2

A Main Goal

Given a sparse matrix, for example, symmetric positive definite (or s.p.d. for short) $A = (a_{i,j})_{i,j=1}^{n}$, our main goal is to devise efficient algorithms for solving the system of linear equations,

$$A\mathbf{x} = \mathbf{b}.$$

In practice, n can be very large (e.g., in the range of millions). A first comment then is that in a massively parallel environment direct solvers are out of the question due to their prohibitive memory requirements.

The fact that A is sparse means that an inexpensive operation is "matrix times vector." Therefore, iterative methods for solving $A\mathbf{x} = \mathbf{b}$ are appealing because they involve computing at every iteration the residual

$$\mathbf{r} = \mathbf{b} - A\mathbf{x},$$

corresponding to a current iterate $\mathbf{x}$. The next iterate is obtained by computing a correction based on $\mathbf{r}$. A stationary iterative method exploits a mapping (sometimes explicitly represented by a matrix) B, which has an easily computable inverse action B^{-1} (in some cases implicitly represented only by a procedure). Then, the new iterate $\mathbf{x}_{\text{new}}$ equals,

$$\mathbf{x}_{\text{new}} = \mathbf{x} + B^{-1}\mathbf{r}.$$

The corresponding new residual $\mathbf{r}_{\text{new}} = \mathbf{b} - A\mathbf{x}_{\text{new}}$ equals

$$\mathbf{r}_{\text{new}} = (I - B^{-1}A)\mathbf{r}.$$

By making the successive residuals $\mathbf{r} \mapsto 0$ (in some norm), we get more accurate approximations $\mathbf{x}$ to the exact solution $A^{-1}\mathbf{b}$.

If B is s.p.d. (symmetric positive definite), the method of choice is the (preconditioned) CG, which initially computes $\mathbf{p} = B^{-1}\mathbf{r}$ and at every successive step updates it based on the preconditioned residual $\overline{\mathbf{r}} = B^{-1}\mathbf{r}$ as $\mathbf{p} := B^{-1}\mathbf{r} + \beta\ \mathbf{p}$ for a proper scalar β. At any rate, major operations here are again (as in a stationary iteration) the

P.S. Vassilevski, *Multilevel Block Factorization Preconditioners*,
doi: 10.1007/978-0-387-71564-3_2,

actions of B^{-1} and matrix vector products with A in addition to some vector inner products. The actual preconditioned CG iteration takes the form:

(0) Initiate: Given a tolerance ϵ and an integer $n_{\text{iter}}^{\max}$ that gives the maximal number of iterations allowed, for an initial iterate $\mathbf{x}$, which we choose $\mathbf{x} = B^{-1}\mathbf{b}$, we
1. Compute $\delta_0 = \mathbf{x}^T\mathbf{b}$.
2. Compute $\mathbf{r} = \mathbf{b} - A\mathbf{x}$,
3. Compute $\overline{\mathbf{r}} = B^{-1}\mathbf{r}$,
4. Let $\mathbf{p} = \overline{\mathbf{r}}$.
5. Compute $\delta = \overline{\mathbf{r}}^T\mathbf{r}$.
6. Test for convergence, if $\delta \leq \epsilon^2\ \delta_0$ stop.
7. Set $n_{\text{iter}} = 0$;

(i) Loop: until convergence or max number of iterations reached
1. Compute $\mathbf{h} = A\mathbf{p}$.
2. Compute $\alpha = \delta/\mathbf{p}^T\mathbf{h}$.
3. Compute $\mathbf{x} := \mathbf{x} + \alpha\mathbf{p}$.
4. Compute $\mathbf{r} := \mathbf{r} - \alpha\mathbf{h}$.
5. Compute $\overline{\mathbf{r}} = B^{-1}\mathbf{r}$.
6. Set $\delta_{\text{old}} = \delta$,
7. Compute $\delta = \overline{\mathbf{r}}^T\mathbf{r}$.
8. Set $n_{\text{iter}} := n_{\text{iter}} + 1$;
9. Check for convergence: if either $\delta \leq \epsilon^2\ \delta_{\text{old}}$ or $n_{\text{iter}} > n_{\text{iter}}^{\max}$, stop.
10. Compute $\beta = \delta/\delta_{\text{old}}$.
11. Compute $\mathbf{p} := \overline{\mathbf{r}} + \beta\ \mathbf{p}$ and go to (i).

We have the following popular convergence rate result after n_{iter} steps,

$$\delta \leq \kappa \left(\frac{2q^{n_{\text{iter}}}}{1+q^{2n_{\text{iter}}}}\right)^2\ \delta_0 \leq 4\kappa\ q^{2n_{\text{iter}}}\ \delta_0,$$

where $q = \sqrt{\kappa} - 1/\sqrt{\kappa} + 1$ and $\kappa = \text{Cond}(B^{-1}A)$.

In practice, we typically prove (spectral equivalence) estimates

$$c_1\ \mathbf{v}^T A\mathbf{v} \leq \mathbf{v}^T B\mathbf{v} \leq c_2\ \mathbf{v}^T A\mathbf{v} \quad \text{for all } \mathbf{v}.$$

Then, because the eigenvalues of $B^{-1}A$ are in $[1/c_2,\ \ 1/c_1]$, we clearly have $\kappa \leq c_2/c_1$.

A simple candidate for an iteration matrix B is based on the diagonal D of A. For example, we can use the diagonal matrix χD, where χ is either a diagonal matrix with entries $\chi_i = \sum_{j:\, a_{i,j}\neq 0} 1$ or just the scalar $\max_i \chi_i$. We have

$$c_1\ \mathbf{v}^T A\mathbf{v} \leq \mathbf{v}^T B\mathbf{v} \leq c_2\ \mathbf{v}^T A\mathbf{v}$$

with $c_1 = 1$. Unfortunately, for f.e. matrices A, such as the discretized Poisson equation, the estimate from below is mesh-dependent; that is, we typically have

$$c_2^{-1} = \min_{\mathbf{v}} \frac{\mathbf{v}^T A\mathbf{v}}{\mathbf{v}^T \chi D\mathbf{v}} \simeq h^2 \mapsto 0.$$

Hence, Cond $(B^{-1}A) \leq c_2/c_1 \simeq h^{-2} \mapsto \infty$ with $h \mapsto 0$.

The goal then is to construct a B such that:

(i) The action of B^{-1} costs as little as possible, the best being $\mathcal{O}(n)$ flops.
(ii) The constants c_1 and c_2 in the spectral equivalence estimate are such that c_2/c_1 is as close as possible to one, for example, being independent of various problem parameters, in particular being independent of n.
(iii)* In a massively parallel computer environment, we also want B^{-1} to be composed of local actions, essentially based on a "hierarchy" of sparse matrix vector products. The latter is achieved by the multilevel preconditioners that are a main topic of the present book.

Based on the convergence estimate, it is clear then that to get an approximate solution to $A\mathbf{x} = \mathbf{b}$ within tolerance ϵ, it is sufficient to perform as many as n_{iter} iterations such that $q^{n_{\text{iter}}} < \frac{1}{2}\kappa^{-1}\,\epsilon$ or $n_{\text{iter}} = \mathcal{O}(\log 1/\epsilon)$. The constant in the $\mathcal{O}$ symbol is reasonable as long as κ is kept under control (not too far away from one).

For large sparse matrices A that come from finite element discretization of elliptic PDEs (partial differential equations), like the Poisson equation, we can achieve both (i) and (ii) (and to a certain extent (iii)*) based on the multilevel preconditioning methods that are the main topic of the present book. Such methods are often referred to as *scalable* iterative methods.

http://www.springer.com/978-0-387-71563-6

Multilevel Block Factorization Preconditioners
Matrix-based Analysis and Algorithms for Solving Finite Element Equations
Vassilevski, P.S.
2008, XIV, 530 p. 34 illus., Hardcover
ISBN: 978-0-387-71563-6