Abstract

This paper summarizes the Orthomin\(^1\) and GMRES\(^2\) linear solvers. It presents a mapping from the Orthomin residual basis over to the GMRES basis that is used to switch solvers dynamically during the linear solve step in a reservoir simulator. This mapping enables a combined algorithm to adapt easily to problem complexity.

Both solvers estimate the same residual norm at the end of each iteration. For simple problems that require few iterations, Orthomin is generally faster. This is because GMRES requires an additional iteration to recover the solution vector, whereas Orthomin updates the solution at each iteration. In contrast, GMRES requires only half the memory and reduced CPU time during the orthogonalization step and is therefore more suited to large or difficult problems.

The criteria for switching over from Orthomin to GMRES is problem specific and is best determined by internal CPU timers.

Introduction

Modern geostatistical packages provide high levels of detail and heterogeneity. When such grids are used inside current generation reservoir simulators, the linear solver becomes a severe bottleneck both in memory and in CPU time. This is because the Jacobian matrix no longer displays a single strong inner direction and preconditioners such as Nested Factorization\(^3\) are less effective. Typically, the iterative solvers take an order of magnitude more iterations and the time spent is dominated by the orthogonalization step. This is very well illustrated by 10\(^{th}\) SPE Comparative Solution Study\(^3\). Attempts to simulate this model using a conventional black oil, fully implicit simulator motivated the current work.

This paper presents a combined Orthomin and GMRES solver that adjusts itself to problem complexity. For small stack sizes Orthomin is used, while for medium sized ones a crossover is made to GMRES. This takes advantage of the single memory stack and reduced computation during the orthogonalization phase.

Overview of Orthomin and GMRES

Both algorithms use an iterative procedure to improve an initial estimate, \(x_0\), of the solution to the linear system

\[ Ax = b \]  

(1)

To determine the solution, each algorithm recasts (1) as the solution to a Least Squares (LS) minimization problem:

Minimize \([b-Ax_k]\) \_2

s.t. \(x_k \in x_0 + K(r_0, A, k)\)

where \(K(r_0, A, k) = \text{span}\{r_0, Ar_0, \ldots, A^kr_0\}\) is the Krylov subspace.

Both algorithms solve (2) by constructing an orthogonal basis that spans \(K(r_0, A, k)\).
The algorithms are mathematically similar and differ mainly in their treatment of the LS minimization. In Orthomin, this is performed at the end of each iteration. In GMRES a Hessenberg reduction of the original system is constructed and the LS problem is solved only after the estimated residual has converged. Both solvers estimate the same value of the residual norm at the end of each iteration.

There are two main consequences of this:

- The basis vectors produced by each algorithm are not the same, even though the same generator and initial vectors, \( r_0 \), are used to create the Krylov space. In Orthomin, the basis is constructed from the optimal step directions taken in the residual space. In GMRES, the basis is the orthonormal Arnoldi vectors. This paper derives the relation between the two sets of basis vectors.

- In Orthomin, the minimum residual and solution vectors are always available. In GMRES the solution vector need only be calculated after the residual norm estimate indicates convergence.

In (1) the matrix \( A \) is typically large, sparse and cannot be readily inverted. A preconditioning matrix \( B \) is used to accelerate the convergence of the iterative solution of (2).

Introducing right preconditioning of the original matrix replaces (2) with

\[
\text{Minimize } \| b - A B^{-1} z_k \|_2 \\
\text{s.t. } z_k \in z_0 + K(r_0 , A B^{-1}, k) \\
\text{where} \\
x_k = B^{-1} z_k
\]

Left preconditioning leads to

\[
\text{Minimize } \| B^{-1} (b - A x_k) \|_2 \\
\text{s.t. } x_k \in x_0 + K(r_0 , B^{-1} A, k)
\]

B is chosen so that the system \( Bx = z \) can be easily solved and that \( B^{-1} \) approximates the inverse of \( A \). In this paper, the examples used Nested Factorization Preconditioning. The preconditioning was designed to ensure that the sum of the elements of the residual vector was initially zero.

Additionally, both algorithms are started after a non-optimal full step of the Orthomin algorithm, This introduces the matrix \( (B-A) \) into the Krylov space generator and ensures that the sum of the residuals remains zero during subsequent iterations.

The reason for not using GMRES to solve all cases is related to the residual norm being minimized. The right preconditioned version of GMRES requires an extra matrix inversion to recover the solution vector (eqn 4). For simple problems, with <10 iterations this is a significant overhead. Since Orthomin uses right preconditioning and updates the solution at each iteration, no such extra inversion is required.

Interestingly, a left preconditioned version of GMRES does not require an extra final inversion. However, for consistency, the initial residual must be transformed to the new norm. This places the additional inversion at the start of the algorithm. Failure to transform the residual will produce a working algorithm that is potentially faster than Orthomin. This algorithm was implemented. Unfortunately, the convergence behaviour differed unpredictably from Orthomin and it is not to be recommended. Furthermore, the choice of termination criteria is more difficult since the \( B^{-1} \) norm is a solution norm and selecting appropriate termination criteria for this norm becomes problem dependent. The residual norm is the one we are most familiar in reservoir simulation.

The idea of restarting GMRES with the latest Orthomin estimate was implemented but the convergence was found to be slower than either algorithm on their own. This motivated the desire to produce a mapping that allowed GMRES to utilise the Orthomin basis at the point of restarting. Fig. 1 illustrates the importance of including this sub space on the convergence rate of subsequent GMRES iterations.
The following sections reproduce the algorithms without derivation in order to establish a common notation prior to introducing the crossover formulae.

**Orthomin**

The minimum residual method introduced by Vinsome$^1$ is as follows:

\[
\begin{align*}
r_0 &= b, \quad x_0 = 0, \quad k = 0 \\
\text{while} \quad ( \| r_k \| > \varepsilon ) \\
\delta x_k &= B^{-1} r_k \\
\delta r_k &= A \delta x_k \\
\delta r_k &= \delta r_k - \sum_{i=0}^{k-1} \beta^i S_i \\
\delta x_k &= \delta x_k - \sum_{i=0}^{k-1} \beta^i S_i \\
S_k &= \delta r_k \\
r_{k+1} &= r_k - \alpha_k S_k \\
x_{k+1} &= x_k + \alpha_k S_k \\
k &= k + 1
\end{align*}
\]

**endwhile**

where

\[
\begin{align*}
\alpha_k &= \frac{(r_k \cdot \delta r_k)}{(\delta r_k \cdot \delta r_k)} \\
\beta^k_i &= \delta r_k \cdot S_i
\end{align*}
\]

The step length, $\alpha_k$, at each iteration is chosen to minimize the $L_2$ norm of the residual $r_{k+1}$. By construction, the $S_i$ are a set of orthogonal vectors which decouple the LS minimization problem from the previous search directions. $\alpha_k$ is determined from a simple 1D minimization along direction $\delta r_k$. 
GMRES

The Generalized Minimum residual method introduced by Saad\textsuperscript{2} is as follows:

\[ g_0 = b, x_0 = 0, h_{l0} = \| g_0 \|_2, k = 0 \]

\textbf{while} \ (h_{k+1,k} > \varepsilon) \ 

\[ v_{k+1} = \frac{g_k}{h_{k+1,k}} \]

\[ k = k + 1 \]

\[ z_k = B^{-1}v_k \]

\[ g_k = Az_k \]

\[ g_k = g_k - \sum_{i=1}^{k} h_{i,k} v_i \]

\[ h_{k+1,k} = \| g_k \|_2 \]

\textbf{endwhile}

where

\[ h_{i,k} = v_i, g_k \]

\textbf{After convergence}

\[ z_k = \sum_{i=1}^{k} v_i y_i \]

\[ x_k = x_0 + B^{-1}z_k \]

The scalar \( y_i \) are chosen so as to minimize

\[ \| h_{i0} e_1 - H_k Y_k \|_2 \]

where

\[ H_k = [h_{i,k}] \]

\[ Y_k = (y_1, y_2, \ldots, y_k) \]

\[ e_1 = (1, 0, \ldots) \]

It can be shown that

\[ \| b - AX_k \|_2 = h_{k+1,k} \]

\( h_{k+1,k} \) is then the residual error at the \( k^{th} \) step. It is this relation that avoids GMRES having to generate the solution vector in order to estimate the current residual norm. GMRES iterates until the \( h_{k+1,k} \) reaches the desired tolerance. The \( x_k \) are determined by solving the upper Hessenberg LS problem. In practice the Hessenberg matrix is converted to upper triangular form using Givens rotations at the end of each iteration.
Orthomin to GMRES Crossover Formulae

This section presents the equations used to implement the crossover from the Orthomin residual stack to the GMRES stack.

The combined algorithm runs with Orthomin until internal timers indicate that the cost of the duplicate orthogonalization phase is dominating. The GMRES algorithm then takes over for the same number of steps as Orthomin, accumulating the Hessenberg factorization, but using the crossover formulae to avoid the costly matrix multiplications and inversions. Thereafter, GMRES continues until convergence is achieved.

This procedure only works because both algorithms start from the same residual and employ the same matrix operator, $M = AB^{-1}$ to generate the orthogonal vectors.

The first two iterations of Orthomin produce:

$$S_1 = Mr_0$$
$$r_1 = r_0 - \alpha_i S_1$$
$$S_2 = Mr_1 - \beta_i^1 S_1$$
$$= S_1 - \alpha_i MS_1 - \beta_i^1 S_1$$

This suggests that $MS_n$ can be expressed as a linear function of the set of existing basis vectors together with the next one to be generated. The following general recurrence relation can be derived:

$$MS_n = \frac{S_1 - S_{n+1} - \sum_{i=1}^{n} \alpha_i MS_i - \sum_{i=1}^{n} \beta_i^1 S_i}{\alpha_n} \quad (5)$$

This relation will be used to avoid the expensive matrix inversions during the crossover phase.

The first two iterations of GMRES, using the known Orthomin residual stack vectors are reproduced below:

$$v_1 = \frac{r_0}{\|r_0\|}$$
$$g_1 = Mv_1 - (Mv_1) v_1$$
$$= \frac{S_1}{\|r_0\|} - (Mv_1) v_1$$
$$v_2 = \frac{g_1}{\|g_1\|}$$
$$g_2 = Mv_2 - (Mv_2) v_1 - (Mv_2) v_2$$
$$= \frac{MS_1}{\|g_1\| \|r_0\|} - \frac{(Mv_1) v_1}{\|g_1\|} - (Mv_2) v_1 - (Mv_2) v_2$$
$$= \frac{MS_1}{\|g_1\| \|r_0\|} - (Mv_1) v_1 - (Mv_2) v_2$$

This suggests that the $g_k$ can be expressed as a linear function of the $S_k$ plus an extra term containing the Arnoldi vectors constructed by GMRES: 

$$g_k = F_k(S) - \sum_{i=1}^{k} (X_i v_i), v_i$$

where
The components of the $F_k(S)$ vector can be calculated using the recursive formula:

$$F_k(S) = \frac{M F_{k-1}(S) - \sum_{i=1}^{k-1} X_{k-1,i} F_i(S)}{\|g_{k-1}\|}$$

with

$$F_1(S) = (f_1^{-1}, 0) = (1.0/\|r_0\|, 0)$$

and

$$X_1 = f_1^{-1} S_1$$

In the above formula $F_k(S)$ and $X_k$ are merely different representations of the same vector. The first term in equation (6), $MF_{k-1}(S)$, is evaluated using the $f_{k-1}$ and recurrence relation (5). $X_{k-1}$ is then determined and the second orthogonalization term in (6) can be evaluated. During the crossover, the Orthomin solution stack, which is no longer needed, is used to temporarily store the $X_k$.

**Conclusions**

The Orthomin and GMRES algorithms are shown to be very similar and that combining both in a single linear solver is not a significant coding or maintenance overhead. The combined solver can exploit the significant speed and memory advantage of GMRES on difficult problems.

The crossover formulae have been applied to standard SPE benchmarks and models built from realistic geological grids. As expected, no significant overhead was incurred during the crossover phase. On large and difficult problems (>40 iterations) GMRES showed a consistent 25% reduction in the CPU usage during the orthogonalization step. This led to a 10% reduction in total linear solver time (including the extra final iteration). On smaller problems, (<10 iterations) the reduction in total CPU usage was not significant.

**Nomenclature**

- $A$ = Jacobian matrix
- $B$ = Preconditioning matrix
- $r$ = Orthomin residual vector
- $g$ = GMRES residual vector
- $S$ = Orthomin residual basis vector
- $v$ = GMRES Arnoldi basis vector
- $x$ = solution vector containing pressures and saturations
- $z$ = preconditioned solution variable in GMRES
- $i,j,k$ = index or iteration counters
- $h_{ij}$ = (ij) element of the Hessenberg factorization of $A$
- $H_k$ = the partial Hessenberg factorization matrix
- $a_k$ = Orthomin step size multiplier, minimizing the $k^{th}$ residual
- $\beta^k$ = Projection of $i^{th}$ Orthomin basis vector onto $k^{th}$ residual estimate
- $F_k(S)$ = Generator function for GMRES iteration, using the Orthomin stack during crossover.
- $f_k^i$ = coefficient of basis vector $S_i$ in generator $F_k(S)$
- $X_k$ = evaluation of $F_k(S)$
Fig 1. Restarted GMRES vs Orthomin with Crossover to GMRES

References
