

B025

Deflation Accelerated Preconditioned Conjugate Gradient Method in Finite Element Methods in Oil Reservoirs

F.J. Vermolen* (Delft University of Technology), P.L.J. Zitha (Delft University of Technology) & C. Vuik (Delft University of Technology)

SUMMARY

Oil reservoirs generally contain several layers with highly varying permeabilities. Hence modeling oil and water flows often involves solving partial differential equations with very large contrasts in the coefficients. Since the injection and production wells are very small compared to dimensions of the reservoir, the injection wells and production wells are modelled by the use of delta-functions appearing in the right hand side of the partial differential equation for the pressure. In the presentation we consider a reservoir that consists of several layers with extreme contrasts of the permeability at the interfaces between the adjacent layers. Further, the finite element mesh is refined in the vicinity of the production and injection wells.

The finite element discretization of the above equation gives a stiffness matrix with extremely varying coefficients and hence the spectrum consists of large eigenvalues and eigenvalues that are almost zero, which gives a very high condition number. Hence a very bad convergence behavior for an iterative solver such as the conjugate gradient method results. A preconditioner, like ILU, removes almost all the small eigenvalues, however, some small eigenvalues due to the large ratio of the coefficients at the interfaces persist. These small eigenvalues are removed by deflation based on a set of vectors that approximate the span of the corresponding eigenvectors.

Herewith the speed of convergence is successfully enhanced and the computational cost are reduced significantly. By the use of a proper choice for the deflation vectors, we show that the speed of convergence of our method does not depend on either the value of the contrasts in the coefficients or on the number of layers with varying coefficients. Further, the method is scalable in a parallel computing environment.

1 Introduction

Commercial reservoir simulators are widely used to model single- and multi-phase flow processes in oil and gas reservoirs, see for instance the work of Aziz and Settari [1] page 18–24. The simulators solve the governing equations taking into account specific constitutive relations and thermodynamic processes. Often the simulators are based on Finite Volume techniques that are used to solve the governing equations. Since the subsurface reservoirs usually contain several layers with abruptly varying properties, such as the permeability, and since these interfaces between the adjacent layers are curved in general, the implementation of Finite Element techniques is more straightforward.

In the present paper, we will consider the Finite Element discretization of an elliptic partial differential equation. This will give a large, but sparse, system of linear equations to be solved at each time-step. For such a large sparse system of linear equations it is more efficient to use an iterative method to solve it.

We only consider symmetric positive definite (SPD) matrices. Presently, direct methods (such as a Choleski-decomposition) are available to solve such a linear system. However, for large sparse coefficient matrices fill-in causes a loss of efficiency (in computer memory and number of floating point operations). For such a case iterative methods are a better alternative. Furthermore, if a time integration is necessary, then the solution of the previous time-step can be used as a starting vector for the algorithm to get the result on the next time-step. This too supports the use of iterative methods.

Iterative methods such as Gauss-Seidel, Jacobi, SOR, and Chebyshev-methods can be used, however, convergence is in general slow and it is often very expensive to determine good estimates of parameters on which they depend. To avoid these problems, the conjugate gradient method is used. We deal with an application from transport in porous media where we encounter extreme contrasts in the coefficients of the partial differential equation or a very large domain of computation for which we favor the use of parallel computing. The large contrasts are caused by the layered domain with extreme contrasts in permeability.

Here a preconditioning is necessary and we use a standard incomplete Choleski factorization as a preconditioner for the conjugate gradient method (ICCG) to improve the convergence behavior. The improvement is caused by the removal of very small eigenvalues resulting from a discretization with a high resolution. For the case that the permeability changes abruptly between adjacent layers, say a factor of the order of 10^7 , then, preconditioning does not remove all very small eigenvalues, and hence, convergence still remains very bad. For this case deflation is applied as a projection to remove effect of the small eigenvalues. An other notorious problem pops up in parallel computation in combination with a block preconditioner. The parallel computation aims at making the problem scalable, that is, the total computation speed should be inversely proportional with the number of processors used. However, due to the inaccuracy of block preconditioning with respect to the preconditioning of the full problem, the number of conjugate gradient iterations increases drastically, which deteriorates the scalability of the problem: the number of needed CG iterations becomes dependent on the number of processors used. Hence, the efficiency of the method decreases. In this paper we will see that the use of deflation will provide a constant number of conjugate gradient iterations, which makes the method scalable.

For references related to the Deflated ICCG method we refer to the overview given in [11] and [10]. The DICCG method has already been successfully used for complicated magnetic field simulations [2]. A related method is recently presented in [7]. In [3] deflation is used

to accelerate block-IC preconditioners combined with Krylov subspace methods in a parallel computing environment. We note that this list of references is far from complete.

The DICCG method is related to coarse grid correction, which is used in domain decomposition methods [4, 7]. Therefore insight in a good choice of the deflation vectors can probably be used to devise comparable strategies for coarse grid correction approaches. In Nabben & Vuik [6] it is shown that deflation is more efficient than the balancing preconditioner introduced by Mandel [5].

Part of the present paper is a compilation of earlier results on deflation by one of the authors and here it serves to introduce the concept of deflation into the computational geosciences community. New is the consideration of anisotropies. The paper is organized as follows. First the model equations are presented as a motivation of the benchmark problem, which has all the mathematical difficulties that occur in the real model. Subsequently the numerical solution method using deflation is introduced. This is followed by some numerical experiments and some conclusions are drawn.

2 Model equations

The elliptic partial differential equation that we use as a benchmark problem for the application of the deflated preconditioned conjugate method is applicable in general in flow in porous media, such as salt/fresh water flow, flow in oil reservoirs *etc.* Therefore, the model is simplified to

$$\begin{aligned} -\operatorname{div} k \operatorname{grad} p &= 0, & \text{for } \mathbf{x} \in \Omega \\ \frac{\partial p}{\partial n} &= 0, \text{ on } \partial\Omega_N, & p = 1, \text{ on } \partial\Omega_D, \end{aligned} \tag{1}$$

where $\partial\Omega = \partial\Omega_N \cup \partial\Omega_D$, $\partial\Omega_N \cap \partial\Omega_D = \emptyset$ and k is allowed to vary abruptly over the interfaces of adjacent subdomains. It is easy to see that the exact solution of this problem is $p = 1$ on Ω .

A weak form of the above benchmark problem is given by

$$(P_1) \begin{cases} \text{Find } p \in H^1(\Omega) \text{ (} p|_{\partial\Omega_D} = 1 \text{) such that} \\ \int_{\Omega} k \operatorname{grad} p \cdot \operatorname{grad} v \, dA = 0 \text{ for all } v \in H_0^1(\Omega) \text{ (} v|_{\partial\Omega_D} = 0 \text{).} \end{cases} \tag{2}$$

This problem is solved by the use of a standard Galerkin Finite Element Method, with $p = \sum_{i=1}^{n+n_D} p_i v_i$ ($v_i|_{\partial\Omega_D} = 0$ for $i \in \{1, \dots, n\}$), with piecewise linear element functions v_i . In our examples we take a set of subdomains with

$$k(\mathbf{x}) = \begin{cases} k_{\min} = 10^{-7}, & \mathbf{x} \in \Omega_L, \\ k_{\max} = 1, & \mathbf{x} \in \Omega_H, \end{cases} \tag{3}$$

where we suppose that the closure of the domain Ω consists of the union of non-overlapping $\overline{\Omega}_L$ and $\overline{\Omega}_H$, that is $\Omega = \Omega_L \cup \Omega_H \cup \partial\Omega_{HL}$ and $\Omega_H \cap \Omega_L = \emptyset$, further $\partial\Omega_{HL}$ denotes the interface between the adjacent subdomains. We assume that the subdomains are not connected. Further, we define $\varepsilon := k_{\min}/k_{\max} = k_{\min}$. In some applications, the high and low

mobility respectively correspond to sand and shale layers. We will also use this terminology to refer to the high and low permeability layers. From the Galerkin discretization it follows immediately that across an interface the coefficients in the discrete equation varies several orders of magnitude.

Discretization by the use of Galerkin's method results into a matrix-vector equation of type

$$A\underline{x} = \underline{b}, \quad (4)$$

where $A \in \mathbb{R}^{n \times n}$, $\underline{x} \in \mathbb{R}^n$ and $\underline{b} \in \mathbb{R}^n$ respectively represent the discretization (or stiffness-) matrix, solution vector and right-hand side vector. Using a FEM approach the discretization matrix is sparse, symmetric and positive definite (SPD). We use preconditioned conjugate gradient methods to solve this problem. Furthermore, the discretization is chosen such that the interfaces between consecutive layers coincide with gridpoints. For the case of large jumps in the coefficient k the condition of the discretization-matrix is very large. The remainder of the paper is devoted to the efficient solution of the above matrix-vector equation when n is large.

3 Deflation

It is known that the use of classical preconditioners does not solve the problem of the slow convergence of the conjugate gradient method if ε is very small. This holds if the parameter k changes abruptly over the interfaces. This also holds if parallel computation is used with blockwise preconditioning even without large contrasts of permeability. In order to deal with the bad convergence, deflation on equation (4) is applied. First we briefly describe the idea of the use of deflation, which is applied to high contrasts and to parallel computing.

Let A be a symmetric positive definite matrix that results from a finite element discretization problem (1), suppose further that the number of layers with high permeability k_{\max} whose boundary does not (partly) co-incide with $\partial\Omega_D$ where the solution is prescribed explicitly. Then, since the quantity $k \frac{\partial p}{\partial n}$ is continuous in the whole domain of computation, the discretization matrix that would only correspond to this layer would almost be singular due to $k_{\min} \ll k_{\max}$. Therefore, the number of excessively small eigenvalues (in the order of ε) due to the abrupt changes of the permeability is given by the total number of high permeability layers whose intersection with $\partial\Omega_D$ is empty. This result is proven as a theorem by Vuik *et al.* [12]. The convergence behavior of the (preconditioned) conjugate gradient method is determined by the condition of the matrix, which is the ratio of the largest and smallest absolute value of the eigenvalues of the matrix. The trick to cope with this bad condition of the matrix causing the slow convergence rate is as follows: The span, that is the linear combination, of the eigenvectors that correspond to these small eigenvalues is approximated by the span of a set of vectors, referred to as deflation vectors, that are produced more easily than the actual eigenvectors and further they will be sparse. Suppose that the number of eigenvalues of $O(\varepsilon)$ is m , and let $\underline{z}_1, \dots, \underline{z}_m$ be the deflation vectors that approximate the span of the eigenvectors corresponding to the m very small eigenvalues of A and let the matrix Z be defined by $Z := [\underline{z}_1 \ \underline{z}_2 \ \dots \ \underline{z}_m]$, and

$$P := I - AZ(Z^T AZ)^{-1}Z^T. \quad (5)$$

Here I is the identity matrix. Suppose that $V = [\underline{v}_{m+1} \ \dots \ \underline{v}_n]$ contains eigenvectors of the matrix A , then, since the eigenvectors of a symmetric matrix are orthogonal, the vectors

v_{m+1}, \dots, v_n are orthogonal, hence these eigenvectors are also eigenvectors of the matrix PA . Furthermore it can be seen by elementary linear algebra that $PAZ = 0$ and that $P^2 = P$, hence P is a projection and the vectors Az_1, \dots, Az_m are within the nullspace of P . Hence, the following holds: Let the spectrum of A be given by $\{\lambda_1, \dots, \lambda_m, \lambda_{m+1}, \dots, \lambda_n\}$, with $|\lambda_i| \ll |\lambda_j|$ for $i \in \{1, \dots, m\}$ and $j \in \{m+1, \dots, n\}$, then, the spectrum of PA is given by $\{0, \dots, 0, \lambda_{m+1}, \dots, \lambda_n\}$. In Vermolen *et al.* [9] it is proved that Az_1, \dots, Az_m is a basis of the nullspace of P . Equation (4) is solved by first solving

$$PA\underline{x} = P\underline{b}, \quad (6)$$

where PA has the same eigenvalues as A except that the excessively small eigenvalues of the order of ε of A are replaced with zero. The matrix PA is singular, hence the solution of the above equation is not uniquely defined. Kaasschieter proved that the effective condition number for the conjugate gradient method is determined by the ratio of the smallest and largest absolute value of the non-zero eigenvalues. The matrix PA is singular and the solution of equation (6) is not uniquely defined, therefore its solution is projected by

$$\underline{x} = (I - P^T)\underline{x} + P^T\underline{x} = Z(Z^T AZ)^{-1}Z\underline{b} + P^T\underline{x}. \quad (7)$$

It is shown in Vermolen *et al.* [9] that $P^T\underline{x}$ is uniquely defined and hence \underline{x} is unique.

Several choices for the deflation vectors z_1, \dots, z_m exist, where the auxiliary condition is that their span should approximate the span of the eigenvectors from the eigenvalues of A that are $O(\varepsilon)$. For a survey on the various deflation vectors, we refer to Vermolen *et al.* [9] and Vuik *et al.* [10]. In the present study we will use the so-called algebraic deflation vectors, which are defined by

$$z_i = \begin{cases} 1, & \mathbf{x} \in \Omega_i, \\ \in \{0, 1\}, & \mathbf{x} \in \partial\Omega_i \\ 0, & \mathbf{x} \in \Omega \setminus \Omega_i \end{cases} \quad (8)$$

Here z_i corresponds to subdomain Ω_i . On the boundary layer, the deflation vector is either one or zero, depending on the permeability of the layer Ω_i . If the permeability is high, *i.e.* $\Omega_i \subset \Omega_H$, then, the interface value is one, else it is zero, *i.e.* $\Omega_i \subset \Omega_L$. In Vermolen *et al.* [9] it is proven that the span of this set of deflation vectors approximates the span of the eigenvectors corresponding to the eigenvalues of $O(\varepsilon)$. This choice of deflation vectors is referred to as algebraic. An other option is to choose the value at the interface equal to $\frac{k}{k_{\min} + k_{\max}}$, this is called algebraic weighting. An other option is to choose the deflation vectors such that they satisfy the Laplace equation in the low permeability layer and and that they are constant in the high permeability layer. For these physics based deflation vectors, only for high permeability layers whose boundary has no intersection with the Dirichlet boundary $\partial\Omega_D$ deflation vectors are attributed.

4 Numerical experiments

The principle of deflation is applied to reservoirs with layers of extreme contrasts in permeability and in parallel computing. In Figure 2 some results for seven layers with alternating permeability are shown. In Figure 2 the error $\|\underline{x}_k - \underline{x}_{\text{true}}\|_2$ is shown as a function of the

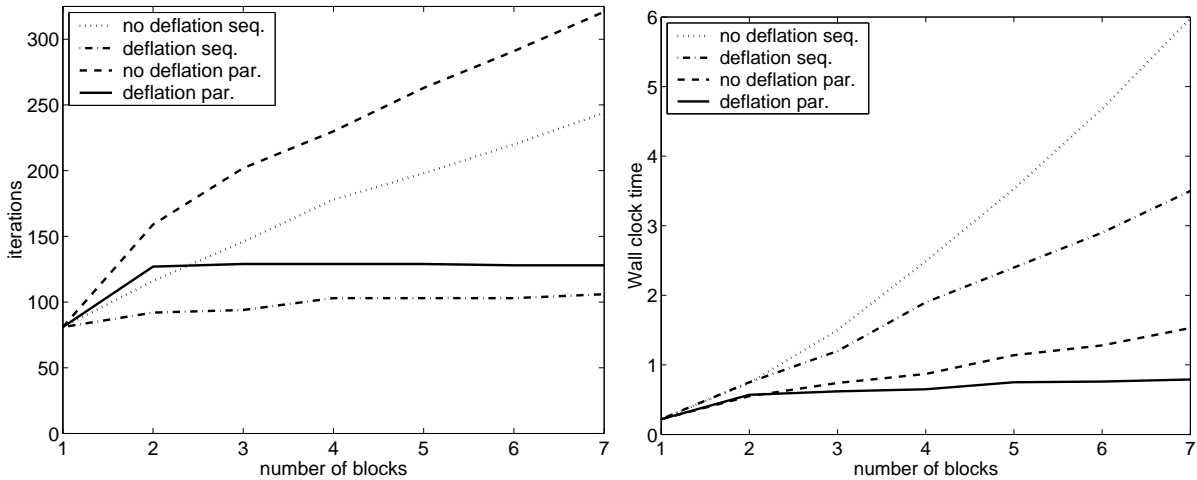


Figure 1: Left: Number of iterations. Right: Wall-clock time

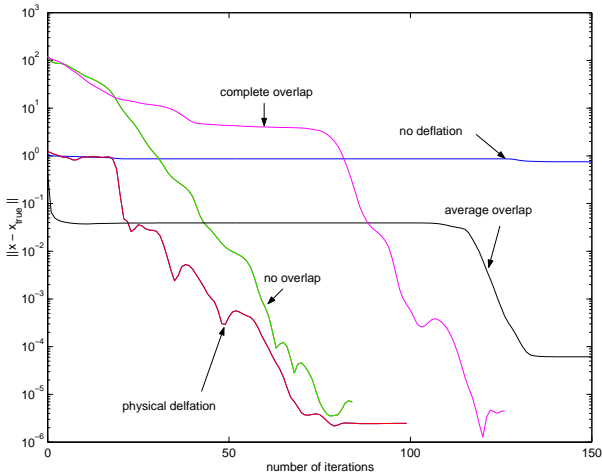


Figure 2: The error as a function of the iteration number for various types of deflation.

iteration number k for various deflation vectors and for the case without deflation. It can be seen that the use of deflation is crucially important

For a description of the parallel implementation of the deflation method, we refer to Vermolen *et al.* [8]. As a preconditioner we use the incomplete Choleski method. The domain is divided into layers of equal permeability. To each layer a processor is attributed. In Figure 1 the number of conjugate gradient iterations and wall clock time have been plotted as a function of the number of blocks. The total domain of computation is enlarged as the number of blocks increases. It can be seen for the deflated methods (both sequential and parallel) that the number of iterations and wall clock time do not or hardly depend on the number of subdomains, that is, the size of the overall domain of computation. We remark that qualitatively the same holds for the coarse grid correction, see Nabben & Vuik, although the actual speed of deflation is larger.

Further, we investigate the influence of anisotropies of the permeability of the medium on

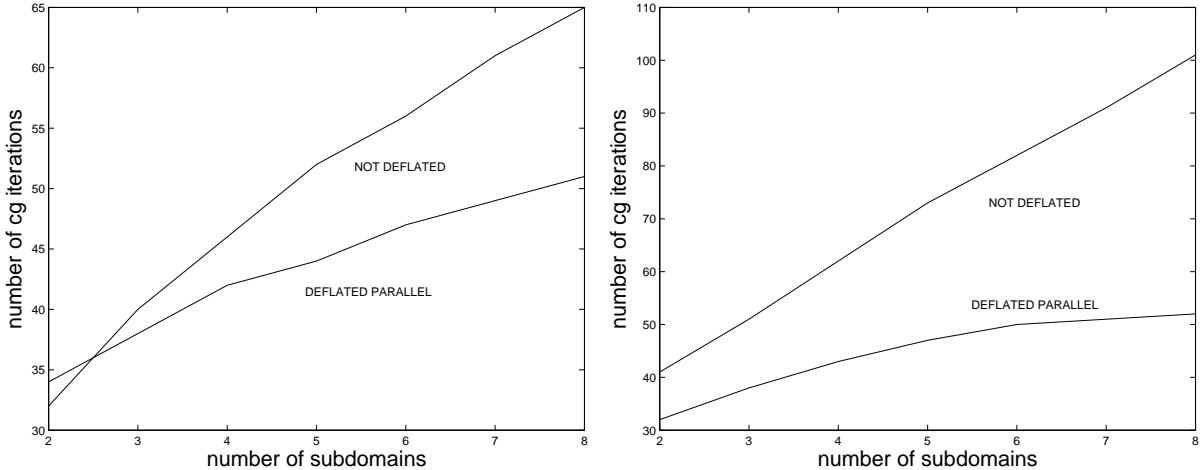


Figure 3: Left: Number of iterations for a nonisotropic medium with $k_1 = 1, k_2 = 5$ and $k_3 = 10$. Right: Number of iterations for an nonisotropic medium with $k_x = 10, k_y = 5$ and $k_z = 1$.

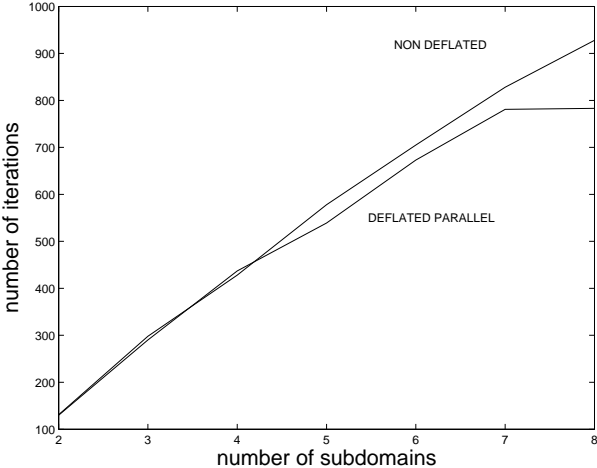


Figure 4: Number of iterations for a anisotropic medium with $k_x = 1, k_y = 5$ and $k_z = 1000$.

the convergence behavior, that is

$$-\operatorname{div} \left(k_x \frac{\partial u}{\partial x}, k_y \frac{\partial u}{\partial y}, k_z \frac{\partial u}{\partial z} \right) = 0, \quad (9)$$

with otherwise the same boundary conditions as in equation (1). As in the other simulations, the stacking has been done in the z -direction. If the permeability is high in the direction of the adjacent layers, then, it is observed that the scalability of the deflated parallel method changes, see Figures 3 (left) and 4. The simulations with more processors indicate that the deflation makes the number of iterations scalable if the number of processors is very large. In Figure 3 (right) it can be seen that choosing the permeability high in a different direction from the stacking direction, influences the scalability less significantly. More research is needed to explain these results.

5 Conclusions

The usual Choleski preconditioning still leaves the discretization matrix with an excessively large condition number and hence the convergence of the preconditioned conjugate gradient is still unacceptably slow. Deflation improves the speed of convergence very much. Deflation even improves the speed of convergence if the adjacent layers do not differ in permeability in a parallel computing environment. Hence for reservoir simulators deflation is a very useful technique because of the large scale computing, which needs a parallel computing setting and because of the abrupt changes of parameters across the interfaces between adjacent layers.

References

- [1] K. Aziz and A. Settari. *Petroleum reservoir simulation*. Applied Science Publishers, London and New York, 1979.
- [2] H. De Gerssem and K. Hameyer. A deflated iterative solver for magnetostatic finite element models with large differences in permeability. *Eur. Phys. J. Appl. Phys.*, 13:45–49, 2000.
- [3] J. Frank and C. Vuik. On the construction of deflation-based preconditioners. *SIAM J. Sci. Comput.*, pages 442–462, 2001.
- [4] C. B. Jenssen and P. Å. Weinerfelt. Coarse grid correction scheme for implicit multiblock Euler calculations. *AIAA Journal*, 33(10):1816–1821, 1995.
- [5] J. Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9:233–241, 1993.
- [6] R. Nabben and C. Vuik. A comparison of deflation and the balancing preconditioner. *SIAM Journal of Scientific Computing*, 27:1742–1759, 2006.
- [7] A. Padiy, O. Axelsson, and B. Polman. Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems. *SIAM J. Matrix Anal. Appl.*, 22:793–818, 2000.

- [8] F.J. Vermolen, C. Vuik, and A. Segal. Deflation accelerated parallel preconditioned conjugate gradient method in finite element problems. *Numerical Mathematics and advanced applications*, pages 825–833, 2004.
- [9] F.J. Vermolen, C. Vuik, and A. Segal. Deflation in preconditioned conjugate gradient methods for finite element problems. *in: Conjugate gradient algorithms and finite element methods, Scientific computation*, pages 103–129, 2004.
- [10] C. Vuik, A. Segal, L. el Yaakoubi, and E. Dufour. A comparison of various deflation vectors applied to elliptic problems with discontinuous coefficients. *Appl. Numer. Math.*, 41:219–233, 2002.
- [11] C. Vuik, A. Segal, and J. A. Meijerink. An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients. *J. Comput. Phys.*, 152:385–403, 1999.
- [12] C. Vuik, A. Segal, J. A. Meijerink, and G. T. Wijma. The construction of projection vectors for a Deflated ICCG method applied to problems with extreme contrasts in the coefficients. *J. Comput. Phys.*, 172:426–450, 2001.

