An efficient CG method for layered problems with large contrasts in the coefficients

Kees Vuik* Guus Segal* Koos Meijerink†

1 Introduction

One of the problems an oil company is confronted with when drilling for oil, is the presence of large excess pressures in underground layers. Knowledge of the excess pressure may be of big help in the prediction of the presence of oil and gas reservoirs. Another reason why the pressure distribution is important, is because of safety and environmental aspects during a drilling process.

A mathematical model for the prediction of excess pressures in a geological time scale is based on conservation of mass and Darcy's law [1]. This leads to a time-dependent diffusion equation, where the region also changes in time. 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.

Since the system of equations is symmetric a preconditioned Conjugate Gradient method (ICCG) [3] is a natural candidate to solve it. Unfortunately a 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}, whereas in sandstone it is of the order 1 to 10^{-4}. 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. Since the convergence rate of ICCG depends on the distribution of the eigenvalues of the matrix [7] 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 means that due to ill-conditioning the standard termination criterion is no longer reliable. To our knowledge this observation has not been made before.

In [6] and [5] a deflated CG method is proposed. In every CG iteration the residual is projected

*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

onto a chosen subspace. The projected CG method used in this work 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. Deflation is also used in iterative methods for non-symmetric systems of equations [4].

2 Statement of the problem and experiments with ICCG

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:

$$-\mathrm{div}\sigma\nabla p = 0\ , \tag{1}$$

with p the excess pressure and σ the permeability. At the earth surface the excess pressure is prescribed and at the artificial boundaries we assume 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 boundary conditions is of course $p = 1$, but if we start with $p = 0$ or a random vector, the 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

$$Ax = b\ . \tag{2}$$

In our first experiment we have solved (2) on a rectangular domain with 7 straight layers (Figure 1), using a standard CG solver 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 [2]. 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. 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 decrease 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.

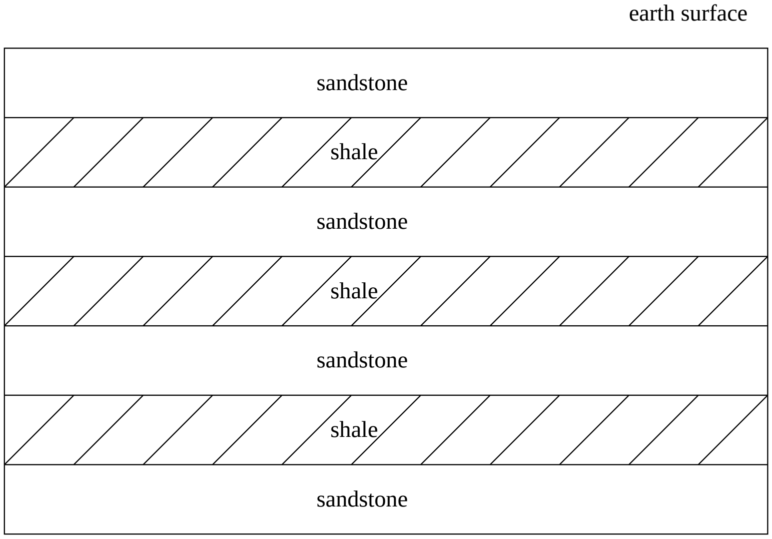

Figure 1: Artificial configuration with 7 straight layers

Repeating the same experiment using an ILU 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. Mark 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).

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 are "discovered".

When we use an ILU 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. The convergence behaviour shown in Figure 3 can be explained by these 3 eigenvalues. Once a small eigenvalue is "discovered" by the CG process, the residual increases considerably. Only when all small eigenvalues are visible to the algorithm, the actual error decreases.

3 The Deflated ICCG method

In this section we derive a Deflated Incomplete Choleski Conjugate Gradient method. 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.

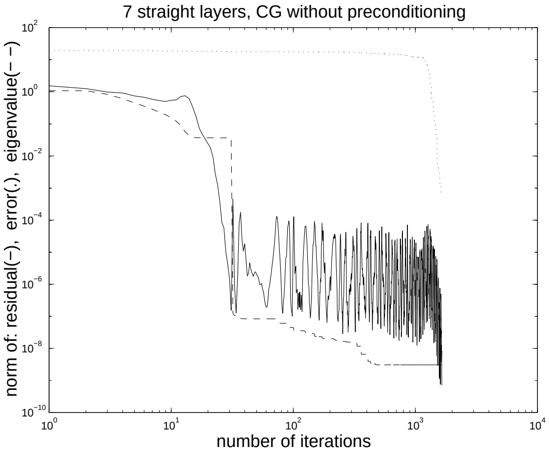

Figure 2: Convergence behaviour of CG without preconditioning

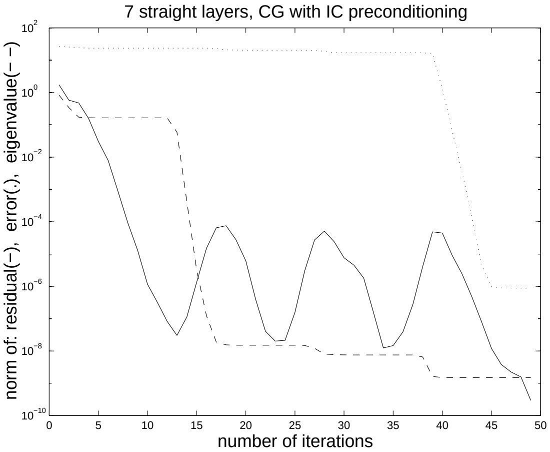

Figure 3: Convergence behaviour of CG with ILU preconditioning

We write the system of equations to be solved as:

$$Ax = b \ ,$$

where A is a symmetric and positive definite (SPD) matrix. An Incomplete Choleski decomposition is determined such that $A \approx LL^T = M$, 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 $v_1, ..., v_m$ are given and form an independent set. These vectors define a space $\mathcal{V} = span\{v_1, ..., v_m\}$ and a matrix $V = [v_1 ... v_m]$. The projection P is defined by

$$P = I - VE^{-1}(\tilde{A}V)^T \quad \text{with} \quad E = (\tilde{A}V)^T V \ .$$

In the remainder of this section 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

$$y = (I - P)y + Py \ . \tag{3}$$

The first part is determined from:

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

To compute the second part (Py) we use

$$\tilde{A}Py = P^T\tilde{A}y = \ P^T\tilde{b} \ ,$$

and solve y from

$$P^T\tilde{A}y = P^T\tilde{b} \ . \tag{4}$$

The singular system (4) has a solution because $P^T\tilde{b}$ is an element of the Range $(P^T\tilde{A})$. A solution y of (4) 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 (4) we get the Deflated ICCG algorithm:

DICCG

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

When the vector y_k is multiplied by P and substituted in (3) we get an approximation of $y = L^T x$. In order to determine the matrix V we have to compute (or approximate) the eigenvectors of the matrix $\tilde{A}$.

4 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 2) 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 3 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. The choice of the projection vectors is 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 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 $span\{\bar{v}_1, \bar{v}_2, \bar{v}_3\}$ is identical to the space $span\{w_1, w_2, w_3\}$, where the vertical cross section of w_i is defined by:

- the value of w_i is one in the $i + 1^{\text{th}}$ sandstone layer and zero in the other sandstone layers,

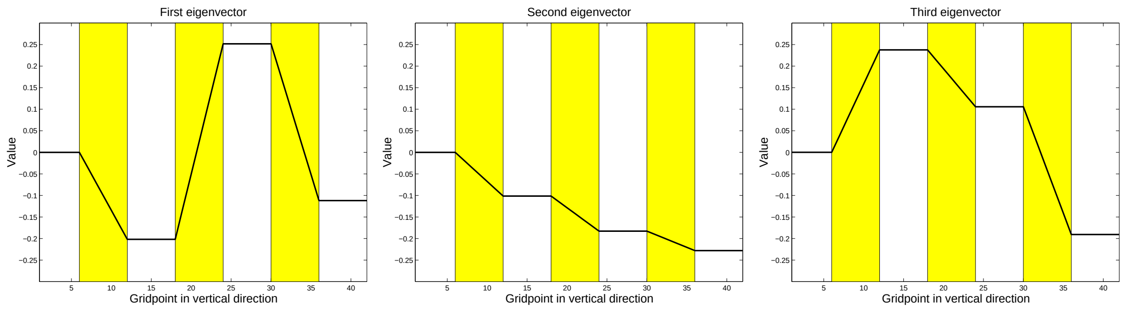

Figure 4: The vertical cross section of the eigenvectors corresponding to the small eigenvalues

- their graph is continuous in the whole domain and linear in the shale layers.

These vectors are used in our DICCG method.

5 Experiments

In order to test the deflated ICCG method we have applied DICCG to the 7 straight layers problem defined in Section 2. The 3 ”approximate” eigenvectors are defined as in the previous section. For this straight layer case this means that they span exactly the space of the 3 eigenvectors corresponding to the small eigenvalues. Figure 5 shows the convergence behaviour of the DICCG 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.

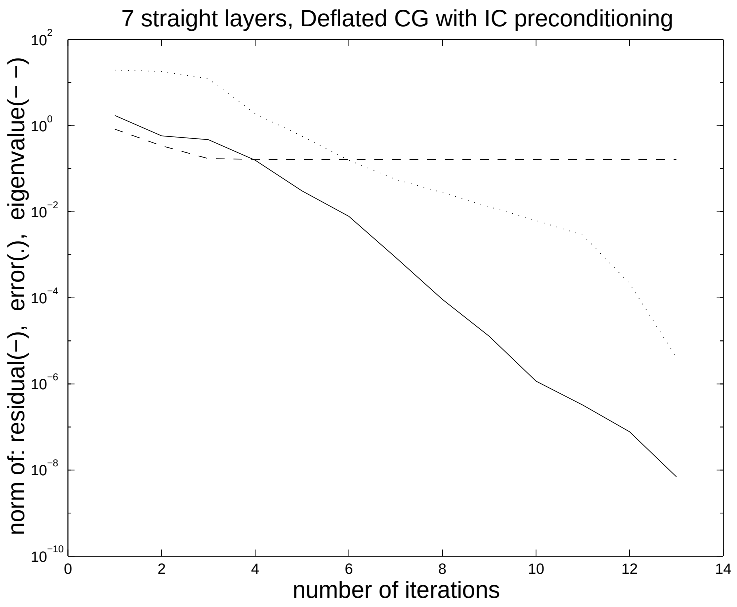

Figure 5: Convergence behaviour of DICCG for the straight layer problem

6 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 eigen vectors 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.

References

[1] J. Bear. *Dynamics of fluids in porous media.* American Elsevier publishing company, New York, 1972.

[2] E.F. Kaasschieter. A practical termination criterion for the conjugate gradient method. *BIT*, 28:308–322, 1988.

[3] 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.

[4] R.B. Morgan. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.*, 16:1154–1171, 1995.

[5] 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.

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

[7] A. van der Sluis and H.A. van der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.*, 48:543–560, 1986.