A MULTIGRID FINITE ELEMENT METHOD FOR THE MESOSCALE ANALYSIS OF CONCRETE

Stefan Häfner and Carsten Könke

Institute of Structural Mechanics,
Bauhaus-University Weimar
Marienstraße 15, 99423 Weimar, Germany
e-mail: stefan.haefner@bauing.uni-weimar.de,
carsten.koenke@bauing.uni-weimar.de

Key words: Concrete, Heterogeneous Material, Finite Element Method, Mesoscale Analysis, Multigrid Method.

Abstract. In classical engineering models concrete structures are described by homogenized values of material properties such as elastic modulus and ultimate stresses. However, as concrete mainly consists of natural aggregates and mortar matrix, it is a heterogeneous material with significant physical differences among its components. For most realistic simulation, we propose to process a mechanical analysis of concrete on the mesoscale. The first important aspect is the geometrical modeling of the heterogeneous material. Besides the possibility of digital image processing, other methods are introduced for generating inclusion–matrix models of concrete. As any real-size concrete specimen includes numerous particles, the geometrical complexity becomes critical for creating an adequate finite element mesh of inclusion surfaces. Instead of that, uniform element grids are used to model the structure and the material properties are assigned at the elements according to the corresponding position of the geometrical model. The resulting, potentially high number of unknowns raised the need of an efficient solver. Various iterative solvers are evaluated and a corresponding multigrid solver is introduced. The algorithms includes mesh adaptivity and cyclic coarse grid correction. Different error influences are considered and assessed for achieving best overall efficiency. Some examples are presented to examine the proposed methods regarding algorithmic aspects, accuracy and computational costs.
1 INTRODUCTION

This article discusses the simulation of concrete material on the mesoscale, which is embedded in the diagnosis of real-sized concrete structures by a multiscale analysis\(^1\). Usually the simulation of concrete structures is based on homogenized materials. In contrast, the analysis on the mesoscale integrates the particular components for a direct modeling of concrete as a heterogeneous material and aims at a more flexible simulation basis.

The article describes a combination of methods for achieving this concept. First of all the geometrical modeling of the heterogeneous concrete material is of major interest (Section 2). Corresponding concrete characteristics such as particle shapes, particle size distribution and physical properties of the particular components are numerically evaluated in [1, 2]. Models on the mesoscale require a high resolution and corresponding numerical discretizations lead to very large, sparse equation systems. Therefore a local grid formulation using rectangular finite elements is introduced (Section 3). Basic iterative solver algorithms are directly adapted to this special discretization and their efficiency is compared on a sample structure (Section 4). The special discretization technique as well as the iterative solver method prepare the discussion on the multigrid method (Section 5). Algorithms of an adaptive strategy and a more effective cyclic coarse grid correction are highlighted and an example with varying mesh refinement and error tolerance allow for a first evaluation of the implemented multigrid method. An effective error control plays an important role to assess the quality and efficiency of the different algorithms connected to the geometrical resolution, finite element analysis and iterative solving (Section 6). The presented examples are restricted to the linear-elastic material law. Further research is necessary to extend the methods to nonlinear constitutive laws taking into account damage processes.

2 GEOMETRICAL MESOSCALE MODEL OF CONCRETE

The data from a concrete section could either directly be derived from a photograph by digital image processing (Section 2.1) or a concrete sample can be created analytically from theoretical information about the concrete mixture [1] (Section 2.2).

2.1 Digital image processing of two-dimensional concrete sections

A first study project revealed that a combination of human interaction and automated operations is most promising to identify aggregates within the mortar matrix at reasonable effort. The applied image format is the uncompressed 24-bit RGB-bitmap, which is defined by a 54-byte header and following 8-bit red, green and blue channels defining one pixel after another up to a defined line width and image length. Global filters of

\(^1\)Project A5 "Simulation of Predamaged Structures using Multiscale Models" of the Collaborative Research Center 524 "Materials and Structures in Revitalization of Buildings" at Bauhaus-University Weimar, Germany; supported by the German Research Foundation (DFG).
brightness and color did not lead to satisfying results. Therefore the user defines a set of mortar matrix pixels on the original image (Fig. 1), which exactly determine certain RGB–combinations. Other pixels are then automatically defined as mortar matrix, if they lie within a defined bandwith to any of these RGB–combinations (but not just within the minimum–maximum range of the different channels).

Another strategy is to select one or more typical mortar matrix pixels and extend the matrix from any pixel to its neighbor pixel recursively, if a certain color gradient from the actual pixel to the neighbor pixel is not exceeded. As there is a lower size limit of aggregates to be clearly identified, smaller areas are dedicated to its surrounding area (Fig. 2). This leads to a clear definition that aggregates above a certain size are explicitly identified, whereas the rest is a mixture of smaller particles and mortar matrix. This information can be directly applied to a computational analysis of this two-dimensional section or further be processed to identify the main characteristics, such as particle size distribution, particle shapes or geometrical packing, which are then applicable to generate larger samples in two or three dimensions with corresponding properties. Further, different methods to identify the mesoscale geometry of concrete are presented in [3] and [4].

2.2 Numerical descriptions of inclusion–matrix models

Most important parameters of the mesoscale geometry are the volume ratio of particles, the particle shapes and the particle size distribution (more details in [2]). While often the aggregates within the matrix are defined by ellipses or ellipsoids, other particle shapes are also applicable [5]. Eq. 1 defines the basic implicit formula for generating different particle shapes in 2D (Fig. 3, top), which can further be enriched by a sum of sine functions $F_i$ (Eq. 2, Fig. 3, bottom).
According to a given particle size distribution the particles are generated up to a ratio of 60% to 90% of the total concrete volume. For generating appropriate particle sizes for the two-dimensional case, Eq. 3 was derived from Fuller’s curve on the assumption of ideal spheres, where \([x_1;x_2]\) defines the particle size interval and \(X_1^{[0;1]}\) and \(X_2^{[0;1]}\) define two independent random variables.

\[
x(x_1^{[0;1]},X_2^{[0;1]}) = \left(x_1^{-1.5} - X_1^{[0;1]}(x_1^{-1.5} - x_2^{-1.5})\right)^{-\frac{1}{1.5}} \sqrt{1 - \left(X_2^{[0;1]}\right)^2}
\]  

(3)

Then the particles are sorted by size before placing them one after another into the matrix. Thereby it continuously has to be tested, if the position of the current inclusion is valid (not overlapping with any former particle). Different methods of such separation checks are introduced in the following Subsections 2.2.1–2.2.5.

### 2.2.1 Fast detection using bounding boxes

A simple and fast method to detect whether two particles are separated, is to compare the distance of their centers to the defined maximum particle size. Alternatively, bounding boxes can be generated as illustrated in Fig. 4. If these boxes are separated, the conclusion is unique, whereas overlapping boxes require to apply a more sophisticated and costly separation check within the overlapping region (Fig. 5). Therewith tight bounding boxes are preferable. Conversely simple conditions assure overlapping of close particles (interior boxes). Partitioning of the object further reduces the cost of the check, if the actual particle overlaps with any other particle. However, these conditions are not sufficient to cover the general arrangement of two particles.
2.2.2 Analytical separation checks

A fast separation check for three-dimensional ellipsoids was proposed in [1], which bases on an algebraic condition (more details in [6]). The advantage of this closed-form solution is the analytical accuracy and the speed. However, it is just applicable to the special form of ellipsoids (or ellipses).

2.2.3 Rastered separation checks in overlapping bounding boxes

Another concept is to detect the overlapping area of two bounding boxes and then to define a raster on which the two implicit formulas are explicitly tested (Fig. 6). This method can be applied to any particle shape, but it is relative expensive (especially for three-dimensional packing). Furthermore, it underlies a certain resolution and therefore it is not an exact method. It can only be sufficient from an engineering point of view. Another difficulty is to determine the particle area of any implicit formula, which is necessary for appropriate sorting by size and especially for determining the correct total volume ratio.

2.2.4 Pointwise separation checks on particle boundaries

A recent study project proposed to process separation checks on previously determined points on the surface of particles (Fig. 7). Therefore an analytical determination of the boundary points can be used for certain functions or in general a bisectional algorithm can be applied. It appears to be more efficient than the previous approach. However, complex shapes may not be detected by this method. Furthermore the method is more complicated to handle with regard to translation and rotation of the particle.
2.2.5 Rastered separation checks on complete domain

Our latest approach proposes to process the separation check directly on the real-world object grid, which will then directly be applied to further mechanical analysis using the finite element method (Algorithm 1).

Algorithm 1: Rastered separation checks on complete domain

1. Create one random particle according to given statistics
2. Determine and save pixels of particle boundary: Search from center point to any pixel of the boundary. Follow the boundary and draw pixels into a virtual domain until the first boundary pixel is found again.

Repeat (1)-(3) until defined particle volume ratio is reached

4. Sort particles by area (pixels).
5. Prepare Domain: Initialize domain grid to ‘0’. Set object boundary pixels to a non-zero number.

Loop (6)-(8) through all particles, start from the largest.

6. Pick random particle position in domain grid
7. Test the position of the current particle by simple if-conditions: if (all boundary pixels are still ‘0’ in the domain) goto (8) else goto (6)
8. Set boundary pixels and fill particle by a non-zero number.

This method can be applied to any random concave or convex particle shape in two or three dimensions (under the condition that an interior point can be found, e.g. the center point). The accuracy is directly adjusted to the subsequent numerical analysis. A potential disadvantage is that the obtained particle configuration requires some effort to be transferred on grids of a different ratio. However, this is principally possible, as the analytical parameters of each particle can be saved. The method is very fast, even for complex particle shapes, as there is no analytical effort after the first loop ((1)-(3)).
It can further be extended to model other objects, just by wrapping the corresponding pixels by a non-zero value, e.g. also placing steel bars into the domain is simply possible without any modification. Therewith the generation of a mesoscale geometry (Fig. 8) is not critical compared to the following mechanical analysis. For crack analysis using finite elements, it might further be useful to require at least on pixel gap between the particles (Fig. 9), which can easily be controlled by this method.

3 FINITE ELEMENT ANALYSIS OF HETEROGENEOUS BODY

After the definition of the macroscopic object, the number of degrees of freedom are defined and corresponding hierarchical orthogonal grids can be created on which the mesoscale geometry will be generated in form of physical parameters applied to each pixel. The parameters are then interpolated on the coarser grids, which play an important role for an efficient solving of this problem by a corresponding multigrid solver (Section 5). In this case the finite element method is preferable over the often in grid analysis applied finite difference method, as it has become more common with respect to special material laws.

3.1 Notes on mechanical background

A comfortable characteristic is that the stiffness matrix $K$ of any isoparametric panel element of constant thickness $t$ does not vary, if the size of the (distorted) finite element is scaled by a factor $\xi$. This is true, because it can be shown for the elastic stiffness matrix $K$

$$K = t \int_t \int_s B^T C B \det J \, dr \, ds;$$  \hspace{1cm} (4)
that the strain-displacement matrix $B$ and the Jacobi determinant $\det J$ follow the subsequent conditions

$$B(\xi x_k, \xi y_l) = \frac{1}{\xi} B(x_k, y_l)$$

(5)

$$\det J(\xi x_k, \xi y_l) = \xi^2 \det J(x_k, y_l)$$

(6)

where $\xi$ is any positive value and $(\xi x_k, \xi y_l)$ are the nodal coordinates for $k = 1..n$ and $l = 1..n$ up to a defined number of nodes $n$. For the mechanical background and common notation of Eq. 4 please refer to [7]. As the elements in the hierarchical grids are linear scaled, only one basic element stiffness matrix can be applied to any element $i$, if it is multiplied by the individual modulus of elasticity $E_i$

$$K(E_i, \mu_c) = E_i K(1, \mu_c)$$

(7)

under the condition, that $\mu_c$ is a constant value (otherwise fifty-one stiffness matrices are enough to cover the theoretical range of $\mu_c = [0.00; 0.50]$ by a sufficient accuracy of $\mu = \pm 0.005$). Concluding any line of the global stiffness matrix can quickly be generated by a predefined assembly of the basic element stiffness matrix and corresponding moduli of elasticity. This favours the use of iterative solvers, which only need to evaluate a certain line of the equation system at a time.

### 3.2 Local numbering scheme

The finite element analysis on orthogonal grids does not require to establish the global stiffness matrix at any time of the computation. Similar to the difference star in the finite difference method [8], a local assembly can be developed for finite elements. For the center node of a 2x2 element patch (with 18 degrees of freedom) all equations to solve the global system can be derived. The hat-symbol $\hat{}$ denotes variables, which are assigned to this system of Fig. 10. The local degrees of freedom of one element are sorted as shown in Fig. 11. The numbering of the basic element stiffness matrix $\hat{K}$ (Eq. 18) and of the displacement vectors $u$, which are marked by a superscript from $^a$ to $^d$, refers to this system of one element. The following equations represent the stiffness of the center node (9 and 10 refer to the x– and y–component, respectively.)

$$\hat{K}_{9,9} = E^a \hat{K}_{1,1} + E^b \hat{K}_{3,3} + E^c \hat{K}_{5,5} + E^d \hat{K}_{7,7}$$

(8)

$$\hat{K}_{9,10} = E^a \hat{K}_{1,2} + E^b \hat{K}_{3,4} + E^c \hat{K}_{5,6} + E^d \hat{K}_{7,8}$$

(9)

$$\hat{K}_{10,10} = E^a \hat{K}_{2,2} + E^b \hat{K}_{4,4} + E^c \hat{K}_{6,6} + E^d \hat{K}_{8,8}$$

(10)
In the special case of four rectangular panel elements with varying modulus of elasticity $E$, these equations can be simplified to

\[ \hat{K}_{9,9} = (E^a + E^b + E^c + E^d) \hat{K}_{1,1} \]  
\[ \hat{K}_{9,10} = (E^a - E^b + E^c - E^d) \hat{K}_{1,2} \]  
\[ \hat{K}_{10,10} = (E^a + E^b + E^c + E^d) \hat{K}_{2,2} \]  

The resulting forces are calculated by the following equations

\[ \hat{f}_9 = E^a \hat{K}_{1,1} \tilde{u}^a_i + E^b \hat{K}_{3,1} \tilde{u}^b_i + E^c \hat{K}_{5,1} \tilde{u}^c_i + E^d \hat{K}_{7,1} \tilde{u}^d_i \]  
\[ \hat{f}_{10} = E^a \hat{K}_{2,1} \tilde{u}^a_i + E^b \hat{K}_{4,1} \tilde{u}^b_i + E^c \hat{K}_{6,1} \tilde{u}^c_i + E^d \hat{K}_{8,1} \tilde{u}^d_i \]

or respectively (after correct assignment of $\tilde{u}^{a..d}$ to $\hat{u}$ and corresponding assembly of $\hat{K}_{9,i}$ and $\hat{K}_{10,i}$ from the Eqs. 14 and 15)

\[ \hat{f}_9 = \hat{K}_{9,i} \hat{u}_i \]  
\[ \hat{f}_{10} = \hat{K}_{10,i} \hat{u}_i \]

The Eqs. 8–17 can be applied to any interior node of an orthogonal grid. Boundary nodes can be included by omitting the terms of inexistent neighbor elements. In the next section these formulations are directly integrated into iterative solver methods.

4 BASIC ITERATIVE SOLVER ALGORITHMS

The general form of the linear equation system

\[ K \mathbf{u} = \mathbf{f} \]

can be solved by iterative methods, whereby essentially only vectors of the order $n$ have to be stored ($n$ denotes the number of degrees of freedom in the global system). While the computational efficiency benefits from this low memory demand, also the internal
handling of operations can be adjusted to the local numbering system. The iterative
solvers of this section are well-known classical approaches, which have generally been
superseded by newer algorithms (like the conjugate gradient method and many more).
However, they become again competitive, if they are applied on hierarchical grids in
connection with multigrid methods. Subsequently, the local finite element equations are
directly integrated into these methods.

4.1 The Block Jacobi method

The first intuitive solving scheme presented in [2] corresponds to the Block Jacobi
method and 2x2 blocks for the x– and y– components at each node. This can be inter-
preted that for any 2x2 element patch within the domain, the incremental displacement
\( u^{inc} \) is calculated by determining the unbalanced (or residual) forces \( f^r \) on the center
node and then calculate exactly the equilibrium point assuming all other nodes of this
patch would be rigid. The general form of the Block Jacobi method (the superscripts
denote the numbering of the blocks within the global system [9])

\[
(u^k)^i = (K^{ii})^{-1} \left( f^i_s - \sum_{j=1, j\neq i}^\beta K^{i,j} (u^{k-1})^j \right)
\]  

(19)
can be rewritten for the local numbering scheme as

\[
\begin{bmatrix}
\hat{u}_9^{inc} \\
\hat{u}_{10}^{inc}
\end{bmatrix} =
\begin{bmatrix}
\hat{K}_{9,9} & \hat{K}_{9,10} \\
\hat{K}_{10,9} & \hat{K}_{10,10}
\end{bmatrix}^{-1}
\begin{bmatrix}
\hat{f}_9^r \\
\hat{f}_{10}^r
\end{bmatrix}
\]  

(20)

As comprehensible later, additional damping by a relaxation factor (e.g. \( \omega=0.8 \) ) is
required for the convergence of this method in the applied test example.

4.2 The Jacobi method

The Jacobi method is similar to the Block Jacobi method. However, the equilibrium is
not calculated as accurate, because the exactly determined unbalanced forces \( f_x^r \) and \( f_y^r \)
here only lead to corresponding displacement increments \( u_x^{inc} \) and \( u_y^{inc} \), respectively. The
possibility that a force \( f_x \) (\( f_y \)) in a heterogeneous media can also cause a displacement
increment in y (x) direction is not included (usually in the numbering system of any
general linear equation system the order of x– and y–directions may not be recoverable).
The general equation of the Jacobi method

\[
\hat{u}_i^k = \frac{1}{\hat{K}_{ii}} \left( f^s_i - \sum_{j=1, j\neq i}^n K_{ij}\hat{u}_j^{k-1} \right)
\]  

(21)
can be adjusted to the local numbering scheme by using the following equations \((f^r\) denotes the residual force and \(f^s\) the externally applied force)

\[
\begin{align*}
\hat{f}^r_9 &= \hat{f}^s_9 - \hat{f}_9 \\
\hat{f}^r_i &= \hat{f}^s_i - \sum_{j=1}^{n} \hat{K}_{9,j} \hat{u}_{j}^{k-1} \\
\hat{u}_{9}^{inc} &= \hat{u}_{9}^{k} - \hat{u}_{9}^{k-1}
\end{align*}
\]  

which results in

\[
\hat{u}_{9}^{inc} = \frac{\hat{f}^r_9}{\hat{K}_{9,9}}
\]

and correspondingly

\[
\hat{u}_{10}^{inc} = \frac{\hat{f}^r_{10}}{\hat{K}_{10,10}}
\]

The Jacobi method, as well as the Gauss–Seidel method, converge, if \(K\) is strictly diagonally dominant (sufficient condition, on details see [9])

\[
|K_{ii}| > \sum_{j=1}^{n} |K_{ij}| \quad \text{for all } i
\]

\subsection{4.3 The Gauss–Seidel method}

The Gauss–Seidel method is almost analog to the Jacobi method. While the Jacobi method adds the displacement increments \textit{simultaneously} after cycling through all nodes, the Gauss–Seidel method adds all increments \textit{successively} as soon as available. Thereby the nodal order changes the convergence behavior of the Gauss-Seidel method. For orthogonal grids there are several different variants, as red-black ordering (like on a chessboard), lexicographical ordering or zebra-line ordering [9]. The general form of the Gauss–Seidel method

\[
u_{i}^{k} = \frac{1}{K_{ii}} \left( f_{i}^{s} - \sum_{j=1}^{i-1} K_{ij} \hat{u}_{j}^{k} - \sum_{j=i+1}^{n} K_{ij} \hat{u}_{j}^{k-1} \right)
\]

again leads to the Eqs. 25 and 26, but the local residual forces \(f^r_9\) and \(f^r_{10}\) depend on successively updated displacements. The Gauss–Seidel method converges if the global stiffness matrix \(K\) is positive definite. This is true, if the finite element model is properly restrained and stable [7].
4.4 The simultaneous over–relaxation method and the successive over–relaxation method

The over-relaxation methods propose to scale each incremental displacement by a relaxation factor \( \omega \). The general form of the over-relaxation methods is given by

\[
 u^k_i = (1 - \omega) u^{k-1}_i + \omega \bar{u}^k_i
\]  

(29)

where \( \bar{u}^k_i \) corresponds to \( u^k_i \) of the Jacobi method in case of the simultaneous over-relaxation method (JOR method) or to \( u^k_i \) of the Gauss–Seidel method in case of the successive over-relaxation method (SOR method). (if \( \omega = 1 \), this corresponds to the Jacobi and the Gauss–Seidel method, resp.). The optimum relaxation factor can theoretically be derived from the spectral radius (largest absolute eigenvalue) of the iteration matrix. But as this is expensive, several methods on the practical determination of \( \omega \) are proposed in [9] and [10]. Usually in finite element analysis the optimum relaxation for the SOR method is between 1.3 and 1.9 [7]. The convergence of the JOR method in the model case was achieved by a heuristic damping factor of \( \omega = 0.8 \).

4.5 Optimized relaxation by minimization of the residual

Depending on the error criteria which might be the vector norm of unbalanced forces, the Jacobi method can be used to calculate a displacement increment, which is then scaled by an optimal factor \( \omega \) such that the resulting vector norm of residual forces minimizes. The force vector increment \( f^{inc} \) which corresponds to \( u^{inc} \) can be determined by Eqs. 16–17. The condition then reads as

\[
 \| f^r + \omega f^{inc} \| = \text{Minimum}
\]  

(30)

which expresses a simple minimization problem with a quadratic expression of \( \omega \) and results in

\[
 \omega = -\frac{\sum_{i=1}^{n} f^r_i f^{inc}_i}{\sum_{i=1}^{n} (f^{inc}_i)^2}
\]  

(31)

This calculation requires additional computational effort. However, corresponding to the resulting displacement vector

\[
 u^k = \omega u^{inc} + u^{k-1}
\]  

(32)

the residual forces of the next step can quickly be calculated by

\[
 (f^r)^k = \omega f^{inc} + (f^r)^{k-1}
\]  

(33)

Depending on the number of digits of the implemented number format, \( (f^r)^k \) should be refreshed by Eqs. 16–17 because of a potential computational approximation error after several iterations. Special algorithms might also be useful, to evaluate the expression of Eq. 31 for a high number of degrees of freedom.
4.6 A sample comparison of convergence

One test example as shown in Fig. 12 has been selected to compare the convergence of the different methods (Fig. 13). It is remarkable, that the JOR method with optimized relaxation factor (J(opt.)) achieves almost the same convergence rate as the Gauss–Seidel method (G(1.0)) (probably due to numerical inaccuracy the J(opt.) graph bends at an error tolerance of 1.0E-13). Increasing the relaxation factor of the SOR method up to 1.9 leads to a significant acceleration. As the optimum relaxation factor is near 1.9, this indicates that the stiffness matrix is well-conditioned. An overestimation of the optimum relaxation factor reduces the convergence speed (ω = 1.99) and a further increase leads to non-convergent oscillating behavior. Concluding the implemented algorithms lead to the theoretically expected results on this test example. (The Damped Block-Jacobi method leads to almost identical results as the Damped Jacobi method.)

5 THE MULTIGRID METHOD

Multigrid methods have been well-known for over twenty years. They are often introduced for structured grids, but could also be used for unstructured grids [11]. A basic idea is to get a good start vector or a next increment for the fine grid by a cheap iterative step on a coarser grid. The multigrid method fits well into our scope of multiscale analysis, as the macroscopic dimensions of the mesoscale structure can be restricted to geometrical simple domains, whereas the heterogeneous character is integrated within the material data of the finite elements. Within this scope an adaptive multigrid method and a cyclic multigrid method are presented, which are applicable to linear problems. Furthermore, the multigrid method is also extensible to the simulation of nonlinear damage behavior,
as ") multi–grid algorithms are perfectly suited for solving nonlinear boundary value problems” (W. Hackbusch [8]).

5.1 An adaptive multigrid method

The previously discussed basic iterative solvers can be integrated into multigrid algorithms. A first simple, but not so effective variant is the adaptive multigrid method, which is of rather descriptive character, as it does not really touch the potential of multigrid methods. This method using mesh refinement as illustrated in Fig. 14 is outlined in Algorithm 2. Therein a good start vector of the finest mesh can be derived from cheap approximations on coarse meshes.

Algorithm 2: An adaptive multigrid method

For each of the hierarchical meshes the boundary conditions are applied and the corresponding material data is generated. The displacement vector of the coarsest mesh is initialized by e.g. $\vec{0}$.

(1) Perform iterative step on current mesh by using an applicable basic algorithm as SOR method or JOR method.

Loop through (1) until the approximation error is less than a defined tolerance. If the actual mesh meets the defined resolution then finish else goto (2).

(2) Refine mesh and interpolate the actual displacements of the coarse mesh onto the fine mesh accordingly to the shape functions.

Continue at (1).
5.2 Coarse grid correction by a cyclic multigrid method

While Algorithm 2 quickly leads to locally, smooth approximations, it slows down towards the global balance. The convergence of the fine grid can be accelerated by interactive calculations on coarse grids (Algorithm 3). Several parameters can be varied for an optimal performance of this method. In this study, the JOR method with optimized relaxation factors was applied, whereas the SOR method might be more efficient. Furthermore, a direct solver can be used to correct the residuals on the coarsest grid.

**Algorithm 3: A cyclic multigrid method**

The convergence is accelerated on the fine grid by calculations on any coarser grid. (Notation: \( f \) denotes fine grid; \( c \) denotes coarse grid.)

1. Determine residual forces on the fine grid \( f^r \)
2. Interpolate the residual forces on the coarse grid corresponding to the shape functions: \( f^r \longrightarrow c^r \)
3. Calculate a displacement increment \( c^{inc} \) on the coarse grid
4. Interpolate the displacement increment onto the fine mesh \( c^{inc} \longrightarrow f^{inc} \)
5. Scale displacement increment by an optimum relaxation factor and add it to displacement vector: \( f^{(k)} = \omega f^{(inc)} + f^{inc} \)

A cyclic selection of the coarse grid leads to the cyclic multigrid method.

5.3 Analysis results

The example shown in Fig. 15 was selected to analyze the adaptive, cyclic multigrid method with regard to an increasing number of degrees of freedom and varying energy error tolerance. (The ratio of the moduli of elasticity of inclusion and matrix was 2:1.) As basic iteration scheme the JOR method with iteratively, minimized residual was applied. The problem was computed on one processor of a SGI Altix 3000 with 64-Bit Linux-Operating System (Itanium 2-Processor, 900MHz). Up to a number of 33.5 million degrees of freedom of a 4096x4096 grid the computation time follows a uniform graph (Fig. 16). Although the error tolerance varies by a factor of 10000 the graphs are relative close, which indicates that this strategy is applicable to achieve accurate solutions. It was observed that the curve with error tolerance \( \epsilon = 1 \cdot 10^{-4} \) (line without symbol) is almost identical...
to the following function of the low exponent 1.3 (cross symbol),

\[ t[s] = \frac{N^{1.285}}{424713} \]  

which essentially labels the multigrid method. As a further result Figs. 17 and 18 have been selected to show the effective stresses (of a comparable heterogeneous geometry). Whereas the macroscopic stress state would be constant, the stresses on the mesoscale are highly nonlinear.

## 6 ERROR ANALYSIS

The error is used as measure of convergence. Different error measures are possible, as e.g. the Euclidian norm of the residual forces, the Euclidian norm of the displacement increment \[7\] or the error of the total energy. For achieving a best possible answer of the structural behavior, several error sources have to be considered and assessed. This is proposed on the comparable basis of error energy:

- **Classical discretization error of the finite element method:** The provided bilinear shape functions restrict the flexibility of the structure and lead to artificial stiffening. As the stresses are not compatible along the element boundaries, this defect can be used to estimate the error energy. By various methods the stresses can be recovered and compared to the finite element solution \[12\]. It should be noted, that not only the discretization (element size), but also the load on the element influences the error. A constant element stress state can always exactly be determined.
- **Geometry error due to insufficient resolution of the heterogeneous material**: The heterogeneous structure has to be sufficiently resolved and numerically represented by an element grid. For a sufficient resolution the element size should be significantly smaller than the smallest particle to be explicitly modeled. As increasing of the resolution of the geometrical model is cheaply possible (Section 2.2.5) by a finer element mesh, again, stress recovery on refined particle boundaries can provide a basis to estimate the error energy.

- **Error in the iterative solution of the linear equation system**: Generally the solution error of iterative methods is calculated by the Euclidean norm of the residual forces. However, the Euclidean norm is not directly comparable to an energy error. Therewith an alternative would be to calculate the error energy by the product of displacement vector and residual forces, which represents the global error energy in the system. However, a restriction of the local energy error might additionally be useful.

It is easy to understand that all of these three errors are important for a successful assessment of the structural behavior. Therewith the strategy of the algorithm can be modified taking all errors into account. It can be totally more efficient to reduce the discretization error and the geometrical error by using a finer mesh while allowing for a larger approximation error of the linear equation system.
7 CONCLUSIONS

This article gives an overview on several methods towards establishing a mesoscale analysis of concrete, while still other important aspects had to be left out, e.g. the physical characteristics of concrete. Various methods to generate mesoscale geometries of concrete were compared and a new fast algorithm based on efficient raster checks was introduced. A local finite element numbering scheme was developed and directly integrated into several basic iterative solver methods, which lead to short and efficient formulations. Thereby, this procedure might provide a good basis to implement newer solver methods. Finally, the discussed multigrid method highlights the possibility to solve very large systems in connection with a high resolution mesoscale analysis. However, there is the need to enhance the algorithms and test them on various examples with different boundary conditions.

As emphasis was placed on a realistic material model and several error sources were considered, finally an engineering judgment of the structural behavior would be essential. This will be relevant, as soon as the nonlinear material model has been integrated into our computational analysis.

ACKNOWLEDGEMENT

This research has been supported by the German Research Foundation (DFG) through Collaborative Research Center 524, which is gratefully acknowledged by the authors.
REFERENCES


