Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **11**, No. 6, pp. 1376-1397

DOI: 10.4208/aamm.OA-2019-0002
December 2019

# Two-Level Hierarchical PCG Methods for the Quadratic FEM Discretizations of 2D Concrete Aggregate Models

Yingxiong Xiao*, Heng Chen and Lingjie Xie

*Civil Engineering and Mechanics College, Xiangtan University, Xiangtan 411105, Hunan, China*

Received 3 January 2019; Accepted (in revised version) 21 March 2019

**Abstract.** The concrete aggregate model is considered as a type of weakly discontinuous problem consisting of three phases: aggregates which randomly distributed in different shapes, cement paste and internal transition zone (ITZ). Because of different shapes of aggregate and thin ITZs, a huge number of elements are often used in the finite element (FEM) analysis. In order to ensure the accuracy of the numerical solutions near the interfaces, we need to use higher-order elements. The widely used FEM softwares such as ANSYS and ABAQUS all provide the option of quadratic elements. However, they have much higher computational complexity than the linear elements. The corresponding coefficient matrix of the system of equations is a highly ill-conditioned matrix due to the large difference between three phase materials, and the convergence rate of the commonly used solving methods will deteriorate. In this paper, two types of simple and efficient preconditioners are proposed for the system of equations of the concrete aggregate models on unstructured triangle meshes by using the resulting hierarchical structure and the properties of the diagonal block matrices. The main computational cost of these preconditioners is how to efficiently solve the system of equations by using linear elements, and thus we can provide some efficient and robust solvers by calling the existing geometric-based algebraic multigrid (GAMG) methods. Since the hierarchical basis functions are used, we need not present those algebraic criterions to judge the relationships between the unknown variables and the geometric node types, and the grid transfer operators are also trivial. This makes it easy to find the linear element matrix derived directly from the fine level matrix, and thus the overall efficiency is greatly improved. The numerical results have verified the efficiency of the resulting preconditioned conjugate gradient (PCG) methods which are applied to the solution of several typical aggregate models.

**AMS subject classifications**: 65N55, 65N22

**Key words**: Concrete aggregate model, hierarchical quadratic elements, ill-conditioned matrix, algebraic multigrid method, preconditioner.

*Corresponding author.
*Email:* xyx610xyx@xtu.edu.cn (Y. X. Xiao)

## 1 Introduction

Concrete is considered as a type of composite consisting of three phases: cement paste, coarse aggregate and interfacial transition zone (ITZ) on meso-level. The macroscopic mechanical properties and failure modes of concrete materials are mainly determined by the properties of component materials, the volume content and mixture ratio of aggregates considered [1]. Therefore, it is necessary to establish appropriate mesoscopic models in order to perform some numerical analysis. There are many methods for establishing random geometric models of concrete, see references such as [2–6]. Recently, a hybrid realization method is proposed in order to rapidly obtain 3D concrete spherical aggregate models with high volume fraction which can reach about 65% by combining FORTRAN and ANSYS softwares [7]. This method can also easily generate the polyhedron aggregate model with different shapes.

After the geometry of concrete aggregate model is obtained, a huge number of elements are often used in finite element computation because of different shapes of aggregate and thin ITZs. Moreover, in order to ensure the high accuracy of the finite element solution near the interfaces, it is necessary to use higher-order elements, for example, the widely used FEM softwares as ANSYS and ABAQUS all provide the option of quadratic elements. However, higher-order elements need more computer storage space and have much higher computational complexity than the linear elements. Since the large difference between three phase materials in practice, the coefficient matrix of the system of equations is a highly ill-conditioned matrix. The efficiency of the commonly used solvers will be rapidly reduced. Therefore, it is important to design more efficient solving methods to improve the overall efficiency of finite element analysis for the concrete models with random aggregates under the condition of ensuring the accuracy of numerical solution.

Multigrid method has recently been one of the most effective numerical methods for solving the system of equations arising from the finite element discretizations of partial differential equations, in which algebraic multigrid (AMG) method has become the research focus since it has some advantages for solving large-scale system of equations on unstructured grids [8–12]. If we can use the known information that is easily available in most finite element applications, for instance, the type of the partial differential equation considered, the number of physical unknowns residing in each grid and nodal coordinates on the finest grid level, more robust and efficient AMG as the geometric-based AMG (GAMG) methods can be obtained. There are quite a lot of research results, see for example AMGe method [13, 14] and AMG method based on element agglomeration scheme [15]. Several GAMG methods were designed and analyzed for some different problems in references [16–18]. We would like to refer readers to [20–23] for the recent efforts to apply AMG methods to the system of equations of low-order finite elements for elasticity problems. Recently, some better research results on GAMG methods have been developed in [24–26] for higher-order discretizations of linear elasticity. Hierarchical elements are often used in designing fast solving algorithms and this can help to reduce the computational cost and improve the operation efficiency [27–29].

In this paper, we firstly obtain four types of aggregate geometric models including circular, elliptic, polygonal and mixed shapes with high volume fraction which can reach between 60% and 70% by using the algorithm based on ANSYS parametric design language (APDL). Two simple and efficient preconditioners are then designed for the system of equations arising from the finite element discretizations of concrete aggregate models by combining the hierarchical structure of the coefficient matrix and the properties of the resulting block diagonal matrices. The basic idea of our method is to essentially turn the quadratic discrete systems into mainly solving the corresponding linear discrete systems. In this way, we can obtain preconditioned conjugate gradient (PCG) method whose efficiency of inner iteration has been greatly improved by using the existing efficient GAMG methods. Finally, numerical experiments have been performed for several typical aggregate models to verify the efficiency and the robustness of the resulting methods.

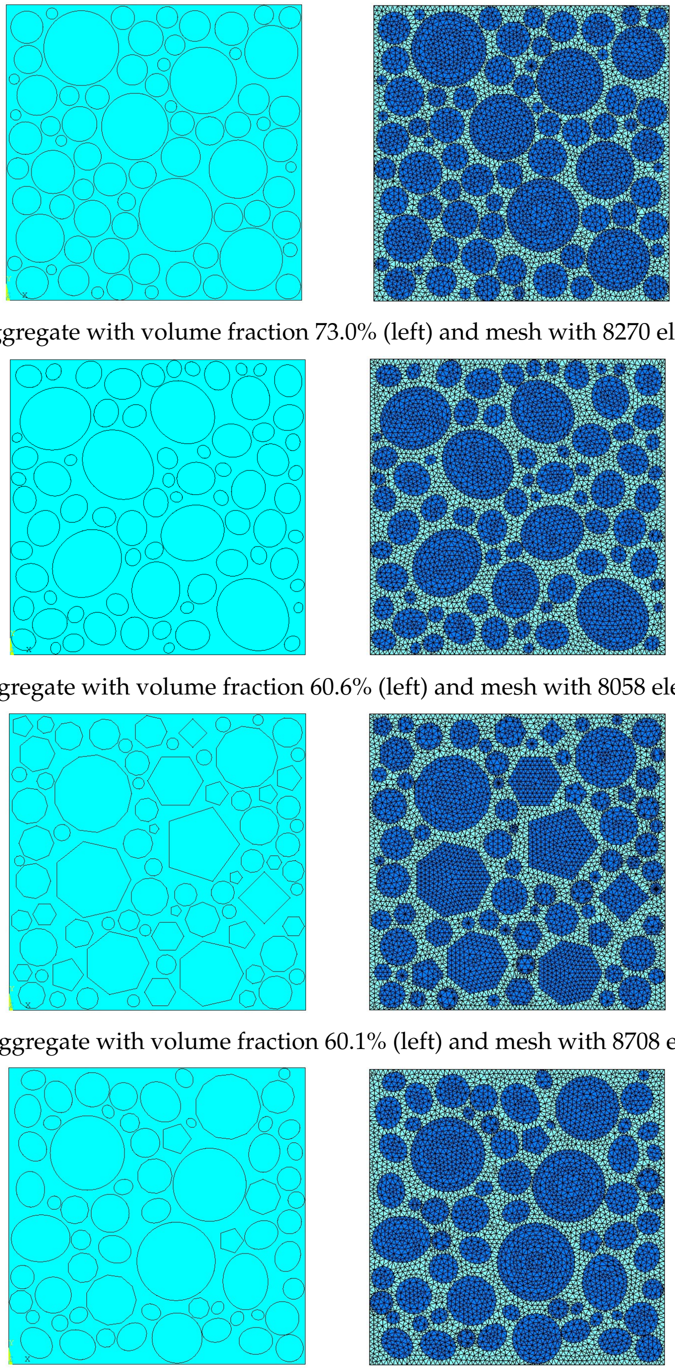
The remainder of the paper is organized as follows. In the next section, we introduce the geometric model of concrete aggregate, mesh generation and the corresponding mathematical model and hierarchical finite element discretization. In Sections 3, two types of preconditioners are introduced for the system of equations arising from the hierarchical quadratic element discretizations. In Section 4, we present several typical examples to verify the efficiency and robustness of the resulting PCG methods. Some concluding remarks are discussed in the final section.

# 2 Aggregate model and hierarchical FEM discretization

## 2.1 Geometric model and mesh generation

In order to establish the geometric model of 3D concrete aggregate, we have designed a special algorithm based on APDL in ANSYS software, see more details in [30]. Here, we directly apply this algorithm to the geometric model of 2D concrete aggregate. The resulting four types of aggregate geometric models are presented in Fig. 1(left) for a square concrete specimen with size of $150mm\times 150mm$, where the particle sizes of circular are determined by two gradation, i.e., the diameter of small stone belongs to $5\sim 20$mm, the diameter of medium stone belongs to $20\sim 40$mm and the corresponding proportion is 5.5:4.5. Then, a conforming quasi-uniform mesh with triangular elements is performed for the determined. geometric model, where we always assume that the interfaces between the aggregates and cement paste are composed of the boundaries of the related triangular elements. The corresponding meshes are also shown in Fig. 1(right).

In addition, the effect of ITZ on the mechanical properties of concrete should be considered in practical application. Concrete aggregate model is regarded as a type of composite consisting of three phases: cement paste, aggregate and ITZ. Four different aggregate geometry models with ITZs and the corresponding meshes are shown in Fig. 2, respectively.



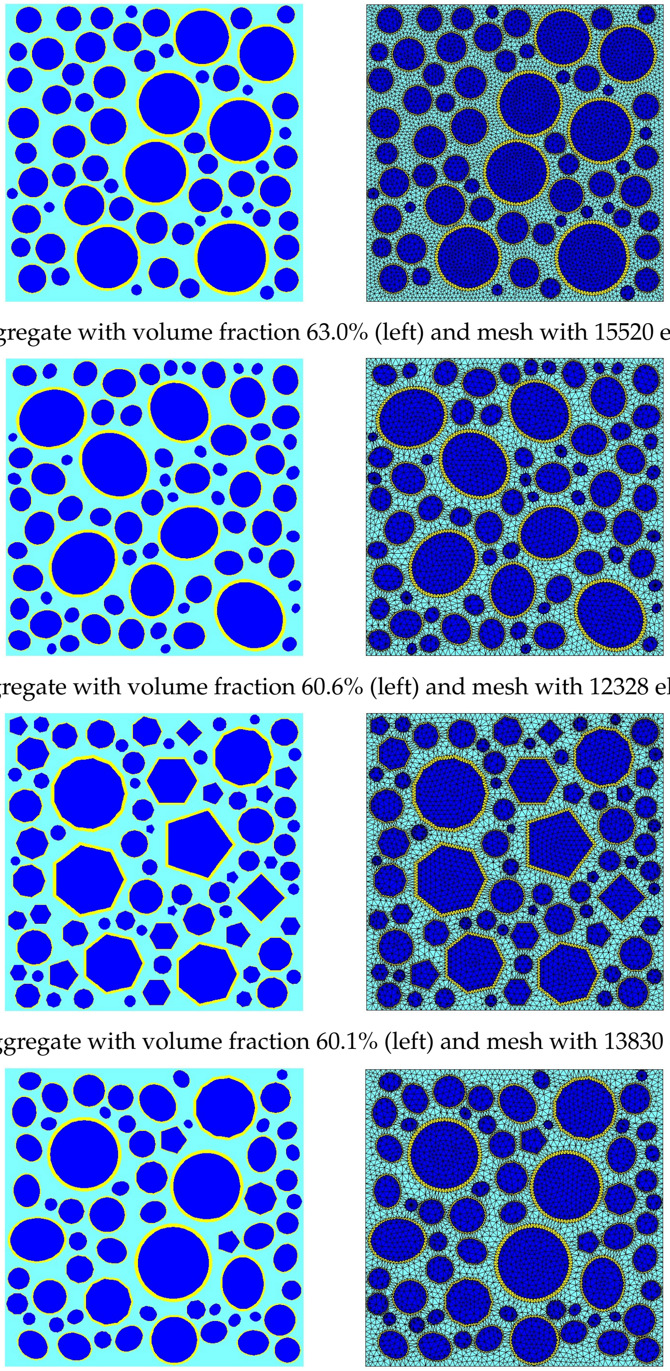
(a) Circular aggregate with volume fraction 73.0% (left) and mesh with 8270 elements (right)

(b) Elliptic aggregate with volume fraction 60.6% (left) and mesh with 8058 elements (right)

(c) Polygonal aggregate with volume fraction 60.1% (left) and mesh with 8708 elements (right)

(d) Mixed aggregate with volume fraction 60.4% (left) and mesh with 8162 elements (right)

Figure 1: Geometric models and conforming quasi-uniform meshes of four types of aggregates.



(a) Circular aggregate with volume fraction 63.0% (left) and mesh with 15520 elements (right)

(b) Elliptic aggregate with volume fraction 60.6% (left) and mesh with 12328 elements (right)

(c) Polygonal aggregate with volume fraction 60.1% (left) and mesh with 13830 elements (right)

(d) Mixed aggregate with volume fraction 60.4% (left) and mesh with 12350 elements (right)

Figure 2: Geometric models and conforming meshes of four types of aggregates with ITZs.

## 2.2 Mathematical model and hierarchical FEM discretization

The concrete aggregate model is considered as a type of weakly discontinuous problem consisting of aggregates, cement paste and ITZ. If we limit the deformation in the linear elastic range and assume that the combination of aggregates, cement paste and ITZs is perfect, the corresponding mathematical model can be expressed as follows

$$\mu\Delta\mathbf{u}+(\lambda+\mu)\nabla\text{div}\,\mathbf{u}=\mathbf{f} \qquad \text{in } \Omega, \tag{2.1a}$$

$$\mathbf{u}=\overline{\mathbf{u}} \qquad \text{on } \Gamma_0, \tag{2.1b}$$

$$\sigma(\mathbf{u})\cdot\mathbf{n}=\mathbf{g} \qquad \text{on } \Gamma_1, \tag{2.1c}$$

where $\Omega\subset\mathbf{R}^2$ is a two-dimensional bounded domain with Lipschitz boundary, $\Gamma_0\bigcap\Gamma_1=\emptyset$ and $mes(\Gamma_0)\neq 0$, $\mathbf{u}$ is the displacement vector, $\mathbf{f}$ is an external force and $\mathbf{g}$ is a surface force on the boundary $\Gamma_1$, whose unit outward normal vector is denoted by $\mathbf{n}$, $\overline{\mathbf{u}}=(\overline{u}_1,\,\overline{u}_2)^{\mathrm{T}}$ is the known displacement function and $\sigma$ is the stress tensor denoted by $\mathbf{u}$. The Lamé constants $\lambda$ and $\mu$ can be expressed with Young's modulus $E$ and Poisson's ratio $\nu$ as follows:

$$\lambda^{(l)}=\frac{E^{(l)}\nu^{(l)}}{(1-(\nu^{(l)})^2)}, \qquad \mu^{(l)}=\frac{E^{(l)}}{2(1+\nu^{(l)})}, \tag{2.2}$$

where $l=0,1,\cdots,M,M+1,\cdots,M+M_I$, the index "0" denotes the cement paste, $M$ is the number of aggregate, $M_I$ is the number of ITZs. The domains of cement paste, aggregate and ITZs are denoted by $\Omega_0$ and $\Omega_l$, $l=1,\cdots,M,M+1,\cdots,M+M_I$, respectively. It is obvious that

$$\Omega_l\subset\Omega, \quad \overline{\Omega}=\bigcup_{l=0}^{M+M_I}\bar{\Omega}_l \quad \text{and} \quad \Omega_l\bigcap\Omega_m=\emptyset,$$

for $l\neq m$.

Let $\mathbf{H}^1_\Gamma:=\{\mathbf{v}\in H^1(\Omega)^2|\mathbf{v}|_{\Gamma_0}=\mathbf{0}\}$. In what follows, we consider the corresponding weak formulation of the problem (2.1) by using the divergence theorem:

Find $\mathbf{u}-\overline{\mathbf{u}}\in\mathbf{H}^1_\Gamma$ such that

$$a(\mathbf{u},\mathbf{v})=\langle\tilde{\mathbf{f}},\mathbf{v}\rangle, \qquad \forall\mathbf{v}\in\mathbf{H}^1_\Gamma, \tag{2.3}$$

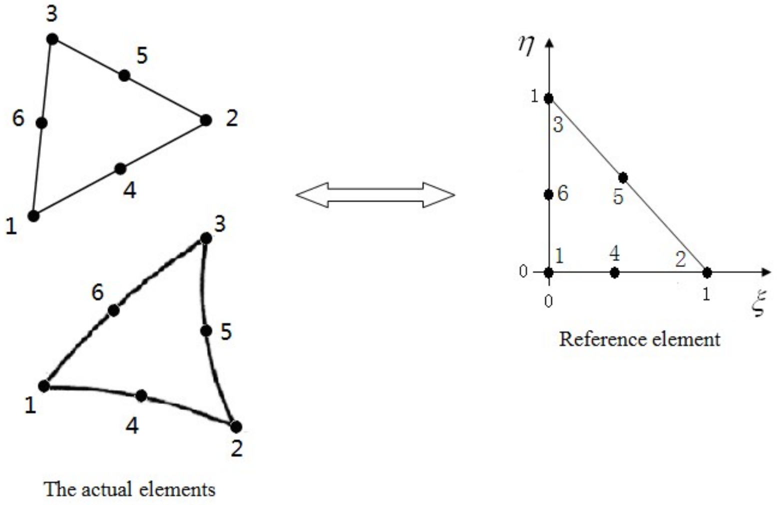
where

$$a(\mathbf{u},\mathbf{v})=\int_\Omega\varepsilon(\mathbf{u}):\sigma(\mathbf{v})d\Omega, \quad \langle\tilde{\mathbf{f}},\mathbf{v}\rangle=\int_\Omega\mathbf{f}\cdot\mathbf{v}d\Omega-\int_{\Gamma_1}\mathbf{g}\cdot\mathbf{v}d\Gamma_1, \quad \varepsilon(\mathbf{u}):\sigma(\mathbf{v})=\textstyle\sum_{ij}\varepsilon(\mathbf{u})_{ij}\sigma(\mathbf{v})_{ij},$$

in which $\varepsilon(\mathbf{u})$ is the strain tensor denoted by $\mathbf{u}$.

Let $T^h$ be a triangle partition of the domain $\Omega$, where $h$ is the maximal diameter of all elements in $T^h$. We introduce the Lagrangian finite element space of $p$ ($p\geq 2$) order as follows

$$\mathbf{V}_h^{(p)}=\{\mathbf{v}_h^{(p)}|\mathbf{v}_h^{(p)}\in C(\bar{\Omega})^2,\ \mathbf{v}_h^{(p)}|_\tau\in P_p(\tau)^2,\ \forall\tau\in T^h\}, \tag{2.4}$$



The actual elements

Reference element

Figure 3: Quadratic elements (left) and the reference element (right).

where $P_p$ is the set of polynomials of degrees not more than $p$.

The finite element solution of problem (2.3) $\mathbf{u}_h^p \in \mathbf{V}_h^p$ satisfies

$$a(\mathbf{u}_h^{(p)},\mathbf{v}_h^{(p)}) = \left\langle \tilde{\mathbf{f}},\mathbf{v}_h^{(p)} \right\rangle, \tag{2.5}$$

for any given $\mathbf{v}_h^{(p)} \in \mathbf{V}_h^{(p)} \bigcap \mathbf{H}_\Gamma^1$.

In the actual engineering computation, we often use quadratic elements with higher precision. For example, the widely used FEM softwares as ANSYS and ABAQUS provide the option of quadratic elements. In general, higher-order elements are not preferred to be used because of the complexity of the calculation and the numerical instability such as Runge phenomenon. Two typical triangle elements with straight and curved edges are presented in Fig. 3(left).

Consider a reference quadratic element with 6 nodes in the local coordinate system $(\xi\eta)$ as shown in Fig. 3(right). In order to establish multi-level iterations or preconditioners easily, we often use hierachical elements. The hierarchical basis is defined in terms of local coordinates on the reference element. In this paper, the corresponding hierarchical basis functions are chosen as follows

$$\begin{array}{lll} H_1 = \xi, & H_4 = 4\xi\eta, & \\ H_2 = \eta, & H_5 = 4\eta(1-\xi-\eta), & 0 \le \xi,\eta \le 1, \\ H_3 = 1-\xi-\eta, & H_6 = 4\xi(1-\xi-\eta). & \end{array}$$

The unknowns associated with the vertices are ordered first and those associated with the midside nodes are ordered last. By assembling the equations for all elements and considering the boundary conditions, we can obtain the global system of linear algebraic equations with some block structure as follows

$$\mathbf{KU}=\mathbf{F}, \tag{2.6}$$

or

$$\begin{bmatrix}\mathbf{K}_{vv} & \mathbf{K}_{vm}\\ \mathbf{K}_{mv} & \mathbf{K}_{mm}\end{bmatrix}\begin{bmatrix}\mathbf{u}_v\\ \mathbf{u}_m\end{bmatrix}=\begin{bmatrix}\mathbf{F}_v\\ \mathbf{F}_m\end{bmatrix}, \tag{2.7}$$

where $\mathbf{U}=[\mathbf{u}_v,\mathbf{u}_m]^{\mathrm{T}}=[U_1,U_2,\cdots,U_N]^{\mathrm{T}}$ is the displacement vector, $\mathbf{F}=[\mathbf{F}_v,\mathbf{F}_m]^{\mathrm{T}}=[F_1,F_2,\cdots,F_N]^{\mathrm{T}}$ is the node load vector, $N=2(n_v+n_m)$ in which $n_v$ and $n_m$ are the number of vertices and the number of midside nodes after the Dirichlet boundary conditions have been applied, respectively. The global stiffness matrix $\mathbf{K}$ is symmetric and positive definite. The block matrices $\mathbf{K}_{vv}$, $\mathbf{K}_{mm}$ and $\mathbf{K}_{vm}=\mathbf{K}_{mv}^{\mathrm{T}}$ are defined by the relations

$$\begin{aligned}
&(\mathbf{K}_{vv}\mathbf{v}_v,\mathbf{w}_v)=a(\mathbf{v}_v,\mathbf{w}_v), &&\forall \mathbf{v}_v,\mathbf{w}_v\in\mathbf{V}_v,\\
&(\mathbf{K}_{mm}\mathbf{v}_m,\mathbf{w}_m)=a(\mathbf{v}_m,\mathbf{w}_m), &&\forall \mathbf{v}_m,\mathbf{w}_m\in\mathbf{V}_m,\\
&(\mathbf{K}_{mv}\mathbf{v}_m,\mathbf{w}_v)=a(\mathbf{v}_m,\mathbf{w}_v), &&\forall \mathbf{v}_v\in\mathbf{V}_v,\quad \mathbf{v}_m\in\mathbf{V}_m,
\end{aligned}$$

where $\mathbf{V}_v$ and $\mathbf{V}_m$ is the FEM spaces spanned by the vertex and midside node basis functions, respectively.

# 3 PCG methods for the hierarchical quadratic FEM discretizations

## 3.1 Analysis of condition numbers

In what follows, we present the condition numbers of the quadratic element matrix $\mathbf{K}$ and two diagonal block matrices $\mathbf{K}_{vv}$ and $\mathbf{K}_{mm}$ for the circular aggregate model. Consider a square concrete specimen with size $150mm\times150mm$, where the particle sizes of circular are determined by two gradation. Assume that $E_1=13.4$GPa and $\nu_1=0.25$ for cement paste, $E_2=74.5$GPa and $\nu_2=0.15$ for aggregates, and $E_3=11.0$GPa and $\nu_3=0.20$ for ITZs. The bottom surface of the domain is fixed and the top surface is applied a vertical downward shear stress $\mathbf{g}=(0,t)^{\mathrm{T}}$ with $t=28N/mm^2$ and not considering the gravity. The corresponding geometric model and one sample of conforming quasi-uniform mesh with 8270 elements is presented in Fig. 1(a), respectively, where the number of element for aggregates is 5506. The geometric model with ITZs and one sample of conforming mesh with 15520 elements is presented in Fig. 2(a), respectively, where the number of element for aggregates is 6498, and the number of element for ITZs is 4152. The condition numbers of the quadratic element matrix $\mathbf{K}$, $\mathbf{K}_{vv}$ and $\mathbf{K}_{mm}$ under five different meshes are summarized in Table 1. Note that the symbol "$Cond(\mathbf{A})$" denotes the condition numbers for any given matrix $\mathbf{A}$, which can be defined by $Cond(\mathbf{A})=\frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}$, where $\lambda_{\max}(\mathbf{A})$

Table 1: Condition number of the quadratic element matrices and two diagonal block matrices.

(a) Several problem sizes without ITZs

| #element | $Cond(\tilde{\mathbf{K}})$ | $Cond(\mathbf{K})$ | $Cond(\mathbf{K}_{vv})$ | $Cond(\mathbf{K}_{mm})$ |
|---|---|---|---|---|
| 8270 | 836096.48 | 254919.23 | 116770.16 | 82.73 |
| 14358 | 1463961.92 | 437218.49 | 228563.64 | 97.62 |
| 47824 | 5220618.08 | 1456830.72 | 780407.58 | 97.76 |
| 101838 | $1.015\times10^7$ | 2972006.39 | 1527511.39 | 93.16 |
| 196084 | $2.117\times10^7$ | 5843756.02 | 3844362.49 | 96.62 |

(b) Several problem sizes with ITZs

| #element | $Cond(\tilde{\mathbf{K}})$ | $Cond(\mathbf{K})$ | $Cond(\mathbf{K}_{vv})$ | $Cond(\mathbf{K}_{mm})$ |
|---|---|---|---|---|
| 15520 | 2825904.67 | 875431.18 | 337904.92 | 122.62 |
| 26958 | 4153749.29 | 1247665.96 | 547499.87 | 109.40 |
| 48778 | 7031946.25 | 2126596.20 | 1060885.99 | 99.85 |
| 102262 | $1.756\times10^7$ | 5037138.95 | 2692571.46 | 114.49 |
| 193198 | $2.963\times10^7$ | 8239802.92 | 5204949.95 | 100.55 |

and $\lambda_{\min}(\mathbf{A})$ is the maximum eigenvalue and the minimum eigenvalue, respectively, here $\mathbf{A}=\mathbf{K},\mathbf{K}_{vv}$ and $\mathbf{K}_{mm}$. In addition, we also present the condition numbers of the quadratic element matrix $\tilde{\mathbf{K}}$ obtained by using the standard node basis functions. The corresponding results are also shown in Table 1.

It can be seen that the condition number of the hierarchical stiffness matrices is much smaller than the standard stiffness matrices. However, the hierarchical stiffness matrices are still highly ill-conditioned with increasing problem size. The efficiency of the conjugate gradient (CG) method will be greatly reduced. The diagonal block matrix $\mathbf{K}_{vv}$ is an ill-conditioned matrix with the increase of the problem size, but another diagonal block matrix $\mathbf{K}_{mm}$ is a well-conditioned matrix whose condition number is independent of the mesh size and only related to the material parameters considered. Besides, the hierarchical stiffness matrix is also a highly ill-conditioned matrix due to the large difference between three phase materials, and the convergence rate of CG method will deteriorate. Therefore, the preconditioned conjugate gradient (PCG) method is necessary to be used to solve the resulting highly ill-conditioned discrete system of equations in order to improve the overall efficiency.

## 3.2 Two preconditioners

Assume that **B** is a preconditioner of matrix **K**, then a simple outline of the PCG algorithm is presented as follows:

**Algorithm 3.1** (PCG iteration)**.**

Step 1 Given an initial guess $\mathbf{U}_0$, compute $\mathbf{r}_0=\mathbf{F}-\mathbf{K}\mathbf{U}_0$, $\mathbf{z}_0=\mathbf{B}\mathbf{r}_0$, $\mathbf{p}_0=\mathbf{z}_0$, $\rho_0=(\mathbf{z}_0,\mathbf{r}_0)$ and let $j=0$.

Step 2 Perform the following sub-steps:

(1) $\alpha_j = \frac{(\mathbf{r}_j,\mathbf{z}_j)}{(\mathbf{K}\mathbf{p}_j,\mathbf{p}_j)}$

(2) $\mathbf{U}_{j+1} = \mathbf{U}_j + \alpha_j \mathbf{p}_j$

(3) $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j \mathbf{K}\mathbf{p}_j$

(4) $\mathbf{z}_{j+1} = \mathbf{B}\mathbf{r}_{j+1}, \quad \rho_{j+1} = (\mathbf{z}_{j+1},\mathbf{r}_{j+1})$

(5) $\beta_j = \frac{(\mathbf{r}_{j+1},\mathbf{z}_{j+1})}{(\mathbf{r}_j,\mathbf{z}_j)}$

(6) $\mathbf{p}_{j+1} = \mathbf{z}_{j+1} + \beta_j \mathbf{p}_j$.

Step 3 If $\frac{\rho_{j+1}}{\rho_0} < \varepsilon$ is true, then stop, where $\varepsilon$ is a given control precision; otherwise, let $j = j+1$, then go to Step 2.

It is obvious that the key of PCG algorithm is to present an appropriate approach of computing $\mathbf{z} = \mathbf{B}\mathbf{r}$, where $\mathbf{r}$ is certain given vector. A good preconditioner should have the properties that the action of $\mathbf{K}^{-1}\mathbf{r}$ is easy to compute and $Cond(\mathbf{BK})$ is much smaller than $Cond(\mathbf{K})$. In what follows, we shall design and analyze two types of preconditioners.

(1) AMG-based block diagonal preconditioner

Using the resulting hierarchical structure of matrix $\mathbf{K}$, we take

$$\mathbf{K}_D = diag(\mathbf{K}) = \begin{bmatrix} \mathbf{K}_{vv} & \\ & \mathbf{K}_{mm} \end{bmatrix}. \tag{3.1}$$

Let $\mathbf{B} = BLP(\mathbf{B}_v,\mathbf{B}_m) = \mathbf{K}_D^{-1}$, then $\mathbf{z} = \mathbf{B}\mathbf{r}$ can be decomposed into two decoupled subsystems as follows:

$$\begin{cases} \mathbf{K}_{vv}\mathbf{z}_v = \mathbf{r}_v, \\ \mathbf{K}_{mm}\mathbf{z}_m = \mathbf{r}_m. \end{cases} \tag{3.2}$$

Firstly, let us consider the first system of equations $\mathbf{K}_{vv}\mathbf{z}_v = \mathbf{r}_v$. Since the matrix $\mathbf{K}_{vv}$ is the linear part of the hierarchical quadratic discretizations, that is to say, $\mathbf{K}_{vv}$ is the matrix obtained by using the usual linear Lagrange polynomial basis functions on all vertices. As we all know, AMG method is one of the most efficient solvers to the system of equations arising from the linear FEM discretizations on unstructured meshes, in which the so-called GAMG methods [16, 18, 19, 21] developed in recent years are widely used in solving the system of equations derived from the vector elliptic partial differential equations because of the high efficiency and better robustness.

In this paper, we shall use GAMG method presented in [21] to solve the system of equations $\mathbf{K}_{vv}\mathbf{z}_v = \mathbf{r}_v$. In what follows, we briefly introduce this GAMG method.

Note that the system of equations $\mathbf{K}_{vv}\mathbf{z}_v = \mathbf{r}_v$ can be rewritten in the following form:

$$\begin{bmatrix} \mathbf{K}_{vv}^{xx} & \mathbf{K}_{vv}^{xy} \\ \mathbf{K}_{vv}^{yx} & \mathbf{K}_{vv}^{yy} \end{bmatrix} \begin{bmatrix} \mathbf{z}_v^x \\ \mathbf{z}_v^y \end{bmatrix} = \begin{bmatrix} \mathbf{r}_v^x \\ \mathbf{r}_v^y \end{bmatrix}, \tag{3.3}$$

where $\mathbf{z}_v^x$ and $\mathbf{z}_v^y$ are the $x$ and $y$ components of the displacement fields. Our coarsening process and the construction of prolongation operator for the system matrix $\mathbf{K}_{vv}$ are mostly based on the application of the scalar AMG for each block matrix $\mathbf{K}_{vv}^{xx}$ and $\mathbf{K}_{vv}^{yy}$. More precisely, the main idea is first to construct prolongations $\mathbf{P}_i^v$, $i=x,y$, for example, by applying RS coarsening techniques given in [10] to each block matrix $\mathbf{K}_{vv}^{xx}$ and $\mathbf{K}_{vv}^{yy}$ in the scalar AMG, and then form the global prolongation operator $\mathbf{P}_{vv}$ for $\mathbf{K}_{vv}$ simply as follows:

$$\mathbf{P}_{vv}=\begin{bmatrix}\mathbf{P}_x^v & \mathbf{0}\\ \mathbf{0}^{\mathrm{T}} & \mathbf{P}_y^v\end{bmatrix}. \tag{3.4}$$

The coarse-grid operator is finally formed by the usual Galerkin approach

$$\mathbf{K}_{vv}^c=\mathbf{P}_{vv}^{\mathrm{T}}\mathbf{K}_{vv}\mathbf{P}_{vv}. \tag{3.5}$$

Note that the coarse-grid matrix obtained in above way is still of block structure as shown in (3.3). Thus, we can get the prolongations and the coarse-grid matrix on next grid by repeating the steps above for $\mathbf{K}_{vv}^c$. We denote the resulting AMG method by GAMG. The corresponding approximate solution is denoted by $\tilde{\mathbf{z}}_v$ after calling only one DAMG iteration to solve the system of equations $\mathbf{K}_{vv}\mathbf{z}_v=\mathbf{r}_v$. For convenience, the resulting preconditioner can be represented as $\tilde{\mathbf{B}}_v=\mathbf{B}_v^{GAMG}$.

Next, we consider the second first system of equations $\mathbf{K}_{mm}\mathbf{z}_m=\mathbf{r}_m$. As we know in Section 3, the matrix $\mathbf{K}_{mm}$ is a well-conditioned matrix, whose condition number is independent of the mesh size but depends on Young's modulus $E$ and Poisson's ratio $\nu$. Therefore, we can call the CG method to solve the system of equations $\mathbf{K}_{mm}\mathbf{z}_m=\mathbf{r}_m$. In practical computation, the number of midside nodes is usually very large. In order to improve the efficiency of computation, we choose

$$\mathbf{C}_{mm}^{-1}=diag(\mathbf{K}_{mm})=\begin{bmatrix}\mathbf{K}_{mm}^{xx} & \\ & \mathbf{K}_{mm}^{yy}\end{bmatrix}. \tag{3.6}$$

Thus, $\mathbf{K}_{mm}\mathbf{z}_m=\mathbf{r}_m$ can be transformed into $\mathbf{C}_{mm}\mathbf{z}_m=\mathbf{r}_m$ or

$$\begin{cases}\mathbf{K}_{mm}^{xx}\mathbf{z}_m^x=\mathbf{r}_m^x,\\ \mathbf{K}_{mm}^{yy}\mathbf{z}_m^y=\mathbf{r}_m^y.\end{cases} \tag{3.7}$$

The corresponding approximate solutions are denoted by $\tilde{\mathbf{z}}_m^x$ and $\tilde{\mathbf{z}}_m^y$ after calling one AMG iteration to solve each system of equations in (3.7), respectively. This AMG method is very efficient for the solution of linear systems arising from the discretization of scalar elliptic PDE. Let $\tilde{\mathbf{z}}_m=(\tilde{\mathbf{z}}_m^x,\tilde{\mathbf{z}}_m^y)^{\mathrm{T}}$, then we can obtain $\mathbf{z}_m\approx\tilde{\mathbf{z}}_m$. The resulting preconditioner can be denoted by $\tilde{\mathbf{B}}_m=\mathbf{B}_m^{AMG}$.

In this way, we can get the approximate solutions $\tilde{\mathbf{z}}=(\tilde{\mathbf{z}}_v,\tilde{\mathbf{z}}_m)^{\mathrm{T}}$ by combining the above two steps. The final preconditioner is obtained as follows

$$\mathbf{B}_1\approx\tilde{\mathbf{B}}=BLP(\tilde{\mathbf{B}}_v,\tilde{\mathbf{B}}_m)=diag(\mathbf{B}_v^{GAMG},\mathbf{B}_m^{AMG}). \tag{3.8}$$

The corresponding algorithm of computing $\mathbf{z}=\mathbf{B}_1\mathbf{r}$ is summarized as follows:

**Algorithm 3.2** (Compute $\mathbf{B}_1\mathbf{r}$).

Step 1 Obtain two block diagonal matrices $\mathbf{K}_{vv}^{xx}$ and $\mathbf{K}_{vv}^{yy}$, and then get the GAMG components as interpolation operators, restriction operators and coarse grid matrices on all grids. By choosing efficient smoothers such as Gauss-Seidel or SOR iteration, establish the GAMG method.

Step 2 Given an initial guess $\mathbf{z}_v^{(0)}$, call one GAMG iteration to solve $\mathbf{K}_{vv}\mathbf{z}_v=\mathbf{r}_v$, and get the approximate solution $\tilde{\mathbf{z}}_v$.

Step 3 Given the initial guesses $(\mathbf{z}_m^{(x)})^{(0)}$ and $(\mathbf{z}_m^{(y)})^{(0)}$, call one AMG iteration to solve $\mathbf{K}_{mm}^{xx}\mathbf{z}_m^x=\mathbf{r}_m^x$ and $\mathbf{K}_{mm}^{yy}\mathbf{z}_m^y=\mathbf{r}_m^y$, respectively, and get the approximate solution $\tilde{\mathbf{z}}_m=(\tilde{\mathbf{z}}_m^x,\tilde{\mathbf{z}}_m^y)^{\rm T}$.

Step 4 Let $\tilde{\mathbf{z}}=(\tilde{\mathbf{z}}_v,\tilde{\mathbf{z}}_m)^{\rm T}$ and get the solution $\mathbf{z}=\mathbf{B}\mathbf{r}\approx\tilde{\mathbf{z}}$.

(2) Parallel preconditioner based on auxiliary variational problem

In this subsection, we present another preconditioner denoted by $\mathbf{B}_2$ as a combination of a smoother and a coarse level solver that requires the solution of the system of equations arising from discretizations by linear conforming elements. The corresponding algorithm and the theoretical analysis for the condition number of the proposed preconditioner are presented in reference [25]. Since the hierarchical basis functions are used, we need not present those algebraic criterions to judge the relationships between the unknown variables and the geometric node types, and the grid transfer operators are also trivial. This makes it easy to design the algorithm for computing the preconditioner. The algorithm of computing $\mathbf{B}_2\mathbf{r}$ can be simply described as follows:

**Algorithm 3.3** (Compute $\mathbf{B}_2\mathbf{r}$).

Step 1 Given an initial guess $\mathbf{z}_h^{(0)}$, compute the residual $\mathbf{w}=\mathbf{r}-\mathbf{K}\mathbf{z}_h^{(0)}$, and let $\mathbf{w}=(\mathbf{w}_v,\mathbf{w}_m)^{\rm T}$.

Step 2 Call one symmetric Gauss-Seidel iteration to solve $\mathbf{K}\mathbf{z}=\mathbf{w}$ on the quadratic FEM space $\mathbf{V}_h^{(2)}$, and the corresponding approximate solution is denoted by $\mathbf{z}_h$.

Step 3 Call one GAMG iteration to solve $\mathbf{K}_{vv}\mathbf{z}_v=\mathbf{w}_v$ on the linear FEM space $\mathbf{V}_h^{(1)}$, and the corresponding approximate solution is denoted by $\mathbf{z}_h^v$.

Step 4 Obtain that $\mathbf{B}_2\mathbf{r}=\mathbf{z}=\mathbf{z}_h+\begin{pmatrix}\mathbf{z}_h^v\\ \mathbf{0}\end{pmatrix}$.

This method needs very little programming effort in that it consists of only one iteration of simple relaxation method and only one iteration of certain existing robust AMG solver for linear conforming FEM equations. In addition, the process of Step 2 and Step

3 in Algorithm 3.3 can be performed in parallel. This implies that the proposed PCG method is appropriate for large-scale computations that often arise in the rapidly developing area.

## 4 Numerical experiments

In this section, some numerical experiments are performed to verify the efficiency and robustness of the resulting PCG methods based on two preconditioners proposed in Section 3 for solving the hierarchical quadratic system of Eq. (4.1) of the concrete aggregate models.

**Example 4.1.** Consider a square concrete specimen with size of $150mm\times 150mm$, where the particle sizes of circular are determined by two gradation. Assume that $E_1=13.4$GPa and $\nu_1=0.25$ for cement paste, $E_2=74.5$GPa and $\nu_2=0.15$ for aggregates, and $E_3=11.0$GPa and $\nu_3=0.20$ for ITZs. The bottom surface of the domain is fixed and the top surface is applied a vertical downward shear stress $\mathbf{g}=(0,t)^{\rm T}$ with $t=28N/mm^2$ and not considering the gravity. The corresponding geometric models without ITZs and with ITZs and one sample of conforming meshes for four types of concrete aggregate models are presented in Fig. 1 and Fig. 2, respectively.

The resulting PCG methods are then applied to the solution of the hierarchical quadratic discrete system of Eq. (4.1). The corresponding numerical results are summed as follows. The corresponding stopping criteria is defined by

$$\left\|\mathbf{B}\mathbf{r}^{(k)}\right\| \Big/ \left\|\mathbf{B}\mathbf{r}^{(0)}\right\| \le 10^{-6}, \tag{4.1}$$

where $\mathbf{r}^{(k)}$ is the residual vector at the $k$-th iteration. We choose zero vector as our initial guess. In order to illustrate the advantages and disadvantages of several PCG methods, we have numerically compared them with several widely used PCG methods, which are the popular ILU(0)-PCG described in [31], which is the preconditioned CG method with a popular ILU(0) preconditioner, the AGMG-FCG method, which is the flexible conjugate gradient (FCG) preconditioned with an aggregation-based algebraic multigrid (AGMG) recently proposed in [32] based on the unsmoothed interpolation matrix, and the AMG-PCG method by selecting coarse points based on MIS technique (i.e., PCG method based on AMG01 presented in [19]). In all the tables, the symbols "# element" denotes the number of elements, "#dof" and "Iter" denote the total degree of freedom and the number of iterations required to achieve control accuracy and "CPU/s" denotes the CPU time in seconds. The numerical experiments in this paper are performed on the PC computer with the Inter(R) Xeon(R) CPU E3-1230 V6 under Windows10 environment.

(1) Case of circular aggregates

The corresponding iteration counts and CPU time of four PCG methods for the hierarchical quadratic system of equations are summed in Table 2.

Table 2: Iteration counts and CPU time of several PCG methods for circular aggregate models.

(a) Several problem sizes without ITZs

| | | ILU(0)-PCG | | AGMG-FCG | | AMG-PCG | | $\mathbf{B}_1$-PCG | | $\mathbf{B}_2$-PCG | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #element | #dof | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s |
| 8270 | 33332 | 218 | 2.51 | – | – | 153 | 4.89 | 25 | 1.38 | 24 | 0.79 |
| 14358 | 57768 | 280 | 5.30 | 213 | 11.05 | 184 | 10.20 | 22 | 2.14 | 22 | 1.22 |
| 47824 | 191896 | 491 | 28.25 | 346 | 53.09 | 347 | 62.20 | 23 | 7.40 | 21 | 3.86 |
| 101838 | 408224 | 701 | 82.51 | 501 | 161.57 | 502 | 193.22 | 23 | 13.62 | 22 | 8.48 |

(b) Several problem sizes with ITZs

| | | ILU(0)-PCG | | AGMG-FCG | | AMG-PCG | | $\mathbf{B}_1$-PCG | | $\mathbf{B}_2$-PCG | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #element | #dof | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s |
| 15520 | 62380 | – | – | – | – | 209 | 12.87 | 26 | 2.77 | 25 | 1.50 |
| 26958 | 108232 | – | – | 276 | 24.95 | 270 | 28.84 | 25 | 4.65 | 24 | 2.46 |
| 48778 | 195712 | 513 | 30.08 | 409 | 64.59 | 371 | 70.97 | 25 | 8.73 | 23 | 4.21 |
| 102262 | 409908 | 786 | 95.54 | 579 | 190.60 | 531 | 209.63 | 29 | 16.87 | 23 | 8.94 |

Table 3: Iteration counts and CPU time of several PCG methods for elliptic aggregate models.

(a) Several problem sizes without ITZs

| | | ILU(0)-PCG | | AGMG-FCG | | AMG-PCG | | $\mathbf{B}_1$-PCG | | $\mathbf{B}_2$-PCG | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #element | #dof | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s |
| 8058 | 32484 | 211 | 2.40 | 171 | 4.61 | 157 | 5.21 | 24 | 1.28 | 23 | 0.73 |
| 15898 | 63948 | 294 | 6.11 | 237 | 12.41 | 214 | 13.75 | 23 | 2.44 | 22 | 1.36 |
| 47498 | 190592 | 449 | 25.99 | 400 | 61.56 | 358 | 66.64 | 23 | 7.35 | 22 | 3.98 |
| 101288 | 406024 | 695 | 81.98 | 577 | 187.81 | 516 | 202.27 | 27 | 15.85 | 24 | 9.32 |

(b) Several problem sizes with ITZs

| | | ILU(0)-PCG | | AGMG-FCG | | AMG-PCG | | $\mathbf{B}_1$-PCG | | $\mathbf{B}_2$-PCG | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #element | #dof | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s |
| 12328 | 49552 | – | – | 216 | 10.18 | 182 | 9.03 | 28 | 2.64 | 27 | 1.40 |
| 26840 | 107760 | – | – | 308 | 27.82 | 267 | 28.63 | 26 | 4.73 | 24 | 2.44 |
| 48760 | 195640 | 505 | 29.72 | 419 | 66.45 | 374 | 65.83 | 24 | 8.75 | 23 | 4.25 |
| 102378 | 410372 | 790 | 93.41 | 535 | 177.26 | 533 | 213.23 | 26 | 15.52 | 24 | 9.33 |

(2) Case of elliptic aggregates

The iteration counts and CPU time of four PCG methods for the hierarchical quadratic system of equations for elliptic aggregate models are summed in Table 3.

(3) Case of polygonal aggregates

The iteration counts and CPU time of four PCG methods for the hierarchical quadratic system of equations for polygonal aggregate models are summed in Table 4.

(4) Case of mixed aggregates

The iteration counts and CPU time of four PCG methods for the hierarchical quadratic system of equations for mixed aggregate models are summed in Table 5.

**Remark 4.1.** The symbol ″–″ in all tables above means that the method chosen has failed to solve the resulting system of equations. For the concrete aggregate models with ITZs,

Table 4: Iteration counts and CPU time of several PCG methods for polygonal aggregate models.

(a) Several problem sizes without ITZs

| | | ILU(0)-PCG | | AGMG-FCG | | AMG-PCG | | $\mathbf{B}_1$-PCG | | $\mathbf{B}_2$-PCG | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #element | #dof | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s |
| 8708 | 35084 | 246 | 3.19 | 183 | 5.68 | 165 | 5.62 | 24 | 1.59 | 23 | 0.89 |
| 16904 | 67972 | 326 | 7.16 | 243 | 13.61 | 218 | 14.09 | 23 | 2.64 | 22 | 1.46 |
| 49438 | 198352 | 485 | 29.13 | 397 | 63.48 | 357 | 66.59 | 23 | 7.64 | 22 | 4.16 |
| 104530 | 418992 | 698 | 88.42 | 570 | 190.78 | 509 | 189.13 | 26 | 15.64 | 22 | 8.96 |

(b) Several problem sizes with ITZs

| | | ILU(0)-PCG | | AGMG-FCG | | AMG-PCG | | $\mathbf{B}_1$-PCG | | $\mathbf{B}_2$-PCG | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #element | #dof | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s |
| 13830 | 55560 | – | – | 268 | 14.26 | 211 | 10.98 | 26 | 2.83 | 26 | 1.54 |
| 27252 | 109408 | – | – | 301 | 29.81 | 285 | 29.43 | 25 | 5.43 | 25 | 2.87 |
| 50892 | 204168 | 554 | 33.96 | 422 | 70.48 | 384 | 72.83 | 23 | 10.02 | 22 | 4.32 |
| 109402 | 438468 | 763 | 94.58 | 612 | 223.61 | 552 | 215.71 | 29 | 18.11 | 24 | 9.96 |

Table 5: Iteration counts and CPU time of several PCG methods for mixed aggregate models.

(a) Several problem sizes without ITZs

| | | ILU(0)-PCG | | AGMG-FCG | | AMG-PCG | | $\mathbf{B}_1$-PCG | | $\mathbf{B}_2$-PCG | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #element | #dof | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s |
| 8162 | 32900 | 231 | 2.84 | – | – | 155 | 5.17 | 26 | 1.44 | 26 | 0.82 |
| 16352 | 65764 | 319 | 6.83 | 235 | 12.98 | 213 | 13.45 | 24 | 2.84 | 23 | 1.47 |
| 47820 | 191880 | 459 | 26.90 | 397 | 61.66 | 356 | 64.59 | 23 | 7.44 | 22 | 3.97 |
| 102452 | 410680 | 672 | 80.57 | 566 | 183.84 | 516 | 202.04 | 26 | 15.52 | 23 | 8.93 |

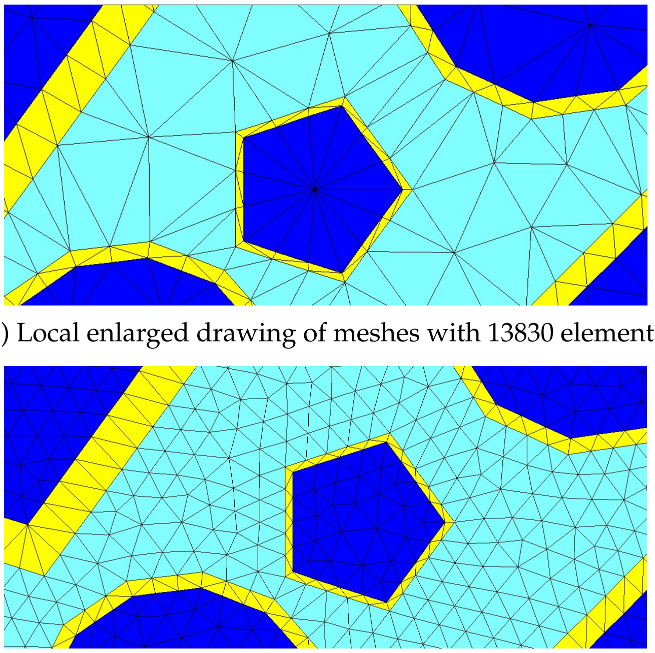
(b) Several problem sizes with ITZs

| | | ILU(0)-PCG | | AGMG-FCG | | AMG-PCG | | $\mathbf{B}_1$-PCG | | $\mathbf{B}_2$-PCG | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #element | #dof | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s |
| 12350 | 49640 | – | – | 215 | 9.98 | 183 | 9.00 | 33 | 2.89 | 33 | 1.68 |
| 26598 | 106792 | 586 | 19.04 | 303 | 27.29 | 276 | 28.73 | 26 | 5.03 | 30 | 2.93 |
| 49092 | 196968 | 516 | 31.29 | 413 | 66.86 | 375 | 70.69 | 24 | 8.95 | 22 | 4.12 |
| 102706 | 411684 | 748 | 92.87 | 583 | 194.45 | 537 | 208.68 | 31 | 17.99 | 25 | 9.58 |

the corresponding mesh quality is poor because of thin ITZs if the number of elements is not much enough or the distorted elements appear on local domains. The ILU(0) preconditioner of hierarchical quadratic element discrete system do not satisfy the positive definiteness, which will lead to failing for the corresponding PCG method. Two local enlarged drawing of meshes are presented in Fig. 4 near a certain aggregate under different element numbers for case of polygonal aggregates with ITZs. It can be seen that the quality of mesh is obviously poor for case of Fig. 4(a), while for case of Fig. 4(b), the quality of mesh is much better and thus five PCG methods are all convergent.

Finally, we present the contour maps of displacement for the mixed aggregate models without ITZs discretized by using hierarchical quadratic elements with 8162 elements as shown in Fig. 5 and Fig. 6, respectively.

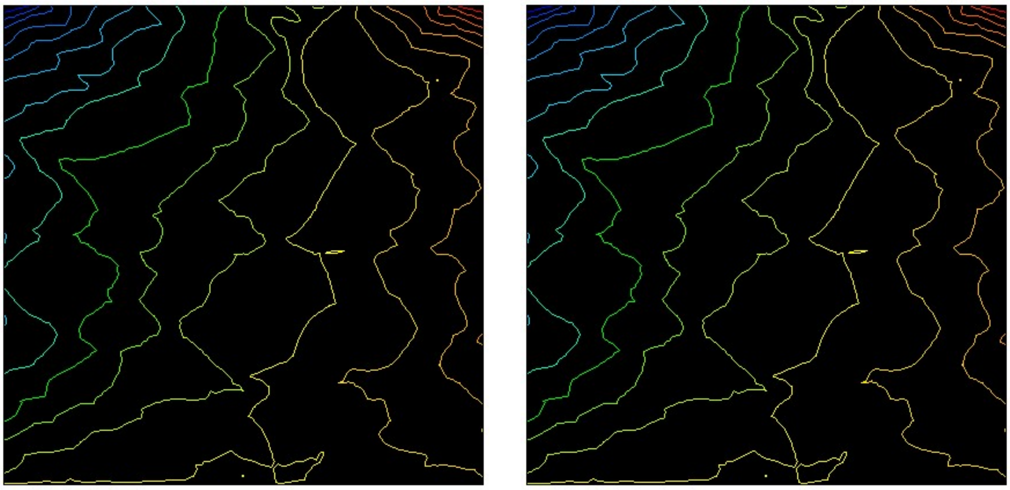
From these numerical results, it can be seen that the hierarchical finite element



(a) Local enlarged drawing of meshes with 13830 elements

(b) Local enlarged drawing of meshes with 50892 elements

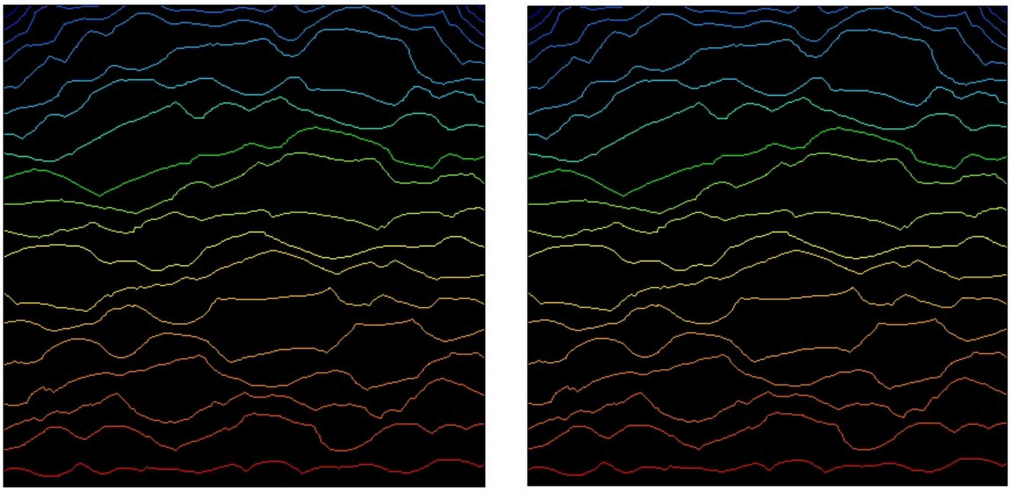
Figure 4: Two local enlarged drawing of meshes near a certain aggregate under different element numbers for case of polygonal aggregates with ITZs.



(a) Standard quadratic element

(b) Hierarchical quadratic element

Figure 5: The contour maps of displacement in the $x$ direction.

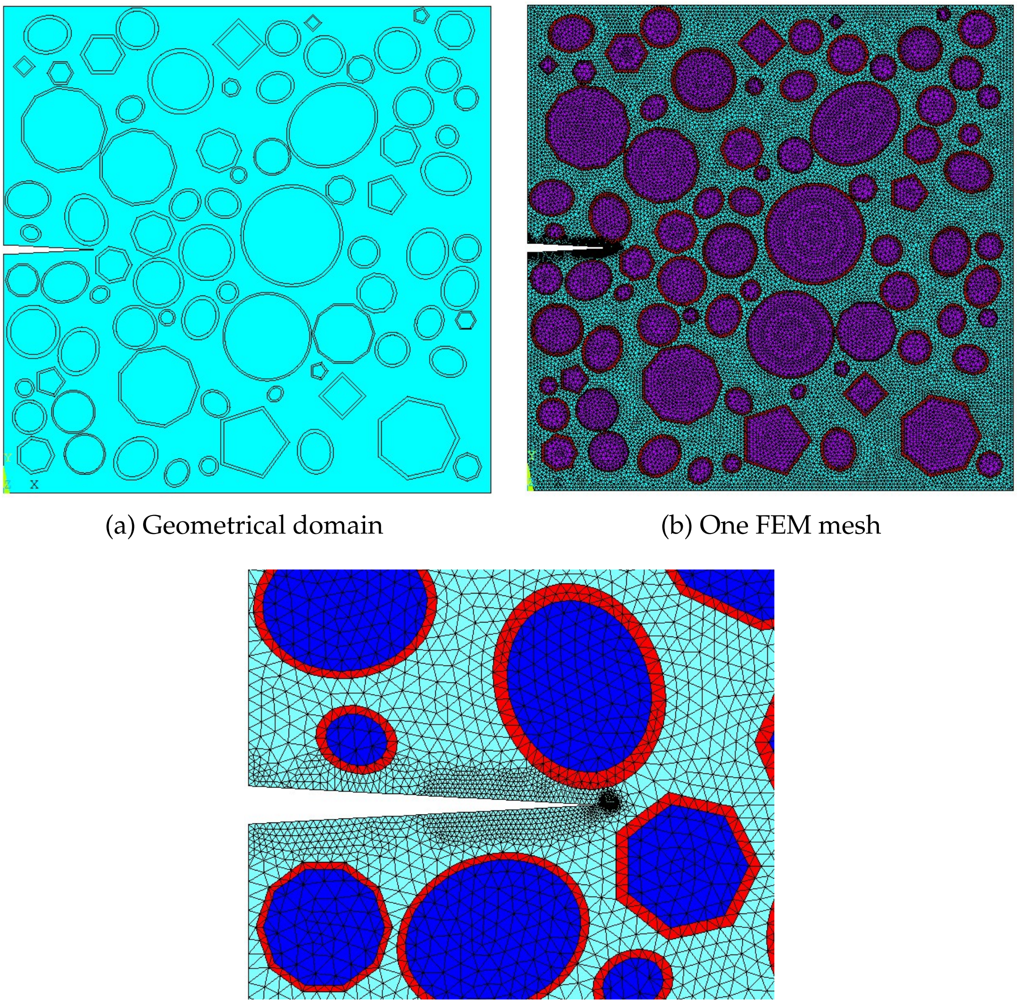


(a) Standard quadratic element (b) Hierarchical quadratic element

Figure 6: The contour maps of displacement in the $y$ direction.

method has the same convergence as the standard finite element method. The proposed preconditioners have good computational efficiency and robustness for solving the hierarchical quadratic element discrete systems of different shape aggregate models, especially for the aggregate models with ITZs. When there are certain poor elements or ill-conditioned elements, ILU(0)-PCG and AGMG-FCG methods fail but the methods designed in this paper still succeed in solving the resulting system of equations considered. Since the hierarchical FEM method is used, we need not to take extra computational cost in constructing grid transformation operators as restriction operator and interpolation operator and the linear FEM matrix used in the two-level PCG method. Considering that the matrix $\mathbf{K}_{vv}$ does not change in running PCG iteration, we generate only one time all the information including the grid transfer operators and the coarse grid matrices needed for the GAMG method. Thus, the efficiency of inner iterative method and further the resulting PCG methods can be greatly improved by fully utilizing the advantages of AMG and GAMG methods in solving large-scale linear FEM system of equations on unstructured grids.

**Example 4.2.** Consider a square concrete specimen with size of $200mm \times 200mm$ as shown in Fig. 7(a), in which there is a small crack at the middle point on the left. The sizes of mixed shape aggregates with ITZ are determined by two gradation. Assume that $E_1 = 30$GPa and $\nu_1 = 0.20$ for cement paste, $E_2 = 60$GPa and $\nu_2 = 0.18$ for aggregates, and $E_3 = 15$GPa and $\nu_3 = 0.25$ for ITZs. The bottom surface of the domain is fixed and the top surface is applied a vertical downward shear stress $\mathbf{g} = (0,t)^{\mathrm{T}}$ with $t = 500N/mm^2$ and not considering the gravity. A conforming mesh with 43137 elements is presented in Fig. 7(b), where local mesh refinement is done near crack tip.



(a) Geometrical domain

(b) One FEM mesh

(c) Local enlarged drawing near crack

Figure 7: Geometric models of mixed aggregates with ITZs and small crack and conforming mesh with 43137 elements.

We first present the condition numbers of the quadratic element matrices $\mathbf{K}$ and $\tilde{\mathbf{K}}$ and two diagonal block matrices $\mathbf{K}_{vv}$ and $\mathbf{K}_{mm}$ of $\mathbf{K}$ as shown in Table 6. The corresponding iteration counts and CPU time of the resulting two PCG methods proposed in this paper

Table 6: Condition number of the quadratic element matrices and two diagonal block matrices for crack problem.

| #element | $Cond(\tilde{\mathbf{K}})$ | $Cond(\mathbf{K})$ | $Cond(\mathbf{K}_{vv})$ | $Cond(\mathbf{K}_{mm})$ |
|---|---|---|---|---|
| 43137 | 6683450.13 | 2103343.79 | 765162.35 | 86.57 |
| 114357 | $1.608\times10^7$ | 4857306.45 | 1823339.72 | 74.46 |

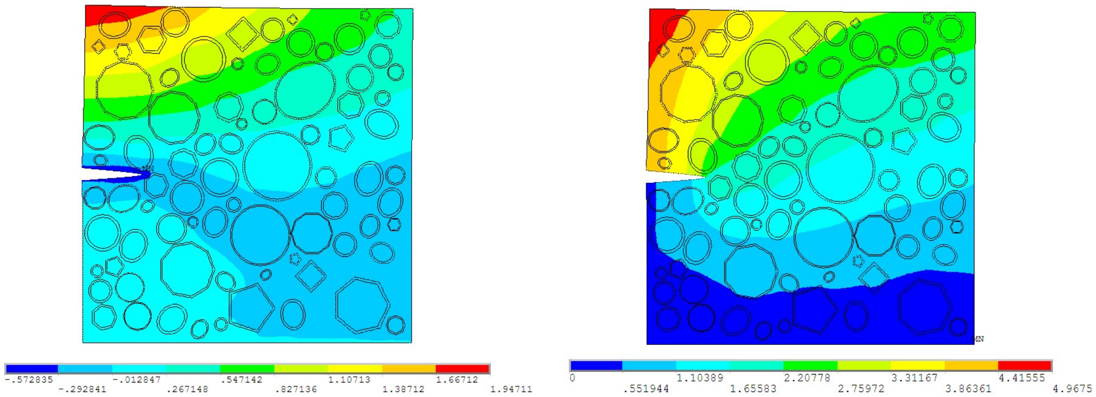
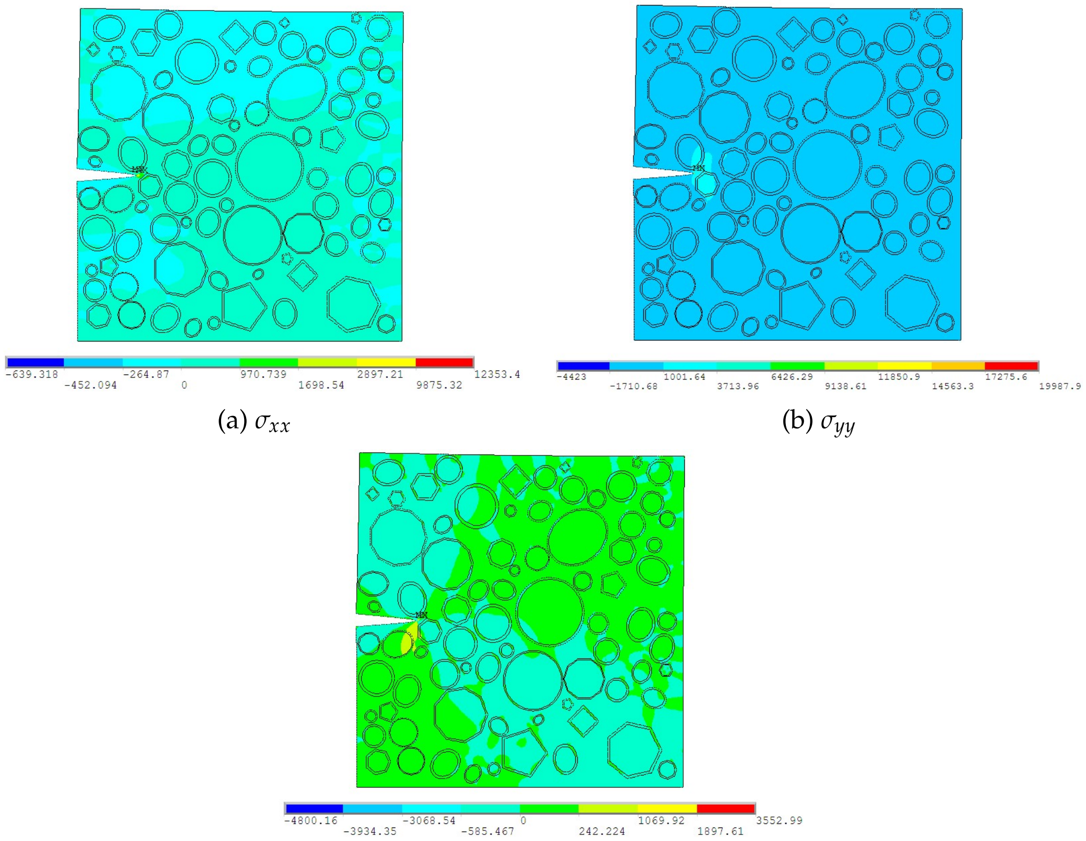


Figure 8: Displacement nephograms the $x$ direction (left) and in the $y$ direction (right) with 43137 elements.



(a) $\sigma_{xx}$

(b) $\sigma_{yy}$

(c) $\tau_{xy}$

Figure 9: Stress nephograms for the aggregate models with ITZs and crack, where #element is 43137.

Table 7: Iteration counts and CPU time of two resulting PCG methods for crack problem.

| | | ILU(0)-PCG | | AGMG-FCG | | AMG-PCG | | $\mathbf{B}_1$-PCG | | $\mathbf{B}_2$-PCG | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #element | #dof | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s | Iter | CPU/s |
| 43137 | 173378 | 517 | 28.24 | 356 | 52.37 | 324 | 52.52 | 24 | 7.91 | 22 | 4.03 |
| 114357 | 458670 | Out of memory | | 538 | 210.99 | Out of memory | | 31 | 18.93 | 25 | 9.79 |

and other three commonly used PCG methods for the hierarchical quadratic system of equations are summed in Table 7. The displacement and stress nephograms under the case of 43137 elements are presented in Fig. 8 and Fig. 9, respectively. It can be seen that the ill-condition of the hierarchical quadratic FEM matrix for the aggregate model with crack is mainly determined by the corresponding linear FEM matrix, and the matrix $\mathbf{K}_{mm}$ assembled by midpoint nodes is a well-conditioned matrix. The two PCG methods are also very efficient for solving hierarchical quadratic system of equations for this crack problem, and by contrast, the $\mathbf{B}_2$- PCG method has better computational efficiency and robustness. By taking the case of 43137 elements as an example, the iteration counts and the CPU times of the $\mathbf{B}_2$- PCG method are 22 and 4.03 seconds, respectively, which are better than the $\mathbf{B}_1$-PCG method with 24 iterations and 7.91 seconds.

# 5 Concluding remarks

In FEM analysis of the concrete aggregate models, using higher-order elements can greatly improve the accuracy of the numerical results, especially can ensure the accuracy of the numerical solutions near the ITZs. It is necessary to design more efficient solvers for the resulting system of equations in order to improve the overall efficiency of FEM method. In this paper, two simple and efficient preconditioners are proposed for the hierarchical quadratic system of equations. The resulting two-level hierarchical PCG methods have better computational efficiency. These preconditioners are essentially deduced into solving the linear system of equations. Thus, the efficiency of inner iterative method and further the resulting PCG methods can be greatly improved by using the existing efficient GAMG methods to solve the linear system of equations.

While only limited numerical experiments have been performed, the numerical results are still quite encouraging. More extensive numerical experiments and theoretical estimate for the condition number of the proposed two preconditioners for the hierarchical matrices of multi-phase materials are further expected. Some robust and efficient preconditioners can be obtained by using the known information that is easily available in most FEM applications, for instance, the type of the partial differential equation considered, the number of physical unknowns residing in each grid and nodal coordinates on the finest grid level and by combining the existing efficient AMG methods. Besides, the proposed algorithms can also be extended to nonlinear FEM numerical simulation of concrete materials and structures, for example, cracking failure, dynamic analysis and so on. Such areas will be a subject of our future research.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant No. 11601462), the Hunan Provincial Natural Science Foundation of China (Grant No. 14JJ2063) and the Scientific Research Fund of Hunan Provincial Education Department (Grant No. 15A183).

## References

[1] MA HUAIFA, CHEN HOUQUN AND LI BAOKUN, *Review on micro-mechanics studies of concrete,* J. China Institute Water, 2(2) (2004), pp. 124–130.

[2] SONG LAIZHONG, SHEN TAO AND YU BO, *The approach to establishing a two-dimensional parameterized aggregate model for concrete simulation,* Eng. Mech., 30(10) (2013), pp. 5–13.

[3] X. F. WANG, Z. J. YANG, J. R. YATES, A. P. JIVKOV AND CH ZHANG, *Monte Carlo simulations of mesoscale fracture modelling of concrete with random aggregates and pores,* Construction and Building Materials, 75 (2015), pp. 35–45.

[4] GAO ZHENGGUO AND LIU GUANGTING, *Two-dimensional random aggregate structure for concrete,* Journal of Tsinghua University (Science & Technology), 43(5) (2003), pp. 710–714.

[5] SONG XIAOGANG AND YANG ZHICHUN, *A new method to simulate round concrete aggregate generation,* Eng. Mech., 27(1) (2010), pp. 154–159.

[6] SIAMAK SHAHBAZI AND IRAJ RASOOLAN, *Meso-scale finite element modeling of non-homogeneous three-phase concrete,* Case Studies in Construction Materials, 6 (2017), pp. 29–42.

[7] GUO RUIQI, XIAO YINGXIONG AND TANG XIANQIONG, *A fast hybrid realization method for three-dimensional concrete aggregate models,* Journal of Civil, Architectural & Environmental Engineering, 39(5) (2017), pp. 100–107.

[8] A. BRANDT, S. MCCORMICK AND J. RUGE, Algebraic Multigrid (AMG) for Sparse Matrix Equations, in Sparsity and its Applications, Evans DJ (ed.), Cambridge University Press: Cambridge, 1984, pp. 257–284.

[9] A. BRANDT, *Algebraic multigrid theory: the symmetric case,* Appl. Math. Comput., 19 (1986), pp. 23–56.

[10] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid,* in Multigrid Methods, Frontiers Appl. Math., S. F. McCormick editor, SIAM, Philadelphia, 3 (1987), pp. 73–130.

[11] K. STÜBEN, An Introduction to Algebraic Multigrid, in Multigrid, Trottenberg U, Oosterlee CW, Sch ller A (eds), Academic Press: London, 2001, pp. 413–532.

[12] M. GRIEBEL, D. OELTZ AND M. A. SCHWEITZER, *An algebraic multigrid method for linear elasticity,* SIAM J. Sci. Comput., 25 (2003), pp. 385–407.

[13] T. CHARTIER, R. D. FALGOUT, V. E. HENSON, J. E. JONES, T. MANEUFFEL, S. F. MCCORMICK, J. W. RUGE AND P. S. VASSILEVSKI, *Spectral AMGe (AMGe),* SIAM J. Sci. Comput., 20(1) (2003), pp. 1–20.

[14] M. BREZINA, A. J. CLEARY, R. D. FALGOUT, V. E. HENSON AND J. E. JONES ET AL., *Algebraic multigrid based on element interpolation (AMGe),* SIAM J. Sci. Comput., 22 (2000), pp. 1570–1592.

[15] T. V. KOLEV AND P. S. VASSILEVSKI, *AMG by element agglomeration and constrained energy minimization interpolation,* Numer. Linear Algebra Appl., 13 (2006), pp. 771–788.

[16] SHI SHU, JINCHAO XU, YING YANG AND HAIYUAN YU, *An algebraic multigrid method for finite element systems on criss-cross grids,* Adv. Comput. Math., 25 (2006), pp. 287–304.

[17] SHI SHU, DUDU SUN AND JINCHAO XU, *An algebraic multigrids for higher order finite element discretizations,* Computing, 77 (2006), pp. 347–377.

[18] Y. X. XIAO, P. ZHANG AND S. SHU, *An algebraic multigrid method with interpolation reproducing rigid body modes for semi-definite problems in two dimensional linear elasticity,* J. Comput. Appl. Math., 200(2) (2007), pp. 637–652.

[19] YINGXIONG XIAO, SHI SHU, PING ZHANG AND MIN TAN, *An algebraic multigrid method for isotropic linear elasticity on anisotropic meshes,* Int. J. Numer. Methods Biomed. Eng., 26(5) (2010), pp. 534–553.

[20] M. GRIEBEL, D. OELTZ AND M. A. SCHWEITZER, *An algebraic multigrid method for linear elasticity,* SIAM J. Sci. Comput., 25 (2003), pp. 385–407.

[21] XIAO YINGXIONG, ZHOU ZHIYANG AND SHU SHI, *Algebraic multigrid methods for 3D linear elasticity problems on some typical meshes,* Eng. Mech., 28(6) (2011), pp. 11–18.

[22] A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, J. E. JONES, T. A. MANTEUFFEL, S. F. MCCORMICK, G. N. MIRANDA AND J. W. RUGE, *Robustness and scalability of algebraic multigrid,* SIAM J. Sci. Comput., 21(5) (2000), pp. 1886–1908.

[23] A. H. BAKER, TZ. V. KOLEV AND U. M. YANG, *Improving algebraic multigrid interpolation operators for linear elasticity problems,* Numer. Linear Algebra Appl., 7 (2010), pp. 495-–517.

[24] A. EL MALIKI, R. GUÉNETTE AND M. FORTIN, *An efficient hierarchical preconditioner for quadratic discretizations of finite element problems,* Numer. Linear Algebra Appl., 18 (2011), pp. 789-–803.

[25] YINGXIONG XIAO AND SHI SHU, *A robust preconditioner for higher-order finite element discretizations in linear elasticity,* Numer. Linear Algebra Appl., 18(5) (2011), pp. 827–842.

[26] XIAO YINGXIONG, WANG BIAO AND LI ZHENYOU, *GAMG method for higher-order finite element discretizations of modeling weak discontinuities problems,* Chinese Journal of Computational Mechanics, 34(1) (2017), pp. 35–42.

[27] RANJBAR AND AMIR, *An approach to construct algebraic two-grid preconditioners on hierarchical triangular,* Arabian J. Sci. Eng., 36(8) (2011), pp. 1621.

[28] ZHIYONG LIU AND YINNIAN HE, *Two-level hierarchical basis preconditioner for elliptic equations with jump coefficients,* J. Comput. Anal. Appl., 16(1) (2014), pp. 515.

[29] S. B. ADRIAN, F. P. ANDRIULLI AND T. F. EIBERT, *A hierarchical preconditioner for the electric field integral equation on unstructured meshes based on primal and dual Haar bases,* J. Comput. Phys., 330(1) (2017), pp. 365–379.

[30] CHEN HENG, XIAO YINGXIONG AND GUO RUIQI, *Numerical simulation for the concrete aggregate models based on p-version adaptive finite element method,* Proceedings of the 27th National Conference on Structural Engineering, 1 (2018), pp. 165–175.

[31] Y. SAAD, Iterative Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[32] Y. NOTAY, *An aggregation-based algebraic multigrid method,* Tech. Rep. GANMN 08-02, Universite Libre de Bruxelles, Brussels, Belgium, 2008, `http://homepages.ulb.ac.be/∼ynotay`; Electronic Transactions on Numerical Analysis, 37(2010), pp. 123–146.