

C. Vuik*, A. Segal*, J.A. Meijerink†

An efficient preconditioned CG method for the solution of layered problems with extreme contrasts in the coefficients

* Delft University of Technology, Faculty of Information Technology and Systems, Department of Technical Mathematics and Informatics, Mekelweg 4, 2628 CD Delft, The Netherlands, **e-mail:** c.vuik@math.tudelft.nl

† Shell International Exploration and Production, P.O. Box 60, 2280 AB Rijswijk, The Netherlands

Summary.

Knowledge of the excess pressure is valuable in the prediction of the presence of oil and gas reservoirs. A mathematical model for the prediction of excess pressures is given by a time-dependent diffusion equation. Application of the finite element method leads to a system of linear equations. A complication is that the underground consists of layers with very large differences in permeability. This implies that the symmetric and positive definite coefficient matrix has a very large condition number. Bad convergence behaviour of the CG method has been observed, moreover a classical termination criterion is not valid in this problem. After scaling of the matrix the number of extreme eigenvalues is reduced and equal to the number of layers with a high permeability. To annihilate the effect of these eigenvalues a deflated CG method is used. The convergence rate improves considerably and the termination criterion is reliable. Finally a cheap approximation of the eigenvectors is proposed.

Keywords: porous media, preconditioned Conjugate Gradients, deflation, Poisson equation, discontinuous coefficients across layers, eigenvectors, finite element method

AMS Subject Classification: 65F10, 65F15, 76S05

1 Introduction

One of the problems an oil company is confronted with when drilling for oil, is the presence of excess fluid pressures within the rock layers of the subsurface. Knowledge of the excess pressure is valuable in the prediction of the presence of oil and gas in reservoirs and a key factor in safety and environmental aspects of drilling a well.

A mathematical model for the prediction of excess pressures in a geological time scale is based on conservation of mass and Darcy's law ([5] and [10]). This leads to a time-dependent diffusion equation, where the region also changes in time, as rocks are deposited or eroded. The Euler Backward is used for the time integration. In order to solve this diffusion equation, the finite element method is applied. As a consequence in each time-step a linear system of equations has to be solved. Due to non-linear effects and the time-dependence of the region the coefficients of the diffusion equation change in each time-step.

In practical applications we are faced with large regions in a three-dimensional space and as a consequence a large number of finite elements is necessary. The matrix itself is sparse, but due to fill-in a direct method requires too much memory to fit in core. Moreover, since in each time-step we have a good start vector, only iterative methods are acceptable candidates for the solution of the linear systems of equations.

Since these equations are symmetric a preconditioned conjugate gradient method (ICCG) [18] is a natural candidate. Unfortunately an extra complication of the physical problem we are dealing with, is that the underground consists of layers with very large differences in permeability. For example in shale the permeability is of the order 10^{-6} to 10^{-11} (D), whereas in sandstone it is of the order 1 to 10^{-4} (D). Hence a contrast of 10^{-7} is common in the system of equations to be solved.

A large contrast in coefficients usually leads to a very ill-conditioned system of equations to be solved. Since the convergence rate of ICCG depends on the distribution of the eigenvalues of the matrix one may expect a slow convergence rate. In Section 2 it is shown that this is indeed the case. An even more alarming phenomenon is that numerical results suggest that ICCG has reached a certain accuracy but that the actual accuracy is in fact orders worse. This is due to the ill-conditioned matrix which results in the standard termination criterion

no longer being reliable. To our knowledge this observation has not been made before.

An analysis of the problem in Section 3 shows that without preconditioning there are many small eigenvalues in the matrix, but using an IC preconditioned matrix this number is reduced to the number of sandstone layers that not reach the earth surface. This surprising observation is also new. This analysis suggests a way of solving the problems mentioned. In Section 4 it is shown that the convergence and reliability of the termination criterion is considerably improved by projecting the solution in each iteration onto the orthogonal complement of the space spanned by the eigenvectors corresponding to the very small eigenvalues of the preconditioned matrix. The idea is that, assuming that the vectors are written as linear combination of eigenvectors, the components corresponding to these specific eigenvectors do not play a role anymore. As a result one may expect a much faster convergence and a reliable termination criterion. A clear disadvantage of this method is of course that one has to compute the specific eigenvectors. In Section 5, however, it is shown, how one can approximate these eigenvectors easily, based on physical arguments. Furthermore it is shown, that even approximate eigenvectors lead to a fast convergence. Finally in Section 6 some numerical evidence of our improved algorithm is given.

The CG [12] method, combined with a preconditioner, is a popular iterative method to solve large algebraic systems of linear equations, when the system matrix is symmetric and positive definite. Many practical preconditioners are based on an Incomplete Choleski factorization. The resulting ICCG method has been first described in [18]. Later on various alternatives have been formulated, such as MICCG [11], RICCG [3], and ILUM [24]. An incomplete factorization preconditioner for singular systems has been investigated in [22]. Recently a number of preconditioners have been proposed for the discretized Poisson equation, where the rate of convergence does not depend on the grid size. Examples are: NGILU [28] and DRIC [23]. A comparison of these and related preconditioners has been given in [6].

It is well known that the convergence rate of CG depends on the ratio of the largest and smallest eigenvalue of the system matrix. To explain the superlinear convergence of CG, Ritz values have to be taken into account [29]. The convergence rate only depends on active eigenvalues. An eigenvalue is active when the error has a non zero component in the corresponding eigenvector. This observation is used to solve singular systems with the CG method (see [2] p. 476-480). In [15] the initial approximation is projected such that the start residual is perpendicular to the kernel of the matrix. In [1] the start approximation is projected in such a way that the error has no components in the eigenvectors corresponding to small eigenvalues. This increases the smallest active eigenvalue and thus the convergence rate. After the projection the original CG method has been used in both papers.

In [21] and [20] a deflated CG method has been proposed. In every CG iteration the residual is projected onto a chosen subspace. The projected CG method used in this work (compare [31]) is closely related to these deflated CG methods. The main difference is the choice of the subspace. We base our choice on the physical properties of the problem considered. Another difference is the implementation. Various implementations are possible to incorporate a projection. We specify an implementation such that the basis of our subspace consist of vectors with many zero elements. Related work [13], [26] has been presented at the Copper Mountain conference on iterative methods.

Deflation is also used in iterative methods for non-symmetric systems of equations [19], [16], [9], [8], [7], [4]. In these papers the smallest eigenvalues have been shifted away from the origin. The eigenvectors are in general obtained from the Arnoldi method. The motivation to use deflation is to enhance the convergence of restarted GMRES [25]. Finally deflation techniques have also been combined with solution methods for systems of non-linear equations [27] and [30].

2 Statement of the problem and experiments with ICCG

As mentioned in the introduction, in each time-step we have to solve a system of equations that arises from the discretization of a 3D time-dependent diffusion equation. In this paper, however, we are only interested in the convergence behaviour of the ICCG process for problems with layers with large contrasts in the coefficients. For that reason we simplify the equation considerably and assume that we have to solve the stationary linearized 2D diffusion equation, in a layered region:

$$-\operatorname{div}(\sigma \nabla p) = 0, \quad (1)$$

with p the excess pressure and σ the permeability. At the earth surface the excess pressure is prescribed. At the artificial boundaries we take a zero flux condition. For our model problem we assume that σ in sandstone is equal to 1 and σ in shale is equal to 10^{-7} . Furthermore the Dirichlet boundary condition at the earth surface is set equal to 1. The solution of equation (1) with these boundary conditions is of course $p = 1$, but if we start with $p = 0$ or a random vector, our linear solver will not notice the difference with a real problem. Numerical experiments show that the choice of start vector has only marginal effects.

Equation (1) is discretized by a standard finite element method using bilinear quadrilateral elements. This results in a system of linear equations to be solved, which will be denoted as $Ax = b$. In our first experiment we have solved this problem on a rectangular domain with 7 straight layers (Figure 1), using CG without preconditioner. The termination criterion is based on the estimation of the smallest eigenvalue during the iterations by a Lanczos method as described by Kaasschieter [14]. Figure 2 shows the norm of the residual, the norm of the error and also the estimation of the smallest eigenvalue as function of the number of iterations. In each layer 10 elements in the horizontal and 5 elements in the vertical direction are used. From this figure the following remarkable observations may be made.

1. The residual decreases monotonously between iterations 1 and 30. For the iterations between 31 and 1650 we have an erratic behaviour of the residual. After iterations 1650 again we have a monotone decreasing of the residual.
2. If we require an accuracy of order 10^{-2} , the process would stop after approximately 25 iterations, since then the residual divided by the estimation of the smallest eigenvalue is small enough. Unfortunately the error is still large because the estimation of the smallest eigenvalue is very inaccurate.
3. In iterations 1-30 it looks as if the smallest eigenvalue is in order 10^{-2} , whereas from iteration 31 it is clear that the smallest eigenvalue is of order 10^{-7} .

So we see that the bad condition leads to a large number of iterations. Moreover, for practical values of the error, the termination criterion is not reliable.

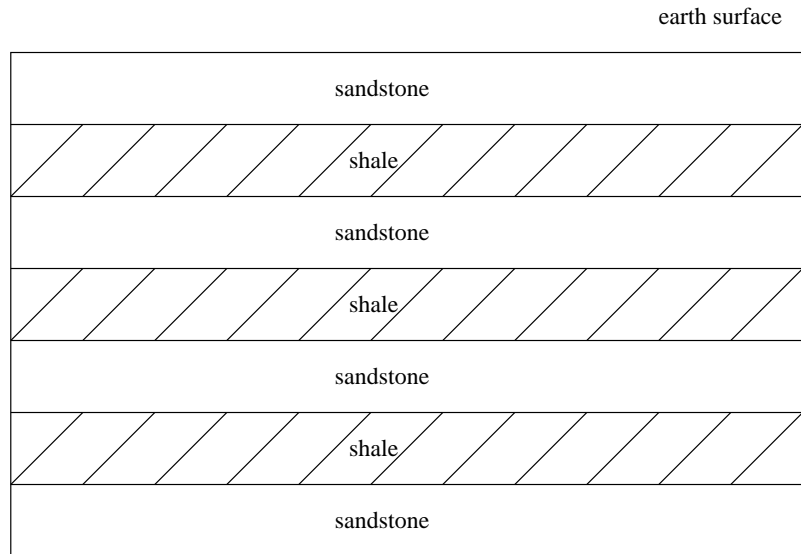


Fig. 1. Artificial configuration with 7 straight layers

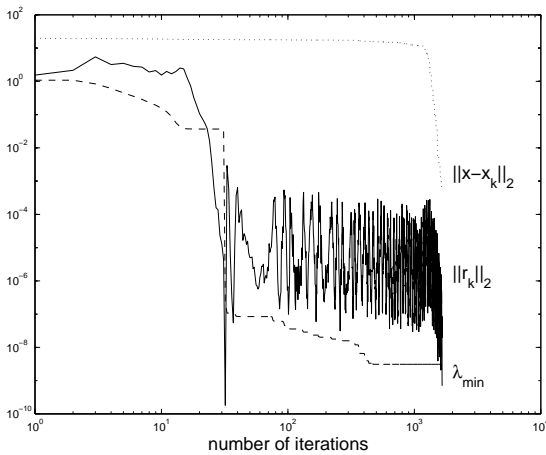


Fig. 2. Convergence behaviour of CG without preconditioning

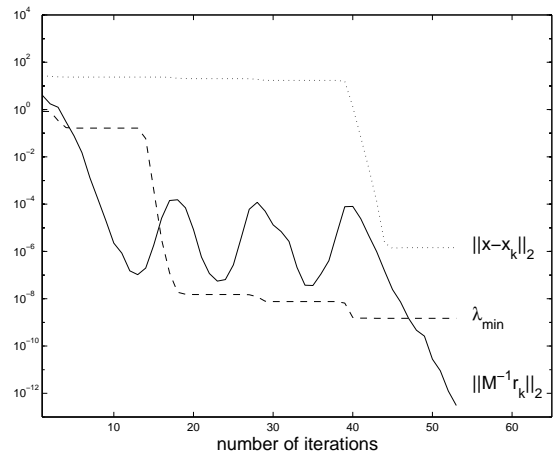


Fig. 3. Convergence behaviour of CG with IC preconditioning

Repeating the same experiment using an IC preconditioning gives a drastic reduction of the number of iterations, but still the same conclusions as for the case without preconditioning can be drawn. Figure 3 shows the convergence behaviour. Note that the horizontal scales in Figures 2 and 3 are quite different. Although the number of iterations (48) is small compared to the non-preconditioned algorithm (1650), still it is quite large compared to the number of unknowns (385).

3 Analysis of the iteration matrix

In order to get more insight in the convergence behaviour, we have investigated the eigenvalues of the matrix. If we compute all eigenvalues of the discretization matrix, then we see that the number of small eigenvalues (i.e. of order 10^{-7}), is equal to the number of nodes that are

entirely in the shale layers plus 3. One can expect that this number is at least equal to the number of internal "shale" nodes, since all non-zero elements in the corresponding rows of the matrix are of order 10^{-7} . The number 3 will be explained later on. The iteration process only converges, once all small eigenvalues have been "discovered".

When we use an IC preconditioner, and compute all eigenvalues of the discretization matrix multiplied by the preconditioning matrix, we see that only 3 eigenvalues are of order 10^{-7} . All other eigenvalues are of order 1. This observation appears to be true for all kinds of preconditioners, even for a simple diagonal scaling. The convergence behaviour shown in Figure 3 can be explained by these 3 eigenvalues. Once a small eigenvalue has been "discovered" by the CG process, the residual increases considerably. Only when all small eigenvalues have become visible to the algorithm, the actual error decreases.

A possible explanation for the fact that there are only 3 small eigenvalues in the preconditioned case is the following. The preconditioner will scale the Laplacian equation per layer in such a way that the rows with small elements at the diagonal, will get elements of the order 1. However, in a shale layer we have a Neumann boundary condition at the "side" walls. But at the top and bottom we have a sandstone layer. Since the permeability in sandstone is much larger than in shale, the pressure in the sandstone may be considered as more or less constant. So from the view of a shale layer we have a kind of Dirichlet boundary condition for the top and bottom. On the other hand for the sandstone layers, the shale layers may be regarded as more or less impermeable. The interface condition is approximately a Neumann boundary condition. So for each sandstone layer between two shale layers we have to solve a Laplacian equation with approximately Neumann boundary conditions. Only at the top layer we have a given Dirichlet boundary condition. Since the solution of the Neumann problem is fixed upon an additive constant we may expect a small eigenvalue for each sandstone layer, that has no explicit Dirichlet boundary conditions. So it is reasonable to expect 3 small eigenvalues in this particular example. A mathematical proof of this observation is given below.

Let our rectangular region consist of a sequence of $2n + 1$ plain layers of equal thickness with at the top a sand layer and further downwards alternatingly shale and sand layers. The permeability of the sand and shale layers is 1 respectively $\epsilon > 0$. We choose a rectangular mesh with a uniform mesh size h in both x and y direction such that the sand/shale interfaces coincide with element boundaries. We discretize (1) by applying the standard first order bilinear FE-method and numerical integration with the element corner-points as the integration points. Numbering the unknowns locally from left to right and top to bottom the element matrix S for a single element in the sand is:

$$S = \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & 1 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix}.$$

The element matrix in shale is ϵS .

Assembling all the element matrices leads to a system of equations $Ax = b$, where A is a symmetric $m \times m$ M-matrix. If we group the equations and unknowns belonging to a single layer together, add the unknowns and equations of each sand/shale interface to the corresponding

$$\hat{A} = \hat{\Delta} + \mathcal{E}, \quad (2)$$

where $\hat{\Delta}$ is the block-diagonal matrix with as first block $\hat{\Delta}_h^T$, the diagonally scaled Δ_h^T , and then further down alternately $\hat{\Delta}_h^D$, the scaled Δ_h^D , and $\hat{\Delta}_h^N$, the scaled Δ_h^N . The eigenvalues $\lambda_j^{\hat{\Delta}}$ of $\hat{\Delta}$ are equal to the eigenvalues of all its diagonal blocks. Let $\{\lambda_j^T\}_{0 \leq j \leq m_1}$, $\{\lambda_j^D\}_{0 \leq j \leq m_2}$ and $\{\lambda_j^N\}_{0 \leq j \leq m_3}$ be the ordered eigenvalues of respectively $\hat{\Delta}_h^T$, $\hat{\Delta}_h^D$ and $\hat{\Delta}_h^N$. It is well known that:

$$\lambda_0^N = 0, \quad (3)$$

and that there exists a $c2(h)$ such that

$$\begin{aligned} c2(h) &\leq \lambda_j^N \leq 2 \text{ for } 1 \leq j \leq m_3, \\ c2(h) &\leq \lambda_j^T \leq 2 \text{ for } 0 \leq j \leq m_2, \\ c2(h) &\leq \lambda_j^D \leq 2 \text{ for } 0 \leq j \leq m_3. \end{aligned} \quad (4)$$

The blocks of the symmetric tridiagonal block matrix \mathcal{E} are given by:

$$\begin{aligned} \mathcal{E}_{1,1} &= \hat{H}_0 - \hat{\Delta}_h^T && \text{and for } 1 \leq i \leq n, \\ \mathcal{E}_{2i-1,2i} &= \hat{I}_i^T, \\ \mathcal{E}_{2i,2i} &= \hat{L}_i - \hat{\Delta}_h^D, \\ \mathcal{E}_{2i,2i+1} &= \hat{J}_i^T, \\ \mathcal{E}_{2i+1,2i+1} &= \hat{H}_i - \hat{\Delta}_h^N. \end{aligned} \quad (5)$$

For $0 \leq i \leq n$, $\mathcal{E}_{2i+1,2i+1}$ contain only non-zero matrix entries which relates nodes in the interface with their neighbouring interior nodes. Their values are

$$\frac{-1 + \sqrt{1 + \epsilon}}{2\sqrt{2} + 2\epsilon} = O(\epsilon). \quad (6)$$

For $1 \leq i \leq n$, $\mathcal{E}_{2i,2i} = 0$ and $\mathcal{E}_{2i-1,2i}$, $\mathcal{E}_{2i,2i+1}$, $\mathcal{E}_{2i,2i+1}$ and $\mathcal{E}_{2i+1,2i}$ have only non zero elements on the off-diagonal, relating interface nodes to their direct neighbour in the low permeable layer. The value of these entries is:

$$\frac{-\sqrt{\epsilon}}{2\sqrt{2} + 2\epsilon} = O(\sqrt{\epsilon}). \quad (7)$$

Let Q be the block-diagonal orthogonal matrix such that $Q^T \hat{\Delta} Q = \Lambda^{\hat{\Delta}}$, and B a block diagonal matrix of which the blocks are defined by:

$$\begin{aligned} B_{2i+1,2i+1} &= (\sqrt{\epsilon}/c3) I && \text{for } 0 \leq i \leq n, \\ B_{2i,2i} &= I && \text{for } 1 \leq i \leq n, \end{aligned} \quad (8)$$

where $c3$ is an arbitrary constant. If we now define $\bar{A} = B^{-1} Q^T \hat{A} Q B$ then

$$\bar{A} = B^{-1} Q^T \hat{\Delta} Q B + B^{-1} Q^T \mathcal{E} Q B = \bar{\Delta} + \bar{\mathcal{E}}. \quad (9)$$

The blocks of $\bar{\Delta}$ just contain the eigenvalues of $\hat{\Delta}$ (which satisfy equation (3) and inequalities (4)) and for the blocks of $\bar{\mathcal{E}}$ we find that

$$\begin{aligned}
& \text{the elements of } \bar{\mathcal{E}}_{2i+1,2i+1} = O(\epsilon), \\
& \text{the elements of } \bar{\mathcal{E}}_{2i-1,2i} \text{ and } \bar{\mathcal{E}}_{2i+1,2i} = O(c3), \\
& \text{the elements of } \bar{\mathcal{E}}_{2i,2i+1} \text{ and } \bar{\mathcal{E}}_{2i+2,2i+1} = O(\epsilon/c3), \\
& \text{the elements of } \bar{\mathcal{E}}_{2i,2i} = 0.
\end{aligned} \tag{10}$$

If we now choose $c3 < c2(h)/4$ and subsequently ϵ small enough, apply Gershgorin's theorem to \bar{A} and account for the fact that each eigenvalue of \hat{A} is also an eigenvalue of \bar{A} and in the interval $(0,2)$ then

$$\begin{aligned}
0 < \lambda_j^{\hat{A}} = O(\epsilon) & \quad \text{for } 0 \leq j \leq n-1, \\
c2(h)/2 + O(\epsilon) \leq \lambda_j^{\hat{A}} < 2 & \quad \text{for } n \leq j \leq m.
\end{aligned} \tag{11}$$

This proves the following theorem:

Theorem 3.1. *For ϵ small enough the diagonally scaled matrix $D^{-1/2}AD^{-1/2}$ has only n eigenvalues of $O(\epsilon)$, where n is the number of sand layers between shale layers.*

4 The Deflated ICCG method

In this section we derive a Deflated Incomplete Choleski Conjugate Gradient method. This method can be used to solve the system of linear equations for the excess pressure. In the previous section it has been shown that the preconditioned matrix has only a small number of very small eigenvalues. The deflation is used to annihilate the effect of these eigenvalues on the convergence of the ICCG method.

Let $Ax = b$ be the system of equations to be solved, where A is a symmetric and positive definite (SPD) matrix. Let M be the Incomplete Choleski decomposition of A satisfying $A \approx LL^T = M$, where L is a sparse lower triangular matrix and M is SPD. ICCG consists of the application of CG to the following preconditioned system

$$L^{-1}AL^{-T}y = L^{-1}b, \quad x = L^{-T}y.$$

Define $\tilde{A} = L^{-1}AL^{-T}$ and $\tilde{b} = L^{-1}b$. Note that \tilde{A} is SPD.

To define the Deflated ICCG method we assume that the vectors v_1, \dots, v_m are given and form an independent set. These vectors define a space $\mathcal{V} = \text{span}\{v_1, \dots, v_m\}$ and a matrix $V = [v_1 \dots v_m]$. A special choice for v_i are the eigenvectors corresponding to the smallest eigenvalues of \tilde{A} hence $\tilde{A}v_i = \lambda_i v_i$, $0 < \lambda_1 \leq \lambda_n \dots \leq \lambda_n$.

The operator P defined by $P = I - VE^{-1}(\tilde{A}V)^T$ with $E = (\tilde{A}V)^T V$, is a projection with the following properties (the matrix $E \in \mathbb{R}^{m \times m}$ is symmetric and positive definite):

Theorem 4.1. *The operator P has the following properties:*

- i $PV = 0$ and $P^T \tilde{A}V = 0$,
- ii $P^2 = P$,
- iii $\tilde{A}P = P^T \tilde{A}$.

Proof. Properties i) and iii) are easily checked. The proof of ii) runs as follows

$$\begin{aligned} P^2 &= (I - VE^{-1}(\tilde{A}V)^T)(I - VE^{-1}(\tilde{A}V)^T) \\ &= P - VE^{-1}(\tilde{A}V)^T + VE^{-1}(\tilde{A}V)^TVE^{-1}(\tilde{A}V)^T = P. \end{aligned}$$

⊠

Corollary 4.1. *The matrix $\tilde{A}P$ is symmetric and positive semi-definite.*

Remark 4.1. When v_i are eigenvectors of \tilde{A} with norm equal to 1 then $P = I - VV^T$ because $v_i^T v_j = \delta_{ij}$.

We assume that the start vector x_0 is zero. If $x_0 \neq 0$ the Deflated ICCG algorithm should be applied to $A(x - x_0) = b - Ax_0$. To speed up the convergence of ICCG we assume that the space \mathcal{V} is chosen such that it contains the slow converging components and split the vector y into two parts

$$y = (I - P)y + Py. \quad (12)$$

The first part $(I - P)y$ is the component of y contained in \mathcal{V} , whereas the second part Py is perpendicular to \mathcal{V} in the $(\cdot, \cdot)_{\tilde{A}}$ inner product. The first part is determined from:

$$(I - P)y = VE^{-1}(\tilde{A}V)^T y = VE^{-1}V^T \tilde{b}. \quad (13)$$

To compute the second part Py we use $\tilde{A}Py = P^T \tilde{A}y = P^T \tilde{b}$, and solve y from

$$P^T \tilde{A}y = P^T \tilde{b}. \quad (14)$$

The singular system (14) has a solution because $P^T \tilde{b}$ is an element of the Range $(P^T \tilde{A})$. A solution y of (14) may contain an arbitrary element of Null $(P^T \tilde{A}) = \mathcal{V}$. Since $PV = 0$, Py is uniquely determined.

When we apply the CG algorithm to the symmetric positive semi-definite system (14) we get the Deflated ICCG algorithm:

DICCG1

$k = 0, y_0 = 0, \tilde{p}_1 = \tilde{r}_0 = P^T L^{-1}b,$

while $\|\tilde{r}_k\| > \varepsilon$ **do**

$k = k + 1;$

$\alpha_k = \frac{(\tilde{r}_{k-1}, \tilde{r}_{k-1})}{(\tilde{p}_k, P^T \tilde{A} \tilde{p}_k)};$

$y_k = y_{k-1} + \alpha_k \tilde{p}_k;$

$\tilde{r}_k = \tilde{r}_{k-1} - \alpha_k P^T \tilde{A} \tilde{p}_k;$

$\beta_k = \frac{(\tilde{r}_k, \tilde{r}_k)}{(\tilde{r}_{k-1}, \tilde{r}_{k-1})};$

$\tilde{p}_{k+1} = \tilde{r}_k + \beta_k \tilde{p}_k;$

end while

In order to get an approximation of $y (= L^T x)$ the vector y_k is multiplied by P and substituted in (12).

In order to determine the matrix V we have to compute (or approximate) the eigenvectors

of the matrix \tilde{A} . Unfortunately these eigenvectors contain many non-zero elements in our application. Furthermore they are hard to predict on physical grounds. The eigenspace of the matrix $L^{-T}L^{-1}A$ corresponding to the m smallest eigenvalues can be approximated by the span of m sparse vectors, which are obtained on physical grounds. For that reason we rewrite the Deflated ICCG algorithm as follows:

Define $\bar{P} = L^{-T}PL^T$, $\hat{r}_k = P^T L^{-1}r_k = L^{-1}\bar{P}^T r_k = L^{-1}\hat{r}_k$, with $\hat{r}_k = \bar{P}^T r_k$, and $z_k = L^{-T}L^{-1}\hat{r}_k$. Since $y_k = L^T x_k$ and $L^T x_k = L^T x_{k-1} + \alpha_p \tilde{p}_k$, we choose $\tilde{p}_k = L^T p_k$. Substitution in DICCG1 leads to

DICCG2

```

 $k = 0, y_0 = 0, \hat{r}_0 = \bar{P}^T r_0, p_1 = z_1 = L^{-T}L^{-1}\hat{r}_0;$ 
while  $\|\hat{r}_k\| > \varepsilon$  do
   $k = k + 1;$ 
   $\alpha_k = \frac{(\hat{r}_{k-1}, z_{k-1})}{(p_k, \bar{P}^T A p_k)};$ 
   $x_k = x_{k-1} + \alpha_k p_k;$ 
   $\hat{r}_k = \hat{r}_{k-1} - \alpha_k \bar{P}^T A p_k;$ 
   $z_k = L^{-T}L^{-1}\hat{r}_k;$ 
   $\beta_k = \frac{(\hat{r}_k, z_k)}{(\hat{r}_{k-1}, z_{k-1})};$ 
   $p_{k+1} = z_k + \beta_k p_k;$ 
end while

```

It is easy to verify that the projection $\bar{P} = L^{-T}PL^T$ has the following properties:

Properties of \bar{P}

1. $\bar{P} = I - \bar{V}E^{-1}(A\bar{V})^T$ where $\bar{V} = L^{-T}V$ and $E = (\tilde{A}V)^T V = (A\bar{V})^T \bar{V}$,
2. $\bar{P}\bar{V} = 0$, and $\bar{P}^T A\bar{V} = 0$,
3. $\bar{P}^T A = A\bar{P}$.

The vector x can be splitted into two parts (compare equation (12)):

$$x = (I - \bar{P})x + \bar{P}x . \quad (15)$$

The first part can be calculated as follows

$$(I - \bar{P})x = \bar{V}E^{-1}\bar{V}Ax = \bar{V}E^{-1}\bar{V}^T b .$$

For the second part we project the solution x_k obtained from DICCG2 to $\bar{P}x_k$.

For the special choice that v_i are eigenvectors of \tilde{A} , \bar{v}_i are eigenvectors of $L^{-T}L^{-1}A$. In that case the projection can be written as $\bar{P} = I - \bar{V}(LL^T\bar{V})^T$.

A well known convergence result for CG applied to $\tilde{A}y = \tilde{b}$ is ([17] p.187):

$$\|y - y_k\|_{\tilde{A}} \leq 2\|y - y_0\|_{\tilde{A}} \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^k , \quad (16)$$

where $K = K_2(\tilde{A}) = \frac{\lambda_n}{\lambda_1}$. Since the results obtained from DICCG1 and DICCG2 are equal we restrict our convergence research to DICCG1. When we choose $V = [v_1 \dots v_m]$ where v_i are the normalized eigenvectors of \tilde{A} , it is easy to verify that

$$\begin{aligned} P^T \tilde{A}v_i &= 0 \quad \text{for } i = 1, \dots, m, \\ P^T \tilde{A}v_i &= \lambda_i v_i \quad \text{for } i = m+1, \dots, n. \end{aligned}$$

On the space $\text{span}\{v_{m+1}, \dots, v_n\}$ the norm $\|\cdot\|_{P^T \tilde{A}} = \|\cdot\|_{\tilde{A}P}$ is well defined. Using this norm together with Inequality (16) it can be proved that

$$\|Py - Py_k\|_2 \leq 2\sqrt{K}\|Py - Py_0\|_2 \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right)^k,$$

where $K = \frac{\lambda_n}{\lambda_{m+1}}$. This inequality predicts a speed up of convergence when $\lambda_m \ll \lambda_{m+1}$.

Since Py and Py_k are elements of $\text{span}\{v_{m+1}, \dots, v_n\}$ the following expression holds:

$$\|P^T \tilde{b} - P^T \tilde{A}y_k\|_2 = \|\tilde{A}Py - \tilde{A}Py_k\|_2 \geq \lambda_{m+1}\|Py - Py_k\|_2. \quad (17)$$

So the following termination strategy can be used: estimate λ_{m+1} via Kaasschieter's method [14] and terminate DICCG1 when

$$\|P^T \tilde{b} - P^T \tilde{A}y_k\|_2 \leq \frac{\varepsilon}{\lambda_{m+1}}.$$

This together with Inequality (17) implies that $\|Py - Py_k\|_2 \leq \varepsilon$. For DICCG2 a comparable termination criterion can be derived.

5 A choice of projection vectors

A good choice of the projection vectors is important to obtain an efficient Deflated ICCG method. In this section we restrict ourselves to the class of problems defined in Section 2. An analysis of the matrix (Section 3) shows that the spectrum of this matrix contains many small eigenvalues (of order 10^{-7}). For the preconditioned matrix, the number of small eigenvalues is drastically reduced. This number is proportional to the amount of sandstone layers. In Section 4 a Deflated ICCG method is given, which is very suitable to problems where the matrix has a small number of extreme eigenvalues.

We consider the problem as shown in Figure 1. As a first choice we take v_1, v_2, v_3 equal to the three eigenvectors of \tilde{A} corresponding to the small eigenvalues. We use DICCG1 with $P = I - VV^T$. The vectors v_i should be stored, so $3n$ extra memory positions are needed. Furthermore in every iteration of DICCG1, the projection P should be applied to a vector, which costs 3 inner products and 3 vector updates extra per iteration.

Drawbacks of this choice are:

1. the determination of the eigenvectors can be expensive,
2. the amount of extra memory and work per iteration grows, when the number of small eigenvalues increases.

For the determination of the eigenvectors an inverse (Krylov) iteration can be used, however this costs more work than the solution of the original system. In our application the excess pressure is needed in every time iteration. The differences in the matrices in consecutive time-steps are relatively small. In such a problem DICCG1, with eigenvectors as projection vectors,

can be feasible when the eigenvectors are only computed at a small number of time steps.

Because of these drawbacks we use another approach, motivated by the properties of the eigenvectors $\bar{v}_i = L^{-1}v_i$ of $L^{-T}L^{-1}A$, corresponding to the small eigenvalues. For the problem considered a vertical cross section of the eigenvectors is plotted in Figure 4. The cross sections

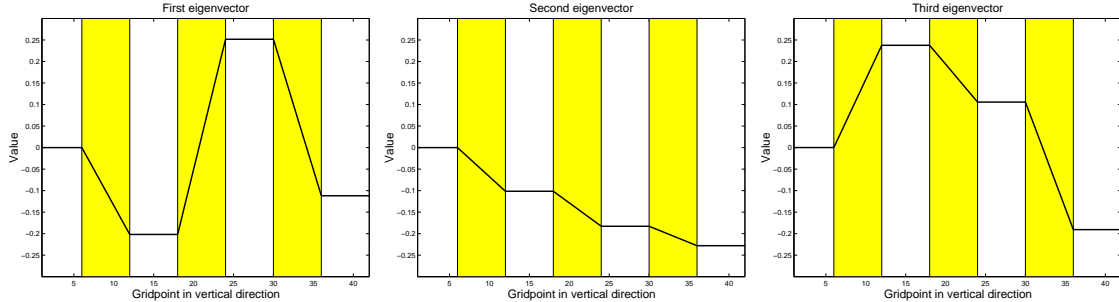


Fig. 4. The vertical cross section of the eigenvectors corresponding to the small eigenvalues

have the following properties:

- their value is constant in sandstone layers,
- their value is zero in the first sandstone layer,
- in the shale layers their graph is linear.

So the space $\text{span}\{\bar{v}_1, \bar{v}_2, \bar{v}_3\}$ is identical to the space $\text{span}\{w_1, w_2, w_3\}$, where the vertical cross sections of w_i are defined by:

- the value of w_i is one in the $i + 1^{\text{th}}$ sandstone layer and zero in the other sandstone layers,
- their graph is continuous in the whole domain and linear in the shale layers.

So instead of DICCG1 with the eigenvectors DICCG2 is applied with $\mathcal{V} = \text{span}\{w_1, w_2, w_3\}$. Since the vectors w_i are no eigenvectors it is necessary to store w_i and Aw_i . Due to the sparseness 2 memory vectors are sufficient to store all w_i . Furthermore the elements of Aw_i are only non-zero at the grid points connected to the interfaces of the i^{th} shale layer. Thus 2 memory vectors are also sufficient to store all vectors Aw_i . In the same way the sparseness can be used to save CPU time. It is possible to implement the projection such that the extra amount of work per iteration is less than 2 inner products and 2 vector updates independent of the number of small eigenvalues. This makes the DICCG2 algorithm very attractive for this kind of problems.

We have also solved problems where shale and sandstone layers are slightly curved. Again DICCG2, with projection vectors defined in the same way as above, proved to be an efficient solution algorithm. If we assume that the sandstone layers without a Dirichlet condition are numbered from 1 to m , then we propose to use DICCG2 with the projection vectors w_i chosen as:

- the value of w_i is one in the i^{th} sandstone layer and zero in the other sandstone layers,
- in the shale layers, w_i satisfies

$$-\text{div}(\sigma \Delta w_i) = 0, \quad (18)$$

and on the interfaces it satisfies a Dirichlet boundary condition equal to the constant value 0 or 1 of the neighbouring sandstone layer.

For our original problem, this choice leads to the same projection vectors as before. The solution of (18) amounts to solving the same system of equations at a much smaller domain without the extreme contrasts in the coefficients. In fact this process is similar to a domain decomposition method.

6 Numerical experiments

In order to test the Deflated ICCG method we have applied DICCG2 to the 7 straight layers problem defined in Section 2. The 3 projection vectors are defined as in the previous section. For this straight layers case these vectors span exactly the space of the 3 eigenvectors corresponding to the small eigenvalues. Figure 5 shows the convergence behaviour of the DICCG2 method, the estimation of the smallest eigenvalue as well as the error. It is clear that we have an enormous improvement compared to the results without projection as shown in Figure 3. Because of the sparse structure of the approximate eigenvectors the overhead per iteration is very moderate. Besides that, the decrease of the residual is now a measure for the error, so that we have a reliable termination criterion. Our intention is to use the

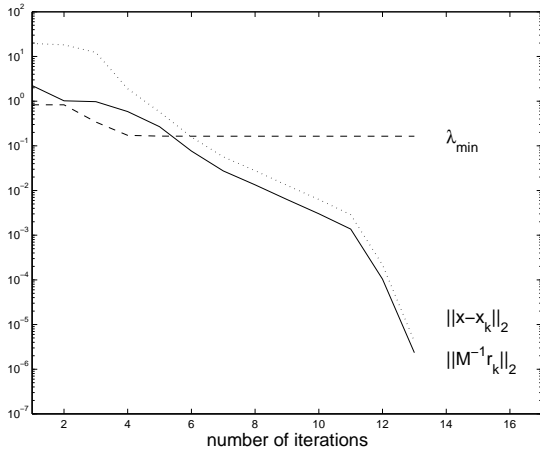


Fig. 5. Convergence behaviour of DICCG2 for the straight layers problem

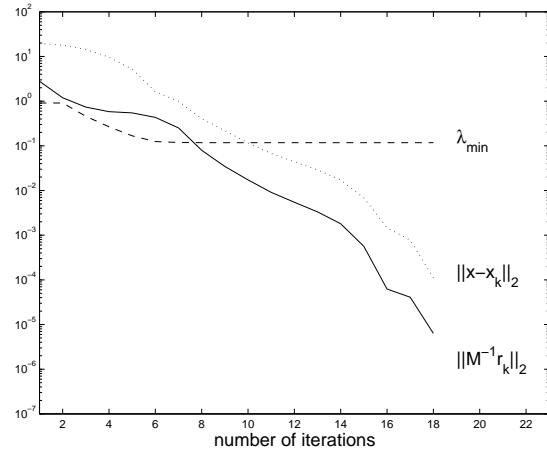


Fig. 6. Convergence behaviour of DICCG2 for the parallel arcs problem

DICCG2 method also for the case that we have "approximate" eigenvectors. Therefore we have replaced the straight layers in our example by curved layers as shown in Figure 7a and 7b. The number of elements is exactly the same as for the straight layers region. For these examples the graphs of the vertical cross sections of the eigenvectors are no longer linear in the shale layers. Nevertheless we use exactly the same projection vectors in DICCG2 as for the straight layers problem. The convergence behaviour of the DICCG2 method applied to the mesh of Figure 7a is shown in Figure 6. The number of iterations has been increased compared to the straight layers case, but the overall behaviour is the same. Application of the DICCG2 method to the mesh of Figure 7b leads to the same convergence behaviour as the one for the straight layers (Figure 5). Presumably, the projection vectors are good approximations

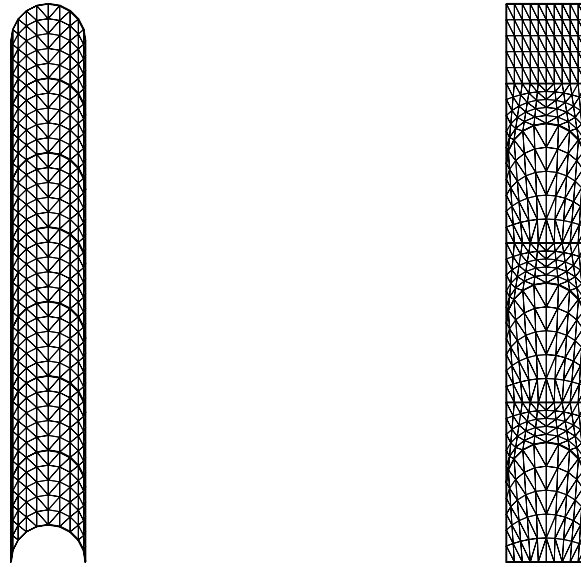


Fig. 7. a) Curved mesh consisting of parallel arcs b) Curved mesh with straight and curved layers

of the eigenvectors in these cases. From our limited number of experiments it is clear that the DICCG2 method is an enormous improvement compared to the classical ICCG method, provided the approximate eigenvectors are a reasonable estimate of the true eigenvectors.

7 Conclusions

It has been shown that the preconditioned Conjugate Gradient method for layered problems with extreme contrasts in the coefficients has a very erratic convergence behaviour. The residual shows large bumps and moreover the decrease of the residual cannot be used as reliable termination criterion. Only when all eigenvectors corresponding to small eigenvalues are detected, the smallest Ritz values are converged to the smallest eigenvalues, the convergence behaviour is more or less as usual. In order to solve this problem a new method called DICCG has been developed that projects the contribution of the eigenvectors corresponding to the small eigenvalues onto the null space. This new method has excellent convergence properties and more important a reliable termination criterion. Even if we use approximations of these eigenvectors based on physical arguments still the deflated ICCG method performs very good.

It is our aim to apply the DICCG method to large time-dependent 3D problems with a realistic number and shape of layers. A point to be solved, however, is how to create the approximate eigenvectors in more general configurations including inclusions. We think that it is sufficient to solve the original problem for each completely enclosed shale layer with approximate boundary conditions. Since we are only dealing with approximate eigenvectors we expect that

the solution of the sub-problem may be done with a moderate accuracy. The choice of the approximate eigenvectors as well as the sensitivity of the method to the accuracy of these approximate eigenvectors is the subject of our present research.

Acknowledgement

The authors thank Mohammed Mokaddam for doing numerical experiments.

1. M. Arioli and D. Ruiz. Block Conjugate Gradient with subspace iteration for solving linear systems. In S.D. Margenov and P.S. Vassilevski, editors, *Iterative methods in linear algebra II, Proceedings of the Second IMACS International Symposium on Iterative Methods in Linear Algebra, Blagoevgrad, Bulgaria, June 17-20, 1995*, IMACS series in computational and applied mathematics 3, pages 64–79, Piscataway, 1996. IMACS.
2. O. Axelsson. *Iterative solution methods*. Cambridge University Press, Cambridge, UK, 1994.
3. O. Axelsson and G. Lindskog. On the eigenvalue distribution of a class of preconditioning methods. *Numer. Math.*, 48:479–498, 1986.
4. J. Baglama, D. Calvetti, G.H. Golub, and L. Reichel. Adaptively preconditioned GMRES algorithms. SCCM- 96-15, Stanford University, Stanford, 1996.
5. J. Bear. *Dynamics of fluids in porous media*. American Elsevier publishing company, New York, 1972.
6. E.F.F. Botta, K. Dekker, Y. Notay, A. van der Ploeg, C. Vuik, F.W. Wubs, and P.M. de Zeeuw. How fast the Laplace equation was solved in 1995. *Appl. Num. Meth.*, 24:439–455, 1997.
7. K. Burrage, J. Erhel, and B. Pohl. A deflation technique for linear systems of equations. *SIAM J. Sci. Comp.*, 1997. To appear.
8. A. Chapman and Y. Saad. Deflated and augmented Krylov subspace techniques. *Num. Lin. Alg. Appl.*, 4:43–66, 1997.
9. J. Erhel, K. Burrage, and B. Pohl. Restarted GMRES preconditioned by deflation. *J. Comp. Appl. Math.*, 69:303–318, 1996.
10. Melvyn R. Giles. *Diagenesis: a quantitative perspective; implications for basin modelling and rock property prediction*. Kluwer, Dordrecht, 1997.
11. I.A. Gustafsson. A class of first order factorization methods. *BIT*, 18:142–156, 1978.
12. M.R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *J. Res. Nat. Bur. Stand.*, 49:409–436, 1952.
13. Victoria E. Howle and Stephen A. Vavasis. Preconditioning complex-symmetric layered systems arising in electrical power modeling. In T.A. Manteuffel and S.F. McCormick, editors, *Fifth Copper Mountain Conference on Iterative Methods, Copper Mountain, Colorado, March 30 - April 3, 1998*, 1998.
14. E.F. Kaasschieter. A practical termination criterion for the conjugate gradient method. *BIT*, 28:308–322, 1988.
15. E.F. Kaasschieter. Preconditioned conjugate gradients for solving singular systems. *J. of Comp. Appl. Math.*, 24:265–275, 1988.
16. S.A. Kharchenko and A.Yu. Yeregin. Eigenvalue translation based preconditioners for the GMRES(k) method. *Num. Lin. Alg. Appl.*, 2:51–77, 1995.
17. D.G. Luenberger. *Introduction to linear and nonlinear programming*. Addison Wesley, New York, 1973.
18. J.A. Meijerink and H.A. Van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31:148–162, 1977.
19. R.B. Morgan. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.*, 16:1154–1171, 1995.
20. S.G. Mulyarchik, S.S. Bielawski, and A.V. Popov. Efficient computational schemes of the conjugate gradient method for solving linear systems. *J. Comp. Phys.*, 110:201–211, 1994.
21. R.A. Nicolaides. Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.*, 24:355–365, 1987.
22. Y. Notay. Incomplete factorizations of singular linear systems. *BIT*, 29:682–702, 1989.
23. Y. Notay. DRIC: a dynamic version of the RIC method. *J. Num. Lin. Alg.*, 1:511–532, 1994.
24. Y. Saad. ILUM: a multi-elimination ILU preconditioner for general sparse matrices. *SIAM J. Sci. Comput.*, 17:830–847, 1996.
25. Y. Saad and M.H. Schultz. GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 7:856–869, 1986.
26. Y. Saad, M. Yeung, J. Ehrel, and F. Guyomarc'h. A deflated version of the Conjugate Gradient method. In T.A. Manteuffel and S.F. McCormick, editors, *Fifth Copper Mountain Conference on Iterative Methods, Copper Mountain, Colorado, March 30 - April 3, 1998*, 1998.

27. G. Shroff and H.B. Keller. Stabilization of unstable procedures: the recursive projection method. *SIAM J. Num. Anal.*, 30:1099–1120, 1993.
28. A. van der Ploeg, E.F.F. Botta, and F.W. Wubs. Nested grids ILU-decomposition (NGILU). *J. Comp. Appl. Math.*, 66:515–526, 1996.
29. A. van der Sluis and H.A. van der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.*, 48:543–560, 1986.
30. H. van der Veen, K. Vuik, and R. de Borst. Post-bifurcation behavior in soil plasticity: eigenvector perturbation compared to deflation. In D.R.J. Owen, E. Oñate, and E. Hinton, editors, *Computational Plasticity; fundamentals and applications, Part 2, Proceedings of the Fifth International Conference on Computational Plasticity, Barcelona, Spain, 17-20 March, 1997*, pages 1745–1751, Barcelona, 1997. CIMNE.
31. Kees Vuik, Guus Segal, and Koos Meijerink. An efficient CG method for layered problems with large contrasts in the coefficients. In T.A. Manteuffel and S.F. McCormick, editors, *Fifth Copper Mountain Conference on Iterative Methods, Copper Mountain, Colorado, March 30 - April 3, 1998*, 1998.