Krylov Subspace Iteration

This survey article reviews the history and current importance of Krylov subspace iteration algorithms.

Since the early 1800s, researchers have considered iteration methods an attractive means for approximating the solutions of large linear systems. They make these solutions possible now that we can do realistic computer simulations. The classical iteration methods typically converge very slowly (and often not at all). Around 1950, researchers realized that these methods lead to solution sequences that span a subspace—the Krylov subspace. It was then evident how to identify much better approximate solutions, without much additional computational effort.

When simulating a continuous event, such as the flow of a fluid through a pipe or of air around an aircraft, researchers usually impose a grid over the area of interest and restrict the simulation to the computation of relevant parameters. An example is the pressure or velocity of the flow or temperature inside the gridpoints. Physical laws lead to approximate relationships between these parameters in neighboring gridpoints. Together with the prescribed behavior at the boundary gridpoints and with given sources, this leads eventually to very large linear systems of equations, $Ax \approx b$. The vector x is the unknown parameter values in the gridpoints, b is the given input, and the matrix A describes the relationships between parameters in the gridpoints. Because these relationships are often restricted to nearby gridpoints, most matrix elements are zero.

The model becomes more accurate when we refine the grid—that is, when the distance between gridpoints decreases. In a 3D simulation, this easily leads to large systems of equations. Even a few hundred gridpoints in each coordinate direction leads to systems with millions of unknowns. Many other problems also lead to large systems: electric-circuit simulation, magnetic-field computation, weather prediction, chemical processes, semiconductor-device simulation, nuclear-reactor safety problems, mechanical-structure stress, and so on.

The standard numerical-solution methods for these linear systems are based on clever implementations of Gaussian elimination. These

1521-9615/00/$10.00 © 2000 IEEE

Henk A. van der Vorst
Utrecht University

methods exploit the sparse linear-system structure as much as possible to avoid computations with zero elements and zero-element storage. But for large systems these methods are often too expensive, even on today's fastest supercomputers, except where A has a special structure. For many of the problems previously listed, we can mathematically show that the standard solution methods will not lead to solutions in any reasonable amount of time. So, researchers have long tried to iteratively approximate the solution x. We start with a good guess for the solution—for instance, by solving a much easier nearby (idealized) problem. We then attempt to improve this guess by reducing the error with a convenient, cheap approximation for A—an *iterative solution method.*

Unfortunately, defining suitable nearby linear systems is difficult—in the sense that each step in the iterative process is cheap, and most important, that the iteration converges sufficiently fast. Suppose that we approximate the $n \times n$ matrix A of the linear system $Ax = b$ by the simpler matrix K. Then, we can formulate the above sketched iteration process as follows: in step i + 1, solve the new approximation x_{i+1} for the solution x of $Ax = b$, from

$$Kx_{i+1}= Kx_i + b - Ax_i .$$

For arbitrary initial start x_0, this process's convergence requirement is that the largest eigenvalue, in modulus, of the matrix $I - K^{-1}A$ is less than 1. The smaller this eigenvalue is, the faster the convergence will be (if $K = A$, we have convergence in one step). For most matrices, this is practically impossible. For instance, for the discretized Poisson equation, the choice $K = diag(A)$ leads to a convergence rate $1 - \mathcal{O}(h^2)$, where h is the distance between gridpoints. Even for the more modern incomplete LU decompositions, this convergence rate is the same, which predicts a very marginal improvement per iteration step. We get reasonable fast convergence only for strongly diagonally dominant matrices. In the mid 1950s, this led to the observation in Ewald Bodewig's textbook that iteration methods were not useful, except when A approaches a diagonal matrix.[1]

Faster iterative solvers

Despite the negative feelings about iterative solvers, researchers continued to design faster iterative methods.

The developments of modern and more successful method classes started at about the same time, interestingly, in a way not appreciated at the time. The first and truly iterative approach tried to identify a trend in the successive approximants and to extrapolate on the last iteration results. This led to the *successive overrelaxation methods*, in which an overrelaxation (or extrapolation) parameter steered the iteration process. For interesting classes of problems, such as convection-diffusion problems and the neutron-diffusion equation, this led to attractive computational methods that could compete with direct methods (maybe not so much in computing time, but certainly because of the minimal computer memory requirements). David Young[2,3] and Richard Varga[4] were important researchers who helped make these methods attractive. The SOR methods were intensively used by engineers until more successful methods gradually replaced them.

The first and truly iterative approach tried to identify a trend in the successive approximants.

The early computers had relatively small memories that made iterative methods still attractive, because you had to store only the nonzero matrix elements. Also, iterative solution, although slow, was the only way out for many PDE-related linear systems. Including iteration parameters to kill dominant factors in the iteration errors—as in SOR—made the solution of large systems possible.

Varga reports that by 1960, Laplacian-type systems of 20,000 could be solved as a daily routine on a Philco-20000 computer with 32,000 words of core storage.[4] This would have been impossible with a direct method on a similar computer. However, the iterative methods of that time required careful tuning. For example, for the Chebyshev accelerated iteration methods, you needed accurate guesses for the matrix's extremal eigenvalues. Also, for the overrelaxation methods, you needed an overrelaxation parameter that was estimated from the largest eigenvalue of some related iteration matrix.

Another iterative-method class that became popular in the mid 1950s was the Alternating Direction method, which attempted to solve discretized PDEs over grids in more dimensions by successively solving 1D problems in each coordinate direction. Iteration parameters steered this process. Varga's book, *Matrix Iterative Analysis*, gives a good overview of the state of the art in

```
Compute $r_0 = b - Ax_0$ for some initial guess $x_0$
for $i = 1, 2, \ldots$
    Solve $z_{i-1}$ from $Kz_{i-1} = r_{i-1}$
    $\rho_{i-1} = r^*_{i-1} z_{i-1}$
    if $i = 1$
        $p_1 = z_0$
    else
        $\beta_{i-1} = \rho_{i-1} / \rho_{i-2}$;
        $p_i = z_{i-1} + \beta_{i-1} p_{i-1}$
    endif
    $q_i = Ap_i$
    $\alpha_i = \rho_{i-1} / p_i^* q_i$
    $x_i = x_{i-1} + \alpha_i p_i$
    $r_i = r_{i-1} - \alpha_i q_i$
    check convergence; continue if necessary
end;
```

Figure 1. The conjugate gradient algorithm.

```
$r = b - Ax_0$, for a given initial guess $x_0$
for $j = 1, 2, \ldots$
    $\beta = \|r\|_2$, $v_1 = r/\beta$; $\hat{b} = \beta e_1$;
        for $i = 1, 2, \ldots, m$
            $w = Av_i$;
            for $k = 1, \ldots, i$
                $h_{k,i} = v_k^* w; w = w - h_{k,i} v_k$ ;
            $b_{i+1,i} = \|w\|_2$ ; $v_{i+1} = w / b_{i+1,i}$;
            for $k = 2, \ldots, i$
                $\mu = b_{k-1,i}$
                $b_{k-1,i} = c_{k-1}\mu + s_{k-1} b_{k,i}$
                $b_{k,i} = -s_{k-1}\mu + c_{k-1} b_{k,i}$
            $\delta = \sqrt{h_{i,i}^2 + h_{i+1,i}^2}$; $c_i = h_{i,i} / \delta$; $s_i = h_{i+1,i} / \delta$
            $r_{i,i} = c_i b_{i,i} + s_i b_{i+1,i}$
            $\hat{b}_{i+1} = -s_i b_i$; $\hat{b}_i = c_i \hat{b}_i$
            $\rho = |\hat{b}_{i+1}| \left( = \left\| b - Ax^{((j-1)m+i)} \right\|_2 \right)$
            if $\rho$ is small enough then
                ($n_r = i$; goto $SOL$);
    $n_r = m$, $y_{n_r} = \hat{b}_{p_r} / h_{n_r,n_r}$
SOL: for $k = n_r - 1, \ldots, 1$
        $y_k = (\hat{b}_k - \sum_{i=k+1}^{n_r} h_{k,i} y_i) / h_{k,k}$
    $x = \sum_{i=1}^{n_r} y_i v_i$; if $p$ small enough quit
    $r = b - Ax$
```

Figure 2: GMRES(m) of Saad and Schultz.

1960.[4] It even mentions a system with 108,000 degrees of freedom. Many other problems with a variation in matrix coefficients, such as electronics applications, could not be solved at that time.

Because of the nonrobustness of the early iterative solvers, research focused on more efficient direct solvers. Especially for software used by nonnumerical experts, the direct methods have the advantage of avoiding convergence problems or difficult decisions on iteration parameters. The main problem, however, is that for general PDE-related problems discretized over grids in 3D domains, optimal direct techniques scale $\mathcal{O}(n^{2.3})$ in floating-point operations, so they are of limited use for the larger, realistic 3D problems. The work per iteration for an iterative method is proportional to n, which shows that if you succeed in finding an iterative technique that converges in considerably fewer than n iterations, this technique is more efficient than a direct solver.

For many practical problems, researchers have achieved this goal, but through clever combinations of modern iteration methods with (incomplete) direct techniques: the ILU preconditioned Krylov subspace solvers. With proper ordering techniques and appropriate levels of incompleteness, researchers have realized iteration counts for convection-diffusion problems that are practically independent of the gridsize. This implies that for such problems, the required number of flops is proportional with n (admittedly with a fairly large proportionality constant). The other advantage of iterative methods is that they need modest amounts of computer storage. For many problems, modern direct methods can also be very modest, but this depends on the system's matrix structure.

The Krylov subspace solvers

Cornelius Lanczos[5] and Walter Arnoldi[6] also established the basis for very successful methods in the early 1950s. The idea was to keep all approximants computed so far in the iteration process and to recombine them to a better solution. This task might seem enormous, but Lanczos recognized that the basic iteration (for convenience we will take $K = I$) leads to approximants x_i that are in nicely structured subspaces. Namely, these subspaces are spanned by the vectors r_0, Ar_0, A^2r_0, ..., $A^{i-1}r_0$, where $r_0 = b - Ax_0$. Such a subspace is a Krylov subspace of dimension i for A and r_0.

Lanczos showed that you can generate an orthogonal basis for this subspace with a very simple three-term recurrence relation if the matrix A is symmetric. This simplified the optimal-solution computations in the Krylov subspace. The attractive aspect is that you can obtain these optimal solutions for approximately the same computational costs as the approximants for the original iterative process, which was initially not recognized as a breakthrough in iterative processes. The early observation was that, after $n - 1$ steps, this process must terminate because the Krylov subspace is of dimension n. For that reason, this Lanczos process was regarded as a direct-solution method. Researchers tested the method on tough (although low-dimensional) problems and soon observed that after $n - 1$ steps the approximant x_n could be quite far away from the solution x, with which it should coincide at that point. This made potential users suspicious.

Meanwhile, Magnus Hestenes and Eduard Stiefel[7] had proposed a very elegant method for symmetric positive definite systems, based on the same Krylov subspace principles: the conjugate gradient method. This method suffered from the same lack of exactness as Lanczos' method and did not receive much recognition in its first 20 years.

The conjugate gradient method

It took a few years for researchers to realize that it was more fruitful to consider the conjugate gradient method truly iterative. In 1972, John Reid was one of the first to point in this direction.[8] Meanwhile, analysis had already shown that a factor involving the ratio of the largest and smallest eigenvalue of A dictated this method's convergence and that the actual values of these eigenvalues play no role.

About the same time, researchers recognized that they could construct good approximations K for A with the property that the eigenvalues of $K^{-1}A$ were clustered around 1. This implied that the ratio of these eigenvalues was moderate and so led to fast convergence of conjugate gradients when applied to $K^{-1}Ax = K^{-1}b$ when K is also symmetric positive definite. This process is called *preconditioned conjugate gradients*. Figure 1 describes the algorithm, where $x * y$ denotes the innerproduct of two vectors x and y (complex conjugate if the system is complex).

David Kershaw was one of the first to experiment with the conjugate gradient method, with incomplete Cholesky factorization of A as a preconditioner for tough problems related to fusion problems.[9] Table 1 quotes iteration numbers for the basic Gauss-Seidel iteration (that is, the basic iteration for K the lower triangular part of A[4]) the accelerated version SOR (actually, a slightly faster variant, Block SOR[4]), and conjugate gradients preconditioned with incomplete Cholesky (also known as ICCG). The iteration numbers were necessary to reduce the initial-residual norm by a factor of 10^{-6}.

Table 1. Kershaw's results for a fusion problem.

Method	Number of iterations
Gauss Seidel	208,000
Block successive overrelaxation methods	765
Incomplete Cholesky conjugate gradients	25

Table 1 shows the sometimes gigantic improvements from the (preconditioned) conjugate gradients. These and other results also motivated the search for other powerful Krylov subspace methods for a more general equation system.

GMRES

Researchers have proposed quite a few specialized Krylov methods, including Bi-CG and QMR for unsymmetric A; MINRES and SYMMLQ for symmetric-indefinite systems; and Orthomin, Orthodir, and Orthores for general unsymmetric systems. The current de facto unsymmetric-system standard is the GMRES method, proposed in 1986 by Youcef Saad and Martin Schultz.[10] In this method, the x_i in the dimension i Krylov subspace is constructed for which the norm $\|b - Ax_i\|_2$ is minimal. This builds on an algorithm, proposed by Arnoldi,[6] that constructs an orthonormal basis for the Krylov subspace for unsymmetric A.

The price for this ideal situation is that you have to store a full orthogonal basis for the Krylov subspace, which means the more iterations, the more basis vectors you must store. Also, the work per iteration increases linearly, which makes the method attractive only if it converges really fast. For many practical problems, GMRES takes a few tens of iterations; for many other problems it can take hundreds, which makes a full GMRES unfeasable.

Figure 2 shows a GMRES version in which a restart occurs after every m iterations to limit the memory requirements and the work per iteration. The application for a preconditioned system $K^{-1}Ax = K^{-1}b$ is straightforward.

```
Compute $r_0 = b - Ax_0$ for some initial guess $x_0$
Choose $\hat{r}_0$, for example $\hat{r}_0 = r_0$
for $i = 1, 2, ...$
    $\rho_{i-1} = \hat{r}_0^* r_{i-1}$
    if $\rho_{i-1} = 0$ method fails
    if $i = 1$
        $p_i = r_{i-1}$
    else
        $\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})(\alpha_{i-1}/\omega_{i-1})$
        $p_i = r_{i-1} + \beta_{i-1}(p_{i-1} - \omega_{i-1}v_{i-1})$
    endif
    Solve $\hat{p}$ from $K\hat{p} = p_i$
    $v_i = A\hat{p}$
    $\alpha_i = \rho_{i-1} / \hat{r}_0^* v_i$
    $s = r_{i-1} - \alpha_i v_i$
    if $\|s\|$ small enough then
        $x^{(i)} = x^{(i-1)} + \alpha_i \hat{p}$ and stop
    Solve $z$ from $Kz = s$
    $t = A z$
    $\omega_i = s^* t / t^* t$
    $x_i = x_{i-1} + \alpha_i \hat{p} + \omega_i z$
    if $x_i$ is accurate enough then quit
    $r_i = s - \omega_i t$
    for continuation it is necessary that $\omega_i \neq 0$
end
```

Figure 3. The Bi-CGSTAB algorithm.

Bi-CGSTAB

The GMRES cost per iteration has also led to a search for cheaper near-optimal methods. Vance Faber and Thomas Manteuffel's famous result showed that constructing optimal solutions in the Krylov subspace for unsymmetric A by short recurrences, as in the conjugate gradients method, is generally not possible. The generalization of conjugate gradients for unsymmetric systems, Bi-CG, often displays an irregular convergence behavior, including a possible breakdown. Roland Freund and Noel Nachtigal gave an elegant remedy for both phenomena in their QMR method. BiCG and QMR have the disadvantage that they require an operation with A^T per iteration step. This additional operation does not lead to a further residual reduction.

In the mid 1980s, Peter Sonneveld recognized that you can use the A^T operation for a further residual reduction through a minor modification to the Bi-CG scheme, almost without additional computational costs. This CGS method was often faster but significantly more irregular, which led to a precison loss. In 1992, I showed that Bi-CG could be made faster and smoother, at almost no additional cost, with minimal residual steps.[11] Figure 3 schematically shows the resulting Bi-CGSTAB algorithm, for the solution of $Ax = b$ with preconditioner K.

It is difficult to make a general statement about how quickly these Krylov methods converge. Although they certainly converge much faster than the classical iteration schemes and convergence takes place for a much wider class of matrices, many practical systems still cannot be satisfactorily solved. Much depends on whether you are able to define a nearby matrix K that will serve as a preconditioner. Recent research is more oriented in that direction than in trying to improve the Krylov subspace methods, although we might see some improvements for these methods as well. Effective and efficient precónditioner construction is largely problem-dependent; a preconditioner is considered as effective if the number of iteration steps of the preconditioned Krylov subspace method is approximately 100 or less.

In this contribution, I have highlighted some of the Krylov subspace methods that researchers have accepted as powerful tools for the iterative solution of very large linear systems with millions of unknowns. These methods are a breakthrough in iterative solution methods for linear systems. I have mentioned a few names that were most directly associated with the development of the most characteristic and powerful methods—CG, GMRES, and Bi-CGSTAB—but these only represent the tip of the iceberg in this lively research area. For more information, see the "Further reading" sidebar.

Another class of acceleration methods that has been developed since around 1980 are the multigrid or multilevel methods. These methods apply to grid-oriented problems, and the idea is to work with coarse and fine grids. Smooth solution components are largely determined on the coarse grid; the fine grid is for the more locally varying components. When these methods work for regular problems over regular grids for PDEs, they can be very fast and are much more efficient than preconditioned Krylov solvers. However, there is no clear separation between the two camps: you can use multigrid as a preconditioner for Krylov methods for less regular problems and the Krylov techniques as smoothers for multigrid. This is a fruitful direction for further exploration.

References

1. E. Bodewig, *Matrix Calculus*, North-Holland, Amsterdam, 1956.
2. D. Young, *Iterative Solution of Large Linear Systems*, Academic Press, San Diego, 1971.
3. D.M. Young, *Iterative Methods for Solving Partial Differential Equations of Elliptic Type*, doctoral thesis, Harvard Univ., Cambridge, Mass., 1950.
4. R.S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Upper Saddle River, N.J., 1962.
5. C. Lanczos, "Solution of Systems of Linear Equations by Minimized Iterations," *J. Research Nat'l Bureau of Standards*, Vol. 49, 1952, pp. 33–53.
6. W.E. Arnoldi, "The Principle of Minimized Iteration in the Solution of the Matrix Eigenproblem," *Quarterly Applied Mathematics*, Vol. 9, 1951, pp. 17–29.
7. M.R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *J. Research Nat'l Bureau of Standards*, Vol. 49, 1954, pp. 409–436.
8. J.K. Reid, "The Use of Conjugate Gradients for Systems of Equations Possessing 'Property A,'" *SIAM J. Numerical Analysis*, Vol. 9, 1972, pp. 325–332.
9. D.S. Kershaw, "The Incomplete Choleski-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations." *J. Computational Physics*, Vol. 26, No. 1, 1978, pp. 43–65.
10. Y. Saad and M.H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nnonsymmetric Linear Systems," *SIAM J. Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.
11. H.A. van der Vorst, "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems," *SIAM J. Scientific and Statistical Computing*, Vol. 13, No. 2, 1992, pp. 631–644.

Further reading

O. Axelsson, *Iterative Solution Methods*, Cambridge Univ. Press, Cambridge, UK, 1994.

R. Barrett et al., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, Soc. for Industrial and Applied Mathematics, Philadelphia, 1994.

G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, 1996.

A. Greenbaum, *Iterative Methods for Solving Linear System*, Soc. for Industrial and Applied Mathematics, Philadelphia, 1997.

G. Meurant, *Computer Solution of Large Linear Systems*, North-Holland, Amsterdam, 1999.

Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.

Y. Saad and H.A. van der Vorst, *Iterative Solution of Linear Systems in the 20th Century*, Tech. Report UMSI 99/152, Supercomputing Inst., Univ. of Minnesota, Minneapolis, Minn., 1999.

P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley & Sons, Chichester, 1992.

Henk A. van der Vorst is a professor in numerical analysis at the Mathematical Department of Utrecht University. His research interests include iterative solvers for linear systems, large sparse eigenproblems, and the design of algorithms for parallel and vector computers. He is associate editor of the *SIAM Journal of Scientific Computing, Journal Computational and Applied Mathematics, Applied Numerical Mathematics, Parallel Computing, Numerical Linear Algebra with Applications,* and *Computer Methods in Applied Mechanics and Engineering.* He is a member of the Board of the Center for Mathematics and Informatics, Amsterdam. He has a PhD in applied mathematics from the University of Utrecht. Contact him at the Dept. of Mathematics, Univ. of Utrecht, PO Box 80.010, NL-3508 TA Utrecht, The Netherlands; vorst@math.uu.nl.