DELFT UNIVERSITY OF TECHNOLOGY

REPORT 09-08

Preconditioned conjugate gradient method enhanced by deflation of rigid body modes applied to composite materials.

T.B. Jönsthövel

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2009

Preconditioned conjugate gradient method enhanced by deflation of rigid body modes applied to composite materials.

T.B Jönsthövel[*], M.B. van Gijzen[†], C.Vuik[†], C. Kasbergen[*], A. Scarpas[*]

June 30, 2009

Abstract

The introduction of computed x-ray tomography allows for the construction of high quality, material-per-element based 3D meshes in the field of structural mechanics. The use of these meshes enables a shift from meso to micro scale analysis of composite materials like cement concrete, rocks and asphalt concrete. Unfortunately, because of the extremely long execution time, memory and storage space demands, the majority of commercially available finite element packages are not capable of handling efficiently the most computationally demanding operation of the finite element solution process, that is, the inversion of the structural stiffness matrix. To address this issue, an efficient iterative method based upon the preconditioned conjugate gradient method has been developed and is presented in this contribution. It is shown that enhancement of the preconditioned conjugate gradient method with information about the rigid body modes of the aggregates results in an aggregate independent convergence behavior. The resulting number of iterations is bounded by the material behavior of the matrix only.

Key words. Deflation, Preconditioned Conjugate Gradient, Static Analysis, Tomography Scans

1 Introduction

Within the mechanics community homogenization is still widely used to simulate composite materials by means of the finite element (FE) method. However, for more accurate simulations of material response, one-material-per-element FE meshes can be necessary. As a consequence, large finite element meshes are needed for an accurate representation of the matrix and the aggregates and hence very large and thus computational intensive, systems of equations.

[*]Delft University of Technology, Faculty of Civil Engineering, Department of Structural Mechancis, 2628CN Delft, the Netherlands ({t.b.jonsthovel,c.kasbergen,a.scarpas}@tudelft.nl)

[†]Delft University of Technology, Faculty of Information Technology and Systems, Department of Applied Mathematical Analysis, 2628CN Delft, the Netherlands ({m.b.vangijzen, c.vuik}@tudelft.nl)

The FE discretization of the linearized virtual work equation gives rise to a linear system $Ku = f$, in which K is the structural stiffness matrix [14]. In general the stiffness matrix is symmetric positive definite (SPD), hence its inverse exists. Moreover, the stiffness matrix is sparse. For small to medium scale problems parallel direct solvers such as MUMPS [1], PARDISO [15] or SuperLU [3] are good choices with respect to cost and efficiency. However, the performance of parallel direct solvers degrades when solving linear systems corresponding to 3D meshes. The bandwidth of the stiffness matrix, hardware limitations, delays in communication due to overhead and latency and the arithmetic complexity (recursion) induce a boundary on the scalability of parallel direct solvers when applied to 3D problems.

In contrast to direct solvers, iterative solvers have favorable properties for solving linear systems for 3D problems. Iterative solvers are fast, they do not require vast amounts of memory and are highly parallelizable without losing their scalability. Drawbacks can be slow convergence or even stagnation due to ill-conditioned matrices. However, preconditioners can speed up the iterative solution process. We construct a parallelizable iterative method which is more resource efficient and faster compared to available parallel direct methods.

Because the stiffness matrix is SPD the conjugate gradient (CG) method [6] is a natural choice for solving the system. CG is composed of only one matrix-vector multiplication and two inner-products per iteration. In exact arithmetic CG constructs the exact solution within n steps where the stiffness matrix has dimension $n \times n$. Although in theory CG always converges, in practice the amount of iterations of CG is determined by the condition number of the stiffness matrix [5]. Linear systems with large, jumps in coefficients, alike the aggregates and bitumen, have a large condition number, hence slow convergence of CG [18].

Preconditioning is the standard technique for improving the convergence of CG. Common choices of preconditioners are diagonal scaling of the stiffness matrix and incomplete Cholesky factorization without fill in, i.e. IC(0) [10]. However, treating the linear system with a traditional preconditioning technique is not sufficient for our type of application. There is a direct correlation between the rigid body modes and the condition number of the stiffness matrix. By removing the rigid body modes of the aggregates from the stiffness matrix we improve the condition number and hence the convergence of CG. The deflation based preconditioners have successfully been applied within the field of computational fluid dynamics, with excellent results on problems with discontinuous jumps in coefficients [17], [4]. We extend the technique of subdomain deflation, introduced by Nicolaides [12], towards rigid body modes deflation or more precisely kernel deflation to remove the effect of the rigid body modes from the linear system. We note that this is the first successful application of deflation based preconditioning applied to coupled systems of partial differential equations.

The structure of this paper is as follows. The first section describes the problem definition and the properties of the stiffness matrix. The second section describes preconditioned CG and the limitations of standard preconditioning with respect to convergence. We illustrate and explain these limitations by introduction of 3-D cases that involve the simulation of asphaltic concrete, which consists of relatively stiff aggregates embedded in a matrix of soft bitumen, resulting in significant differences in the stiffness between the bitumen

and aggregate elements especially at higher temperatures. The third section describes preconditioned CG enhanced by kernel deflation. We describe how to construct the deflation based preconditioner and we show theoretically and experimentally convergence rates independent of the number of aggregates and the differences in stiffness coefficients. Moreover, we will see that the choice of deflation vectors is based on both sound mathematical and physical arguments.

2 Problem definition

Until recently, because of the extremely long execution time, memory and storage space demands, the majority of FE simulations of composite materials such as asphalt concrete, rocks and cement concrete were performed by means of homogenization techniques. Unfortunately these techniques do not provide an understanding of the actual interaction between the components of the material. Nevertheless, it is known that component interaction is the most critical factor in determining the overall mechanical response of the composite material and, that, by being able to control and specify the characteristics of the interaction, the material designer can not only optimize the mechanical performance but, also, tailor the short and long term response to address specific environmental and/or loading demands.

In the framework of this contribution, asphalt concrete shall be used as an example of a composite material. It consists of a mixture of bitumen, aggregates and air voids. Obviously, the difference between the stiffness of bitumen and the aggregates is significantly large especially at higher temperatures. Moreover, plastic and viscous behavior is not likely to be observed for stone aggregates under normal conditions. There are many benefits when modeling asphalt concrete as a composite since then aging effects, cracking and deterioration of the material due to moisture penetration can be carefully modeled thus providing the desired insight as to why material properties degrade or improve when the individual components of the composite change. This reduces the need for extensive and expensive testing of the material. Different models and parameters for plasticity, elasticity and viscosity can be applied to the different materials of the mixture. Changing the amount of aggregates in the mix does not influence the elastic, plastic and viscous material parameters of the bitumen.

Even though simulation of asphalt concrete at micro scale has strong physical advantages, there are also trade offs to be considered. First of all, the construction of the FE mesh is not as simple as in the homogeneous material case. We have to select a sample of material and make a tomography scan. Finite element meshes have to be developed representing the geometry, the location and the mechanical characteristics of the individual components. Typically this can be achieved by means of Computed Tomography (CT) X-ray scans. Figure 1 shows a typical CT scan of a slice of a cylinder of asphalt concrete. The aggregates, the bitumen and the air voids are clearly visible. Many successive CT slices are necessary for construction of high quality 3D surface renderings by means of specialized software tools like Simpleware ScanFE [16]. Then, additional software like CUBIT [7] is necessary for the generation of 3D material-per-element FE meshes.

Up until recently, the technical limitations of medium range computer hardware and software and the forbidding costs of high end computers did not allow

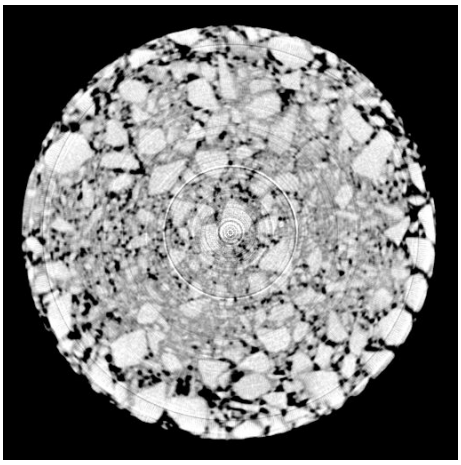

Figure 1: Tomography scan of column of asphaltic material.

the utilization of CT scan produced FE meshes by the majority of academic researchers, since the FE solution of such meshes calls for very large memory and powerful CPUs.

The current developments on computer hardware lead us to the development of scalable, parallel algorithms that can handle large linear systems from both a time and resource (memory and CPU) perspective. The challenge lies in developing an algorithm that is robust, stable and efficient. We will see that by using material specific information we can construct an algorithm that meets all previous requirements.

This contribution addresses the solution of large linear systems that originate from FE discretizations of the CT scans. Only linear elastic material is taken into account. Therefore the structural stiffness matrix remains unchanged during static, distributed load analysis. We consider a system that is subjected to an external load by means of small load steps. Deformation of the system is the natural response to the applied external forces. Hence, at each load step a new force equilibrium has to be established. We compute the force balance with a Newton-Raphson procedure. For elastic problems this reduces to solving equation (1),

$$K\Delta u = \Delta f. \tag{1}$$

Here Δu represents the change of displacement and Δf the force unbalance. The stiffness matrix K is symmetric positive definite for elastic, constrained systems, hence $\forall u \neq 0 : u^T K u > 0$ and all eigenvalues of K are positive. Within the context of mechanics, $\frac{1}{2}u^T K u$ is the strain energy stored within the system for displacement vector u, [2]. Energy is defined as a non-negative entity, hence the strain energy must be non-negative also. When matrix K represents an unconstrained mechanical problem, e.g. a rigid body, the strain energy equals zero for the rigid body displacements as the system remains undeformed and the matrix is positive semi-definite, $\forall u : u^T K u \geq 0$. More specifically, the rigid body modes of any unconstrained volume coincide with the zero-valued eigenvalues of its corresponding stiffness matrix. When a matrix has zero-valued

eigenvalues the kernel $\mathcal{N}(A)$ consists of non-trivial null solutions. Moreover the basis vectors of the kernel of a stiffness matrix represent the principal directions of the rigid body modes. In general, two types of rigid body modes exist, translations and rotations. In three dimensions this implies six possible rigid body modes and hence six kernel vectors can be associated with the rigid body modes at most.

3 Preconditioned CG

Because K is SPD, CG [6] will be used to solve (1) iteratively. The CG method is based on minimizing the energy error of the $i-th$ solution over the Krylov subspace,

$$\mathcal{K}^{i-1}(K; r_0) = \text{span}\{r_0, Kr_0, ..., K^{i-1}r_0\}. \tag{2}$$

The energy norm is defined as $\|u\|_K = \left(u^T K u\right)^{\frac{1}{2}}$. We should note that minimizing the error in the K-norm is in fact minimizing the strain energy over the Krylov subspace $\mathcal{K}^{i-1}(K; r_0)$. This implies that for a given static distributed load we construct a displacement vector that has an optimal distribution of the force over the material.

In [5, Theorem 10.2.6] it is stated that after i iterations the error of CG is bounded by,

$$\|u - u_i\|_K \leq 2\|u - u_0\|_K \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i, \tag{3}$$

where $\kappa = \kappa(K) = \frac{\lambda_n}{\lambda_1}$ is the condition number of K and λ_n, λ_1 are the largest and smallest eigenvalues respectively. The error reduction capability of CG is limited when the condition number is large. The condition number of K will increase when the number of elements increases or when the stiffness of the materials changes. For plastic and viscous behavior this can result in a series of increasing number of iterations as the stiffness changes every load or time step. However, this is out of the scope of this paper but will need future research as plasticity and viscosity are key to realistic simulations.

The convergence of CG is not only affected by the condition number but also by the number and distribution of very small eigenvalues, which has been shown in [18]. The eigenvectors corresponding to the smallest eigenvalues do have a significant contribution to the global solution but may need a significant number of iterations to convergence locally. Hence, very small eigenvalues can increase the number of iterations. We will see that the number of aggregates has a direct correlation with the number of smallest eigenvalues of K. Increasing the number of aggregates may therefore result in more very small eigenvalues and deterioration of the convergence rates.

3.1 Preconditioning

To improve the performance of CG we change the linear system resulting into more favorable extreme eigenvalues and/or clustering. The most efficient way to do this is by preconditioning of the linear system. Preconditioners are well known for their capabilities of improving the performance of iterative solvers and

no Krylov iterative solver can perform well without one. It is because Krylov subspace methods like CG only rely on the eigenvalues that the performance of the solvers is not measured by the choice of the solver but by the choice of the preconditioner [13].

The preconditioned stiffness matrix reads

$$M^{-1}Ku = M^{-1}f, \tag{4}$$

where matrix M is the left preconditioner and assumed to be symmetric, positive definite too. The CG iteration bound of equation (3) also applies to the preconditioned matrix. The preconditioning matrix must satisfy the requirements that it is cheap to construct and it is inexpensive to solve the linear system $Mv = w$. This is because preconditioned algorithms need to solve the linear system $Mv = w$ every iteration step. A rule of thumb is that M must resemble the original matrix K to obtain eigenvalues that cluster around 1. Obviously $M = K$ would be the best but most expensive choice and is equivalent to solving the original system. However, common choices of M are the diagonal of K, which is known as diagonal scaling, and the Incomplete Cholesky factorization with no fill in, IC(0).

3.2 Motivating numerical experiments

All experiments in this section involve the same domain and an equal number of elements. We only consider relatively small problems in order to be able to analyse the eigenvalues with MATLAB [11]. All matrices and material problems were generated by the CAPA-3D finite element simulation software [8]. The aim of the experiments is to find out how the convergence rate of CG depends on material properties and geometry of the volumes. Hence, there will be an emphasis on the relation between the number of CG iterations and the material stiffness and number of aggregates. Increasing the number of elements will result in a more ill-conditioned problem and therefore the number of CG iterations increases.

The experiments will involve four different set-ups.

I. Homogeneous material (bitumen) and no aggregates
II. One aggregate in bitumen layer
III. Four aggregates in bitumen layer
IV. Eight aggregates in bitumen layer

For all experiments only elastic material behavior is taken into account. The bituminous material is considered as a rubber like material, with low stiffness. Furthermore, we assume that under the same conditions, the aggregates do not deform and float in a sea of bituminous material. We use 3D meshes with a finite element discretization of the virtual work equation that can be found in [14]. We use 20 noded, cubic elements with three directions of displacement at every node. At the main diagonal of the stiffness matrix of equation (1) we have the elastic node contributions. The dominating term for the main diagonal is the elastic modulus over the compressibility (Poisson ratio), $\frac{E}{1-2\nu}$ where $0 < \nu \leq \frac{1}{2}$. We only consider normal compressible materials, $0.2 \leq \nu \leq 0.45$, therefore E is the parameter of interest. The E of bitumen will be kept at a constant value of 200 MPa. The E of the aggregates will vary between $\mathcal{O}(10^5)$ and $\mathcal{O}(10^9)$. The

results will be related to the ratio between the E of the bitumen and aggregates respectively.

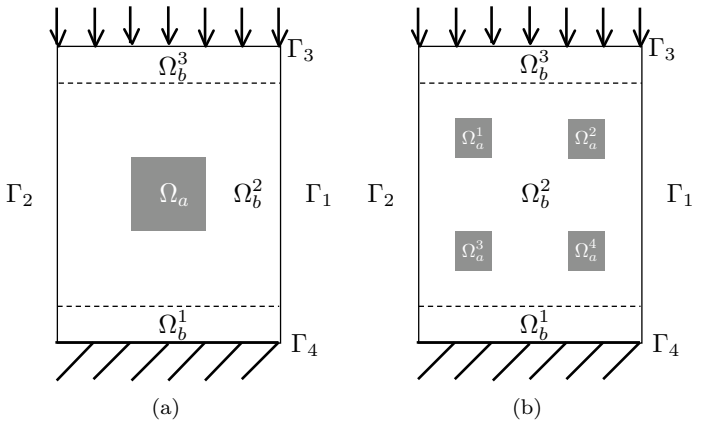

Figure 2: Schematic 2D representation of 3D test cases (II) and (III), figure (a) and (b) contain one and four aggregates respectively.

For reasons of simplicity, Figure 2 shows a 2D representation of 3D test cases (II) and (III). Domain Ω is divided into Ω_a and Ω_b which represent the aggregate and bitumen subdomains respectively. We have $\Omega = \Omega_a \cup \Omega_b$, where $\Omega_a = \bigcup_i^n \Omega_a^i, n = 1, 4$ and $\Omega_b = \bigcup_i^3 \Omega_b^i$. The aggregates are only placed within domain Ω_b^2 such that the boundary conditions at Γ_3 and Γ_4 are not acting on the aggregate elements directly. The aggregates are considered as rigid bodies. We emphasize that the aggregates are actually groups of elements, which share the same material properties.

As the total number of elements remains the same, the size of the aggregates must decrease when the number of aggregates increases. The boundary conditions which do not change are prescribed displacements (Dirichlet b.c.) at boundary Γ_3 and fixed support, i.e. no displacements (Dirichlet b.c.), at boundary Γ_4. At boundaries Γ_1, Γ_2 there is unconstrained displacement (homogeneous Neumann b.c.) in every principal direction.

3.2.1 Convergence results

Figure 3 shows the convergence results of CG with diagonal scaling for all cases (I) to (IV). Incomplete Cholesky (IC) preconditioning with zero fill-in IC(0) has also been tested, but this preconditioner could not be computed by MATLAB due to loss of positive definiteness because of the large jumps in stiffness. Using a drop tolerance does work but is expensive and no fixed bounds can be given on the fill-in of the computed factorization and is therefore not considered as an alternative. We note that case (I), the red dotted line, is used as a benchmark for the other cases. We can conclude that there is a direct correlation between the number of iterations, the material stiffness, and the number of aggregates. As the ratio between the elastic moduli of the bitumen and aggregates increases, the extreme eigenvalues shift in opposite directions, the condition number increases and the number of iterations increases. And when the number of aggregates increases the condition number remains unchanged in order of magnitude but

the number of iterations still increases. We have seen that preconditioning will reduce the condition number and therefore the number of iterations. But the introduction of more aggregates has clearly an effect on the spectrum of eigenvalues. Figure 4 shows the smallest eigenvalues of $M^{-1}K$ of all four cases. We should note that the aggregates are independent sub-domains relative to each other. Hence, no aggregate contains nodes from other aggregates. In this way we can consider the aggregates as rigid bodies within a layer of bitumen. It can be shown that all rigid body modes of the aggregates correspond to the smallest eigenvalues of K. In three dimensions we have six rigid body modes, hence we expect 6, 24 and 48 smallest eigenvalues that correspond to the rigid body modes for cases (II), (III) and (IV) respectively. This is precisely what is observed for all cases. Moreover, the increase in very small eigenvalues is clearly visible as there is a jump between the values of the largest eigenvalues corresponding to the rigid body modes and the remaining eigenvalues in the spectrum of $M^{-1}K$.

The extreme eigenvalues and condition numbers of the (preconditioned) stiffness matrices of test cases (I), (II) and (III) are given in Table 1. Two observations stand out when interpreting the results. The smallest eigenvalue of the non-preconditioned problem is almost invariant with respect to increasing E ratio. The largest eigenvalue of the non-preconditioned problem is not invariant with respect to an increasing E ratio. The order of the largest eigenvalue increases proportionally to the increase in E ratio. Obviously, both observations do not hold when a diagonal scaling preconditioner is applied. In contrary, the inverse effect is observed. The smallest eigenvalue becomes even smaller as the E ratio increases and the largest eigenvalue is a constant value. We have seen that an upperbound for convergence of CG is related to the condition number of the stiffness matrix. Hence, we can expect an increasing number of iterations when the E ratio increases.

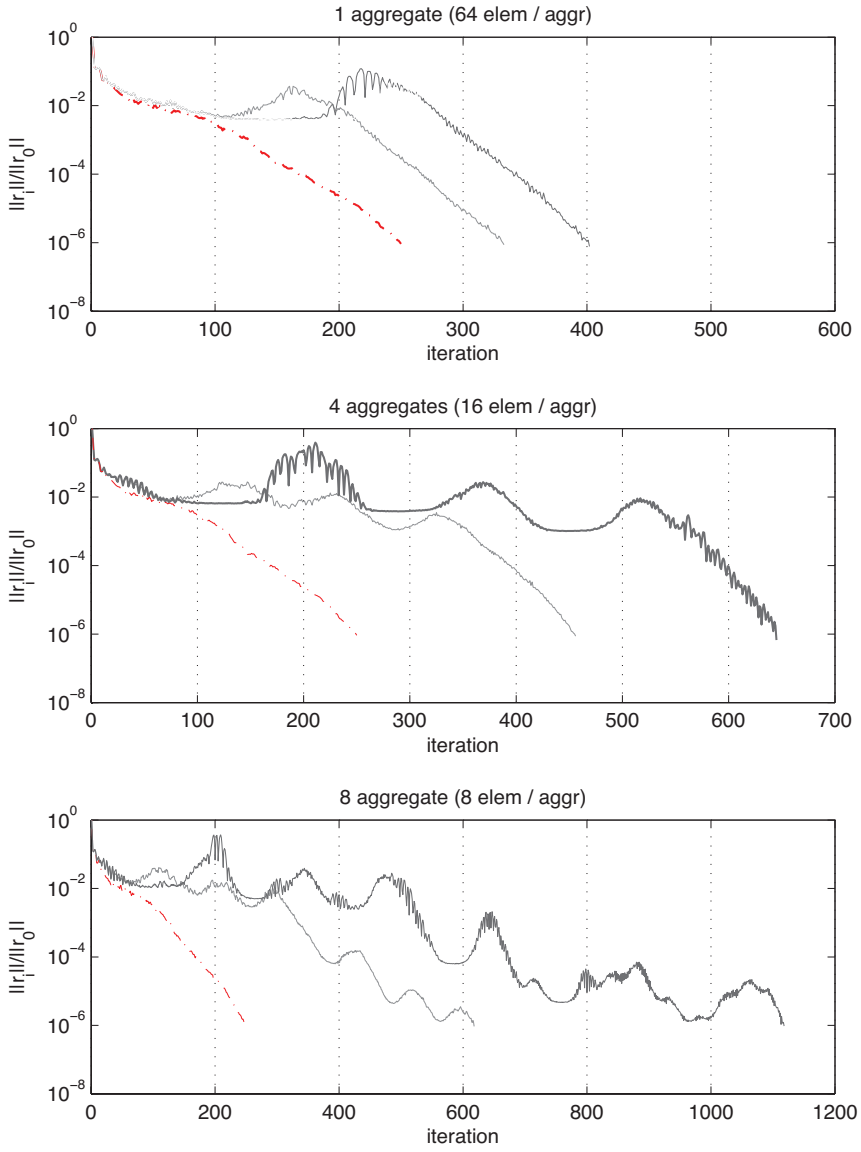

Figure 3: Deterioration of rates of convergence of PCG for increasing number of aggregates and stiffness. -.- homogeneous material, – E ratio $\mathcal{O}(10^3)$, – E ratio $\mathcal{O}(10^5)$.

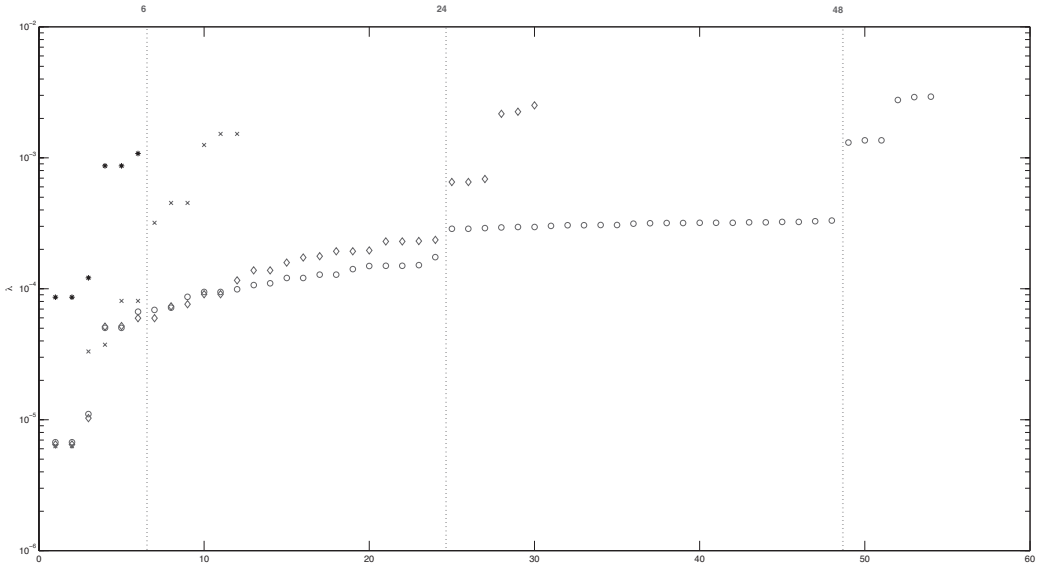

Figure 4: Spectrum of $M^{-1}K$, $M = \text{diag}(K)$. $\times$: 1 aggregate, $\diamond$: 4 aggregates, $\circ$: 8 aggregates, $*$: homogeneous material

As the condition number of the stiffness matrix is of the same order for a different number of aggregates we do not expect a large increase of iterations for an increasing number of aggregates based on the condition number. However, from Figure 4 we expect a slow converging solution due to the large number of smallest eigenvalues, which is indeed observed in Figure 3. The clustering of eigenvalues is clearly visible for cases (III) and (IV), we observe small plateaus which correspond to the slow converging components of the solutions. The number of plateaus increases when the number of aggregates increases due to the number of smallest eigenvalues, i.e. rigid body modes.

4 Deflated Preconditioned CG

As we have shown in the previous Section, the number of iterations to convergence for preconditioned CG is highly dependent on the number of aggregates as well as the ratio of the E moduli. Increasing the number of aggregates introduces more rigid body modes, hence more clustered small eigenvalues. We conclude that the rigid body modes, thus the smallest eigenvalues, are responsible for the plateaus in Figure 3, delaying the convergence of CG. Moreover, we know from [18] that the smallest eigenvalues correspond to the slow converging components of the solution. From a physical point of view it would be natural to try to remove those rigid body modes from the solution process. In fact, the principle directions of the rigid body modes can be computed in advance. When we keep in mind that CG finds a minimal error solution over the Krylov subspace $\mathcal{K}^i(K; r_0)$, we could augment the subspace by the pre computed rigid body modes to improve convergence as these displacements are eliminated from

Case (I)		λ_{min}	λ_{max}	κ
$\mathcal{O}(10^2)$	K	0.0425	$4.15 \cdot 10^3$	$9.76 \cdot 10^4$
	$M^{-1}K$	$8.62 \cdot 10^{-5}$	5.9248	$6.87 \cdot 10^4$
Case (II)				
$\mathcal{O}(10^3)$	K	0.0460	$1.30 \cdot 10^6$	$2.82 \cdot 10^7$
	$M^{-1}K$	$6.29 \cdot 10^{-6}$	5.9248	$9.42 \cdot 10^5$
$\mathcal{O}(10^5)$	K	0.0460	$1.30 \cdot 10^8$	$2.82 \cdot 10^9$
	$M^{-1}K$	$6.64 \cdot 10^{-8}$	5.9248	$8.92 \cdot 10^7$
Case (III)				
$\mathcal{O}(10^3)$	K	0.0450	$1.01 \cdot 10^6$	$2.24 \cdot 10^7$
	$M^{-1}K$	$6.48 \cdot 10^{-6}$	5.9239	$9.15 \cdot 10^5$
$\mathcal{O}(10^5)$	K	0.0450	$1.01 \cdot 10^8$	$2.24 \cdot 10^9$
	$M^{-1}K$	$6.90 \cdot 10^{-8}$	5.9239	$8.59 \cdot 10^7$

Table 1: 2304 elements: Compare the extreme eigenvalues and condition number of preconditioned stiffness matrices. $\mathcal{O}\left(10^n\right)$ represents the jump in E modulus of aggregates and bitumen.

the iterative process.

The deflation technique can be used in conjunction with ordinary preconditioning techniques such as diagonal scaling or Incomplete Cholesky factorization. This is a two-level approach, treating the smallest eigenvalues and largest eigenvalues with deflation and preconditioning respectively. By choosing a smart combination of deflation and preconditioning a more favorable spectrum is obtained, yielding a smaller condition number and less iterations.

For the description of deflation we split the solution of (1) into two parts [4]

$$u = \left(I - P^T\right) u + P^T u, \tag{5}$$

and let us define the projection P by,

$$P = I - KZ(Z^T KZ)^{-1} Z^T, \quad Z \in \mathbb{R}^{n \times m} \tag{6}$$

where Z is the deflation subspace, i.e. the space to be projected out of the residual, and I is the identity matrix of appropriate size. We assume that $m \ll n$ and Z has rank m. Under this assumption $K_c \equiv Z^T KZ$ may be easily computed and factored and is symmetric positive definite. Hence,

$$\left(I - P^T\right) u = ZK_c^{-1} Z^T Ku = ZK_c^{-1} Z^T f \tag{7}$$

can be computed immediately. We only need to compute $P^T u$. Because KP^T is symmetric,

$$KP^T = PK, \tag{8}$$

we solve the deflated system,

$$PK\hat{u} = Pf \tag{9}$$

for $\hat{u}$ using the CG method and multiply this by P^T. We should note that (9) is singular. However, the projected solution $P^T\hat{u}$ is unique, it has no consequence when $\hat{u}$ contains components of the null space, $\mathcal{N}(PK) = \text{span}\{Z\}$. Moreover, from [9] and [18] we learn that the null space of PK never enters the iteration process and the corresponding zero-eigenvalues do not influence the solution. For singular systems, the condition number will go to infinity as the smallest eigenvalue approaches zero. To obtain a useful bound for the error of CG we define the effective condition number of a semi-definite matrix $C \in \mathbb{R}^{n\times n}$ with corank m to be the ratio of the largest and smallest positive eigenvalues,

$$\kappa_{\text{eff}}(C) = \frac{\lambda_n}{\lambda_{m+1}}. \tag{10}$$

Theorem 2.2 of [4] implies that a bound on the condition number of PK can be obtained and leads to the choice of Z applied to the rigid body modes problems. We assume a splitting $K = C + R$ such that C and R are symmetric positive semi-definite with $\mathcal{N}(C) = \text{span}\{Z\}$ the null space of C. Then,

$$\lambda_i(C) \leq \lambda_i(PK) \leq \lambda_i(C) + \lambda_{\max}(PR). \tag{11}$$

Moreover, the effective condition number of PK is bounded by,

$$\kappa_{\text{eff}}(PK) \leq \frac{\lambda_n(K)}{\lambda_{m+1}(C)}. \tag{12}$$

Because we will use a symmetric preconditioner $M = LL^T$, e.g. diagonal scaling, with Theorem 2.3 of [4] we have,

$$\kappa_{\text{eff}}(L^{-1}PKL^{-T}) \leq \frac{\lambda_n(L^{-1}KL^{-T})}{\lambda_{m+1}(L^{-1}CL^{-T})}. \tag{13}$$

To fully comprehend the construction of the deflation vectors and a specific choice of Z we use the following experiment. Assume that we have a cube of bitumen containing one aggregate which is shown in Figure 5. The sub-domains Ω_1 and Ω_2 can be considered as bitumen and aggregates respectively. Clearly, without the constraints of the surrounding bitumen material, the aggregate of Ω_2 will act as a rigid body. With kernel deflation we aim to solve on both sub-domains separately. We separate sub-domain Ω_1 from sub-domain Ω_2 and apply new boundary conditions to the domains. We assume that the aggregates that are not influenced by the boundary conditions of the whole domain act as rigid bodies, therefore we assume homogeneous Neumann boundary conditions. The bitumen will be restricted by the aggregates and we apply homogeneous Dirichlet boundary conditions.

The stiffness matrix is assembled from element stiffness matrices which come from the finite element formulation of the virtual work equation. Assume that we mesh the domain in Figure 5 with k elements. The mesh consists of 20 noded, cubic elements yielding element stiffness matrices $K_e \in \mathbb{R}^{60\times 60}$. We introduce the element operator $N_e \in \mathbb{R}^{60\times n}$ that maps a global vector to an element vector, $u_e = N_e u$. The stiffness matrix K is assembled by,

$$K = \sum_e^m N_e^T K_e N_e. \tag{14}$$

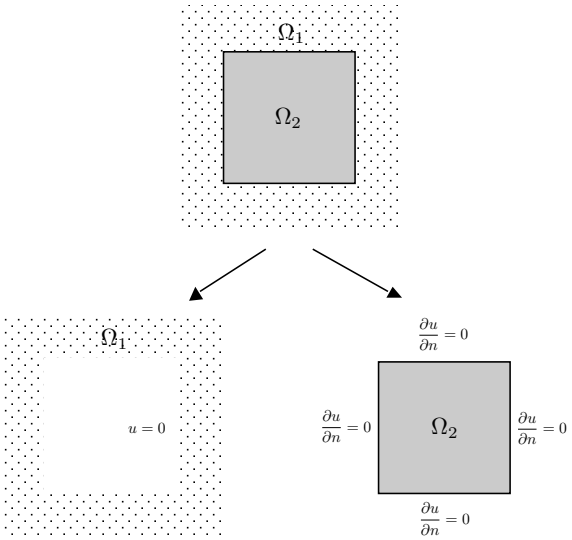

Figure 5: Principle of kernel deflation

Assume that $K_b = \sum_{e\in\Omega_1} N_e^T K_e N_e$, $K_a = \sum_{e\in\Omega_2} N_e^T K_e N_e$ and $K_\Gamma = \sum_{e\in\Omega_1\cap\Omega_2} = N_e^T K_e N_e$. We assume $K = C + R$ according to Theorem 2.2 of [4] where $C = K_b + K_a - K_\Gamma$ and $R = K_\Gamma$. Matrix C consists of two independent block matrices which correspond to the bitumen and aggregate domains. Matrix R consists of only those node contributions of elements from Ω_1 that lie on the intersection of domain Ω_1 and Ω_2. We should note that by the removal of the bitumen-aggregates boundary nodes from the bitumen sub-domain, Dirichlet and Neumann boundary conditions are automatically imposed on the bitumen and aggregate sub matrices in matrix C. The matrix C contains one singular sub matrix corresponding to the aggregate and one positive definite sub matrix corresponding to the bitumen. Moreover, because of the Dirichlet boundary conditions, the bitumen domain is statically determined and numerically well conditioned.

We apply Theorem 2.2 of [4] to $C = K_b + K_a - K_\Gamma$ and $R = K_\Gamma$. We have, $Z = \mathcal{N}(C) = \text{span}\{Z_a\}$ where $Z_a = \{z_a^1, ..., z_a^6\}$ with z_a^j the j-th base vector of the null space of K_b which correspond to all six rigid body modes of the aggregate. We must emphasize that by this choice of deflation subspace Z the rigid body modes are eliminated from the iterative solution process and removes the newly acquired Neumann boundary conditions from the aggregate sub-domains.

Extension of the previous experiment to an arbitrary number of aggregates, is straightforward. Assume that we only consider problems where there are r independent aggregates and one homogeneous layer of bitumen. Two arbitrary aggregates do not share elements and thus nodes. We apply the splitting $K = C + R$ where $C = K_b + \sum_r K_{a_r} - K_\Gamma$ and $R = K_\Gamma$. Matrix C contains r independent singular sub matrices that correspond to the aggregates and one positive definite sub matrix that corresponds to the bitumen. The deflation subspace $Z = \mathcal{N}(C) = \bigcup_r \text{span}\{Z_{a_r}\}$ with $Z_{a_r} = \{z_{a_r}^1, ..., z_{a_r}^6\}$. The dimension of Z will be $r \times 6$. With respect to the software implementation, because all

aggregates are independent there is no overlap of the non zero elements in the deflation vectors. Hence, we can store Z for any problem size within just six vectors.

One can question the amount of work for the computation of the rigid body modes of the aggregates. However, we can assume that the stiffness of the aggregates does not change during the simulation, i.e. the Newton-Raphson iterative process for the computation of the virtual work. As the stiffness sub-matrices corresponding to the aggregates do not change, the rigid body modes will not change and the deflation space remains unchanged. Hence, for any simulation where the geometry and stiffness of the aggregates does not change only one evaluation of the rigid body modes is needed. Moreover, parallel direct solvers can be used to compute the null spaces of the (relatively small) sub-domain matrices. Future research has to be done on predetermination of the rigid body modes on physical grounds, some work has been done on fluids in [17].

4.1 Numerical experiments

All numerical experiments use Deflated Preconditioned CG (DPCG) based on the rigid body modes of the aggregates. The results are benchmarked against the homogeneous material case (I) in which there are no aggregates. The benchmark is CG with diagonal scaling and represented by the red dotted lines. The DPCG algorithm is taken from [17] and included in Appendix A.

4.1.1 Convergence results

Figure 6 shows the performance of PCG and DPCG applied to case (II) and (III).

The results of Figure 6 show what we expected from the deflation. The rigid body modes are no longer active in the iterative process and only the deformation of the bitumen has to be computed. Hence, the performance of DPCG is identical compared to the homogeneous benchmark case. Moreover, in Figure 6 the plots for case (II) and (III) show the same performance but for a different amount of aggregates. Therefore, adding more aggregates to the domain has no influence on the performance of CG. The same behavior can be observed in Figure 7 which shows the performance of PCG and DPCG applied to case (II), (III) and (IV). We should note that the results of Figure 6 are also included in Figure 7.

We have seen that the effective condition number of the deflated matrix PK depends on the smallest positive eigenvalue of C which is identical to the smallest eigenvalue of the bitumen sub matrix. Hence, the number of iterations of CG is only bounded by the material properties of the bitumen and not by the stiffness of the aggregates.

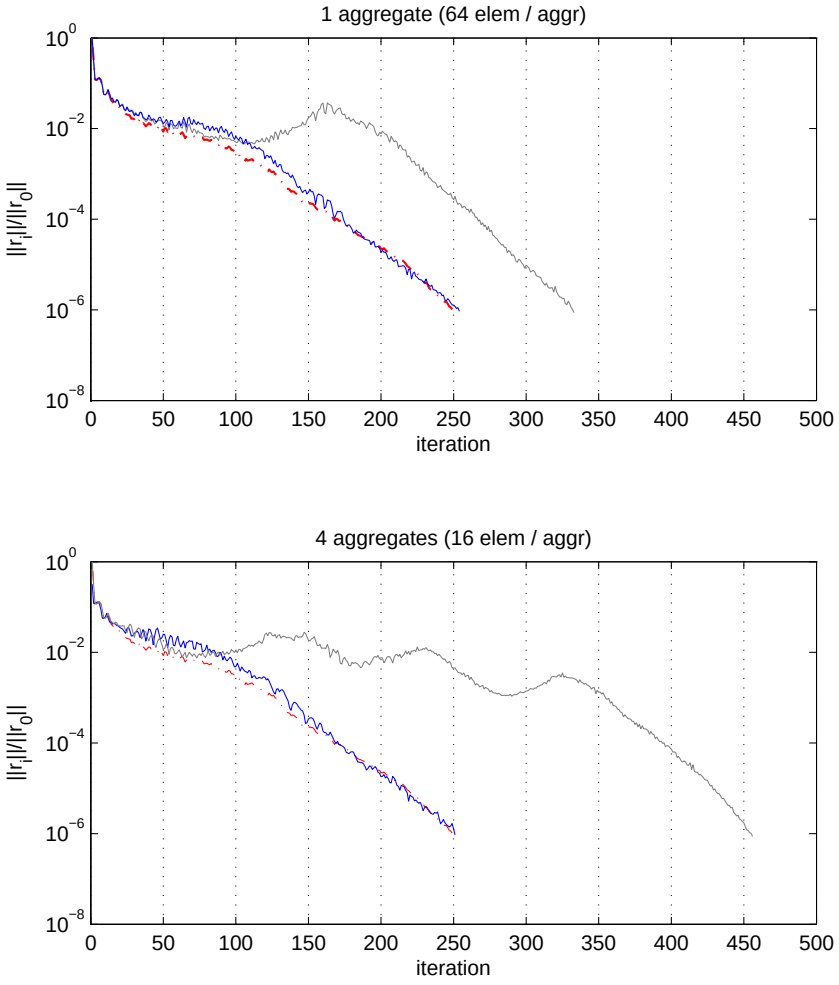

Figure 6: Convergence of PCG and kernel DPCG for 1 and 4 aggregates respectively. -.- PCG (homogeneous material), – PCG (E ratio $\mathcal{O}(10^3)$), – kernel DPCG (E ratio $\mathcal{O}(10^3)$).

5 Conclusions

With the introduction of meshes that result from CT scans, the FE discretization of composite materials yields large, sparse linear systems. Conventional direct and iterative solution methods are not capable of handling such large systems due to ill-conditioned systems and hardware limitations. Moreover, with respect to future developments of computer hardware, parallellization of the solvers will be necessary to benefit from multicore, GPU and grid computing.

All 3D meshes in this research contained 20 noded cubic elements with three displacement directions per node. The linearized virtual work equation gives a symmetric positive definite (SPD) stiffness matrix. Because of these properties of the stiffness matrix we use preconditioned CG (PCG) as the initial solver. The PCG solver has an excellent performance on SPD matrices, is cheap in terms of work and is highly parallelizable. The error of PCG is bounded by the condition number of the stiffness matrix. Increasing the stiffness of the materials results in a higher condition number, yielding worse performance. However, the number of aggregates seems to have no influence on the value of the condition number.

The numerical experiments show that the performance of CG with a diagonal scaling preconditioner is poor. Increasing the stiffness as well as the number of aggregates results in a deterioration of the convergence rates of PCG. The results also show plateaus in the convergence behavior which indicates the existence of small eigenvalues related to slow converging components. Analysis of the spectrum of the preconditioned stiffness matrix shows that the smallest eigenvalues correspond to the domains containing aggregates. Moreover, the number of small eigenvalues is equal to the number of rigid body modes of the aggregates.

The performance of PCG can be improved by removing the smallest eigenvalues from the spectrum of the stiffness matrix. We used deflation to filter out those values. All vectors in the deflation subspace are projected out of the residual of the iterative process. We have used the rigid body modes for the disjunct aggregates as the deflation subspace. The performance of Deflated Preconditioned CG (DPCG) is very good. The removal of the rigid body modes results in a mechanical and mathematical well defined problem that only depends on the material properties of the bitumen. The convergence behavior of DPCG is identical to the convergence behavior of the homogeneous material benchmark case. Adding more and stiffer aggregates has no effect on the performance of DPCG. We have constructed an iterative solver which has aggregate independent convergence behavior. Furthermore, this is the first application of deflation based preconditioning applied to coupled systems of partial differential equations.

Further research has to be done on the parallellization of the current algorithm. Also, as the performance of DPCG now only depends on the bituminous materials, the effects of plasticity and viscosity have to be taken into account. Hence, more research on advanced preconditioning will be needed to obtain reasonable performance. The Deflated CG algorithm is to be embedded into the CAPA-3D finite element system [8].

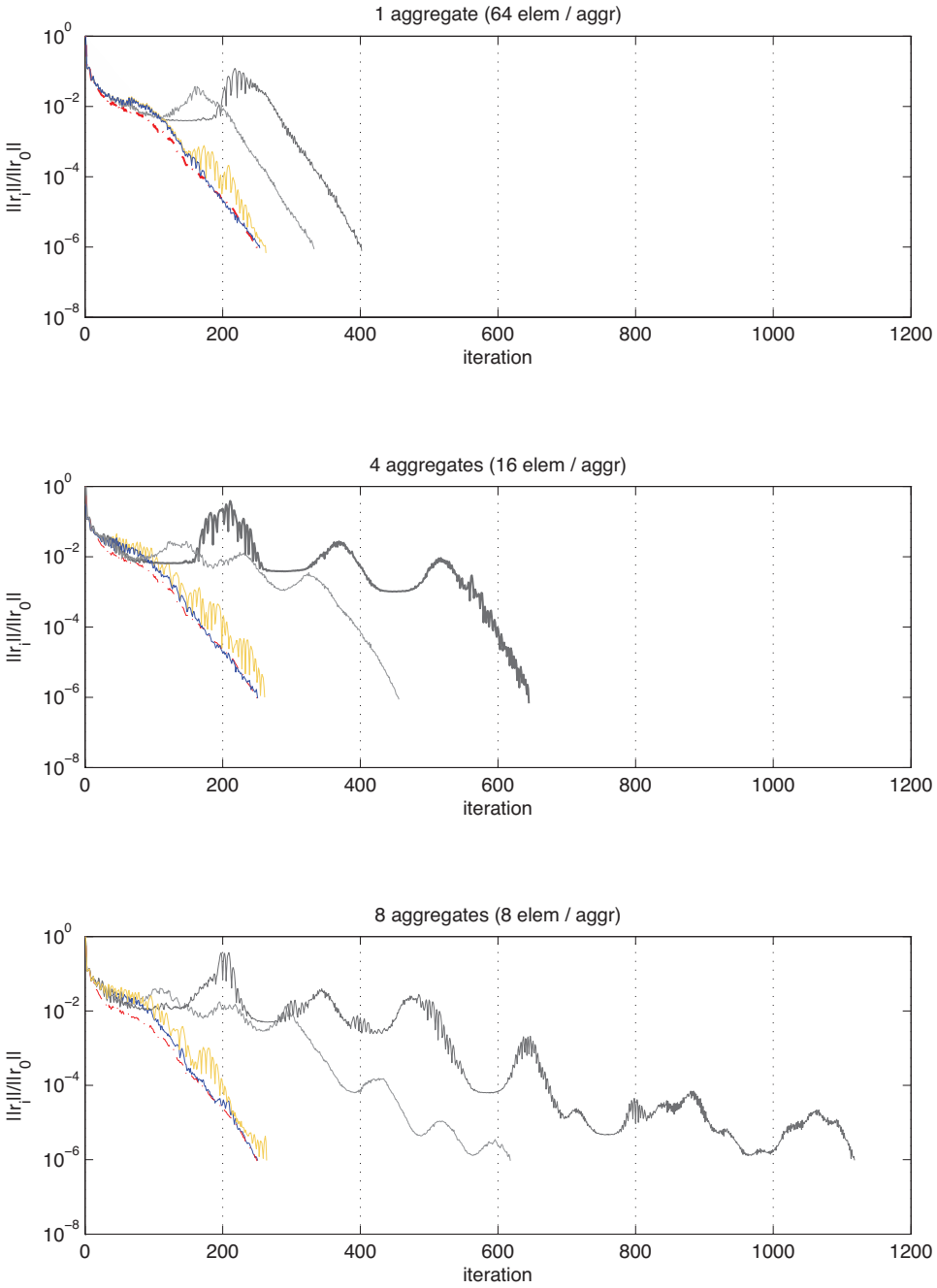

Figure 7: Convergence of PCG and kernel DPCG for 1, 4 and 8 aggregates respectively with increasing value of aggregate stiffness. -.- PCG (homogeneous material), – PCG (E ratio $\mathcal{O}(10^3)$), – PCG (E ratio $\mathcal{O}(10^5)$),– kernel DPCG (E ratio $\mathcal{O}(10^3)$), – kernel DPCG (E ratio $\mathcal{O}(10^5)$).

A Appendix

The deflated preconditioned CG algorithm [17].

Algorithm 1 Deflated preconditioned CG solving $K\mathbf{u} = \mathbf{f}$

Select $\mathbf{u}_0$. Compute $\mathbf{r}_0 = (\mathbf{f} - K\mathbf{u}_0)$, set $\hat{\mathbf{r}}_0 = P\mathbf{r}_0$ and $\mathbf{p}_0 = \hat{\mathbf{r}}_0$
Solve $M\mathbf{y}_0 = \hat{\mathbf{r}}_0$ and set $\mathbf{p}_0 = \mathbf{y}_0$
for $j = 0, 1, ...$ until convergence **do**
  $\hat{\mathbf{w}}_j = PK\mathbf{p}_j$
  $\alpha_j = \frac{(\hat{\mathbf{r}}_j, \mathbf{y}_j)}{(\hat{\mathbf{w}}_j, \mathbf{p}_j)}$
  $\hat{\mathbf{u}}_{j+1} = \hat{\mathbf{u}}_j + \alpha_j \mathbf{p}_j$
  $\hat{\mathbf{r}}_{j+1} = \hat{\mathbf{r}}_j - \alpha_j \hat{\mathbf{w}}_j$
  Solve $M\mathbf{y}_{j+1} = \hat{\mathbf{r}}_{j+1}$
  $\beta_j = \frac{(\hat{\mathbf{r}}_{j+1}, \mathbf{y}_{j+1})}{(\hat{\mathbf{r}}_j, \mathbf{y}_j)}$
  $\mathbf{p}_{j+1} = \mathbf{y}_{j+1} + \beta_j \mathbf{p}_j$
end for
$\mathbf{u} = ZK_c^{-1}Z^T\mathbf{f} + P^T\hat{\mathbf{u}}_{j+1}$

References

[1] P. R. Amestoy, I. S. Duff, and J.-Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Methods Appl. Mech. Eng.*, 184:501–520, 2000.

[2] K. J. Bathe. *Finite Element Procedures.* Prentice Hall, 2 revised ed edition, June 1995.

[3] J. W. Demmel, J. Gilbert, and X. S. Li. SuperLU users' guide. Technical Report CSD-97-944, 1997.

[4] J. Frank and C. Vuik. On the construction of deflation-based preconditioners. *SIAM J. Sci. Comput.*, 23(2):442–462, 2001.

[5] G. H. Golub and C. F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences).* The Johns Hopkins University Press, Baltimore, October 1996.

[6] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, Dec 1952.

[7] Sandia Labs http://cubit.sandia.gov/. Cubit, geometry and mesh generation toolkit.

[8] Delft University of Technology http://www.capa 3d.org. Capa-3d computer aided pavement analysis.

[9] E. F. Kaasschieter. Preconditioned conjugate gradients for solving singular systems. *J. Comput. Appl. Math.*, 24(1-2):265–275, 1988.

[10] J. A. Meijerink and H. A. van der Vorst. Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems. *Journal of Computational Physics*, 44(1):134–155, 1981.

[11] C. B. Moler. MATLAB user's guide. Technical report, University of New Mexico. Dept. of Computer Science, November 1980. This describes use of Classic Matlab, the prototype for the very-much expanded professional Matlab from The MathWorks. Classic Matlab is no longer available.

[12] R. A. Nicolaides. Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.*, 24(2):355–365, 1987.

[13] Y. Saad. *Iterative Methods for Sparse Linear Systems, Second Edition.* Society for Industrial and Applied Mathematics, Philadelphia, April 2003.

[14] A. Scarpas. *Mechanics based computational platform for pavement engineering.* PhD Thesis TU Delft, 2004.

[15] O. Schenk, K. Gärtner, W. Fichtner, and A. Stricker. Pardiso: a high-performance serial and parallel sparse linear solver in semiconductor device simulation. *Future Generation Computer Systems*, 18(1):69–78, 2001.

[16] Simpleware. http://www.simpleware.com.

[17] J.M. Tang. *Two-Level Preconditioned Conjugate Gradient Methods with Applications to Bubbly Flow Problems*. PhD Thesis TU Delft, 2008.

[18] A. Van der Sluis and H.A. Van der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.*, 48(5):543–560, 1986.