Lapack Working Note 156 / ICL-UT-02-03 / UT-CS-02-477

Polynomial acceleration of optimised multi-grid smoothers; basic theory

Victor Eijkhout*

March 2002

1 Introduction

It is possible to implement multi-grid smoothers such as the Gauss-Seidel and SOR method in such a way that there is considerable cache-reuse [2]. The resulting code can have a performance several times higher than that of a naive implementation. The crux to this optimisation is the out-of-order execution of several iterations at one time: the part of the residual in cache is subjected to several smoothing steps before the same iterations are applied to another part of the vector. Such a reordering of the operations does not change the semantics of the floating point numbers produced in any way.

While from a numerical point of view one might want to accelerate the smoother by a polynomial method such as CG or GMRES, in practice this cache-aware performance optimisation precludes such a numerical optimisation, since the inner products make out-of-order iterating impossible. However, it is possible to store several iterations worth of the Stationary Iteration (SI) vectors, and reconstruct from them the iterates that would have been produced by an orthogonality-based method with the smoother as preconditioner.

We will derive the relevant mathematics here, and indicate various computational aspects of the algorithm.

The advantages of such an approach to constructing CG iterates are twofold. First of all we optimise the computation of the matrix vector product and the preconditioner application, giving them a measure of cache re-use. Secondly, the vector operations that occur in normal CG are now largely bundled into a QR decomposition, raising their efficiency from Blas Level 1 to Level 3.

2 Mathematical underpinnings

Sequences of residuals constructed by different iterative methods, but from the same starting vector, all span the same Krylov space. Thus, it is possible in principle to

* This research was funded by SciDAC: TeraScale Optimal PDE Simulations (TOPS), DoE Contract #DE-FC02-01ER25480, and DoE Contract #W-7405-ENG-48 B519921.

construct CG residuals and iterates from SI residuals and iterates by taking linear combinations. The problem is how to find the exact coefficients of the combinations.

Here we are helped by the following lemma.

Lemma 1 *Let $R^{(1)}$ and $R^{(2)}$ be the sequences of residuals of two different iterative methods with the same coefficient matrix A, preconditioner M, and starting vector r_1, then each sequence consists of convex combinations of the other:*

$$r_i^{(2)} = \sum_{j=1}^{i} \alpha_{i,j} r_j^{(1)}, \qquad \sum_j \alpha_{i,j} = 1.$$

Proof. Let $X^{(1)}$ the series of iterates of which $R^{(1)}$ are the residuals. The iterates satisfy a relation

$$x_{i+1}^{(1)} = x_1^{(1)} + \pi_i^{(1)}(M^{-1}A)M^{-1}r_1,$$

(where $\{\pi_i^{(1)}\}_i$ are a series of inhomogeneous polynomials), from which

$$M^{-1}r_{i+1}^{(1)} = M^{-1}r_1^{(1)} + M^{-1}A\pi_i^{(1)}(M^{-1}A)M^{-1}r_1$$

That is, $r_i^{(1)} = \tilde{\pi}_i^{(1)}(AM^{-1})r_1$ where $\tilde{\pi}_i^{(1)}(0) = 1$. We can also write this as

$$R^{(1)} = KU^{(1)}$$

where K is the Krylov sequence based on AM^{-1} and r_1, and $U^{(1)}$ is an upper triangular matrix with $u_{1j}^{(1)} \equiv 1$. Likewise, $R^{(2)} = KU^{(2)}$ for a similarly normalized upper triangular matrix $U^{(2)}$.

This gives us $R^{(2)} = R^{(1)}V$ with $V = {U^{(1)}}^{-1}U^{(2)}$. Writing $U^{(2)} = U^{(1)}V$, we find

$$1 = u_{1j}^{(2)} = \sum_k u_{1k}^{(1)} v_{kj} = \sum_k v_{kj},$$

that is, V has column sums $\equiv 1$. •

If the residuals of one method are convex combinations of those of another method, the iterates follow from the same combination process.

Lemma 2 *Let $R^{(1)}$ and $R^{(2)}$ be residual sequences such that $R^{(2)} = R^{(1)}U$ where U has unit column sums, then the iterates sequences $X^{(1)}$ and $X^{(2)}$ follow the same relation: $X^{(2)} = X^{(1)}U$.*

Proof. We can write in matrix notation $R^{(1)} = AX^{(1)} - fe^t$ where $e^t = (1, \ldots)$. Noting that $e^tU = e^t$, we get

$$\begin{aligned} R^{(2)} &= R^{(1)}U \quad \Leftrightarrow \\ AX^{(2)} - fe^t &= (AX^{(1)} - fe^t)U \\ &= AX^{(1)}U - fe^t \end{aligned}$$

which by nonsingularity of A gives the stated result. •

Although lemma 1 implies that any iterative method can be formed by re-combining any other method, it is not constructive in giving the coefficients of the combination process. In the case of conjugacy-based methods we are helped out by the orthogonality properties of the method.

The specific iterative method we are constructing from the SI iterates is a function of the inner product under which we orthogonalise. For CG (or its nonsymmetric generalisation OrthoRes) we observe that the residuals are orthogonal under the M^{-1} inner product; for MinRes or GMRES we use the fact that the residuals are orthogonal under the AM^{-1} inner product.

Lemma 3 *Let R and X be the residuals and iterates from a stationary iterative process. Let $R = QU^{-1}$ be a QR decomposition under a certain inner product. Define $d^t = e^t U$ be the scaling to arrive at convex combinations. Then the sequences $\bar{R} = Q\,\mathrm{diag}(d)^{-1}$ and $\bar{X} = XU^{-1}\,\mathrm{diag}(d)^{-1}$ consist of the residuals and iterates respectively of*

Conjugate Gradients or OrthoRes *if the M^{-1} inner product is used, and*
MinRes or GMRES *if the AM^{-1} inner product is used*

for the QR factorisation.

This follows from the definition of the scaling by use of the previous lemma, and the definitions of the respective iterative methods.

Note that the matrix U is small, of size of the number of iterations performed, and its inversion negligible in cost.

3 Practical considerations

3.1 Construction of the residual sequences of SI

If we try to construct CG iterates, as observed above we need to orthogonalise the residuals under the M^{-1} inner product. In other words, we need to orthogonalise the sequences R and $M^{-1}R$. Normally these sequences are not directly computed in SI methods, as opposed to in CG-like methods. Additionally, for the construction of GMRES iterates we need to orthogonalise under the AM^{-1} inner product, so $AM^{-1}R$ is needed as well.

To obtain R, $M^{-1}R$, and $AM^{-1}R$ from SI methods, we note that

- In stationary iterative methods $x_{i+1} - x_i = M^{-1}r_i$, which gives us the $M^{-1}R$ sequence. Iterating with $i = 1 \dots k$, this gives us $M^{-1}r_1 \dots M^{-1}r_{k-1}$.
- The residuals can be easily constructed as a byproduct of such methods as Gauss-Seidel: with the exception of $a_{ii}x_i$ all other components have to be calculated anyway.
  Specifically, in the relaxation from x_i to x_{i+1} the method forms $A_U x_i$ and $A_L x_{i+1}$. Therefore, if we iterate $i = 1 \dots k$, Ax_i is fully formed for $i = 2 \dots k-1$, and we need additional computation of $A_L x_1$ to match the formation of $M^{-1}r_1 \dots M^{-1}r_{k-1}$ in the previous point.

- Computing $AM^{-1}r_i$, proceeds likewise: $AM^{-1}r_i = Ax_{i+1} - Ax_i$, and the components of Ax_i are formed already in the computation of x_{i+1}, as we observed above. Iterating with $i = 1 \ldots k$ and additional computation of $A_L x_1$ and $A_U x_k$ we can form $AM^{-1}r_1 \ldots AM^{-1}r_{k-1}$.

Thus, at the cost of some extra storage, we can retain the necessary intermediate quantities that allow construction of the CG iterates and residuals. Since we need to iterate SI for one iteration more than the number of CG iterations we want to construct, this constitutes an additional overhead of one iteration in scalar cost over the straightforward computation of the CG method.

Above we noted that computation of $A_L x_1$ and $A_U x_k$ is needed. Alternatively we can dispense with these and compute only $r_2 \ldots r_{i-2}$, but this probably increases the overhead too much: the CG iteration will now be based on a subspace of dimension lower by 3 than the SI process.

3.2 QR decomposition

In implementing the QR decomposition itself we are faced with two demands. First of all, we do not want to degrade performance, so unmodified Gram-Schmidt is preferable over modified GS. On the other hand, the QR process is essentially applied to the result of a matrix power iteration, so is likely to be badly conditioned. In our tests we used a double application of the GS orthogonalisation [1], and found this to be satisfactory. The operation count is double that for simple GS, but execution is fairly efficient and the stability was satisfactory.

3.3 Further possibilities for savings

If we aim at constructing the exact CG residuals and iterates, we have to perform the orthogonalisation of the R against the $M^{-1}R$ sequence. However, in some circumstances it may be as feasible to orthogonalise $M^{-1}R$ against itself. Whether this process makes sense depends on the application and the spectrum of the matrix. In all cases it implies a savings: the QR factorisation of one sequence against itself is performed at half the cost of that of one sequence against another. Also, not having to construct the R sequence brings about a savings in workspace, a marginal savings in work, and a further savings in the complexity of the algorithm. Other inner products, for instance induced by diagonal or small-banded matrices based on the coefficient matrix, can also be considered.

References

[1] J.W. Daniel, W.B. Gragg, L. Kaufman, and G.W. Steward. Reorthogonaliation and stable algorithms for updating the Gram-Schmidt QR factorization. *Mathematics of Computation*, 30:772–795, 1976.

[2] C.C. Douglas, J. Hu, W. Karl, M. Kowarschik, U Rüde, and C. Weiss. Fixed and adaptive cache aware algorithms for multigrid methods. In *Multigrid Methods VI, E. Dick, K. Riemslagh and J. Vierendeels (eds); Lecture Notes in Computer Sciences*, pages 87–93. Springer-Verlag, Berlin, 2000.