}\right) & \text{if } \Delta E > 0\end{cases}, \tag{4}$$

where T represents the temperature of the system. The swapping process begins with a starting T_0 determined by the algorithm of Ben-Ameur [33]. T_0 is tuned such that the average acceptance rate of swaps that result in $\Delta E > 0$ is 0.5 during the first iteration. Next, T is decreased throughout swapping according to the following cooling scheme:

$$T = T_0 \lambda^{itt}, \tag{5}$$

where λ is a tuning factor, between 0 and 1, and itt is the number of iterations. This temperature cooling helps the process to transition from a more exploratory phase to pure optimization, both of which contribute to an optimal reconstruction for a given amount of iterations. The swapping continues until one of the following three stopping criteria is fulfilled: (i) a predefined number of iterations (N_{ite}) has been reached; (ii) a predefined number of consecutive rejected swaps (N_{sto}) has been performed; (iii) E becomes smaller than a predefined threshold (E_{thr}).

The computational cost incurred by SA depends on the cost of computing ΔE per swap and the required number of swaps. The number of swaps is controlled by the number of pixels that are considered for a swap at each iteration (e.g., limitations to boarder pixels), and the grid dimension [34]. In this work, the cost associated with the evaluation of ΔE and the performed number of swaps are reduced as follows: to begin with, the structural descriptors are not fully recalculated after each iteration. Instead, only the descriptors' variations caused by a given swap are computed. To decrease the number of pixels considered in the swapping process, the approach suggested by Karsanina *et al.* [19] for selecting the pair of pixels to be swapped is used. This starts by choosing a random location within a randomly chosen phase. From this location, a direction (main axis or diagonals) along which two pixels will be swapped is randomly selected. A random pair of grid points that fulfills the following conditions is considered for a swap: (i) the two grid elements must be located at the interface between two or more phases; (ii) they must belong to two different phases, and (iii) they cannot be direct neighbors. Finally, our algorithm uses the multiresolution approach

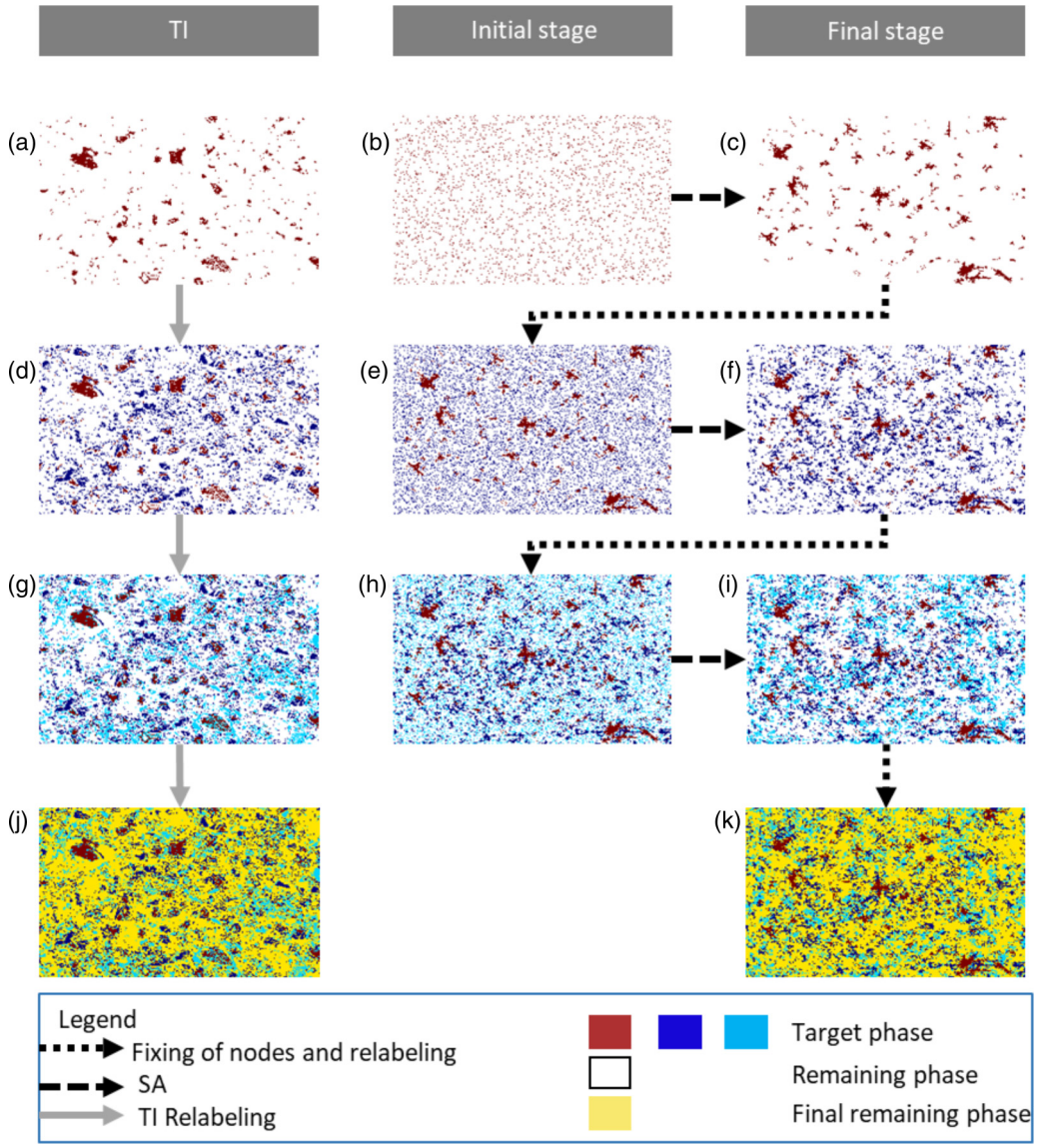

FIG. 3. Main principle of the hierarchical approach for simulated annealing. The multiphase TI (j) is generated by simulating one target phase after another. To do so pixels from the current target phase and the remaining phase are swapped. Once one target phase is simulated the pixels of the previously simulated phase are fixed and cannot be changed anymore and the simulation of the next target phase begins.

described in Sec. II B 3 to reduce the required number of iterations.

2. Hierarchical simulated annealing (H-SA)

We propose to perform multiphase SA reconstruction using a hierarchical approach, by simulating one phase after another. The main principle of the approach is depicted in Fig. 3. At the beginning of the simulation, the multiphase TI [see Fig. 3(j)] is transformed into a binary image by assigning one category to the phase that will be simulated first [colored red in Fig. 3(a)] and the other category (which we call "remaining phases," colored white) to all the other phases. This TI is then used to produce a binary realization as described in Sec. II B 1. Once this binary realization has been generated, the pixels associated with the first simulated phase, e.g., the red-colored pixels in Fig. 3(c), are fixed. These pixels can thus no longer be swapped. In the next step, the remaining is split into two subgroups: the phase that will be simulated next [colored dark blue in Fig. 3(e)] and the other remaining pixels. The TI is now accordingly turned into a three-level categorical image. To generate a realization that honors this new TI, pixel values from the two subgroups [blue- and white-colored pixels in Fig. 3(e)] are considered for swapping. Afterward, the remaining phase is again split into two subgroups, and the TI is updated accordingly. This process continues until all phases but one present in the original TI have been simulated. As the last phase equals the remaining phases from the simulation of the last but one target phase, no additional simulation is required in the final step [Figs. 3((i) and 3(k)].

We suggest simulating the phases in decreasing order of their PPA, with the matrix being simulated last. Doing so ensures that the algorithm has enough space to reconstruct the larger particles. Additionally, it also speeds up the computations.

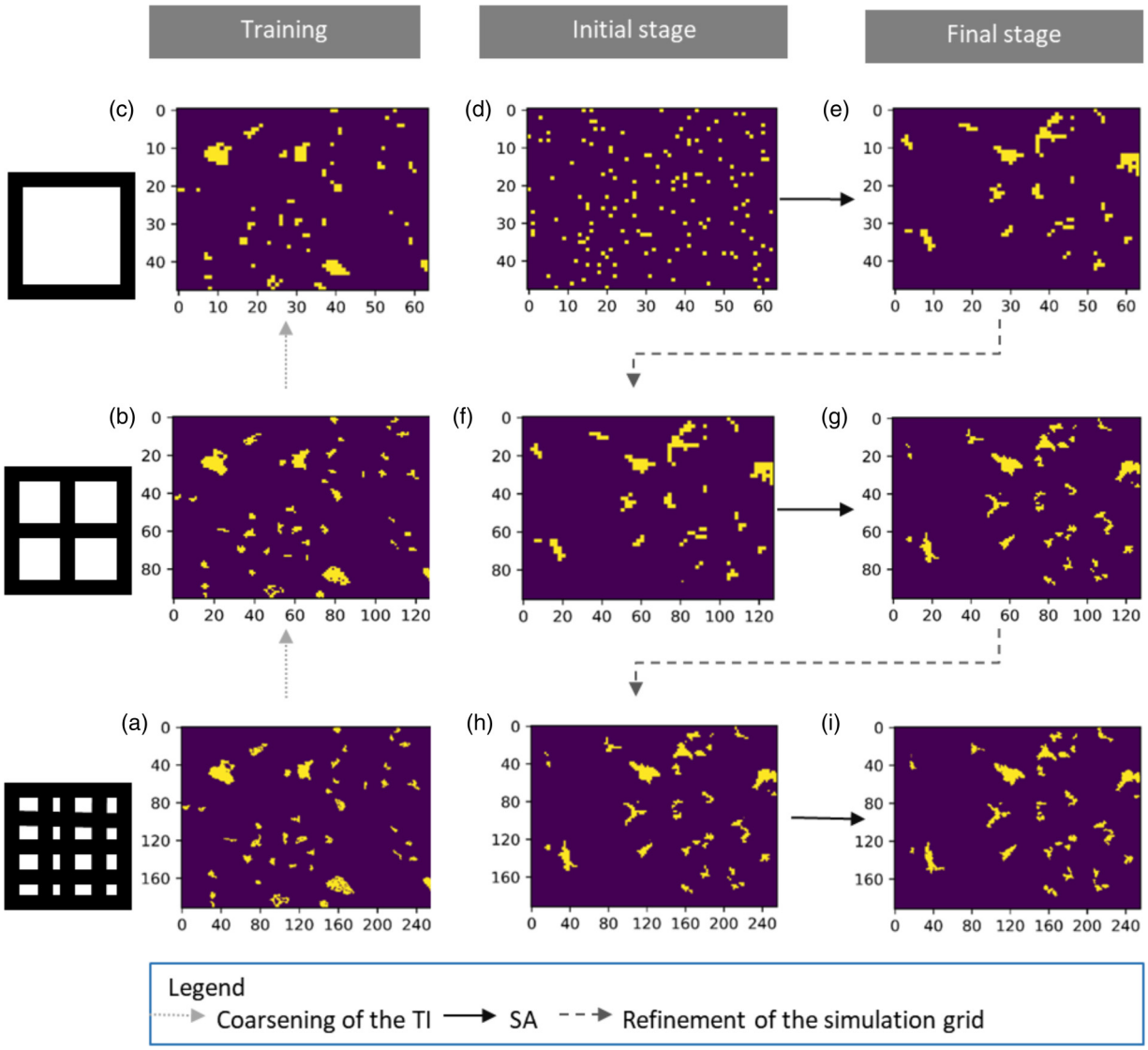

FIG. 4. Concept of the multiresolution approach to decrease the computational cost. Initially the TI is coarsened to a coarser scale representation, while merging the information from a 2×2 patch by a single pixel. Once the reconstruction is initialized on the coarse resolution, the structure is optimized by SA (black arrow). Once the desired accuracy is obtained, the structure is refined (dark gray dashed arrow) and reoptimized by SA using the appropriate TI.

Moreover, if a phase consists of aggregates of strongly different shapes and sizes, then we recommend to split this phase into several subphases. Doing so increases the information content of the structural descriptors and hence the reconstruction quality. More details about this approach can be found in Sec. II B 4.

3. Multiresolution simulated annealing (MSA)

a. Binary multiresolution simulated annealing

This section details the multiresolution component of our algorithm. For clarity, a 2D binary case is considered and exemplified in Fig. 4. The approach, however, naturally extends to 3D reconstruction. The extension to multiphase categorical images is discussed in Sec. II B 3 b.

To lower the computational demand of SA, Pant *et al.* [20], Campaigne and Fieguth [34], and Karsanina and Gerke [35] suggested a multiresolution approach. The underlying idea of the multiresolution approach is to first generate a realization of the reconstruction on the coarser scale and to refine the structure afterwards. Once this coarse scale resolution is obtained, the reconstruction grid is iteratively refined (this procedure is represented by the dark gray dashed arrow in Fig. 4). Next, the structure is optimized by SA to account for small changes due to the enhanced resolution (black arrows in Fig. 4). To obtain the structural descriptors for the coarse scale representation two different approaches can be used. In the one of Karsanina and Gerke [35], the structural descriptors are simply recalculated while taking each 2^n entrance of the original curve, where n is the number of different resolutions that are used. In the second approach, the ones used by Pant *et al.* [20] and Campaigne and Fieguth [34], the original TI is a coarsened process that is represented by the light gray dotted arrows in Figs. 4(a)–4(c). The structural descriptors are then derived from the coarsened image. We decided to use the latter approach for this paper as the computational time spent on the

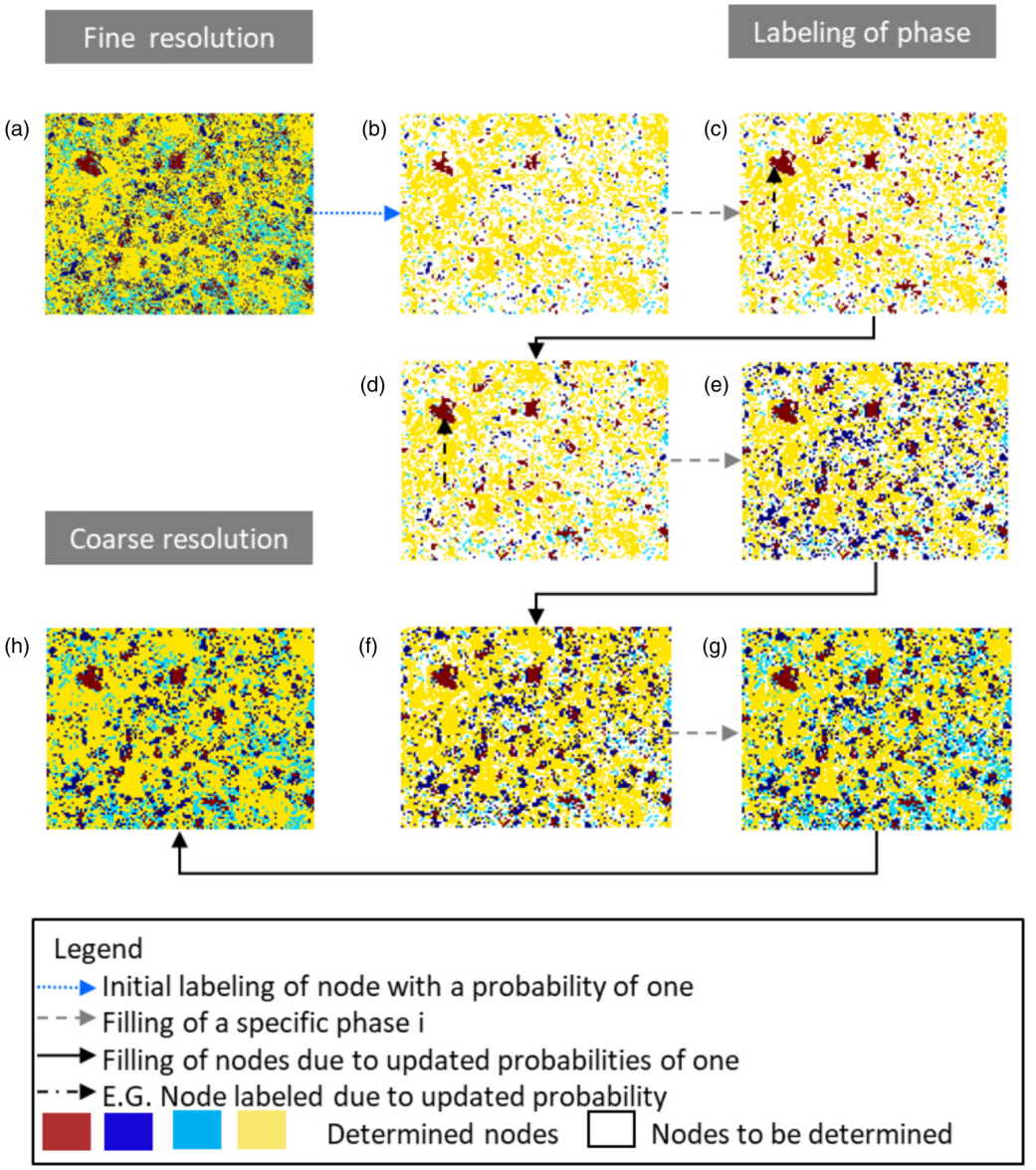

FIG. 5. Sketch of the TI coarsening scheme. The figure represents the evolution of the TI during coarsening.

coarsening of the TI is negligible, while the derived structural descriptors are in our opinion physically more meaningful. Nevertheless all steps in the reminder of the paper could also be performed using the approach of Karsanina and Gerke [35].

The use of the multiresolution approach has two main advantages. First, coarser resolution has a much smaller number of pixels. Larger structures can be generated within fewer iterations, compared to using pixels that have to be investigated for the calculation of the structural descriptors, thereby decreasing the computational cost per iteration.

In the refinement process, each pixel of the coarse resolution is replaced by a specific number of pixels on the next finer resolution. In the case where a 2×2 template is used, each coarse-resolution pixel is replaced by four pixels on the next, finer-resolution grid. At this stage, the finer-scale reconstruction process is already guided in the appropriate direction as the largest clusters have been generated already. Hence, only modifications at the interface of particles are necessary to account for minor changes due to resolution. To ensure that the fine-scale long-range connectivity is not destroyed, only pixels located at the border of particles on the fine resolution are allowed to be exchanged. The associated speed gain increases with the number of resolution used in the reconstruction process. However, the pixel size of the coarsest resolution grid should not be smaller than the size of the smallest particle in the original TI.

For a deeper understanding of the grid-coarsening step of the multiresolution approach by Pant *et al.* [20], consider a binary 2D TI where the solid particles and pores are assigned values of 0 and 1, respectively. To decrease the grid resolution by a factor of 4 Pant and co-workers used the average grid value over a 2×2 template that corresponds to the considered coarse resolution pixel. If this average is larger than 0.5, the coarse resolution pixel is assigned a 1. If it is smaller, it is assigned a 0, while averages of 0.5 are randomly assigned a 0 or a 1 in such a way that the abundance stays constant within

the different resolutions. Note that this may slightly change the abundance due to the digitized, and hence discrete, nature of the image [20].

Despite its advantages over standard SA, the Pant *et al.* [20] method has two major drawbacks. First, it is restricted to binary images. Indeed, the averaging in the coarsening step does not extend naturally to the multiphase case. Second, it does not handle well multimodal PPA within one phase. The reason for this is that to generate the larger inclusions efficiently, several resolutions need to be used or, alternatively, larger templates. On the coarsest resolution, the pixel distance will consequently be larger than the smallest inclusions. In order to keep the abundance constant between the resolution, the contributions from several smaller inclusions are merged into one pixel value on the coarse resolution After the grid refinement, the contributions of several smaller grains will still be represented by one large inclusion, for example, a 2×2 inclusion on the fine resolution due to the pixel switching rule suggested by Karsanina *et al.* [19], it is almost impossible to resplit these artificially large inclusions into several smaller ones. Thus the final realization features too little small inclusions compared to the TI. To mitigate this problem, one could choose as coarsest resolution the size of the smallest inclusion found in the TI. In this case, the multiresolution will likely no longer show any speedup compared to traditional SA. However, subphase splitting might provide a solution (see Sec. II B 4).

b. Multiresolution hierarchical simulated annealing (MH-SA)

To address the two problems highlighted above, we propose a nested multiresolution hierarchical SA algorithm. Its three main innovations are (i) a hierarchical coarsening and refining component that can handle multiphase categorical images, (ii) a subphase splitting process that prevents the generation of artificially large patterns for the multimodal phases, and (iii) a phase merging scheme to handle complex phase relationships. While the first innovation is detailed in this section the remaining ones are discussed in Sec. II B 4.

Grid coarsening and refining

The new grid coarsening and refining scheme has two variants, depending on whether or not our proposed hierarchical approach (see Sec. II B 2) is used. The latter is important for the simulation of multiphase categorical images.

Let us start with the simpler, nonhierarchical variant. For simplicity, the 2D case is considered here. The 3D case will be addressed later in this section. In addition, in this example, the resolution of the original TI is reduced at each step by a factor of 4 (i.e., a factor of 2 in both spatial directions). However, using any other coarsening factor is straightforward.

Nonhierarchical coarsening and refining

The central idea of our coarsening approach is to represent the contribution from multiple pixels (e.g., four pixels, 2×2) on the fine grid by one pixel on the coarser resolution. The pseudocode of the algorithm that coarsens the images is described in (Algorithm 2) and works as follows:

(i) First, the number of pixels occupied by each phase at the original (fine) resolution (n_{ini}) is calculated (line 2 in Algorithm 2).

(ii) Second, the target number of pixels per phase at the coarse resolution (n_{targ}) is obtained by dividing n_{ini} by the reduction factor (e.g., 4 for a patch of 2×2). This makes sure that the abundance stays constant (line 3 in Algorithm 2).

(iii) Third, after splitting the fine resolution grid into 2×2 patches, the occurrence probability p_{ij} of each phase i is calculated for each patch j. p_{ij} is the probability that the pixel representing the fine-resolution patch j on the coarse resolution belongs to phase i. Using a reduction factor of 4, it can only take five values: 0, 0.25, 0.5, 0.75, and 1 (line 4 in Algorithm 2).

(iv) In step 4, all pixels with a probability of 1 for a given phase are filled accordingly (blue dotted arrow in Fig. 5). At this stage, the number of already labeled pixels belonging to a specific phase, n^i_{curr}, will be smaller than n^i_{targ} [Fig. 5(c) includes still white pixels] (lines 14–16 in Algorithm 2).

(v) In the final step, the filling process is done phase by phase. While the filling can be done in any order, it is suggested to start with the least frequent one over the whole TI (let us call this phase i_0). The algorithm determines the difference in the number of pixels, N^i_{dif}, between N^i_{targ}

Algorithm 2. The grid-coarsening approach.

1: **procedure** $coar_{im}$=GRID-COARSENING(fin_{im}, ORDER, HIERARCHICAL, $coar_{imold}$)
▷ $coar_{im} \equiv$ Coarsened version of the image
▷ $fin_{im} \equiv$ Input image
▷ ORDER = 1D-array featuring labels in which order the phases are simulated
▷ HIERACHICAL ≡ boolean determining the use of a hierarchical approach
▷ $coar_{imold} \equiv$ representation of the previous simulation stage
2: Calculate phase occurrence of each phase on fine resolution (n^i_{ini})
3: Calculate target occurrence on the coarse resolution ($n^i_{targ} = \frac{n^i_{ini}}{4}$)
4: Calculate probability maps (pro^i_{map})
5: Initialize the current count on coarse resolution (n^i_{curr})
6: **If** HIERARCHICAL ==1 **then**
7: Fill previous fixed pixels
8: update (n^i_{curr})
9: update (pro^i_{map})
10: **end if**
11: **for** i in 1 to number of phases in order **do**
12: **for** j in each phase of order **do**
13: **if** $n^j_{curr} < n^j_{targ}$ **then**
14: Fill all points in $coar_{im}$ where $pro^j_{map} == 1$
15: update (n^j_{curr})
16: update (pro_{map})
17: **end if**
18: **end for**
19: **if** $n^i_{curr} < n^i_{targ}$ **then**
20: Determine difference (n^i_{diff}) between n^i_{targ} and n^i_{curr}
21: Determine n_{diff} points with the highest probabilities in pro^i_{map} and label them accordingly
22: update (n^i_{curr})
23: update (pro_{map})
24: **end if**
25: **end for**
26: **end procedure**

and N^i_{curr}, that is, the amount of undetermined pixels on the coarser resolution that should be labeled as phase i in order to keep the phase distribution constant. Consecutively, the N^i_{dif} undetermined pixels on the coarsest resolution that have the highest probability to be associated with phase i are labeled accordingly (gray dashed arrow in Fig. 5). If several pixels have the same probability, the pixels are selected randomly. Once N^i_{targ} equals N^i_{curr}, the p_{ij} of the yet undetermined coarse-grid pixels are updated to account for the fact that the i_0 phase has been simulated (lines 19–24 in Algorithm 2). The updated p_{ij} values that have become 1 lead to additional filling of yet undetermined pixels [e.g., the white pixel that becomes blue, marked by the black dot-dashed arrow in Fig. 5(d)] and to the corresponding updates of N^i_{curr} (lines 13–17 in Algorithm 2). This process continues with the remaining phases until all coarse-res pixels have been labeled.

To move from a coarser to a finer resolution, each coarse pixel is replaced by 2×2 pixels with the same phase, on the finer resolution. Eventually, some pixels are randomly relabeled in order to honor the phase distributions frequency.

The extension to three dimensions is straightforward. One just needs to represent the contribution from eight pixels ($2 \times 2 \times 2$) on the fine grid by one pixel on the coarser resolution. To do so n_{ini} is first determined and n_{targ} calculated by dividing n_{ini} by a factor of 8. Afterward, the fine resolution grid is split in $2 \times 2 \times 2$ patches, of which the individual phase probabilities are calculated. The rest of the coarsening process works the same as in two dimensions.

Hierarchical coarsening and refining

In the hierarchical variant, once the first phase is simulated it is necessary to adapt the coarsening approach for fixed pixels. The reason for this is that, if an image is coarsened by the algorithm twice independently, the two realizations will slightly differ due to the random selection of pixels having the same probability. Consequently, the previously simulated phase would not occupy the same pixels in a coarsened two-phase realization as in the three-phase realization. The hierarchical approach, however, requires that previously simulated phases stay unchanged.

To address this problem in hierarchical coarsening, the pseudocode of Algorithm 2 is modified. The difference is that the algorithm takes the previous representation of the coarse scale as an additional input. Additionally, between lines 6 and 10 the algorithm puts all previously fixed pixels at the same spot in the new empty grid [e.g., red pixels in Fig. 6(b3)].

Similarly, once the first phase in the hierarchical approach is simulated, a representation of the finer resolution level, from the previously simulated phase(s), which incorporates fixed pixels, that needed to be honored throughout the simulation of all phases, will always be available. Hence, the grid refining process needs to be updated, analogous to the grid coarsening. The principle of the algorithm described in Algorithm 3 is as follows:

(i) Filling of the fixed phases: In the yet empty fine resolution grid, all fixed pixels of the old fine resolution representation are placed at the same pixels on the new fine resolution realization (e.g., see the red pixels in [Figs. 6(b5) and 6(b7)]. In doing so all fixed phases have been filled and will not be altered anymore (line 4 in Algorithm 3).

Algorithm 3. The grid-refining approach.

1: **procedure** fin_{im}=GRIDREFINING($coar_{im}$, n^i_{targ}, HIERARCHICAL, ORDER fin_{imold})
   ▷ fin_{im} ≡ fine scale representation of the image
   ▷ $coar_{im}$ ≡ input image
   ▷ $fine_{imold}$ ≡ representation of the previous fine simulation stage
2: Initialize fin_{im} having 2 x 2 the size of $coar_{im}$
3: **if** HIERACHICAL == 1 **then**
4: Fill fixed pixels from fin_{imold} in fin_{im}
5: Fill empty pixels in fin_{im} represented by nonfixed pixel in $coar_{im}$
6: **for** i in 1 to number of phases in order **do**
7: Calculate n^i_{curr}
8: Calculate n^i_{diff}
9: Fill n^i_{diff} of the remaining pixels accordingly
10: update (n^i_{curr})
11: **end for**
12: **else**
13: Fill all pixels in fin_{im} according to fin_{imold}
14: **for** i in 1 to number of phases in order **do**
15: Calculate n^i_{curr}
16: Calculate n^i_{diff}
17: **if** $n^i_{diff} < 0$ **then**
18: label n^i_{diff} occupied by this phase as empty
19: update (n^i_{curr}) and n^i_{diff}
20: **end if**
21: **end for**
22: **for** i in 1 to number of phases in order **do**
23: **if** $n^i_{diff} > 0$ **then**
24: label n^i_{diff} empty pixels accordingly
25: update (n^i_{curr}) and n^i_{diff}
26: **end if**
27: **end for**
28: **end if**
29: **end procedure**

(ii) First filling of the target phases. All empty pixels on the new fine resolution grid which correspond to a nonfixed pixel on the coarse resolution get the same labels as the coarse pixel (line 5 in Alg orithm 3).

(iii) Filling of the remaining empty pixels. The current fine resolution grid still features empty pixels. These pixels correspond to a fixed phase pixel representation on the coarse resolution but do not correspond to a fixed pixel on the old fine resolution image. These pixels are randomly labeled with one of the two currently simulated phases, in such a way that their abundance stays constant (line 9 in Algorithm 3).

After detailing the hierarchical and multiresolution approaches separately, let us now join them to explain the basic principles of our multiresolution hierarchical simulated annealing algorithm. To start the generation of the structure, one needs to provide the algorithm for each phase with the appropriate resolution to generate the particles efficiently. One should choose the number of resolution such that the biggest particles of a phase are represented by a few pixels on the coarsest resolution. Concerning the coarsening and refining, it has to be stated that for all phases the same patch size is used, the only difference to achieve the appropriate resolution

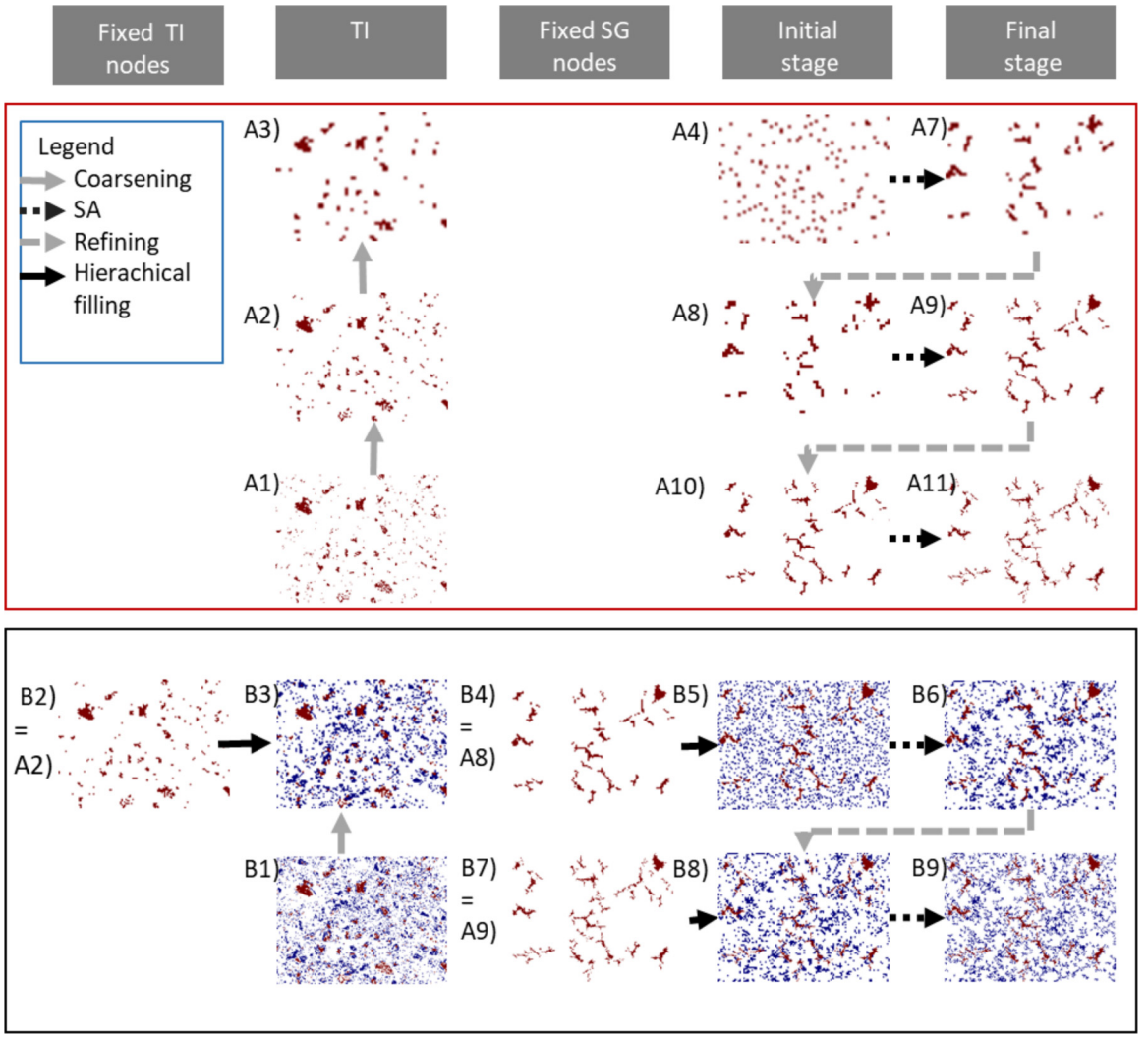

FIG. 6. Sketch of the hierarchical multiresolution approach. In the upper red rectangle, the simulation of the first phase is presented. At this stage, the algorithm does not have to handle fixed pixels, hence the principle is the same as described in Fig. 4. In the lower black rectangle, the simulation of the second phase (blue pixels) is illustrated. In both the coarsening of the TI and the refining of the simulation grid, the algorithms needs to honor the fixed (red) pixels, from the first simulation stage at the same resolution.

is the number of consecutive coarsenings. Once the required number of resolutions has been determined for each phase, the individual phases will be simulated in a hierarchical way, in decreasing order of their number of required resolution, with the most abundant phase coming last. The individual phases are simulated sequentially from the coarsest resolution to the finest (original) one.

To illustrate the algorithm let us consider a medium consisting of three phases [as shown in Fig. 6(b1)]: phase 1 (red), phase 2 (blue), and the matrix (white). Phase 1 features inclusions which require three resolutions [Fig. 6(a)], while phase 2 contains inclusions that require two resolutions [Fig. 6(b)].

In step 1, the algorithm transforms the ternary TI [Fig. 6(b1)] in a binary version [Fig. 6(a1)] assigning one category to phase 1. The rest consists of the matrix and phase 2. In step 2, the algorithm coarsens the binary TI twice and generates an initial realization on the coarsest resolution [Figs. 6(a1)–6(a3)], which honors the different phase fractions found in the TI. This structure is afterward optimized using SA (black dotted arrow) until the reconstruction resembles the coarse TI sufficiently well. Next, the structure is refined for the first time (gray dashed arrow) and optimized again to resemble the intermediate resolution TI. Subsequently, the simulated image is refined again to fit the original resolution and optimized [Figs. 6(a10) and 6(a11)] to match the chosen descriptors associated with the original TI. All pixels belonging to phase 1 are then fixed at all grid levels.

To start the simulation of the phase 2, the TI is transformed back to its ternary version. This TI is then coarsened once using the hierarchical approach described above (Fig. 6). Next, the nonfixed (white) pixels from the intermediate reconstruction of phase 1 [see Fig. 6(a9)] are randomly transformed into phase 2 and matrix in such a way that the obtained image has the same phase distribution as the TI [Fig. 6(b5)]. Once this is done, the algorithm swaps pixels from phase 2 and the matrix in order to minimize E. After the optimization process, the simulated structure is refined to its original resolution honoring the frozen pixels that belong to phase 1 [Fig. 6(b7)].

Last, the structure is optimized by SA again to provide the final realization.

4. Nested simulated annealing (NSA)

a. Subphase splitting

The problem with the incorporation of multimodal PPA within the approach of Pant *et al.* [20] is that different particles would need a different number of resolutions in order to be generated efficiently. However, the approach can only use one number of grid levels per phase i. Moreover, the structural descriptors (at least as we use them here) just represent an average over the different modes, which will prohibit reproduction of the classes in the simulations.

Our proposed algorithm deals with this problem in the following way: we group all particles from a given phase in independent subphases, purely based on their size, and simulate them as if they would be a multiphase medium. Once the different subphases have been generated independently on the finest resolution grid, the subphases are remerged together into a single class. To end the generation, the algorithm runs again an optimization process, where pixels from the remerged phase and the matrix are swapped. This ensures that also the average structural descriptors are well reproduced. While in theory, the optimization process can use its own stopping criteria, we use the same accuracy as for the individual subphases and to only decrease the max number of iterations, for consistency.

The splitted TI has to be provided as an input to the algorithm, with the information which subphases have to be remerged at the end.

b. Phase merging to simulate phase relationships

The hierarchical approach, in combination with the set of structural descriptors used in this paper, has difficulties in generating structures that honor complex topological relationships between phases, for instance, a phase A that always grows around a phase B, or small inclusions of B inside A, or just two phases which always occur together, as represented by the light blue and red phases in Fig. 7(a). The reason for this is that S_2 and L_2 in the multiphase case are phase specific and do not include information about the topological relationships (except for the matrix).

Subphase splitting addresses this problem only partly; it does not constrain topology. To solve this issue, our algorithm initially merges the two phases into a superphase, e.g., AB' (dark blue phase in Fig. 7). Assuming that the phase AB' is enclosed in the matrix, the approach will then first simulate phase AB' by swapping the corresponding yellow and blue pixels. Once the AB' structure is generated, AB' is split back into A and B (black dashed arrow in Fig. 7). Next, only the A and B pixels are exchanged by SA (i.e., the remaining phase pixels are considered fixed). During the phase exchange, the algorithm is allowed to change pixels across various clusters. As a result of the subphase splitting, using the S_2 and L_2 descriptors only is sufficient to generate realizations with the appropriate topological relationships.

The identification of such phase relationships has to be done beforehand by the user. This information is then provided to the algorithm through the use of an additional TI.

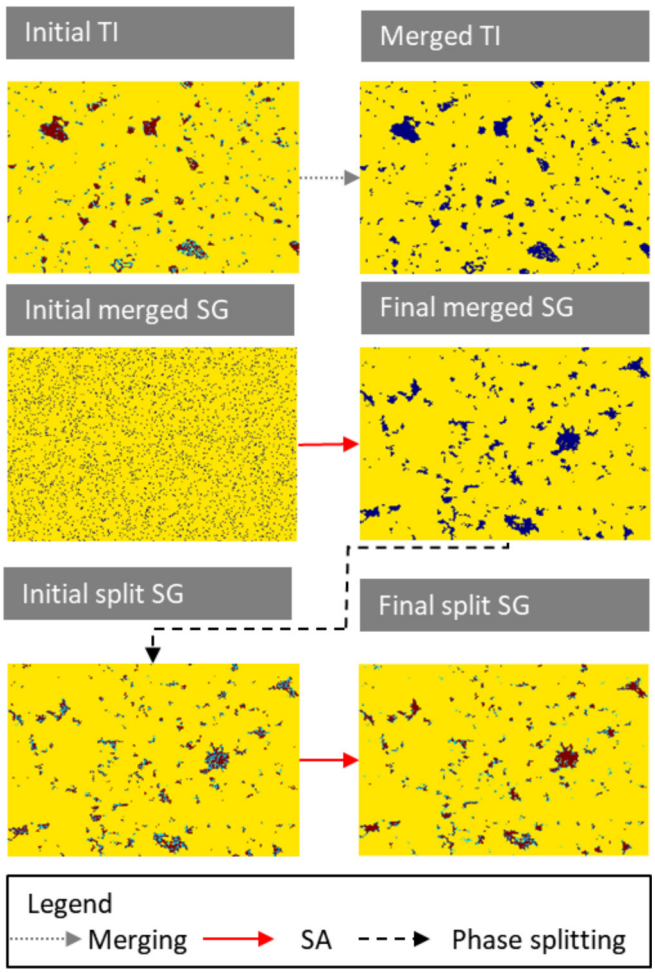

FIG. 7. Phase merging principle to simulate phase relationships. Initially, the red and blue phases are merged to form the new yellow superphase. This phase is reconstructed using SA. In the phase splitting step the dark blue phase is split back into red and light blue pixels, which are eventually optimized by SA. SG = simulation grid.

A phase merging similar in principle to ours was applied by Gerke *et al.* [36] to merge information from images at different resolution.

5. Nested multiresolution hierarchical simulated annealing (NMH-SA)

Our full algorithm is described in Algorithm 4. In case the user wants to apply subphase splitting and merging, he needs to transform the TI and provide both the transformed TIs and the original one. The number of transformed TIs is thereby equal to the number of nested levels. The algorithm will need three TIs for a multiphase reconstruction with both subphase splitting and merging. During the simulation process, the algorithm will then simulate first the individual (sub)phases sequentially using the hierarchical multiresolution approach as depicted in Fig. 6, from the coarsest to the finest resolution (Algorithm 4, lines 2–21). Once all individual (sub)phases have been simulated, the algorithm will then resplit and optimize the individual phases by SA, which were merged during

Algorithm 4. Nested hierarchical multiresolution simulated annealing.

procedure SG=NHMSA(TI,ORDER, λ, Nr, N_{ite}, N_{sto}, E_{thr}, MERGING, Ti_{mer}, $Order_{mer}$, $Phase_{spl}$, TI_{spl}, $Order_{spl}$)
▷ Order ≡ 1D array featuring the labels in which order the subphases should be simulate
▷ Nr ≡ 1D array featuring Nr of resolution per subphase
▷ N_{ite} ≡ = 1D array featuring max number of iterations per subphase
▷ N_{sto} ≡ = 1D array featuring max number of consecutively denied pixel swaps per subphase
▷ $E_{thr\equiv}$ 1D array featuring the Energy threshold per subphase
▷ Merging = Boolean representing if Merging was applied
▷ Ti_{mer} ≡ Ti in which initially merged phases are split
▷ $Order_{mer}$ ≡ List featuring labels in which order phases should be split and simulated
▷ $Phase_{spl}$ ≡ Boolean representing if phase splitting was applied
▷ Ti_{spl} ≡ Ti in which initially split phases are merged
▷ $Order_{spl}$ ≡ List featuring the labels in which order the phases should be merged and simulated
    Transform TI to a binary image
    for i in 1 to Nr[1]-1 **do**
        Coarsen TI non hierarchically (Algorithm 2)
    end for
    Initialize SG
    for i in 1 to Nr[1]-1 **do**
        Apply SA (Algorithm 1)
        Refine SG by one resolution non hierarchically (Algorithm 3)
    end for
    for J in 2 to length of order **do**
        Transform TI to current number of phases (Algorithm 2)
        for i in 1 to Nr[J]-1 **do**
            Coarsen TI using hierarchical approach (Algorithm 2)
        end for
        Initialize SG on coarsest resolution hierarchically
        for i in to Nr[J]-1 **do**
            Apply SA (Algorithm 1)
            Refine SG by one resolution hierarchically (Algorithm 3)
        end for
    end for
    if Merging is == 1 **then**
        for i in $Order_{mer}$ **do**
            Split phases initially merged according to Ti_{mer}
            Apply SA (Algorithm 1)
        end for
    end if
    if subphase splitting == 1 **then**
        for i in $Order_{Spl}$ **do**
            Merge phases initially split according to Ti_{spl}
            Apply SA (Algorithm 1)
        end for
    end if
end procedure

a potential merging step. Finally, the algorithm will merge and optimize the individual phases, if subphase splitting was applied.

III. RESULTS

A. Cement paste

This section studies the performance of our proposed nested multiresolution hierarchical simulated annealing (NMH-SA) approach. All simulations were performed sequentially on a Dell Latitude E5570 Notebook with 16 GB of RAM and a i7-6600U CPU. Two different case studies are considered: cement paste and Boom Clay. Since the purpose of this paper is to present and demonstrate our algorithm, we decided to use images presented in the literature [8,30] as TIs for our test. We are fully aware that these two TIs might contain artifacts caused by the way they were acquired but this is not important for this study.

For each case study, the performance of NMH-SA is compared with that of the popular DS MPS algorithm. A temporary academic license of the DS MPS code can be obtained upon request to one of its developers (Grégoire Mariethoz, Philippe Renard, Julien Straubhaar). To further investigate the capabilities of our algorithm and the different options discussed above, three different settings are used: The first uses only hierarchical simulated annealing (H-SA). The second adds the coarsening scheme (MH-SA). This variant can be seen as a multiphase version of the approach proposed by Pant *et al.* [20]. Our third setting is the full NMH-SA and thus fully takes advantage of the approaches described earlier in this paper.

1. 2D reconstruction

An SEM image of hardened cement paste using ordinary Portland cement is used as a TI [Fig. 1(a)] [30]. The sample has a water-cement ratio of 0.42. Based on manual thresholding in MATLAB the initial grey scale image was transformed in four different phases (pore, portlandite, clinker, and C-S-H). The voxel size is half a micrometer. Both the TI and considered simulation domain have a size of 192 × 256 pixels. More details about the sample casting, imaging, and segmentation can be found in [30].

A total of 50 reconstructions was produced with each of the three different SA settings. The reconstruction scheme used to generate the NMH-SA reconstructions is depicted in Fig. 8. Randomly chosen reconstructions derived by the three settings are displayed in Figs. 9(b)–9(d). The same S_2 and L_2 descriptors and maximum investigation length (maximum lag distance used in the descriptor calculations) were used for all reconstructions. For the reconstruction of an image with half the original resolution, the number of pixels used to calculate the S_2 and L_2 descriptors on the coarse resolution was halved as well. Figure 10 shows the descriptor misfit over (50) the different realizations.

It can be seen in Fig. 9(c) that MH-SA produces visually worse results than H-SA [Fig. 9(b)] and NMH-SA [Fig. 9(d)]. This is confirmed by the associated MPH descriptor values (Fig. 10). The mean MPH misfit between the TI and the realizations is one order of magnitude larger for MH-SA than for the two other variants. This quite large MPH error for MH-SA is due to the frequent occurrence of larger C-S-H areas without inclusion in the reconstruction [Fig. 9(c)]. These large C-S-H clasts are not found in the TI [Fig. 9(a)]. The

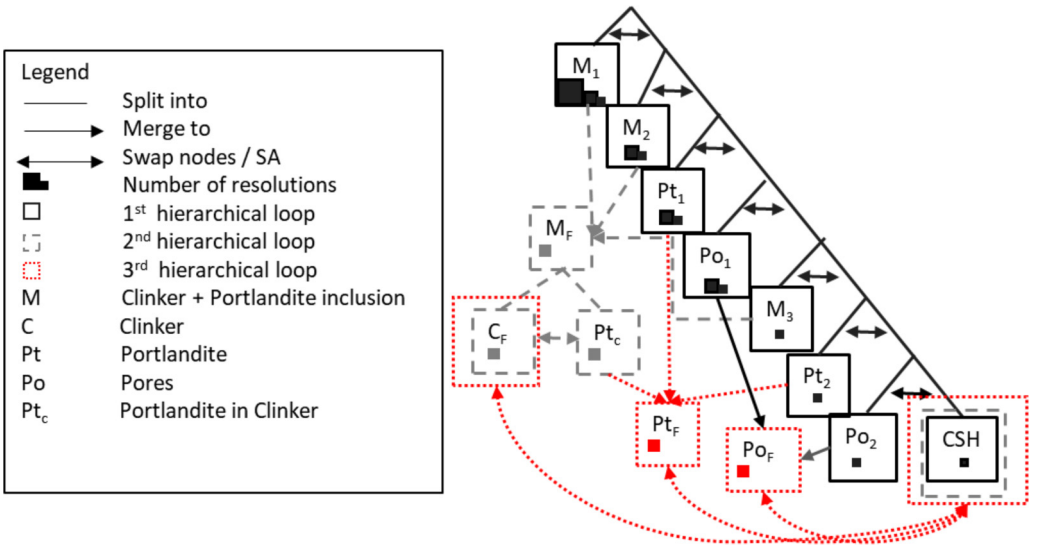

FIG. 8. Settings for the NMH-SA simulation, in order to generate the hardened cement paste. In the first loop (black solid), the individual subphases were generated in a hierarchical order using different resolutions. In the second loop (dashed line), the portlandite inclusion within the clinker were split from the clinker as part of the used phase merging step. In the last loop (dotted) the subphase splitting process was finished.

clasts appear visually so prominent because the MH-SA reconstructions [Fig. 9(c)] are missing the very small inclusions of the other phases found in the TI. This observation is also confirmed by the PPA and P misfit that are one magnitude higher compared to the other two methods.

The phase merging capabilities of the NMH-SA algorithm can be observed by comparing the clinker grains (red-colored areas), Figs. 9(b) and 9(d). In the NMH-SA reconstruction [Fig. 9(d)], turquoise blue areas, portlandite can be found within the larger grains. In contrast, H-SA does reproduce this feature less frequently.

Besides that the reconstructions derived by H-SA and NMH-SA both visually look almost equally good [compare Figs. 9(b) and 9(d)]. The MPH error, however, shows that any of the NMH-SA (purple box plots in Fig. 10) realizations is of better quality than the best MH-SA realization (green box

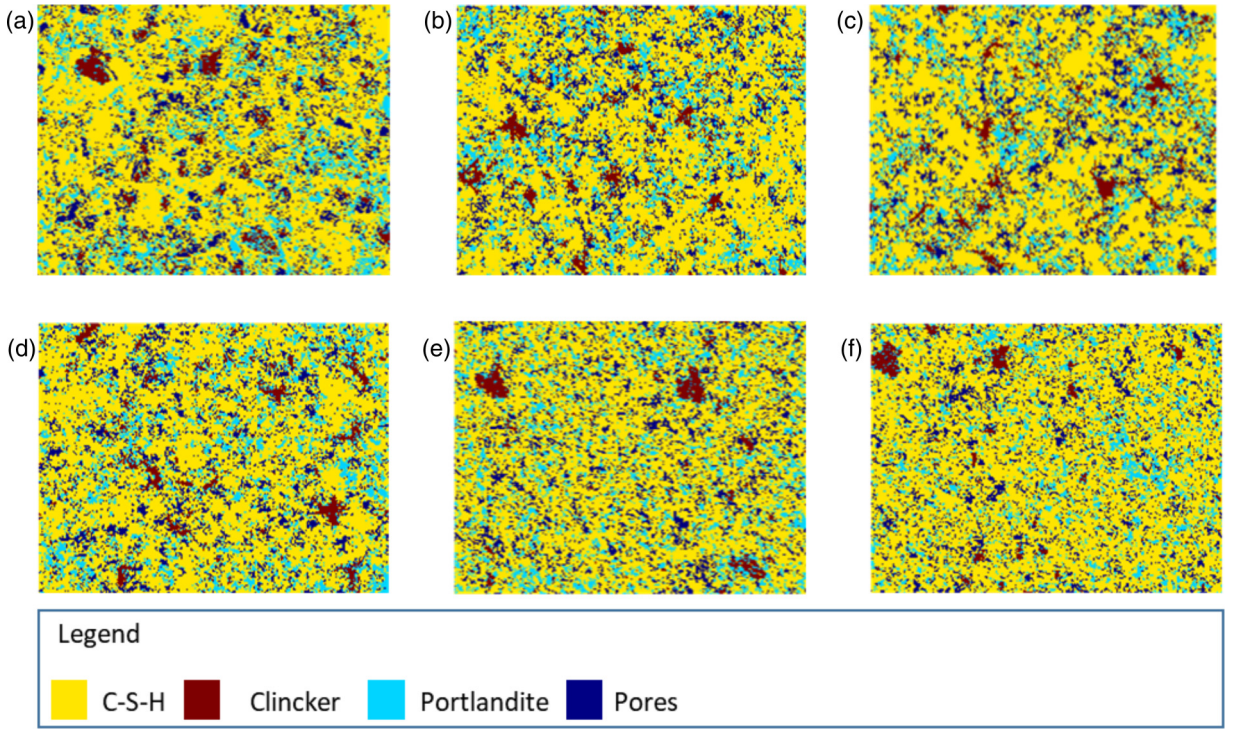

FIG. 9. Selected results for reconstruction of hardened cement paste. (a) Training image, (b) H-SA, (c) MH-SA, (d) NMH-SA, (e) DS reconstruction with perturbation distance 4, (f) DS reconstruction with perturbation distance 6. The perturbation distance is, thereby, defined as the average distance in pixels until which the probability constraints influence the phase choice by the DS algorithm.

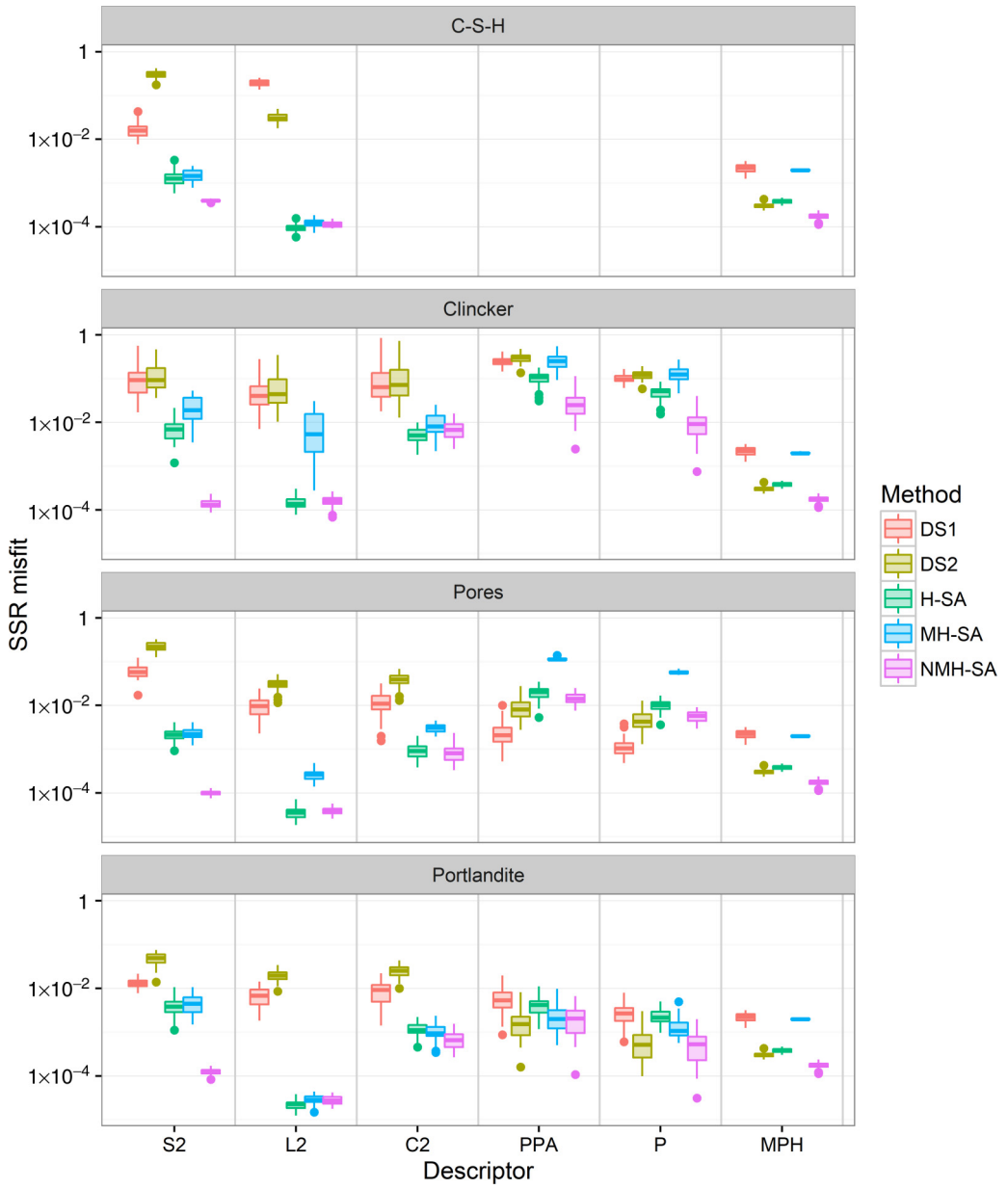

FIG. 10. Box plots of the descriptor misfits of the different cement paste realizations. The box plot is based on 50 realizations and represents the SSR misfit of the individual structural descriptors (S_2 = 2 point probability curve, L_2 = lineal path function, C_2 = cluster function, PPA = particle area, P = perimeter, MPH = multiple point histogram). The four individual subplots represent the different mineralogical phases. Within a single subplot, the different box plots are always grouped in stacks of five bars, next to each other, where each color represents a different methodology. One stack of five bars always represents one structural descriptor. In the C-S-H box plot C, PPA, and P are not represented, as the representation of those descriptors is physically meaningless for the matrix phase.

plots). In addition, while both approaches fit the L_2 function equally well for all phases, NMH-SA is more than a factor 10 better in the reproduction of S_2 for the clinker, portlandite, and C-S-H phases. This can also be deduced from Fig. 9 where NMH-SA is visually better in reproducing the cluster behavior of the larger portlandite particles, or in other words the fact that individual grains always appear as parts of bigger groups.

Overall, NMH-SA also outperforms the other SA variants with respect to PPA and P.

Figure 10 also shows there is nearly a constant ratio between that C_2 and L_2, for all approaches. This may indicate that C_2 only adds limited extra information to L_2 given that the two curves have almost identical shapes (Fig. 1).

Although the main motivation for our multiresolution approach was to decrease the computational effort, it can be seen in Fig. 11 that the computations take a longer time for NMH-SA than for MH-SA. Also, the difference in CPU time between NMH-SA and MH-SA becomes larger as the considered number of grid pixels increases. This is because the TI features a high number of particles, which are so small that they can be reconstructed at the finest resolution only. Additionally, the remerging of the phases also consumes time.

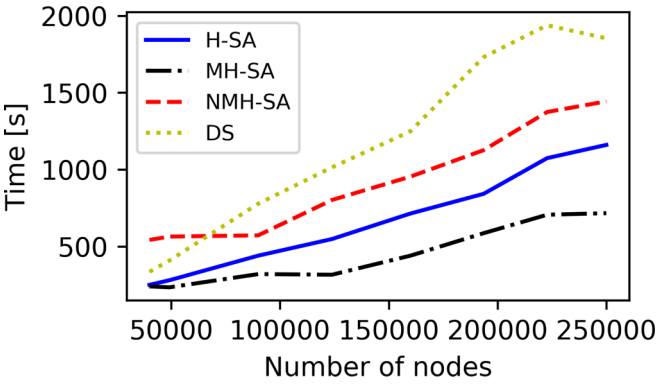

FIG. 11. Cement reconstructions: computational time as function of the number of nodes in the considered generation grid.

Hence, the only reason to use NMH-SA for simulating cement paste is the improved reconstruction quality.

For the reconstruction of the cement paste by DS, we used a neighborhood search tree of 50 pixels. Furthermore, it was necessary to use DS in combination with phase probability constraints as introduced by Mariethoz [37]. Without probability constraints, DS determines the filling of a pixel purely based on the pattern of already determined pixels, surrounding the one that is currently filled. With the addition of the probability constraints, the filling is constrained in such a way that the reconstruction has to fit the individual phase abundances additionally to the pattern. Without the probability constraints, DS was found to produce reconstructions with a too high proportion of the C-S-H phase (yellow-colored area in Fig. 9) and a near-zero fraction of the clinker phase (red-colored area in Fig. 9). While the addition of the probability constraints leads to a better fit of the MPH histogram, it also reduces the long-range connectivity in the DS-based realizations. This observation is more prominent for the smaller perturbation distance of 4 [DS1, Fig. 9(e)] compared to the one of 6 [DS2, Fig. 9(f)]. The perturbation distance is, thereby, defined as the average distance in pixels until which the probability constraints influence the phase choice by the DS algorithm.

For the considered TI, only the C-S-H (yellow matrix in Fig. 9) contains information about the long range connectivity. The problems of the DS1 reconstruction with the long-ranged connectivity is confirmed by the lineal path function of the C-S-H where the misfit of DS1 is one magnitude higher than DS2 and three magnitudes higher than the one of SA. Furthermore, DS1 is not capable of reconstructing the patch behavior of portlandite (turquoise particles in Fig. 9). The situation is already better for DS2 reconstruction but is still visually worse than for the three SA reconstructions [compare Figs. 9(f) and 9(b)–9(d)]. One could argue that comparing NMH-SA and DS using the L_2 descriptor is unfair as NMH-SA actually optimizes the L_2 property during reconstruction while DS does not. Nevertheless, the worst NMH-SA realization performs also better than the best DS realizations with respect to the MPH. Additionally, all SA approaches perform better in reconstructing C_2. The only criterion for which DS outperforms the NMH-SA approach is the P and PPA of the pores. Also, the shape of the clinker grains (red particles in Fig. 9) is much smoother in the realizations by DS1 and DS2 compared to SA approaches and this is closer to the patterns found in the TI. Nevertheless, it is seen that for the DS1 and DS2 reconstructions these clinker grains are almost verbatim copies of those found in the training image [compare Figs. 9(a) with 9(e) and 9(f)].

Regarding CPU time (Fig. 11), it is observed that for small generation domains, DS is nearly as fast as the MH-SA and H-SA variants and slightly faster than the full NMH-SA algorithm. For larger domain sizes, however, DS leads

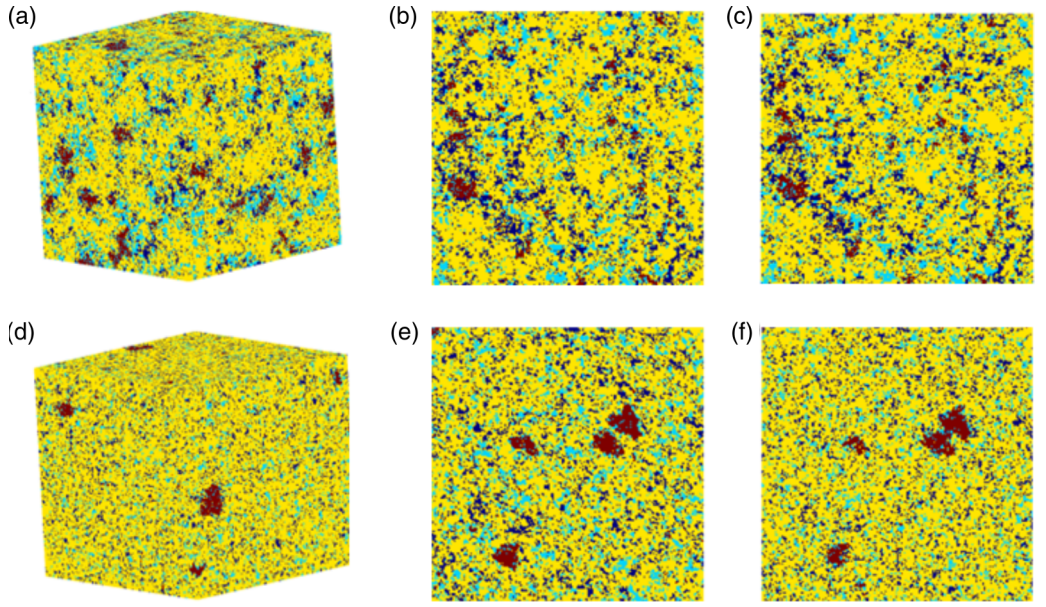

FIG. 12. 2D to 3D reconstruction. A) 3D reconstruction using SA, B) and C) are two adjacent slices from A in z direction. D) 3D reconstruction using DS, E and F) are two adjacent slices from D in z direction. Slice E was generated at the beginning of the reconstruction and hence featured almost no hard data, slice F in contrast was generated at end of the generation and featured a high amount of conditional data which decreased the reconstruction quality significantly.

to a stronger increase in computational time than every SA variant.

2. 2D to 3D reconstruction

Cement paste is generally in literature assumed to be an isotropic material [38,39]. Therefore, we use this material to compare the performance of our simulated annealing algorithm against DS for 2D to 3D reconstruction. Based on the results from the 2D experiments, only the best performing NMH-SA (for SA) and DS2 (for DS) codes are considered. During the NMH-SA optimization process, the S_2 and L_2 structural descriptors were calculated in nine different directions (along the x, y, z planes, and along the two main diagonals in each of these three planes). The size of the reconstruction domain is $200 \times 200 \times 200$. The same NMH-SA algorithm settings were used as for the 2D case except for the maximum number of iterations, which was changed from 10 000 to 1 000 000. With respect to DS, we used the s2Dcd approach introduced by Comunian *et al.* [40] which basically consists of sequential two-dimensional MPS simulations with conditioning data. In this approach, the 3D structure is generated by alternatively simulating perpendicular 2D slices, in which previously simulated pixels are used as conditioning data for the current simulation.

It is noted that the NMH-SA reconstruction [Fig. 12(a)] visually resembles much better the TI [Fig. 9(a)] than the DS2 reconstruction [Fig. 12(d)]. This visual assessment is confirmed by the S_2, L_2, C_2, and MPH misfit depicted in Fig. 13. While the NMH-SA reconstruction features the patch behavior of the portlandite particles (turquoise-colored areas in Fig. 12), the patch behavior is completely missed in the DS reconstruction. The problems encountered by DS to reconstruct the portlandite phase translates into 100 times larger S_2, L_2, and C_2 misfits compared to NMH-SA (Fig. 13). Overall, DS performs ten times worse than NMH-SA for S_2, L_2, and C_2 (Fig. 13). For PPA and P, in contrast, no significant differences between the two methodologies are observed (Fig. 13).

Comparing adjacent slices [Figs. 12(b) and 12(c) for NMH-SA and Figs. 12(e) and 12(f) for DS-s2Dcd] in the generated 3D volume reveals that the NMH-SA simulation also has a higher consistency. Furthermore, while comparing the misfit for the cluster function and the MPH error of the 2D to 3D reconstructions with the 2D reconstruction, no significant differences can be determined (compare Figs. 10 and 13). This means that generating a 3D reconstruction does not decrease the image quality at least for those two descriptors. For P and PPA, the 3D reconstruction is of lowest quality, as it features too many small particles, or in other words isolated pixels (Figs. 12 and 13).

As opposed to NMH-SA, DS-s2Dcd has problems to maintain the consistency of the portlandite grains and pores over consecutive slices [Figs. 12(e) and 12(f)]. The slice shown in Fig. 12(e) was generated at the beginning of the reconstruction and features almost no conditioning data. Therefore, its quality is visually the same as the one generated by the DS2 approach in the 2D case [see Fig. 9(f)] as both DS simulations rely on the same parameters. In contrast, the slice displayed in Fig. 12(f) was produced almost at the end of the simulation run and hence contains a lot of conditioning data

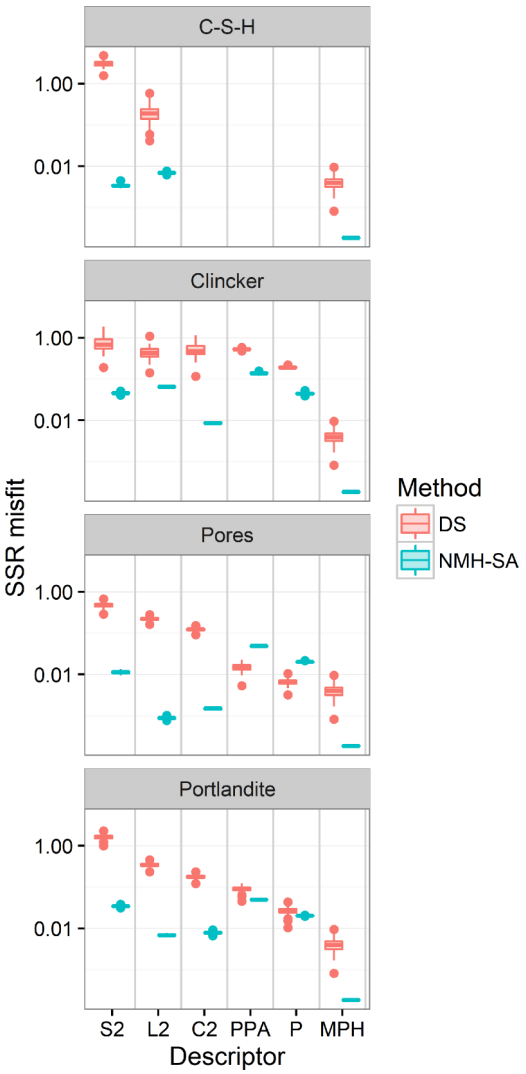

FIG. 13. Box plots of the descriptor misfits of the different cement paste 3D realizations. The box plot is based on 50 realizations and represents the SSR misfit of the individual structural descriptors (S_2 = 2 point probability curve, L_2 = lineal path function, C_2 = cluster function, PPA = particle area, P = perimeter, MPH = multiple point). The four individual subplots represent the different mineralogical phases. Within a single subplot, the different box plots are always grouped in stacks of two bars, next to each other, where each color represents a different methodology. One stack of two bars always represents one structural descriptor. In the C-S-H box plot C, PPA, and P are not represented, as the representation of those descriptors is physically meaningless for the matrix phase.

to honor. The inclusion of these conditioning data strongly reduces the quality of the DS-s2Dcd reconstruction. Note that lower reconstruction quality may also be partly due to shortcomings of the approach by Comunian *et al.* [40] and better results might possibly be obtained by using other 2D to 3D reconstruction methodologies.

Regarding computational effort, both NMH-SA and DS incur a significant computational cost. For the considered $200 \times 200 \times 200$ cubic generation domain DS-s2Dcd needs 13 h to produce a single reconstruction. The NMH-SA algorithm is faster but still needs almost 5 h to do the same task.

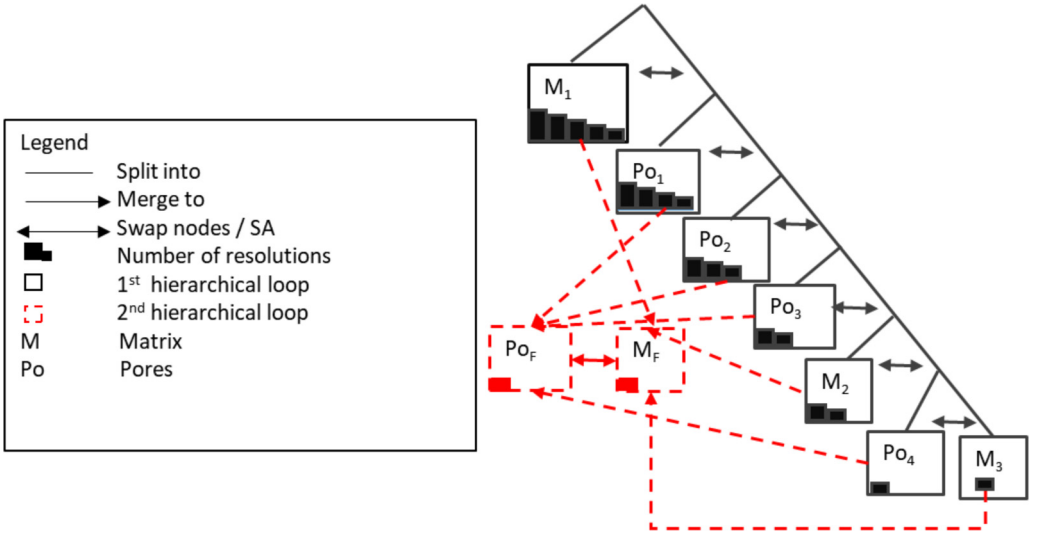

FIG. 14. Reconstruction scheme of the Boom Clay reconstruction.

B. Boom Clay

The Ti used for this case study originates from [8], but was cropped and artificially coarsened to have a resolution of 0.8 micrometer and a size of 400 × 400 pixels. More details about how the image was acquired, processed, and segmented can be found in Hemes *et al.* [8]. For the reconstruction, we start again by comparing the three SA approaches against each other before including DS in the comparison. For each approach, 50 realizations were generated. For H-SA and MH-SA, the TI was treated as a single-phase image and the structural descriptors were calculated for lag distances up to 100 pixels. For NMH-SA, the matrix was subdivided into three different classes (larger quartz grains, smaller quartz grains, and clay matrix) and the pores space was split into four different subclasses based on sizes. At the end of the NMH-SA simulation run, the reconstruction was back transformed into a binary image (see the NMH-SA simulation steps depicted in Fig. 14).

To make a detailed quantitative quality assessment, we transformed the binary realizations derived by each approach into multiphase realizations using five different classes [Figs. 15(b), 15(d) 15(f), 15(h), 15(j), and 15(l)]. The first class consists of pores smaller than 20 pixels, the second class includes pores with a size between 20 and 100, and the third class features pores larger than 100 pixels. The matrix was split into clay particles and clasts. This was done using an automatic thresholding procedure. Connected matrix components with a minimum distance of six pixels to a pore pixel and which feature at least one pixel with a distance to a pore pixel larger than 9.5 pixels are treated as quartz; the rest are considered as clay particles. Pixels initially defined as clay that are included within quartz are also labeled as quartz.

The reason for this transformation from a binary into a multiphase image for quantitative quality assessment is that the three different SA reconstructions are found to fit the binary structural descriptors almost equally well although clear differences in quality are visually observed [see Figs. 15(a), 15(c) 15(e), 15(g), 15(i), and 15(k)]. While splitting the binary reconstructions into multicategorical ones allows for a better quality evolution for S_2, L_2, and C_2, it decreases the information content of the PPA and P descriptors. This is because using five subclasses makes the within-class PPA and P values more similar. Therefore, we used the so-called mean relative histogram difference (HIS) per phase as an additional structural descriptor.

The H-SA approach appears to fail in reconstructing the large quartz grains (or what is the rough corresponder of quartz) or in other words the long-range connectivity [Fig. 15(c)]. Indeed, the MH-SA reconstruction contains almost no quartz at all [green particles in Fig. 15(d)]. This happens even though the H-SA reconstruction fits the L_2 descriptor as well as the MH-SA and NMH-SA reconstructions in the binary case (not shown). The H-SA reconstruction features the long segments without any black pixels, but the S_2 criterion does not encode sufficient information to group those segments together.

The long-range connectivity of the matrix is better reproduced by MH-SA [Figs. 15(e) and 15(f)]. The MH-SA reconstruction features more and larger green clasts than the H-SA one. Compared to that associated with H-SA, the HIS misfit decreases by almost a factor of 5 for MH-SA for quartz (Fig. 16). Additionally the reconstruction quality with respect to S_2, L_2, and C_2 is increased by a factor 5 and each realization of MH-SA is better than the best reconstruction generated by H-SA for the same subclass (Fig. 16). While MH-SA improves the reconstruction quality of the matrix components compared to H-SA, the reconstruction quality of the pores decreases. MMSA performs worse than MH-SA for all pore subclasses and considered descriptors except PPA and P (Fig. 16).

The visually best SA generations are again those produced by NMH-SA [Figs. 15(g) and 15(h)]. For instance Fig. 15(h) has the largest quartz (green) particles. This visual observation is confirmed by the HIS, S_2, L_2, C_2, and MPH descriptors for which the misfit is always the smallest for NMH-SA (Fig. 16). The advantages of NMH-SA become even clearer

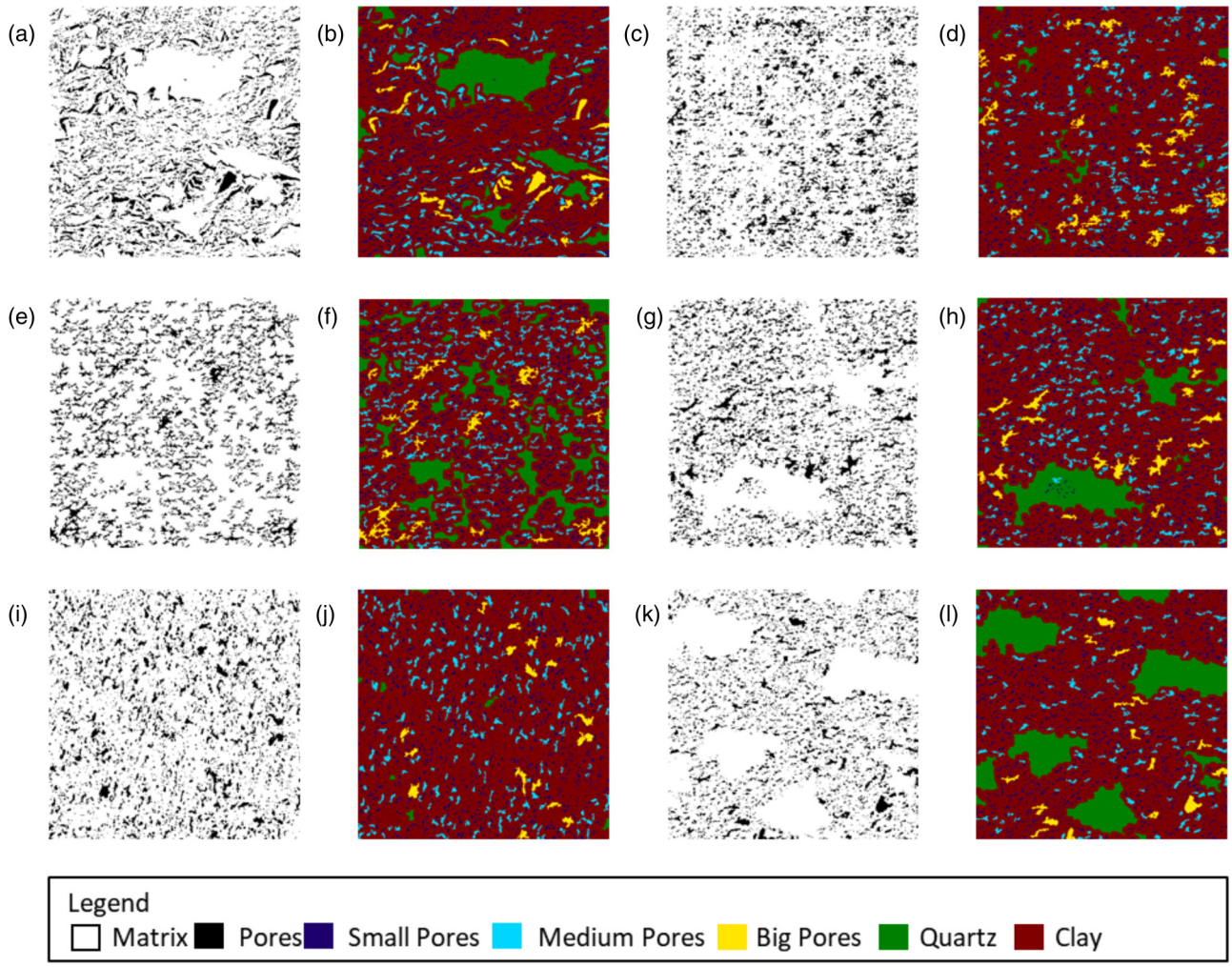

FIG. 15. Boom Clay training image and reconstructions (rec.). (a) TI, (b) relabeled TI, (c), (d) H-SA rec., (e), (f) MH-SA rec., (g), (h) NMH-SA rec., (i), (j) DS rec using binary TI, (k), (i) DS rec. using multiphase TI.

when looking at the quartz and clay phases' descriptor misfit (Fig. 15). Here the L_2 descriptor misfit for both phases is more than one order of magnitude smaller than for their H-SA and MH-SA counterparts.

Although outperforming H-SA and MH-SA, NMH-SA nevertheless does not capture well the bedding behavior of the Boom Clay. The TI clearly has crenulating behavior of the clay fabric, with the pores following the orientation of the quartz grains. This pattern is not found in the reconstruction [Figs. 15(g) and 15(h)], although some fabric can be identified.

Figure 17 depicts the computational time of the individual realizations as function of the number of nodes. Among the SA variants, MH-SA is the fastest. This is because the Boom Clay TI features a high amount of larger particles, which are efficiently generated on the coarse resolution. Yet as explained above MH-SA leads to a much poorer reconstruction quality than H-SA and NMH-SA. The NMH-SA is almost as fast as MH-SA, while providing the largest reconstruction quality.

Similarly to the SA, DS cannot reconstruct the Boom Clay as a binary image [Figs. 15(i) and 15(j)]. In order for DS to deal with the larger pores, we found it necessary to limit the number of closest pixels enclosed in each calculation step to a maximum of 20. However, by doing so the reconstruction almost does not include any quartz grains [Fig. 15(e)]. Moreover, DS BIN (DS using a binary TI) is found to perform even worse than MH-SA (Fig. 15).

Therefore, we used Fig. 15(b) as a TI for a multiphase reconstructions (DSMP). For consistency with the other approaches, the reconstructions were also first transformed into a binary image and afterward relabeled. Using the DS-MP approach allows for DS to generate the big quartz grains in [Figs. 15(k) and 15(l)]. As of quality, the best DS reconstructions were obtained using a neighborhood of 20 points. Using a larger value led to a lower reconstruction quality (not shown).

DSMP and NMH-SA approaches lead to a similar mean MPH error, but DS shows a broader error distribution (Fig. 16). Furthermore, DSMP and NMH-SA induce similar descriptor's misfit values for the different pore phases in L_2, C_2, PPA, and P, which means that they reconstruct equally well the shape of the individual particles. For the large particles DS is even almost one order better in reproducing P and PPA. Nevertheless, NMH-SA is better in reconstructing

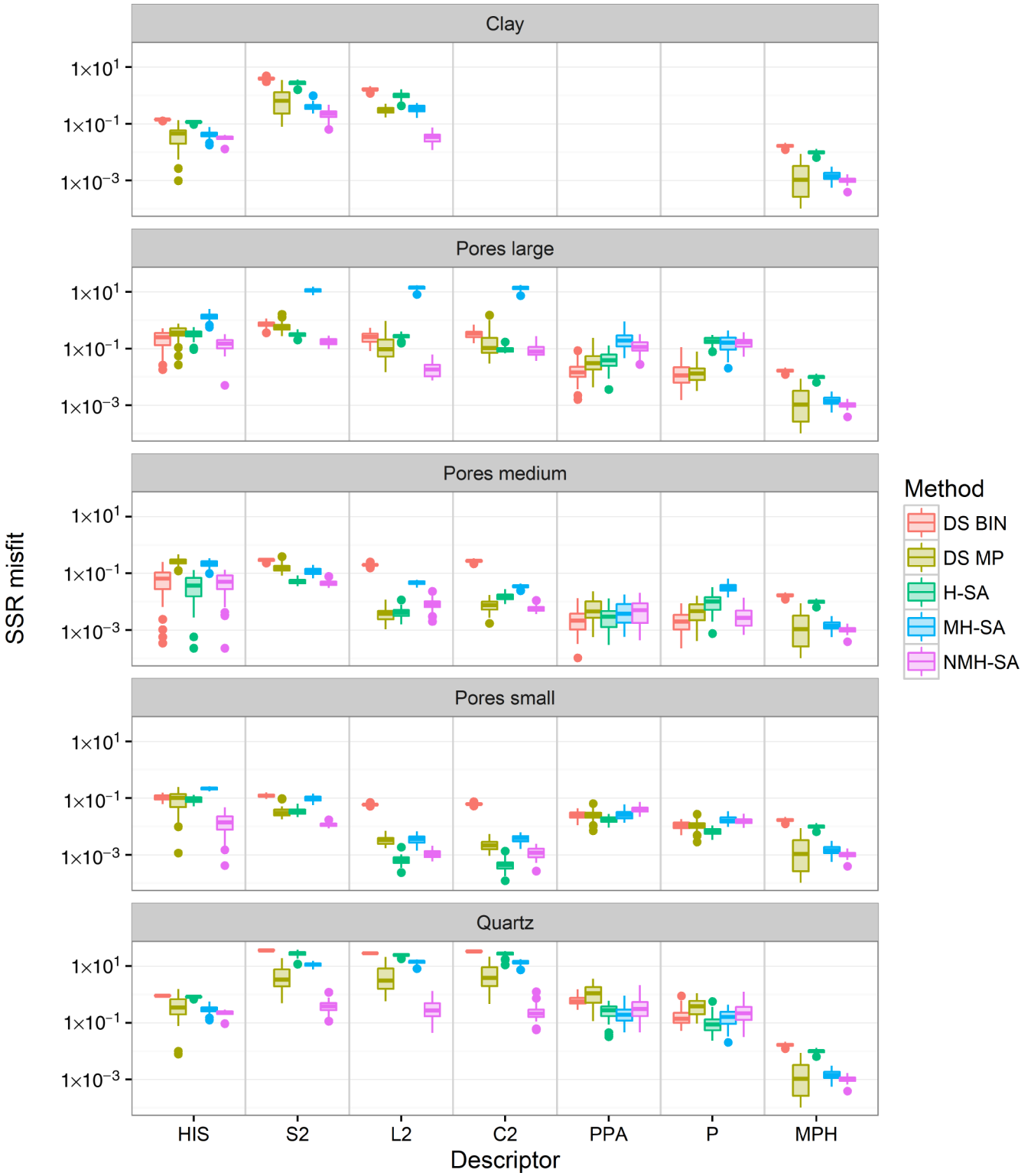

FIG. 16. Box plots of the descriptor misfits of the different Boom Clay realizations. The box plot is based on 50 realizations and represents the SSR misfit of the individual structural descriptors (HIS = mean relative histogram difference, $S_2 = 2$ point probability curve, L_2 = lineal path function, $C - 2$ = cluster function, PPA = particle area, P = perimeter, MPH = multiple point histogram). The five facets represent the different phases. Within a single subplot, the different box plots are always grouped in stacks of five bars, next to each other, where each color represents a different methodology applied. In the Clay box plot C, P, and PPA are not represented, as the representation of those descriptors is physically meaningless for the matrix phase.

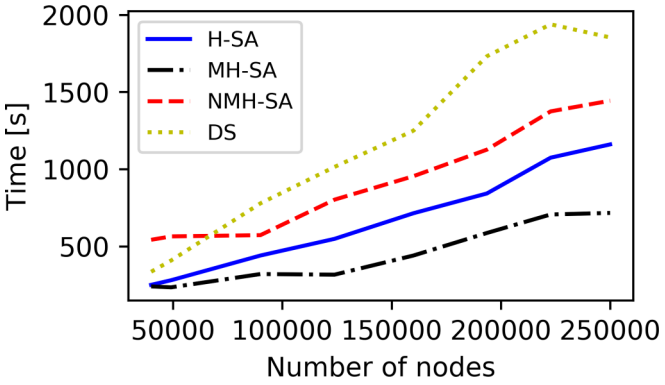

FIG. 17. Boom Clay reconstructions computational time as function of the number of nodes.

the orientation of the individual pore particles to each other. This conclusion can be drawn from the decreased SSR misfit of. Indeed, NMH-SA leads to smaller S2 and HIS misfits. This difference can also been noticed visually. In Fig. 15(l) it is seen that the DSMP reconstruction has too many small pores and too few light blue and yellow particles. While the difference in descriptor values for the pores are minor, NMH-SA performs about ten times better than DS for the quartz particles. (Fig. 17). Additionally, DSMP has the tendency to generate too many larger quartz grains Fig. 15(l). In addition, compared to NMH-SA, DS visually provides a worse reproduction of the bedding related pattern observed in the TI [compare Figs. 15(g) and 15(k)]. The strength of DS however is its computational efficiency where it is three times faster than DS (Fig. 17).

IV. DISCUSSION

In this work we propose an SA algorithm termed NMH-SA for nested multiresolution hierarchical simulated annealing. Our NMH-SA algorithm is able to handle the multiphase case and the multiresolution approach. To our knowledge the only other study using multiphase simulated annealing is the one of of Jiao and Chawala [41]. Compared to their method our algorithm calculates additionally the descriptor values in the diagonal directions which clearly improves reconstruction quality. Additionally they did not use a hierarchical and multiresolution approach which increased their computational cost significantly. Moreover, NMH-SA also improves upon the state of the art for the binary case. For the considered two case studies, cement paste and Boom clay, NMH-SA is found to substantially outperform state-of-the-art SA (Figs. 9, 10, 12, 13, 15, and 16). This much higher reconstruction quality compared to other SA approaches comes with a moderate 1.5–2 times increase in CPU time per realization (Figs. 11 and 17) for the presented 2D case studies. We would like to stress that if for the Boom Clay case study the traditional approach is used in combination with subphase splitting without using the multiresolution approach, NMH-SA becomes seven times faster.

Among 50 realizations by each method, the best single Boom Clay reconstruction is generated by DS (Fig. 16). Yet the reconstruction quality statistics are overall larger for the 50 NMH-SA realizations than for the 50 DS realizations (Fig. 16). For the computed MPH criterion, the MPS distributions derived by NMH-SA and DS show the same median value but the NMH-SA distribution is much narrower. We believe such a small variation in quality between realizations is a desired property for a reconstruction algorithm. The larger spread in the MPH distribution by the DS for the Boom Clay is a result of the nonstationary TI. Indeed, in this work we used a small template of 20 pixels. This leads to a relatively good simulation of the different pore fractions [Figs. 15(k)–15(l)]. However the corresponding quartz grains tend to be too big and the long-range connectivity is relatively badly reproduced (see the spacing between the individual quartz grains and the large error for S_2 and C_2 in Fig. 16). To improve the long-range connectivity it would therefore be necessary to use a larger template size. The larger template size would however lead to the generation of verbatim copies, as only few patterns in the TI would match the search pattern of the reconstruction. To overcome this problem it would, hence, be necessary to use a much larger TI in combination with a larger neighborhood template. This larger Ti, however might not always be available and it would also increase the computational time of DS significantly.

Consequently, it can be stated that our algorithm is more attractive than DS, in case of TIs which feature inclusions of various sizes and frequencies, as the target statistics can be set on a phase, or even sub- or superphase basis.

As any other algorithm based on SA, our proposed NMH-SA has the limitation that it i.e., the algorithm, can only reconstruct patterns which can be well described and efficiently quantified by a set of (easy to calculate) structural descriptors. For example, none of the structural descriptors used in this study is capable of capturing the crenulating bedding behavior of the Boom Clay. That is why even if it is the only tested algorithm that generates realizations which feature some fabric, NMH-SA nevertheless poorly reproduces the bedding patterns found in the TI (Fig. 15).

The following extensions could improve NMH-SA:

(i) The simulation time could be further decreased by distributing the computational burden. The current version of NMH-SA starts with the simulation of a new phase once the previous phase has been simulated from the coarsest to the finest resolution. In a parallel implementation, the simulation of the newest phase could already start with the coarse resolution as soon as the previous phase has been simulated on this same resolution. In doing so, the computational cost could be strongly decreased, certainly in the case of a large cascade of calculations, when many different (sub - or super)phases are involved.

(ii) The problem that the 3D reconstruction does not reproduce the particle size and shape as well as in two dimensions (Figs. 10 and 13), which is related to a too high number of small isolated pixels, might be resolved using the pixel selection rule suggested by Tang [42]. While the current NMH-SA randomly selects any pixel with nonmatching neighboring pixels, the algorithm of Tang [42] preferably selects pixels, with high numbers of nonmatching, and consequently tends to generate realizations without isolated pixels.

V. CONCLUSION

We propose a SA algorithm, termed NMH-SA, that overcomes the typical limitations of SA for microstructure generation. NMH-SA is shown to outperform both state-of-the-art SA and the multiple-point statistics DS method for a multiphase cement paste image and a binary Boom Clay image. These two case studies are characterized by strong differences in size and frequencies between the different phases. The advantages of NMH-SA are found to be most prominent for the 2D to 3D reconstruction of cement paste. For Boom Clay, our NMH-SA approach has difficulties in reproducing the quite complex bedding patterns present in the training image.

ACKNOWLEDGMENTS

This research has received partial funding from the European Union European Atomic Energy Community (Euratom) Horizon 2020 Programme (NFRP-2014/2015) under Grant Agreement No. 662147-CEBAMA. The code used in this study is available [43].

[1] N. Seigneur, A. Dauzeres, M. Voutilainen, V. Detilleux, P. Labeau, and A. Dubus, J. Porous Media **20**, 29 (2017).
[2] S. Torquato, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties* (Springer Science & Business Media, New York, 2013), Vol. 16.
[3] M. J. Blunt, B. Bijeljic, H. Dong, O. Gharbi, S. Iglauer, P. Mostaghimi, A. Paluszny, and C. Pentland, Adv. Water Resour. **51**, 197 (2013).
[4] H. Okabe and M. J. Blunt, Phys. Rev. E **70**, 066135 (2004).
[5] H. Izadi, M. Baniassadi, A. Hasanabadi, B. Mehrgini, H. Memarian, H. Soltanian-Zadeh, and K. Abrinia, J. Pet. Sci. Eng. **149**, 789 (2017).
[6] G. C. Conroy and M. W. Vannier, Science **226**, 456 (1984).
[7] T. M. Buzug, *Computed Tomography: From Photon Statistics to Modern Cone-Beam CT* (Springer Science & Business Media, New York, 2008).
[8] S. Hemes, G. Desbois, J. L. Urai, B. Schröppel, and J.-O. Schwarz, Microporous Mesoporous Mater. **208**, 1 (2015).
[9] R. F. Egerton *et al.*, *Physical Principles of Electron Microscopy* (Springer, New York, 2005).
[10] L. A. Giannuzzi *et al.*, *Introduction to Focused Ion Beams: Instrumentation, Theory, Techniques and Practice* (Springer Science & Business Media, New York, 2004).
[11] P.-E. Øren and S. Bakke, Transp. Porous Media **46**, 311 (2002).
[12] K. Van Breugel, Cement Concrete Res. **25**, 522 (1995).
[13] D. P. Bentz, P. V. Coveney, E. J. Garboczi, M. F. Kleyn, and P. E. Stutzman, Modell. Simul. Mater. Sci. Eng. **2**, 783 (1994).
[14] K. Ankit, J. L. Urai, and B. Nestler, J. Geophys. Res.: Solid Earth **120**, 3096 (2015).
[15] N. Prajapati, M. Selzer, B. Nestler, B. Busch, C. Hilgers, and K. Ankit, J. Geophys. Res.: Solid Earth **123**, 6378 (2018).
[16] S. Torquato and B. Lu, Phys. Rev. E **47**, 2950 (1993).
[17] C. L. Y. Yeong and S. Torquato, Phys. Rev. E **57**, 495 (1998).
[18] K. J. Kearfott and S. E. Hill, IEEE Trans. Medical Imaging **9**, 128 (1990).
[19] M. V. Karsanina, K. M. Gerke, E. B. Skvortsova, and D. Mallants, PLoS One **10**, e0126515 (2015).
[20] L. M. Pant, S. K. Mitra, and M. Secanell, Phys. Rev. E **92**, 063303 (2015).
[21] F. B. Guardiano and R. M. Srivastava, in *Geostatistics Troia'92* (Springer, New York, 1993), pp. 133–144.
[22] S. Strebelle, Math. Geol. **34**, 1 (2002).
[23] G. Mariethoz, P. Renard, and J. Straubhaar, Water Resour. Res. **46**, W11536 (2010).
[24] P. Tahmasebi, M. Sahimi, and J. Caers, Comput. Geosci. **67**, 75 (2014).
[25] L. Mosser, O. Dubrule, and M. J. Blunt, Transp. Porous Media **125**, 81 (2018).
[26] E. Laloy, R. Hérault, J. Lee, D. Jacques, and N. Linde, Adv. Water Resour. **110**, 387 (2017).
[27] E. Laloy, R. Hérault, D. Jacques, and N. Linde, Water Resour. Res. **54**, 381 (2018).
[28] J. Feng, Q. Teng, X. He, and X. Wu, Acta Mater. **159**, 296 (2018).
[29] R. Cang, Y. Xu, S. Chen, Y. Liu, Y. Jiao, and M. Y. Ren, J. Mech. Des. **139**, 071404 (2017).
[30] Q. T. Phung, Ph.D. thesis, Ghent University, 2015.
[31] T. C. Coburn, Technometrics **42**, 437 (2000).
[32] X. Tan, P. Tahmasebi, and J. Caers, Math. Geosci. **46**, 149 (2014).
[33] W. Ben-Ameur, Comput. Optim. Appl. **29**, 369 (2004).
[34] W. R. Campaigne and P. W. Fieguth, IEEE Trans. Image Process. **22**, 1486 (2013).
[35] M. V. Karsanina and K. M. Gerke, Phys. Rev. Lett. **121**, 265501 (2018).
[36] K. M. Gerke, M. V. Karsanina, and D. Mallants, Sci. Rep. **5**, 15880 (2015).
[37] G. Mariethoz, J. Straubhaar, P. Renard, T. Chugunova, and P. Biver, Environ. Modell. Softw. **72**, 184 (2015).
[38] C. Hua, P. Acker, and A. Ehrlacher, Cement Concrete Res. **25**, 1457 (1995).
[39] Y. Xi and H. M. Jennings, Mater. Struct. **30**, 329 (1997).
[40] A. Comunian, P. Renard, and J. Straubhaar, Comput. Geosci. **40**, 49 (2012).
[41] Y. Jiao and N. Chawla, J. Appl. Phys. **115**, 093511 (2014).
[42] T. Tang, Q. Teng, X. He, and D. Luo, J. Microsc. **234**, 262 (2009).
[43] https://github.com/LaLemmens.