Cement and Concrete Research 39 (2009) 266–274

Contents lists available at ScienceDirect

Cement and Concrete Research

journal homepage: http://ees.elsevier.com/CEMCON/default.asp

μic: A new platform for modelling the hydration of cements

Shashank Bishnoi *, Karen L. Scrivener

Laboratory of Construction Materials, Ecole Polytechnique Fédérale de Lausanne EPFL-STI-IMX-LMC, Station 12, 1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 31 July 2008
Accepted 9 December 2008

Keywords:
Microstructure
Modelling
Platform
Cement
Hydration

ABSTRACT

A new modelling platform, called μic has been developed to model the microstructural evolution of hydrating cement paste. The platform uses the vector approach and can be used for modelling particulate reactions including the hydration of many different cementitious systems involving millions of particles. In this paper, the ideas behind the development of μic and its main features are presented. The complexity of cement hydration and the gaps in our current understanding of cement played an important role in its design, so the platform has the primary objective of aiding, rather than replacing experiments. The platform is highly customisable as users can define materials, particles and reactions and choose or create external plugins to define models of microstructural development. The platform can be used to test the validity of hypotheses by easily formulating them as input and comparing simulations with experimental results. This paper presents the design of μic and examples that demonstrate the important features of its performance and design.

1. Introduction

The tremendous advances in computing technology in the last few decades have provided a great impetus to computer based scientific research. Increased speeds and availability of computers have made complex numerical simulations of various materials and processes possible. Computer based numerical models also provide a methodology to study the mechanisms underlying the performance of a material by allowing comparison of model derived values against those experimentally observed. The use of numerical modelling is arguably best justified for complex composite materials like cement paste, where analytical models entail over-simplification of the problem and may overlook some of the features that could significantly affect the result.

A new modelling platform called μic (pronounced Mike, available for free in open source [1]) has been developed to enable simulations of the complex processes and interactions involved in the hydration of cement. Instead of providing rules that are hard-wired in the programmes, the platform allows its users to define reactions and customise models of reaction mechanisms. The microstructural information generated by the simulations can be used for subsequent computations and for analysis of other properties, e.g. mechanical and transport properties of the reacted system. It is hoped that by enabling development and testing of new models related to cement hydration, μic will improve our understanding of cement hydration and, in the long term, provide a platform suited to predicting properties of hydrating cement pastes.

This paper presents the design of μic, including a discussion on the choice of approaches and simplifications used in μic. Examples are presented to demonstrate the importance of the large number of particles that can be handled by this platform and how the flexibility of the platform can be used to numerically study different microstructural growth processes.

2. Currently available microstructural models

Several models that can simulate, with certain assumptions, the evolution of cement microstructure during hydration are available. All models are by definition a simplified representation of reality and are valid only under the range of assumptions and simplifications made during the development of the model. Simplifications do not necessarily reduce the value of the models as long as they are made keeping the end objectives of the model in mind and the limitations of the approach are understood by the user. As every assumption or simplification can be considered to be a different approach to solving the problem, various approaches to modelling cement hydration have emerged.

Based on the simplifications required for an efficient and practically useful representation of the cement microstructure, the discretization approach and the vector approach, are the two main approaches which have been used in microstructural models of cement hydration.

2.1. Discretization approach

In the discretization approach, similar to the finite element method, the entire volume of the microstructure is filled with smaller elements, placed with their faces in contact. While each of these elements can be individually assigned shape, size and material

* Corresponding author.
E-mail addresses: shashank.bishnoi@epfl.ch (S. Bishnoi), karen.scrivener@epfl.ch (K.L. Scrivener).

0008-8846/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cemconres.2008.12.002

properties, it is common to divide the computational volume into a uniform grid of cubic volume-pixels (voxels). CEMHYD3D [2,3] is one of the best-known and most widely-used models for cement that uses such a voxel approach. While this approach allows easy representation of spatial distributions of different phases and different shapes, only particles of sizes larger than the mesh-size can be modelled. Nevertheless, due to the simple shape and the uniform size of the voxels, such an assembly can be computationally more effective. Compared to other mesh types, an assembly of cubes can also be convenient for a direct analysis of the microstructure [4] or the application of methods such as the finite-element method on the microstructure [5].

To simulate the evolution of cement microstructure, CEMHYD3D uses an approach similar to the cellular automata (CA) approach, which was originally developed to study biological self-replication [6]. In the implementation used in CEMHYD3D, the diffusion of species is simulated using random-walks of voxels, which lead to collisions and reactions. In the early ages of hydration, cement particles break up leading to an increase in the roughness of cement particles and consequently an increase in the rate of hydration per step of the simulation. The accumulation of hydrates around the cement particles, and the later consumption of materials, leads to a reduction in the probability of collision events reducing the reaction rates. The steps in the simulation are then related to real time of hydration using calibration with experiments and the evolution of the microstructure can be studied.

Although the size and number of voxels in the simulations are adjustable, the total number of voxels in these simulations is in the order of one to ten million for most computations. It has been shown that the calculated properties can depend on the resolution used and that the optimal resolution required may depend on the application [7]. For example, at least one billion voxels will be required in order to represent particles of 0.1 µm size in a computational volume of 100 µm size and at least one trillion voxels will be required to calculate connectivity including the effect of the finer capillary pores down to 0.01 µm size.

2.2. The vector approach

The vector approach is relatively resolution-free and does not require the division of the features in the microstructures into smaller elements. Similar to the discrete element approach [8], in this approach, information concerning the position, orientation and size of individual features is stored. This approach enables the storage and processing of different features independent of their relative sizes, without imposing a resolution limit on the information. The vector approach has been used in several previous hydration models [9–12]. Johnson and Jennings [9,13] first presented the idea of modelling cement hydration as nucleation and growth of spherical particles in three-dimensional space and wrote a program to achieve this goal in late 1980s. Although the model produced promising results it was not developed further. Hymostruc [10] and the Integrated Particle Kinetics Model [11,14–16] (IPKM) are two of the later models using this approach. Both these models essentially simulate cement hydration as growth of spherical particles with overlaps.

In Hymostruc hydration is simulated only by the growth of new layers of product on the cement particles. This limitation in Hymostruc prevents the representation of products such as portlandite that tend to form new clusters in the pores, rather than depositing on the cement particles [17]. This setup also leads to a more heterogeneous distribution of the products leading to several side-effects such as the deprecolation of the porosity occurring only at 3% — a much lower value than expected from experiments [18]. This limitation in the Hymostruc model, therefore, limits the possibilities in microstructural development and prevents an accurate representation of the microstructure.

In IPKM, the evolution of alite hydration is modelled by the deposition of C–S–H on the surface of C_3S particles and the growth of new portlandite particles in the pores. One of the unique features of the approach used in IPKM is that it is possible to account for the interactions between the particles and the evolution of individual particles can be calculated based on their neighbourhood. This approach was considered better adapted for modelling cement hydration as it enables the modelling of the evolution of each particle independently, without the necessity of approximations using statistically calculated correction factors. For this reason, in µic, the vector approach was implemented in a way similar to that used in IPKM.

However, due to the explicit calculation of all possible interactions, the simulations were slow in IPKM, and only simulations on a relatively small number of particles from reduced particle size distributions were possible. The number of particles modelled in the simulations in the model was limited to around 20,000, although the actual number of particles in a computational volume of typical sizes could be in millions. This limitation restricted the usability of this model as the finer cement particles could not be modelled and the main advantage of the vector approach, that features of all sizes can be modelled, was lost.

The complexity of the calculation of interactions is increased due to the periodic boundary conditions assumed in the simulations. In the periodic boundary conditions, the simulated computational volume is assumed to be a small part of the actual structure, which can be reconstructed by repeatedly placing adjacent copies of the computational volume. Boundary effects are removed by allowing particles to cross any of the boundaries and re-enter the volume from the opposite side, possibly overlapping with particles present on the opposite side. While the periodic boundary conditions can significantly reduce the total size of the computational volume, the calculations are more complex compared to computational volumes of the same size without periodic boundary conditions as the crossing over of particles over boundaries has to be accounted for explicitly.

Since the vector approach preserves information about features that scale over several orders of magnitude, it is considered to be more appropriate for application to multi-scale systems like cement, and was, therefore, chosen for the development of µic. In order to overcome the possibly high computational costs associated with the vector approach, µic was built with support libraries that enable quick calculation of interactions, considerably improving the performance of the vector approach. These libraries are discussed in more detail in the next section.

One of the important effects of the high computational costs of the vector approach is that most models using this approach are limited to spherical particles. This is because, compared to other shapes, spheres require a small amount of information to be stored and the calculation of the distance between two spheres is fast. While this limitation could alter the geometry of the microstructure, the need to keep computational costs for systems with large number of particles within practically acceptable limits necessitates the approximation of particles by spheres. Depending on the application, different properties, such as volume, surface area, largest dimension, etc., may be used for the conversion of other shapes to spheres. Assemblies of many adjacent or overlapping spheres can also be used to obtain better approximations for the shapes, as will be seen in the example presented in Section 5.

3. Design of µic

Two main issues were addressed in the design of µic. The first issue addressed was the relatively slower speed associated with the vector approach. It was aimed to speed up the simulations so as to allow modelling of the millions of particles that are typically present in cement samples of 100 µm in size. The second issue addressed is related to the large amount of hard-wired information and assumptions in most models. Since hydration is a complex process, many aspects of which are still not well understood, μic was designed to allow its users the flexibility to define, modify and customise most aspects of the simulations, including the models for the mechanisms.

μic has been developed using object oriented programming in Java. Although the speed of programmes using Java is often questioned, Java was chosen as it was considered to provide the best balance between ease of learning and efficiency of programming. Java encourages clean coding practices, making the programmes easier to read and edit. Java is also more accessible to users as most Java compilers and development environments are available for free in open source and being platform independent, the compiled programmes are not restricted to a particular operating system or platform. Also, Java provides the capability to dynamically load classes at run-time, which, as explained later, allowed the development of some important features in μic.

3.1. Improving performance using grid subdivisions

The calculation of interaction between particles is the most computationally expensive part of the calculations when the vector approach is used. For example, using a naïve approach, the identification of particles that overlap with a particular particle would require the calculation of distance of all particles from the particle, after accounting for the periodic boundary conditions. Using this method the time needed for the calculation of all interactions in the volume would be proportional to the square of the number of particles.

In order to make the calculation of the interactions faster, the volume was divided into smaller cubes using a regular grid that is laid over the vector information, as shown in Fig. 1. Similar methods are routinely used in computer graphics and gaming [19,20]. All cubes contain a list of particles that lie in the cube and the search for neighbours of a particle can be limited to the particles in the lists contained in the cubes in which the reference particle is present. For example, in Fig. 1, when looking for overlaps with particle 1, particle 3 can be ignored as it does not share a grid cube with particle 1. μic has been built with libraries that can be invoked to create, update or access the information in the grid cubes. The size of the grid has to be optimised according to the size of the particles, as too few divisions would create large zones that have to be searched and too many divisions would require running through information in a large number of zones. Although, by default, μic sets the size of the grid elements to the median particle diameter, the user can customise this value.

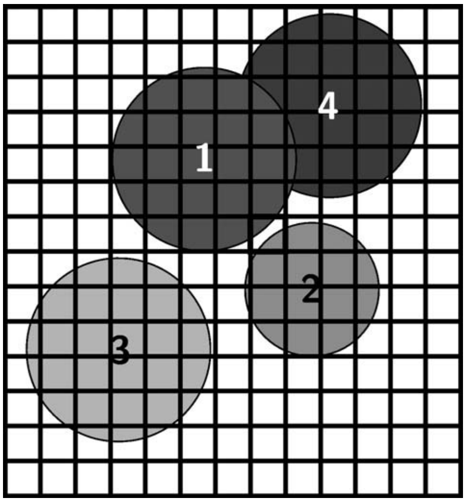


Fig. 1. A grid of cubic voxels overlaid on the vector information in *μ*ic.

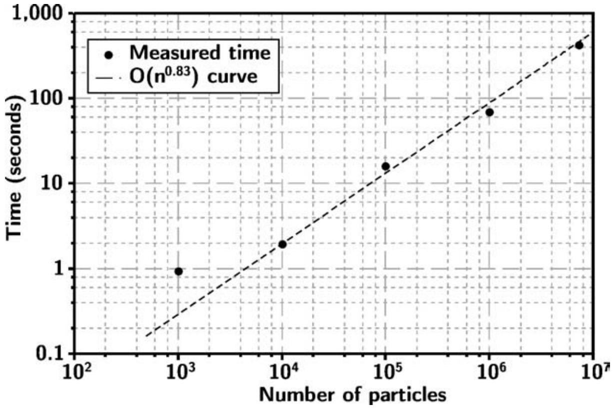


Fig. 2. Relationship between number of particles and time needed for packing different particle size distributions at 0.4 w/c.

The implementation of this method in μic enables simulations with millions of particles within a few hours. The improvement can be understood by considering that a non-overlapping random parking of 2.75 million particles, on a workstation purchased in the year 2008, can take more than two weeks without using the subdivisions and less than three minutes using the subdivisions on the same computer. Fig. 2 shows the observed increase in the time required to pack the particles as finer particle size distributions (larger number of particles) were packed at 0.4 w/c. It was observed that the time required for the packing increased in the order of $n^{0.83}$, where n is the number of particles. The observed exponent is less than 1 because, although the number of particles is larger for finer particles size distributions, it is easier to place finer particles in the space remaining between the larger particles. As the total time of computation for the first point is less than one second, calculation overheads could not be separated and therefore this point lies farthest from the trend-line.

In μic, the interactions of all particles are calculated individually and new product is allowed to form only on the free surface of the particles. The free surface is calculated by sampling the surface at a fixed number of uniformly distributed random points [21,22]. To do this, the overlap of each sampling point with spheres in the grid-cube in which the point is present is checked. It is also possible to sample the surfaces at points lying on a regular grid [12] on the surface of the sphere, but a larger number of sampling points compared to the random approach are required to obtain the same average degree of accuracy. For most cases sufficient accuracy was achieved using 90 to 100 points on each sphere.

3.2. Customisable elements and plugins in μic

Unlike IPKM, simulations in μic are not limited to alite and it should be possible to use μic for virtually all cementitious materials. Indeed, μic may be useful for simulation of systems other than cement since the design of the platform removes most of the hard-wired information that is generally hidden from the user, providing extra flexibility to the user. μic gives users flexibility at two levels. At the first level, a user can add and customise the properties of the materials, particles and reactions in the simulations. At the second level, the user can customise the models for processes that control microstructural development using plugins. Plugins are implementations of models which can be used by μic to calculate values during the simulation. Plugins will be discussed in more detail later in this section.

Fig. 3 summarises some of the important customisable elements in μic and their relationship. In addition to the depicted features, the object-oriented design of μic enables relatively easy development of the programmes and addition of new plugin types that may be

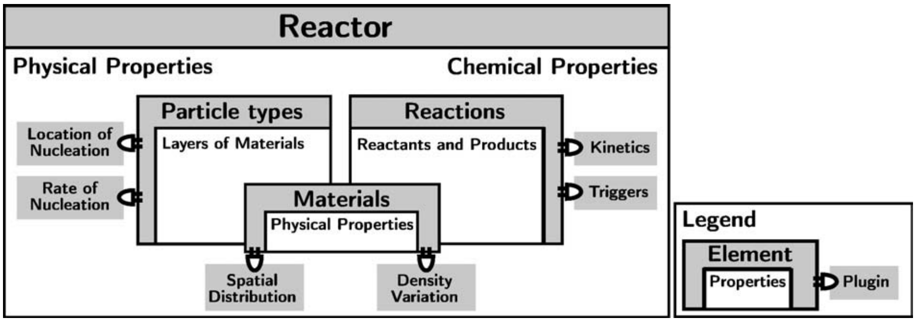


Fig. 3. Elements in µic with examples of customisable properties and plugins.

required. Some of the important customisable elements in µic and the properties that can be controlled using plugins are discussed in more detail in the following sub-sections.

3.2.1. Materials

In order to allow simulation of different types of materials, including cements with different phases; the presence of other materials such as fillers and mineral additives, µic allows definition of different materials for the simulations. All the reactants, products and possible intermediate products are defined as materials. For example, in the system in Fig. 4, Alite, Aluminate, Limestone, CH, Inner C–S–H, Outer C–S–H and Filler C–S–H are the materials defined. Each material can be assigned a name and its properties such as density and initial amount are defined. The density of the material defined here is used to represent reactions in terms of volumes, as discussed in Section 3.2.3. Plugins can be used to vary the average density of materials for individual particles.

Plugins can be also defined to control the initial proportions of materials in individual cement particles and the distribution of materials produced from a reaction to individual particles. For example, a material can be distributed to different particles in proportion of their available surface area and the presence of a particular cement phase could be limited to particles lying within a given size-range.

3.2.2. Particles

Particles of cement can contain many different phases interspersed together into a single grain and during hydration layers of other phases may deposit over these particles. In order to allow simulations with the various possible arrangements of phases in particles, in µic, particles can be defined to contain concentric layers of different materials with clear boundaries between the materials. The layers can also be defined to contain many materials, without a clear boundary between them with the assumption that the materials are distributed uniformly throughout the volume of the layer. The amounts and densities of different materials can be different for different particles.

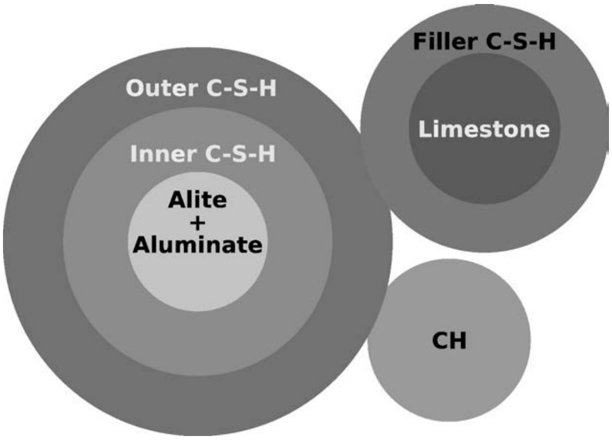


Fig. 4. Example of different materials and particle types defined in a simulation.

For example in the system in Fig. 4, three types of particles are defined. In the first type of particle, Alite and Aluminate are defined to be interspersed together in the same layer, while Inner C–S–H and Outer C–S–H are defined in separate layers. The second particle type in this example is defined to have a single layer of the material CH and the third particle is defined to contain an inner layer of Limestone and an outer layer of Filler C–S–H.

The user can define the initial particle size distribution of a particle type present at the beginning of the simulation. The number of new particles to be formed for each particle type can also be defined using plugins and the location of the new particles can be controlled. For example, new particles can be forced to grow on the free surface of another particle.

3.2.3. Reactions

Different reactions in the system can be defined by using the materials defined earlier as reactants and products. As µic simulates the evolution of the geometry, the proportions of various materials involved in the reactions are defined in terms of volumes. These volumes can be calculated using the stoichiometry of the reaction and the base value of the density of individual materials.

Plugins can be used to control the rates of reactions for individual particles. The reaction rate of a particle can depend on its own condition, its neighbourhood or global values such as time. Reactions can also be defined to be active only under certain conditions. For example, the reaction of a mineral additive may be defined to occur only after a defined minimum amount of a alite has reacted.

3.2.4. Plugins

Plugins are usually short, external, separately compiled encapsulations of algorithms that can be loaded by µic at run-time and can act as an integral part of the model. Users can choose from a library of plugins available with µic or write their own plugins. µic loads plugins at run time and can do so from external accessible sources owing to the capability of Java to dynamically load classes.

Plugins define the localised behaviour of various elements at the level of individual particles. For example, a reaction kinetics plugin would calculate and return the rate of reaction of a particle under a given condition. The calculation may depend on the size of the particle, the availability of a specific material in the particle, the available free surface of the particle, etc. To maximise flexibility, the plugins have access to all microstructural information such as the size of individual layers in the particles, the amounts and densities of different materials in the layers and the neighbourhood of the particles.

Apart from the uses of plugins for individual materials, particles and reactions discussed earlier, plugins can also be used to define the initial packing algorithm of particles, to make state-dependent

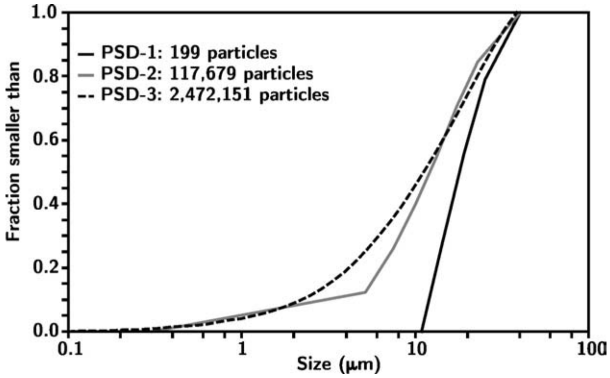


Fig. 5. Particle size distributions used in the simulations.

adjustments in the calculations and to store information after each step of the simulation.

To illustrate the use of plugins, the example below shows how a plugin would be written to control the distribution of a material being produced in a reaction to different particles. The code-segment shown calculates and returns the weight to be given to the given particle when the fraction allocated to this particle is calculated.

```
public double getFraction(Particle particle, float time) {
    if (particle.radius<minRad) {
        return minFraction;
    } else {
        return particle.getFreeArea();
    }
}
```

In the code-segment above, the weight assigned to the given particle is calculated based on the surface area of the particle that is not covered by other particles. The code also gives a bias to smaller particles to encourage them to grow by assigning a minimum weight to particles that have a radius smaller than a prescribed size.

As μic continues to be developed, latest information about development and the software can be found on the μic web page [1]. In the following sections, example simulations are presented to demonstrate the importance of the faster speed of μic and the versatility gained from the design of the platform.

4. Example 1: classical model of alite hydration – influence of particle sizes

In the following examples, alite samples with different particle size distributions are simulated based on rules for hydration described earlier by Pignat et al. [16]. It must be noted that these rules are customisable in μic and have been used as input only for the current simulations, in order to reproduce conditions traditionally simulated, especially those used in the IPKM. In fact the main object of this example is to demonstrate the influence of the particle size distribution on microstructure and therefore the importance of the increased speed of calculations in μic. From this perspective the exact kinetics used have little impact on the results which are presented at different degrees of hydration rather than hydration times.

Table 1
Number of alite particles and machine time for hydration simulations and pore-size distribution calculations at different degrees of hydration and resolutions.

Simulation	Number of alite particles	Machine time for simulations (s)			
		Hydration	Pore size distribution		
			70%; 0.2 μm	70%; 0.1 μm	90%; 0.1 μm
PSD-1	199	110	297	12,460	5337
PSD-2	117,679	412	102	2785	1647
PSD-3	2,472,151	4928	118	929	738

4.1. Mechanisms and rules for hydration

To summarise the rules used for the simulations, spherical particles are generated to match the input particle size distributions and are parked at random locations in the microstructure. The rates of hydration of individual particles are calculated using arbitrarily chosen hydration kinetics. The models for the kinetics used, which have little effect on the comparative results presented here can be found elsewhere [16].

The amounts of materials produced are calculated by converting Eq. (1) below into volumetric terms, as shown in Eq. (2), assuming the density of C_3S to be 3.15 g/cc, 1.0 g/ml for H_2O and 2.24 g/cc for CH. While variations in the literature exist, the value of 2.0 g/cc was used as the density for C–S–H as it is widely accepted as the average density of the product.

$$C_3S + 5.3H \rightarrow C_{1.7}SH_4 + 1.3CH \qquad (1)$$

$$1.0\ V_{C_3S} + 1.318\ V_{H_2O} \rightarrow 1.57\ V_{C\text{–}S\text{–}H} + 0.593\ V_{CH} \qquad (2)$$

The C–S–H is assumed to grow concentrically over the cement particles and the CH is assumed to form new particles in the pore-space. Based on experimental data given by Jennings and Parrott [23], the number of CH particles is assumed to grow at an exponentially reducing rate. The maximum number of CH particles in all systems is arbitrarily set to ten thousand in a computational volume of 100 μm size. The C–S–H formed from a particle is deposited around the same particle and the CH formed is distributed in random proportions to all CH particles present. It is stressed that all parameters can be customised in μic and the values used here are only for comparative purposes.

4.2. The simulations

The hydration of powders with three different particle size distributions, shown in Fig. 5, was simulated. Cubic computational

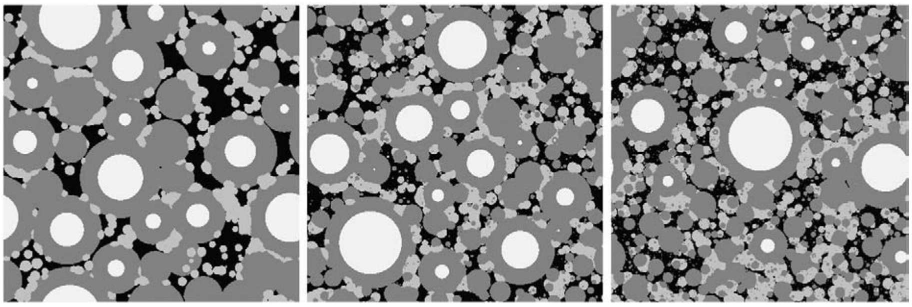
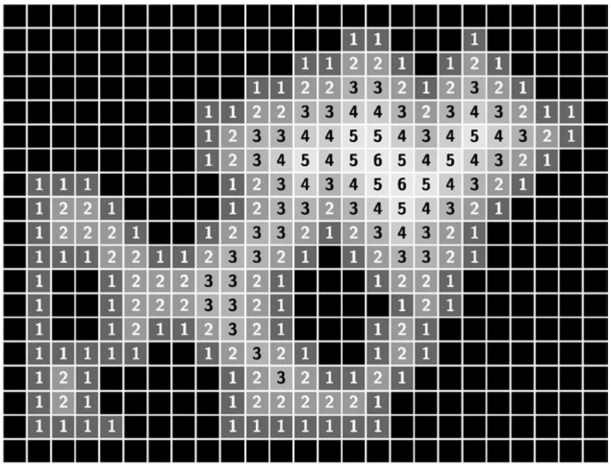


Fig. 6. Microstructures at 80% hydration for PSD-1 (left), PSD-2 (middle) and PSD-3 (right), with C_3S in lightest grey-scale, followed by CH and C–S–H and pores in black.



														1	1				1					
												1	1	2	2	1		1	2	1				
										1	1	2	2	3	3	2	1	2	3	2	1			
								1	1	2	2	3	3	4	4	3	2	3	4	3	2	1	1	
								1	2	3	3	4	4	5	5	4	3	4	5	4	3	2	1	
								1	2	3	4	5	4	5	6	5	4	5	4	3	2	1		
	1	1	1						1	2	3	4	3	4	5	6	5	4	3	2	1			
	1	2	2	1					1	2	3	3	2	3	4	5	4	3	2	1				
	1	2	2	2	1			1	2	3	3	2	1	2	3	4	3	2	1					
	1	1	1	2	2	1	1	2	3	3	2	1		1	2	3	3	2	1					
	1			1	2	2	2	3	3	2	1				1	2	2	1						
	1			1	2	2	2	3	3	2	1					1	2	1						
	1			1	2	1	1	2	3	2	1				1	2	1							
	1	1	1	1	1			1	2	3	2	1			1	2	1							
	1	2	1						1	2	3	2	1	1	2	1								
	1	2	1						1	2	2	2	2	2	1									
	1	1	1	1					1	1	1	1	1	1	1									

Fig. 7. Erosion to identify pore-skeleton using pixels. The solids are shown in black and the pores are shown in shades of grey with numbers indicating the number of steps required to reach the given pore pixel from the closest solid pixel.

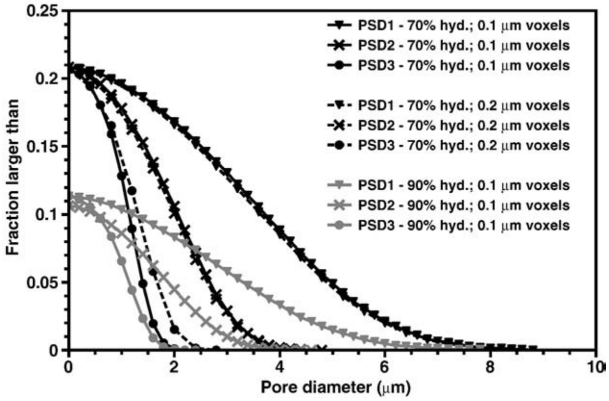


Fig. 8. Pore size distribution at 70% and 90% degree of hydration; the simulations at 70% hydration have been carried out at two voxel sizes, 0.1 μm and 0.2 μm.

volumes, 100 μm on the side, were packed with 44.25% of solids, equivalent to 0.4 water to cement ratio. The first particle size distribution contains mainly large particles, with a total of 199 alite particles in the volume. The third particle size distribution is from a commercial cement, with a total of 2,472,151 alite particles in the volume. The second particle size distribution is a close approximation of the third particle size distribution, although containing only 117,679 particles. A snapshot of the three microstructures around 80% degree of hydration are shown in Fig. 6. The details of the simulations, including the processor time for the simulations are listed in Table 1. The simulation of hydration with PSD-3 took less than 2 h on a desktop computer. The times listed for the simulations include the initial packing, and the calculation of overlaps, implementation of reactions, the creation of new particles and creation and saving of images and other output files for all the 30 steps of hydration simulated.

4.3. Approximate pore-size distributions

The pore-size distributions of the microstructures obtained were calculated using a simplified three-dimensional voxel-erosion method [24]. In the original model, Pignat et al. [16] presented a three-dimensional vector approach to characterise the porosity. However, this approach can at present only be implemented up to around 200,000 particles. For this reason this approach is not used here.

In the method used here, the vector microstructure was overlaid with a cubic mesh and the voxels were marked to be solid or pore based on their occupancy. The pore voxels sharing a face with a solid voxels were marked with the number 1, and in the next step the pore voxels sharing a face with the voxels marked 1, were marked with 2, and so on until all voxels were marked with the number of steps required to reach them from solid boundaries. This method allows identification of voxels farthest away from the solid walls in each pore, which are defined as the centres of the pores. A schematic of this algorithm is shown in Fig. 7. The extent of each pore can be identified by a walk in the reverse direction from the pore-centres.

While this method is only one of the many possible definitions of pores and pore-sizes, and ignores information regarding pore-topology, it can be used to obtain quick comparative results and is widely used elsewhere [24,25].

The simulations were carried out at 70% and 90% hydration to study the change in the pore-size distribution with hydration. A voxel size of 0.1 μm was used in all simulations, resulting in 1 billion voxels in each simulation. The results obtained from this analysis are shown in Fig. 8. Although the total porosity, which is defined by the degree of reaction and the water to cement ratio, is the same, the pore-size distributions are seen to be significantly different for the three particle size distributions.

The results at 70% hydration were also compared to those obtained using a 0.2 μm voxel size to study the convergence with mesh size. While the results for PSD-1 and PSD-2 were close for the two resolutions, there was a slight difference in the results for PSD-3, which is expected to have a larger proportion of pores close to the resolution of 0.1 μm.

Finer pore-size distributions were obtained for the finer particle size distributions. Although the second particle size distribution appears to be close to the third particle size distribution, the pore-sizes of the microstructures are significantly different demonstrating that the particle size distribution can have a significant effect on the microstructure obtained from the simulations. The results also

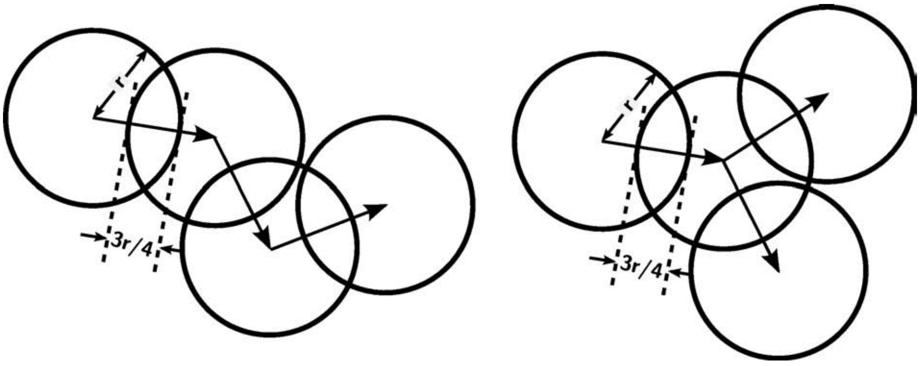


Fig. 9. Schematic for normal (left) and branching growth (right).

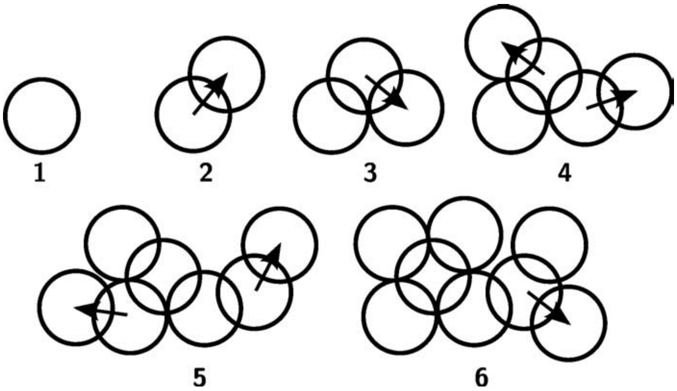


Fig. 10. Steps in branching growth.

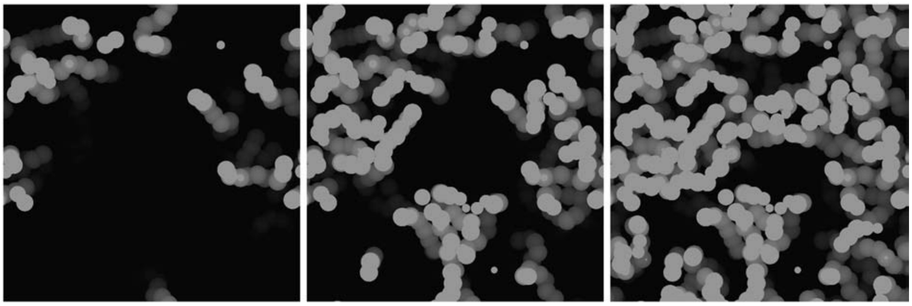


Fig. 11. Evolution of the microstructure with non-branching fibres with solid-volume fraction of 13.3% (left), 22.7% (middle) and 32.0%(right).

demonstrate that the faster implementation of the vector approach in μic, which allows a more representative number of particles to be included in the simulations, can be crucial to a more accurate representation of the model microstructure. The truncation of the particle size distribution to obtain a more manageable number of particles can lead to the calculated microstructure not being representative of the reality.

The voxel size used in this analysis is much smaller than those used in earlier models, but is still large relative to the actual pore sizes present in cement pastes. However, it is important to note that the vector information contained in the model is resolution-free and also contains information about much finer pores in the model microstructure.

5. Example 2: simulations of fibrous growths using μic

In these examples, μic is used to simulate fibrous growth, and the overall growth kinetics and fill fraction are simulated based on the mechanism described below. Two examples of fibrous growth, without and with branching, are presented here. As micrographs have shown C–S–H forming a fibrillar structure [26], the structures simulated here could be relevant to the microstructural development of C–S–H. The simulations presented here demonstrate the versatility gained from the design of the platform.

5.1. Mechanisms and rules

In these simulations, spherical seeds occupying only a small fraction of the volume were placed in the computational space at the start of the simulation. The growth was simulated by the addition of new spherical particles of the same size at each step of the simulation, to form a chain of overlapping spheres. In the first simulation, new particles are added only at the end of the chains, leading to a non-branching growth, while in the second case, new particles are also added overlapping with the inner particles in the chain, leading to a branching growth. The centre of the new particles is generated at a random angle at the specified distance from the end particles in the first case, and from all particles in the second case. A new particle is

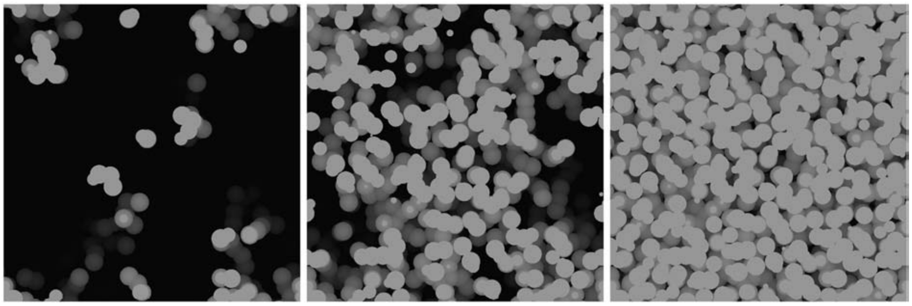


Fig. 12. Evolution of the microstructure with branching fibres with solid-volume fraction of 7.1% (left), 31.5% (middle) and 65.7%(right).

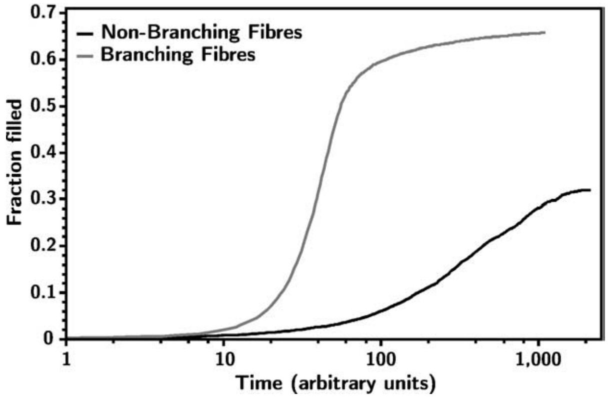


Fig. 13. Evolution of the volume filled by solids with normal and branching fibres.

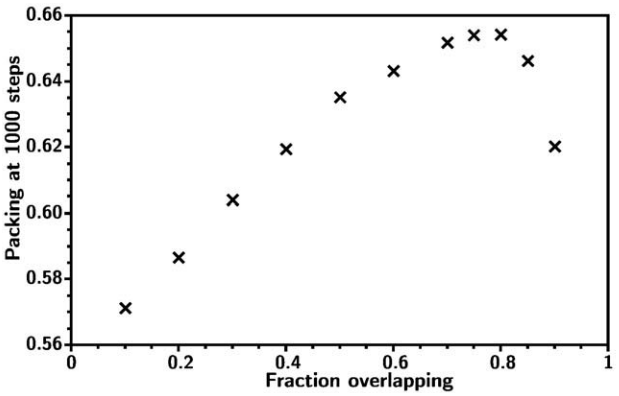


Fig. 15. Dependence of fill fraction at 1000 steps of simulation on fraction of radii of adjacent spheres overlapping.

allowed to form if there is no overlap with a particle other than its parent particle. The random generation of the locations of the new particles leads to a growth in random directions. Similar growth patterns have also been used to study other inorganic and biological growths [27–29]. The main steps in the simulation are listed below.

- Initial spherical particles occupying a small fraction of the volume are randomly placed.
- At each step of the simulation new particles are added to the system, overlapping by a known amount with one of the particles already present. The overlap is made in order to have shapes closer to a fibre than a chain of spheres.
- In the non-branching case, random locations for new particles are attempted only next to the particles at the ends of the chains, while in the branching case the new locations are attempted next to all particles in all chains.
- The new particle is placed only if it can be placed without overlaps with particles other than its parent particle. For this reasons, growth might not occur at some of the seeds at some steps in the simulation.

Schematics of the growth are shown in Figs. 9 and 10.

While the mechanisms modelled here are relatively simple, obtaining quantitative measures of values such as the rate of growth and the final fraction of volume filled may be difficult. This example, therefore, shows how μic can be used to obtain quantitative information about processes that may be too complex to solve analytically.

5.2. The simulations

In the simulations, initial spherical seed particles occupied 0.1% of the computational volume were placed at random locations. The size of the computational volume is 20 units on each side and the diameter of the seeds is 1 unit. The growth is simulated by the addition of new spherical particles of the same size in the volume. The new particles overlap with the seeds by 75% of their radii (Fig. 9). Three-dimensional images of the the evolution of the microstructures for the two simulations are shown in Figs. 11 and 12.

5.3. Results

Figs. 13 and 14 show the evolution of the volume filled and the rate of growth in the simulations. It was observed that the volume is filled much faster in the branching case and also reaches a higher fill-density. In the case of non-branching fibres, the final packing density achieved was only around 32.0% and in the case of branching growth the final packing volume of around 65.7% was achieved. It is interesting to note that the final packing achieved in the latter case is closed to the reported solid fraction in C–S–H [30]. As seen in Fig. 14, in the case where branching is not allowed, the fibres grow at an almost constant rate until proximity with other fibres, or other segments of the same fibre, prevents growth in some directions, after which the rate of growth continues to increase. In the second case, since growth can happen at any particle in the chain, the number of points where growth can occur increases initially, accelerating the calculated total rate of growth. Later, similar to the first case, the rate of growth reduces as the less space available for growth reduces. This acceleration and subsequent deceleration in the rate of growth is similar to the hydration kinetics observed for cement.

The amount of overlap prescribed between neighbouring spheres was varied to study its affect on the maximum packing fraction, as

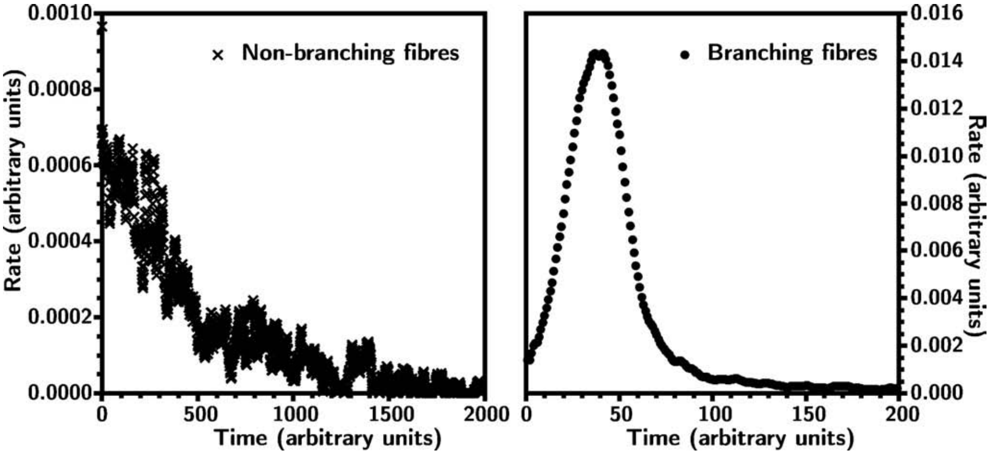


Fig. 14. Rate of reaction with non-branching (left) and branching (right) fibres.

shown in Fig. 15. A lower overlap leaves more empty spaces between individual particles reducing the packing, while a larger overlap prevents branching due to a larger chance of overlap with other particles that have the same parent. The highest maximum packing fraction was found at an overlap of 0.8 times the radius of the particles. Since the chains of overlapping spheres are only an approximation for fibres with uniform cross-section, this overlap gives the closest to fibres with uniform circular cross-sections.

It can be seen from the above example that the versatility of μic provides a relatively simple way to simulate complex processes. Taking the local mechanism of a semi-random process as input, the overall growth rate of fibres was simulated. This was possible in μic as not only the rate of a reaction, but also the location of nucleation can be controlled. The process is relatively complex as the fibres tangle when they grow, thereby inhibiting the growth of other fibres. Still, the overall progress of the reaction can be simulated after accounting for all the interactions and the final volume filled by the particles could be calculated.

6. Conclusions

A new free open source platform, called μic, has been developed to enable the modelling of microstructural evolution of cementitious materials. The vector approach has been used to allow representation of particles and other features with sizes varying over several orders of magnitude. In μic, the performance of the vector approach has been improved to allow simulations with millions of particles within reasonable times. Unlike other currently available models, μic is not restricted to our current understanding of cement or to the information available to the developer of the programmes. The platform has been designed to enable customisation of all aspects of microstructural development and although μic has been developed for cementitious materials, it could be useful for simulations of other particulate growths.

In the examples presented in this study, the importance of the faster algorithms used in μic in order to properly represent cement microstructure by accounting for all particle sizes was highlighted. It was also shown using simple examples that the versatile architecture of μic can be used to study complex processes. While the first set of examples show how μic can be used to reproduce local effects, such as pore-sizes, the second set of examples show how localised mechanisms can be used to study the overall progress of the microstructure. By simulating the widely different processes presented in the two examples using the same modelling platform, the versatility of μic is also highlighted. It can be seen that due to its performance and customisability, μic provides an effective tool to study reaction mechanisms and microstructural development of growth processes which can prove useful to study cement hydration and other particulate systems.

Acknowledgement and authors note

The authors gratefully acknowledge the financial support from the Swiss National Science Foundation project number 200020-112063/1. Latest information about the development status of μic, user documentations and freely downloadable source codes and executables of μic and its graphical user interface, including all the plugins and modules used in this publication can be found on the μic web page [1].

References

[1] S. Bishnoi, Webpage of μic the modelling platform, www.micthemodel.org.
[2] D.P. Bentz, Three-dimensional computer simulation of portland cement hydration and microstructure development, Journal of the American Ceramic Society 80 (1997) 3–21.
[3] D.P. Bentz, Modelling cement microstructure: pixels, particles, and property prediction, Materials and Structures 32 (1999) 187–195.
[4] D.P. Bentz, Cement hydration: building bridges and dams at the microstructure level, Materials and Structures 40 (2006) 397–404.
[5] C.-J. Haecker, E.J. Garboczi, J.W. Bullard, R.B. Bohn, Z. Sun, S.P. Shah, T. Voigt, Modeling the linear elastic properties of Portland cement paste, Cement and Concrete Research 35 (2005) 1948–1960.
[6] J. von Neumann, The Theory of Self-reproducing Automata, Univ. of Illinois Press, USA, 1966.
[7] E.J. Garboczi, D.P. Bentz, Effect of statistical fluctuation, finite size error, and digital resolution on the phase percolation and transport properties of the NIST cement hydration model, Cement and Concrete Research 31 (2001) 1501–1514.
[8] P.A. Cundall, O.D.L. Strack, Discrete numerical model for granular assemblies, Geotechnique 29 (1979) 47–65.
[9] S.K. Johnson, H.M. Jennings, Computer simulated hydration of a cement model, Proceedings of the 10th CIB congress, International Council for Building Research, Studies, and Documentation, Washington D.C., U.S.A., 1986, pp. 2086–2095.
[10] K. van Breugel, Numerical simulation of hydration and microstructural development in hardening cement-based materials (I): Theory, Cement and Concrete Research 25 (1995) 319–331.
[11] P. Navi, C. Pignat, Simulation of cement hydration and the connectivity of the capillary pore space, Advanced Cement Based Materials 4 (1996) 58–67.
[12] M. Stroeven, P. Stroeven, SPACE system for simulation of aggregated matter application to cement hydration, Cement and Concrete Research 29 (1999).
[13] H.M. Jennings, S.K. Johnson, Simulation of microstructure development during the hydration of a cement compound, Journal of the American Ceramic Society 69 (1986) 790–795.
[14] P. Navi, C. Pignat, Three-dimensional characterization of the pore structure of a simulated cement paste, Cement and Concrete Research 29 (1999) 507–514.
[15] P. Navi, C. Pignat, Effects of cement size distribution on capillary pore structure of the simulated cement paste, Computational Materials Science 16 (1999) 285–293.
[16] C. Pignat, P. Navi, K. Scrivener, Simulation of cement paste microstructure hydration, pore space characterization and permeability determination, Materials and Structures 38 (2005) 459–466.
[17] E. Gallucci, K. Scrivener, Crystallisation of calcium hydroxide in early age model and ordinary cementitious systems, Cement and Concrete Research 37 (2007) 492–501.
[18] G. Ye, K. van Breugel, A.L.A. Fraaij, Three-dimensional microstructure analysis of numerically simulated cementitious materials, Cement and Concrete Research 33 (2003) 215–222.
[19] P.M. Hubbard, Collision detection for interactive graphics applications, IEEE Transactions on Visualization and Computer Graphics 1 (1995) 218–230.
[20] D.J. Kim, L.J. Guibas, S.Y. Shin, Fast collision detection among multiple moving spheres, IEEE Transactions on Visualization and Computer Graphics 4 (1998) 230–242.
[21] J.M. Cook, Technical notes and short papers: rational formulae for the production of a spherically symmetric probability distribution, Mathematical Tables and other Aids to Computation 11 (1957) 81–82.
[22] E.W. Weisstein, Sphere point picking, from MathWorld—A Wolfram web resource http://mathworld.wolfram.com/SpherePointPicking.html.
[23] H.M. Jennings, L.J. Parrott, Microstructural analysis of hardened alite paste, Part II: Microscopy and reaction products, Journal of Materials Science 21 (1986) 4053–4059.
[24] W.K. Pratt, Digital Image Processing, Wiley, New York, 1991.
[25] C.A. Baldwin, A.J. Sederman, M.D. Mantle, P. Alexander, L.F. Gladden, Determination and characterization of the structure of a pore space from 3D volume images, Journal of Colloid and Interface Science 181 (1996) 79–92.
[26] I.G. Richardson, The calcium silicate hydrates, Cement and Concrete Research 38 (2008) 137–158.
[27] R.M.J. Cotterill (Ed.), Models of brain function, Cambridge University Press, 1989.
[28] O. Lastow, Simulation of dendrite formations of aerosol particles on a single fibre, Journal of Aerosol Science 25 (Supplement 1) (1994) 199–200.
[29] M.R. Wilson, J.M. Ilnytskyi, L.M. Stimson, Computer simulations of a liquid crystalline dendrimer in liquid crystalline solvents, The Journal of Chemical Physics 119 (2003) 3509–3515.
[30] H.M. Jennings, A model for the microstructure of calcium silicate hydrate in cement paste, Cement and Concrete Research 30 (2000) 101–116.