A Brief Introduction to Krylov Space Methods for Solving Linear Systems

Martin H. Gutknecht[1]

ETH Zurich, Seminar for Applied Mathematics mhg@math.ethz.ch

With respect to the "influence on the development and practice of science and engineering in the 20th century", Krylov space methods are considered as one of the ten most important classes of numerical methods [1]. *Large sparse linear systems of equations* or *large sparse matrix eigenvalue problems* appear in most applications of scientific computing. Sparsity means that most elements of the matrix involved are zero. In particular, discretization of PDEs with the *finite element method (FEM)* or with the *finite difference method (FDM)* leads to such problems. In case the original problem is nonlinear, linearization by Newton's method or a Newton-type method leads again to a linear problem. We will treat here systems of equations only, but many of the numerical methods for large eigenvalue problems are based on similar ideas as the related solvers for equations.

Sparse linear systems of equations can be solved by either so-called *sparse direct solvers*, which are clever variations of Gauss elimination, or by iterative methods. In the last thirty years, sparse direct solvers have been tuned to perfection: on the one hand by finding strategies for permuting equations and unknowns to guarantee a stable LU decomposition and small fill-in in the triangular factors, and on the other hand by organizing the computation so that optimal use is made of the hardware, which nowadays often consists of parallel computers whose architecture favors block operations with data that are locally stored or cached.

The *iterative methods* that are today applied for solving large-scale linear systems are mostly *preconditioned Krylov (sub)space solvers*. Classical methods that do not belong to this class, like the successive overrelaxation (SOR) method, are no longer competitive. However, some of the classical matrix splittings, e.g. the one of SSOR (the symmetric version of SOR), are still used for preconditioning. Multigrid is in theory a very effective iterative method, but normally it is now applied as an inner iteration with a Krylov space solver as outer iteration; then, it can also be considered as a preconditioner.

In the past, Krylov space solvers were referred to also by other names such as *semi-iterative methods* and *polynomial acceleration methods*. Some of them can also be used as (fixed) preconditioners, in which case they are known as *polynomial preconditioners*; but in this function they are reportedly not competitive. *Flexible preconditioning* allows us to apply any Krylov space solver as a preconditioner of another Krylov space solver. Such combinations, called *inner-outer iteration* methods, may be very effective; see, e.g., [14].

Krylov space methods for solving $\mathbf{Ax} = \mathbf{b}$ have the special feature that the $N \times N$ matrix $\mathbf{A}$ needs only be given as an operator: for any N-vector $\mathbf{y}$ one must be able to compute $\mathbf{Ay}$; so $\mathbf{A}$ may be given as a function (or procedure or subroutine). We refer to the operation $\mathbf{Ay}$ as *matrix-vector product (MV).* In practice, it may be much more complicated than the multiplication of a vector by a sparse matrix; e.g., this operation may include the application of a preconditioner, which may also require the solution of a large linear system.

1 From Jacobi iteration to Krylov space methods

The simplest iterative method is *Jacobi iteration.* It is the same as diagonally preconditioned *fixed point iteration*: if the diagonal matrix $\mathbf{D}$ with the diagonal of $\mathbf{A}$ is nonsingular, we can transform $\mathbf{Ax} = \mathbf{b}$ into

$$\mathbf{x} = \widehat{\mathbf{B}}\mathbf{x} + \widehat{\mathbf{b}} \quad \text{with} \quad \widehat{\mathbf{B}} :\equiv \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}\,, \quad \widehat{\mathbf{b}} :\equiv \mathbf{D}^{-1}\mathbf{b} \tag{1}$$

and apply the fixed point iteration $\mathbf{x}_{n+1} :\equiv \widehat{\mathbf{B}}\mathbf{x}_n + \widehat{\mathbf{b}}$.

It is easy to show by considering the powers of the Jordan canonical form of $\widehat{\mathbf{B}}$ that the following convergence result is valid for Jacobi iteration:

$$\mathbf{x}_n \to \mathbf{x}_\star \text{ for every } \mathbf{x}_0 \quad \Longleftrightarrow \quad \rho(\widehat{\mathbf{B}}) < 1\,, \tag{2}$$

where $\mathbf{x}_\star :\equiv \mathbf{A}^{-1}\mathbf{b}$ and $\rho(\widehat{\mathbf{B}}) :\equiv \max\{|\lambda| \bigm| \lambda \text{ eigenvalue of } \widehat{\mathbf{B}}\}$ is the *spectral radius* of $\widehat{\mathbf{B}}$. From the suggested proof of (2) one actually sees that the asymptotic root convergence factor is bounded by $\rho(\widehat{\mathbf{B}})$ (if less than 1), and that this bound is sharp:

$$\limsup_{n\to\infty} \|\mathbf{x}_n - \mathbf{x}_\star\|^{1/n} \leq \rho(\widehat{\mathbf{B}})\,, \tag{3}$$

and equality holds for some $\mathbf{x}_0$.

Unfortunately, even the simplest examples of boundary value problems, like $u'' = f$ on $(0, 1)$ with $u(0) = u(1) = 0$, discretized by a standard finite difference method, show that Jacobi iteration may converge extremely slowly.

Unless we know the solution $\mathbf{x}_\star$ we cannot compute the nth *error (vector)*

$$\mathbf{d}_n :\equiv \mathbf{x}_n - \mathbf{x}_\star\,. \tag{4}$$

Thus, for checking the convergence we normally use the nth *residual (vector)*

$$\mathbf{r}_n :\equiv \mathbf{b} - \mathbf{A}\mathbf{x}_n\,. \tag{5}$$

Note that $\mathbf{r}_n = -\mathbf{A}(\mathbf{x}_n - \mathbf{x}_\star) = -\mathbf{A}\mathbf{d}_n\,.$

Assuming $\mathbf{D} = \mathbf{I}$ and letting $\mathbf{B} := \mathbf{I} - \mathbf{A}$ we have

$$\mathbf{r}_n = \mathbf{b} - \mathbf{A}\mathbf{x}_n = \mathbf{B}\mathbf{x}_n + \mathbf{b} - \mathbf{x}_n = \mathbf{x}_{n+1} - \mathbf{x}_n \,,$$

so we can rewrite the Jacobi iteration as

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \mathbf{r}_n \,. \tag{6}$$

Multiplying this by $-\mathbf{A}$, we obtain a recursion for the residual:

$$\mathbf{r}_{n+1} := \mathbf{r}_n - \mathbf{A}\mathbf{r}_n = \mathbf{B}\mathbf{r}_n \,. \tag{7}$$

So we can compute $\mathbf{r}_n$ either according to definition (5) or by the recursion (7); either way it requires one MV. From (7) it follows by induction that

$$\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \;\in\; \mathsf{span}\,\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^n\mathbf{r}_0\} \,, \tag{8}$$

where $p_n(\zeta) = (1-\zeta)^n$ is a polynomial of exact degree n. From (6) we have

$$\mathbf{x}_n = \mathbf{x}_0 + \mathbf{r}_0 + \cdots + \mathbf{r}_{n-1} = \mathbf{x}_0 + q_{n-1}(\mathbf{A})\mathbf{r}_0 \tag{9}$$

with a polynomial q_{n-1} of exact degree $n-1$. So, $\mathbf{x}_n$ lies in the affine space $\mathbf{x}_0 + \mathsf{span}\,\left\{\mathbf{r}_0, \dots, \mathbf{A}^{n-1}\mathbf{r}_0\right\}$ obtained by shifting the subspace of $\mathbf{r}_{n-1}$.

Computing $q_{n-1}(\mathbf{A})\mathbf{r}_0$ and $p_n(\mathbf{A})\mathbf{r}_0$ requires $n+1$ MVs, because we need to build up the subspace $\mathsf{span}\,\left\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{n-1}\mathbf{r}_0\right\}$. This is the main work. We may ask: *Is there a better choice for* $\mathbf{x}_n$ *in the same affine space?*

The subspace that appears in (8) and (9) is what we call a Krylov space:

Definition 1. *Given a nonsingular* $\mathbf{A} \in \mathbb{C}^{N\times N}$ *and* $\mathbf{y} \neq \mathbf{o} \in \mathbb{C}^N$, *the nth* ***Krylov (sub)space*** $\mathcal{K}_n(\mathbf{A}, \mathbf{y})$ *generated by* $\mathbf{A}$ *from* $\mathbf{y}$ *is*

$$\mathcal{K}_n := \mathcal{K}_n(\mathbf{A}, \mathbf{y}) := \mathsf{span}\,(\mathbf{y}, \mathbf{A}\mathbf{y}, \dots, \mathbf{A}^{n-1}\mathbf{y}). \tag{10}$$

Clearly, $\mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \mathcal{K}_3 \subseteq \dots$, and the dimension increases at most by one in each step. *But when does the equal sign hold?* Moreover, it seems clever to choose the nth approximate solution $\mathbf{x}_n$ in $\mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$. *But can we expect to find the exact solution* $\mathbf{x}_\star$ *in one of those affine space?* These questions are answered in the following lemmas and corollaries that we cite without proof.

Lemma 1. *There is a positive integer* $\nu := \nu(\mathbf{y}, \mathbf{A})$, *called grade of* $\mathbf{y}$ *with respect to* $\mathbf{A}$, *such that*

$$\mathsf{dim}\;\mathcal{K}_n(\mathbf{A}, \mathbf{y}) = \begin{cases} n & \text{if} \quad n \leq \nu \,, \\ \nu & \text{if} \quad n \geq \nu \,. \end{cases}$$

Corollary 1. $\mathcal{K}_\nu(\mathbf{A}, \mathbf{y})$ *is the smallest* $\mathbf{A}$*–invariant subspace that contains* $\mathbf{y}$.

Lemma 2. *The nonnegative integer* ν *of Lemma 1 satisfies*

$$\nu = \min\left\{n \;\middle|\; \mathbf{A}^{-1}\mathbf{y} \in \mathcal{K}_n(\mathbf{A}, \mathbf{y})\right\} \,.$$

Corollary 2. *Let $\mathbf{x}_\star$ be the solution of $\mathbf{Ax} = \mathbf{b}$ and let $\mathbf{x}_0$ be any initial approximation of it and $\mathbf{r}_0 := \mathbf{b} - \mathbf{Ax}_0$ the corresponding residual. Moreover, let $\nu := \nu(\mathbf{r}_0, \mathbf{A})$. Then*

$$\mathbf{x}_\star \in \mathbf{x}_0 + \mathcal{K}_\nu(\mathbf{A}, \mathbf{r}_0)\,.$$

The idea behind Krylov space solvers is to generate a sequence of approximate solutions $\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$ of $\mathbf{Ax} = \mathbf{b}$, so that the corresponding residuals $\mathbf{r}_n \in \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0)$ "converge" to the zero vector $\mathbf{o}$. Here, "converge" may also mean that after a finite number of steps, $\mathbf{r}_n = \mathbf{o}$, so that $\mathbf{x}_n = \mathbf{x}_\star$ and the process stops. This is in particular true (in exact arithmetic) if a method ensures that the residuals are linearly independent: then $\mathbf{r}_\nu = \mathbf{o}$. In this case we say that the method has the finite termination property.

It is not so easy to give a formal definition of Krylov space solvers that covers all relevant cases and is not too general in the sense that it no longer grasps some relevant aspects of Krylov space solvers. Here is our proposal:

Definition 2. *A* ***(standard) Krylov space method for solving a linear system*** $\mathbf{Ax} = \mathbf{b}$ *or, briefly, a* ***Krylov space solver*** *is an iterative method starting from some initial approximation $\mathbf{x}_0$ and the corresponding residual $\mathbf{r}_0 := \mathbf{b} - \mathbf{Ax}_0$ and generating for all, or at least most n, until it possibly finds the exact solution, iterates $\mathbf{x}_n$ such that*

$$\mathbf{x}_n - \mathbf{x}_0 = q_{n-1}(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \tag{11}$$

with a polynomial q_{n-1} of exact degree $n-1$. For some n, $\mathbf{x}_n$ may not exist or q_{n-1} may have lower degree.

A similar statement can be made for the residuals, if they exist:

Lemma 3. *The residuals $\mathbf{r}_n$ of a Krylov space solver satisfy*

$$\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \subseteq \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0)\,, \tag{12}$$

where p_n is a polynomial of degree n, called nth residual polynomial, which is related to the polynomial q_{n-1} of (11) by

$$p_n(\zeta) = 1 - \zeta q_{n-1}(\zeta)\,. \tag{13}$$

In particular, it satisfies the consistency condition $p_n(0) = 1$.

The vague expression "for all, or at least most n" in Definition 2 is needed because in some widely used Krylov space solvers (e.g., BiCG) there may exist exceptional situations, where for some n the iterate $\mathbf{x}_n$ and the residual $\mathbf{r}_n$ are not defined. In other Krylov space solvers (e.g., CR), there may be indices where $\mathbf{x}_n$ exists, but the polynomial q_{n-1} is of lower degree than $n-1$.

There are also *nonstandard Krylov space methods* where the search space for $\mathbf{x}_n - \mathbf{x}_0$ is still a Krylov space but one that differs from $\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$.

When applied to large real-world problems Krylov space solvers often converge very slowly — if at all. In practice, Krylov space solvers are therefore nearly always applied with *preconditioning*: $\mathbf{A}\mathbf{x} = \mathbf{b}$ is replaced by

$$\underbrace{\mathbf{C}\mathbf{A}}_{\widehat{\mathbf{A}}}\,\mathbf{x} = \underbrace{\mathbf{C}\mathbf{b}}_{\widehat{\mathbf{b}}} \quad \text{or} \quad \underbrace{\mathbf{A}\mathbf{C}}_{\widehat{\mathbf{A}}}\,\underbrace{\mathbf{C}^{-1}\mathbf{x}}_{\widehat{\mathbf{x}}} = \mathbf{b} \quad \text{or} \quad \underbrace{\mathbf{C}_L\mathbf{A}\mathbf{C}_R}_{\widehat{\mathbf{A}}}\,\underbrace{\mathbf{C}_R^{-1}\mathbf{x}}_{\widehat{\mathbf{x}}} = \underbrace{\mathbf{C}_L\mathbf{b}}_{\widehat{\mathbf{b}}}\,.$$

The first two cases are referred to as *left and right preconditioning*, respectively, while in the last case we apply a *split preconditioner* $\mathbf{C}_L\mathbf{C}_R$. Actually, here $\mathbf{C}$ and $\mathbf{C}_L\mathbf{C}_R$ are *approximate inverses* of $\mathbf{A}$. Often we use instead of $\mathbf{C}$ a preconditioner $\mathbf{M} \approx \mathbf{A}$ with the property that the system $\mathbf{M}\mathbf{y} = \mathbf{z}$ is easily solved for any $\mathbf{z}$. Then, in the above formulas we have to replace $\mathbf{C}$ by $\mathbf{M}^{-1}$ and $\mathbf{C}_L$, $\mathbf{C}_R$ by $\mathbf{M}_L^{-1}$, $\mathbf{M}_R^{-1}$. Applying a preconditioned Krylov space solver just means to apply the method to $\widehat{\mathbf{A}}\widehat{\mathbf{x}} = \widehat{\mathbf{b}}$.

2 The conjugate gradient method

The *conjugate gradient* (CG) *method* is due to Hestenes and Stiefel [7]. It should only be applied to systems that are symmetric positive definite (spd) or Hermitian positive definite (Hpd), and it is still the method of choice for this case. We will assume real data here.

CG is the archetype of a Krylov space solver that is an *orthogonal projection method* and satisfies a minimality condition: the error is minimal in the so-called *energy norm* or $\mathbf{A}$-*norm* of the error vector $\mathbf{d} := \mathbf{x} - \mathbf{x}_\star$,

$$\|\mathbf{d}\|_{\mathbf{A}} = \sqrt{\mathbf{d}^{\mathsf{T}}\mathbf{A}\mathbf{d}}\,.$$

In nature, stable states are characterized by minimum energy. Discretization leads to the minimization of a quadratic function:

$$\Psi(\mathbf{x}) := \tfrac{1}{2}\,\mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} - \mathbf{b}^{\mathsf{T}}\mathbf{x} + \gamma \tag{14}$$

with an spd matrix $\mathbf{A}$. Ψ is convex and has a unique minimum. Its gradient is $\nabla\Psi(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} = -\mathbf{r}$, where $\mathbf{r}$ is the residual corresponding to $\mathbf{x}$. Hence,

$$\mathbf{x} \text{ minimizer of } \Psi \quad \Longleftrightarrow \quad \nabla\Psi(\mathbf{x}) = \mathbf{o} \quad \Longleftrightarrow \quad \mathbf{A}\mathbf{x} = \mathbf{b}\,. \tag{15}$$

If $\mathbf{x}_\star$ denotes the minimizer and $\mathbf{d} := \mathbf{x} - \mathbf{x}_\star$ the error vector, and if we choose $\gamma := \frac{1}{2}\,\mathbf{b}^{\mathsf{T}}\mathbf{A}^{-1}\mathbf{b}$, it is easily seen that

$$\|\mathbf{d}\|_{\mathbf{A}}^2 = \|\mathbf{x} - \mathbf{x}_\star\|_{\mathbf{A}}^2 = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{\mathbf{A}^{-1}}^2 = \|\mathbf{r}\|_{\mathbf{A}^{-1}}^2 = 2\,\Psi(\mathbf{x})\,. \tag{16}$$

In summary: *If* $\mathbf{A}$ *is spd, to minimize the quadratic function* Ψ *means to minimize the energy norm of the error vector of the linear system* $\mathbf{A}\mathbf{x} = \mathbf{b}$. *The minimizer* $\mathbf{x}_\star$ *is the solution of* $\mathbf{A}\mathbf{x} = \mathbf{b}$.

The above discussion suggest to find the minimizer of Ψ by descending on the surface representing Ψ by following the direction of steepest descent. If we took infinitely many infinitesimal steps, we would find the minimum by following a curved line. However, each determination of the gradient requires an MV, and therefore we need to take long steps and follow a piecewise straight line. In each step we go to the lowest point in the chosen descent direction.

Since $\mathbf{A}$ is spd, the level curves $\Psi(\mathbf{x}) = \mathsf{const}$ are just concentric ellipses if $N = 2$ and concentric ellipsoids if $N = 3$. As can be seen from a sketch, *even for a* 2×2 *system many steps may be needed to get close to the solution.*

We can do much better: by choosing the second direction $\mathbf{v}_1$ *conjugate* or $\mathbf{A}$-*orthogonal* to $\mathbf{v}_0$, the first one, i.e., so that $\mathbf{v}_1^{\mathsf{T}}\mathbf{A}\mathbf{v}_0 = 0$, we find the minimum in at most two steps, because for ellipses, any radius vector and the corresponding tangent direction are conjugate to each other.

How does this generalize to N dimensions, and what can be said about the intermediate results? We choose *search directions* or *direction vectors* $\mathbf{v}_n$ that are *conjugate* ($\mathbf{A}$–orthogonal) to each other:

$$\mathbf{v}_n^{\mathsf{T}}\mathbf{A}\mathbf{v}_k = 0\,, \qquad k = 0, \dots, n-1, \tag{17}$$

and define

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \mathbf{v}_n\omega_n\,, \tag{18}$$

so that $\mathbf{r}_{n+1} = \mathbf{r}_n - \mathbf{A}\mathbf{v}_n\omega_n$. Here, ω_n is again chosen such that the $\mathbf{A}$-norm of the error is minimized on the line $\omega \mapsto \mathbf{x}_n + \mathbf{v}_n\omega$. This means that

$$\omega_n := \frac{\langle \mathbf{r}_n, \mathbf{v}_n \rangle}{\langle \mathbf{v}_n, \mathbf{A}\mathbf{v}_n \rangle}\,. \tag{19}$$

Definition 3. *Any iterative method satisfying* (17)*,* (18)*, and* (19) *is called a* ***conjugate direction*** (CD) ***method****.*

By definition, such a method chooses the step length ω_n so that $\mathbf{x}_{n+1}$ is locally optimal on the search line. But does it also yield the best

$$\mathbf{x}_{n+1} \in \mathbf{x}_0 + \mathsf{span}\,\{\mathbf{v}_0, \dots, \mathbf{v}_n\} \tag{20}$$

with respect to the $\mathbf{A}$-norm of the error? By verifying that

$$\Psi(\mathbf{x}_{n+1}) = \Psi(\mathbf{x}_n) - \omega_n\mathbf{v}_n^{\mathsf{T}}\mathbf{r}_0 + \tfrac{1}{2}\,\omega_n^2\mathbf{v}_n^{\mathsf{T}}\mathbf{A}\mathbf{v}_n\,.$$

one can prove that this is indeed the case [3]:

Theorem 1. *For a conjugate direction method the problem of minimizing the energy norm of the error of an approximate solution of the form* (20) *decouples into* $n+1$ *one-dimensional minimization problems on the lines* $\omega \mapsto \mathbf{x}_k + \mathbf{v}_k\omega$*,* $k = 0, 1, \dots, n$*. Therefore, a conjugate direction method yields after* $n+1$ *steps the approximate solution of the form* (20) *that minimizes the energy norm of the error in this affine space.*

In general, conjugate direction methods are not Krylov space solvers, but with suitably chosen search directions they are. Since

$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \mathbf{v}_0\omega_0 + \cdots + \mathbf{v}_n\omega_n \in \mathbf{x}_0 + \mathsf{span}\ \{\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n\}\ ,$$

we need that

$$\mathsf{span}\ \{\mathbf{v}_0, \ldots, \mathbf{v}_n\} = \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0)\,, \qquad n = 0, 1, 2, \ldots\ . \tag{21}$$

Definition 4. *The* ***conjugate gradient*** (CG) ***method*** *is the conjugate direction method with the choice* (21)*.*

Theorem 1 yields now the main result on CG:

Theorem 2. *The* CG *method yields approximate solutions* $\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$ *that are optimal in the sense that they minimize the energy norm* (**A**-*norm*) *of the error* (*i.e., the* $\mathbf{A}^{-1}$-*norm of the residual*) *for* $\mathbf{x}_n$ *from this affine space.*

Associated with this minimality is the *Galerkin condition*

$$\mathcal{K}_n \perp \mathbf{r}_n \in \mathcal{K}_{n+1}\,, \tag{22}$$

which implies that the residuals $\{\mathbf{r}_n\}_{n=0}^{\bar{\nu}-1}$ are orthogonal to each other and form an orthogonal basis of $\mathcal{K}_{\bar{\nu}}$. On the other hand, by assumption, the search directions $\{\mathbf{v}_n\}_{n=0}^{\bar{\nu}-1}$ form a conjugate basis of $\mathcal{K}_{\bar{\nu}}$.

Here is a formulation of the standard version of CG due to Hestenes and Stiefel, which is sometimes called OrthoMin or OMin version now.

Algorithm 1 (OMin form of the CG method) *For solving* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *let* $\mathbf{x}_0$ *be an initial approximation, and let* $\mathbf{v}_0 := \mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$ *and* $\delta_0 := \|\mathbf{r}_0\|^2$*. Then, for* $n = 0, 1, 2, \ldots,$ *compute*

$$\delta'_n := \|\mathbf{v}_n\|^2_{\mathbf{A}}\,, \tag{23a}$$

$$\omega_n := \delta_n / \delta'_n\,, \tag{23b}$$

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \mathbf{v}_n\omega_n\ , \tag{23c}$$

$$\mathbf{r}_{n+1} := \mathbf{r}_n - \mathbf{A}\mathbf{v}_n\omega_n\ , \tag{23d}$$

$$\delta_{n+1} := \|\mathbf{r}_{n+1}\|^2, \tag{23e}$$

$$\psi_n := -\delta_{n+1}/\delta_n\ , \tag{23f}$$

$$\mathbf{v}_{n+1} := \mathbf{r}_{n+1} - \mathbf{v}_n\psi_n\ . \tag{23g}$$

If $\|\mathbf{r}_{n+1}\| \leq \mathrm{tol}$*, the algorithm terminates and* $\mathbf{x}_{n+1}$ *is a sufficiently accurate approximation of the solution.*

The *conjugate residual* (CR) *method* is defined in analogy to the CG method, but the 2-norm is replaced by the **A**-norm. So what is minimized is now the $\mathbf{A}^2$-norm of the error, which is the 2-norm of the residual. Here, the residuals $\{\mathbf{r}_n\}_{n=0}^{\bar{\nu}-1}$ form an **A**-orthogonal basis of $\mathcal{K}_{\bar{\nu}}$ and the search directions $\{\mathbf{v}_n\}_{n=0}^{\bar{\nu}-1}$ form a $\mathbf{A}^2$-orthogonal basis of $\mathcal{K}_{\bar{\nu}}$. The challenge is to find recurrences so that still only one MV is needed per iteration.

3 Methods for nonsymmetric systems

Solving nonsymmetric (or non-Hermitian) linear systems iteratively with Krylov space solvers is considerably more difficult and costly than symmetric (or Hermitian) systems. There are two different ways to generalize CG:

- Maintain the orthogonality of the projection and the related minimality of the error by constructing either orthogonal residuals $\mathbf{r}_n$ ($\rightsquigarrow$ *generalized* CG (GCG)) or $\mathbf{A}^\mathsf{T}\mathbf{A}$-orthogonal search directions $\mathbf{v}_n$ ($\rightsquigarrow$ *generalized* CR (GCR)). Then, *the recursions involve all previously constructed residuals or search directions and all previously constructed iterates.*
- Maintain short recurrence formulas for residuals, direction vectors and iterates ($\rightsquigarrow$ *biconjugate gradient* (BiCG) *method, Lanczos-type product methods* (LTPM)). The resulting methods are at best *oblique projection methods. There is no minimality property of error or residuals vectors.*

3.1 The biconjugate gradient (BiCG) method

While CG (for spd $\mathbf{A}$) has mutually *orthogonal residuals* $\mathbf{r}_n$ with

$$\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \quad \in \mathsf{span}\ \{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^n\mathbf{r}_0\} =: \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0)\,,$$

BiCG constructs in the same spaces residuals that are orthogonal to a dual Krylov space spanned by *"shadow residuals"*

$$\widetilde{\mathbf{r}}_n = \widetilde{p}_n(\mathbf{A}^\mathsf{T})\widetilde{\mathbf{r}}_0 \in \mathsf{span}\ \left\{\widetilde{\mathbf{r}}_0, \mathbf{A}^\mathsf{T}\widetilde{\mathbf{r}}_0, \dots, (\mathbf{A}^\mathsf{T})^n\,\widetilde{\mathbf{r}}_0\right\} =: \mathcal{K}_{n+1}(\mathbf{A}^\mathsf{T}, \widetilde{\mathbf{r}}_0) =: \widetilde{\mathcal{K}}_{n+1}\,.$$

The initial shadow residual $\widetilde{\mathbf{r}}_0$ can be chosen freely. So, BiCG requires two MVs to extend $\mathcal{K}_n$ and $\widetilde{\mathcal{K}}_n$: one multiplication by $\mathbf{A}$ and one by $\mathbf{A}^\mathsf{T}$. But there are still short recurrences for $\mathbf{x}_n$, $\mathbf{r}_n$, and $\widetilde{\mathbf{r}}_n$. Now there are two Galerkin conditions

$$\widetilde{\mathcal{K}}_n \perp \mathbf{r}_n \in \mathcal{K}_{n+1}\,, \qquad \mathcal{K}_n \perp \widetilde{\mathbf{r}}_n \in \widetilde{\mathcal{K}}_{n+1}\,,$$

but only the first one is relevant for determining $\mathbf{x}_n$.

The residuals $\{\mathbf{r}_n\}_{n=0}^m$ and the shadow residuals $\{\widetilde{\mathbf{r}}_n\}_{n=0}^m$ form *biorthogonal bases* or *dual bases* of $\mathcal{K}_{m+1}$ and $\widetilde{\mathcal{K}}_{m+1}$:

$$\langle \widetilde{\mathbf{r}}_m, \mathbf{r}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta_n \neq 0 & \text{if } m = n. \end{cases}$$

The search directions $\{\mathbf{v}_n\}_{n=0}^m$ and the *"shadow search directions"* $\{\widetilde{\mathbf{v}}_n\}_{n=0}^m$ form *biconjugate bases* of $\mathcal{K}_{m+1}$ and $\mathcal{K}_{m+1}$:

$$\langle \widetilde{\mathbf{v}}_m, \mathbf{A}\mathbf{v}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta_n' \neq 0 & \text{if } m = n. \end{cases}$$

BiCG goes back to Lanczos [8], but was brought to its current, CG-like form later. For a detailed discussion of versions and difficulties such as breakdowns and a possibly somewhat erratic convergence see [6].

3.2 Lanczos-type product methods (LTPMs)

Sonneveld [17] found with the *(bi)conjugate gradient squared method* (CGS) a way to replace the multiplication with $\mathbf{A}^\mathsf{T}$ by a second one with $\mathbf{A}$. The nth residual polynomial of CGS is p_n^2, where p_n is still the nth BiCG residual polynomial. In each step the dimension of the Krylov space and the search space increases by two. Convergence is nearly twice as fast as for BiCG, but even more erratic.

BiCGStab by Van der Vorst [18] includes some local optimization and smoothing, and therefore the residual norm histories tend to be much smoother. The nth residual polynomial is $p_n t_n$, where now t_n satisfies the recursion

$$t_{n+1}(\zeta) = (1 - \chi_{n+1}\zeta)t_n(\zeta)$$

with χ_{n+1} chosen by residual minimization. However, in BiCGStab all zeros of t_n are real (if $\mathbf{A}$, $\mathbf{b}$ are real). If $\mathbf{A}$ has a complex spectrum, it is better to choose two possibly complex new zeros in every other iteration. This is the idea behind BiCGStab*2* [5]. Further generalizations of BiCGStab include BiCGStab*(ℓ)* by Sleijpen and Fokkema [15], [16] and GPBI-CG by Zhang [19]; see [6] for references to yet other proposals.

In view of the form of the residual polynomials, this family of methods has been called *Lanczos-type product methods*. These are often the most efficient solvers. They have short recurrences, they are typically about twice as fast as BiCG, and they do not require $\mathbf{A}^\mathsf{T}$. Unlike in GMRes, the memory needed does not increase with the iteration index n.

3.3 Solving the system in coordinate space: MinRes, SymmLQ, GMRes, and QMR

There is yet another class of Krylov space solvers, which includes well-known methods like MinRes, SymmLQ, GMRes, and QMR. It was pioneered by Paige and Saunders [10]. Their approach was later adapted to more general cases by Saad and Schultz [12] and Freund and Nachtigal [2].

The basic idea is to successively construct a basis of the Krylov space by combining the extension of the space with Gram-Schmidt orthogonalization or biorthogonalization, and to update at each iteration the approximate solution of $\mathbf{Ax} = \mathbf{b}$ in coordinate space. There are essentially three cases:

- symmetric Lanczos process ⇝ MinRes, SymmLQ [10]
- nonsymmetric Lanczos process ⇝ QMR [2]
- Arnoldi process ⇝ GMRes [12]

3.4 Further reading

For further study we suggest recent books on Krylov space solvers such as those of Greenbaum [4], Meurant [9], and Saad [11], as well as the review article by Simoncini and Szyld [13], which also covers developments of the last ten years. Nonsymmetric Lanczos-type solvers were reviewed in [6].

References

1. J. Dongarra and F. Sullivan. Guest editors' introduction to the top 10 algorithms. *Computing in Science and Engineering*, 2(1):22–23, 2000.
2. R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numer. Math.*, 60:315–339, 1991.
3. G. H. Golub and C. F. van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, 3nd edition, 1996.
4. A. Greenbaum. *Iterative Methods for Solving Linear Systems.* SIAM, Philadelphia, PA, 1997.
5. M. H. Gutknecht. Variants of BiCGStab for matrices with complex spectrum. *SIAM J. Sci. Comput.*, 14:1020–1033, 1993.
6. M. H. Gutknecht. Lanczos-type solvers for nonsymmetric linear systems of equations. *Acta Numerica*, 6:271–397, 1997.
7. M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bureau Standards*, 49:409–435, 1952.
8. C. Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bureau Standards*, 49:33–53, 1952.
9. G. Meurant. *Computer solution of large linear systems*, volume 28 of *Studies in Mathematics and its Applications.* North-Holland, Amsterdam, 1999.
10. C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12:617–629, 1975.
11. Y. Saad. *Iterative Methods for Sparse Linear Systems.* SIAM, Philadelphia, 2nd. edition, 2003.
12. Y. Saad and M. H. Schultz. Conjugate gradient-like algorithms for solving nonsymmetric linear systems. *Math. Comp.*, 44:417–424, 1985.
13. V. Simoncini and D. B. Szyld. Recent developments in krylov subspace methods for linear systems. *Numer. Linear Algebra Appl.* To appear.
14. V. Simoncini and D. B. Szyld. Flexible inner-outer Krylov subspace methods. *SIAM J. Numer. Anal.*, 40(6):2219–2239 (electronic) (2003), 2002.
15. G. L. G. Sleijpen and D. R. Fokkema. BiCGstab(l) for linear equations involving unsymmetric matrices with complex spectrum. *Electronic Trans. Numer. Anal.*, 1:11–32, 1993.
16. G. L. G. Sleijpen, H. A. van der Vorst, and D. R. Fokkema. BiCGstab(l) and other hybrid Bi-CG methods. *Numerical Algorithms*, 7:75–109, 1994.
17. P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 10:36–52, 1989.
18. 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. Sci. Statist. Comput.*, 13:631–644, 1992.
19. S.-L. Zhang. GPBI-CG: generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 18(2):537–551, 1997.