An Inner Product-Free Conjugate Gradient-Like Algorithm for Hermitian Positive Definite Systems

Bernd Fischer[†] Roland W. Freund[‡]

Abstract

The conjugate gradient (CG) algorithm for the solution of Hermitian positive definite linear systems requires the computation of inner products. On modern architectures, inner products often represent a bottleneck, and it is desirable to reduce or even eliminate all the inner products. In this note, an inner product-free CG-like algorithm is presented that simulates the standard CG method by approximating the CG orthogonal polynomials by suitably chosen orthogonal polynomials from the Bernstein-Szegő class.

1 Introduction

The classical conjugate gradient (CG) algorithm of Hestenes and Stiefel [4] is one of the most widely used iterative methods for the solution of systems of linear equations

$$Ax = b, \tag{1}$$

where $A \in \mathbb{C}^{N \times N}$ is Hermitian positive definite. The CG algorithm is a polynomial-based iteration. Given any initial guess $x_0 \in \mathbb{C}^N$ and setting $r_0 = b - Ax_0$, CG generates a sequence of approximations to the solution of (1) of the form

$$x_n = x_0 + \psi_n(A) r_0, \quad n = 1, 2, \ldots, \tag{2}$$

where ψ_n is a polynomial of degree at most $n-1$. The residual vectors associated with the iterates x_n are given by

$$r_n = b - Ax_n = \varphi_n(A) r_0, \quad \text{where} \quad \varphi_n(\lambda) = 1 - \lambda \psi_n(\lambda).$$

Note that φ_n is a polynomial of degree at most n, and it is normalized such that $\varphi_n(0) = 1$. The polynomials φ_n are called the residual polynomials of the iteration.

The CG iterates are optimal, in the sense that $r_n^H A^{-1} r_n$ is minimal for all possible iterates of the form (2). Furthermore, the CG iterates can be computed efficiently, by means of recurrences of the form

$$p_n = r_n + p_{n-1}\beta_n, \quad x_{n+1} = x_n + p_n \alpha_n, \quad \text{and} \quad r_{n+1} = r_n - Ap_n\alpha_n. \tag{3}$$

For CG, the coefficients in (3) are given by

$$\beta_n = \frac{r_n^H r_n}{r_{n-1}^H r_{n-1}} \quad \text{and} \quad \alpha_n = \frac{r_n^H r_n}{p_n^H A p_n}.$$

and hence CG requires the computation of two inner products at each iteration step. On modern architectures, calculating inner products often represents a bottleneck.

[†]Institut für Angewandte Mathematik, Universität Hamburg, Bundesstrasse 55, D–20146 Hamburg, Federal Republic of Germany.

[‡]AT&T Bell Laboratories, Room 2C–420, 600 Mountain Avenue, Murray Hill, New Jersey 07974–0636.

2 The CG Residual Polynomials

The residual polynomials φ_n associated with the CG algorithm satisfy the orthogonality relations

$$\langle \varphi_n, \varphi_m \rangle := r_0^H \varphi_n(A)\varphi_m(A) r_0 = 0 \quad \text{for all} \quad n \neq m = 0, 1, \ldots . \tag{4}$$

Using the expansion

$$r_0 = \sum_{j=1}^{L} \sigma_j v_j, \quad \text{where} \quad A v_j = \lambda_j v_j, \quad \|v_j\|_2 = 1, \quad \lambda_1 < \lambda_2 < \cdots < \lambda_L,$$

of r_0 into orthonormal eigenvectors v_j of A, the inner product in (4) can be rewritten as the following Riemann-Stieltjes integral:

$$\langle \varphi_n, \varphi_m \rangle = \int_{\mathbb{R}} \varphi_n(\lambda)\varphi_m(\lambda) d\sigma(\lambda). \tag{5}$$

Here, $\sigma(\lambda)$ is a distribution function defined by

$$\sigma(\lambda) := \sum_{j=1}^{k} \sigma_j^2 \quad \text{for all} \quad \lambda_k \leq \lambda < \lambda_{k+1}, \quad k = 0, 1, \ldots, L, \tag{6}$$

where, for the cases $k = 0$ and $k = L$, we set $\lambda_0 := -\infty$ and $\lambda_{L+1} := +\infty$, respectively. In view of (4) and (5), the CG residual polynomials are orthogonal with respect to the distribution function (6).

Of course, the function $\sigma(\lambda)$ is usually not known. However, very good approximations to $\sigma(\lambda)$ can be constructed from the quantities generated by relatively few steps of the CG algorithm. This was first observed by Freund [3] and then exploited in [1] to construct efficient polynomial preconditioners for Hermitian positive definite linear systems (1). We now show how one can simulate the CG algorithm itself based on a suitably constructed approximation to the distribution function (6).

3 A CG-Type Iteration Based on Bernstein-Szegő Polynomials

The basic idea of our inner product-free CG-type algorithm is as follows. We approximate the unknown CG distribution function σ in (6) by a distribution function $\widetilde{\sigma}$ for which the corresponding orthogonal polynomials $\widetilde{\varphi}_n$ are known explicitly, and we then use $\widetilde{\varphi}_n$ as the residual polynomials of our CG-like scheme. More precisely, we use approximate distribution functions $\widetilde{\sigma}$ from the so-called Bernstein-Szegő class (see, e.g., [2, 6]). Written in terms of the unit interval $[-1, 1]$, the Bernstein-Szegő distribution functions are of the form

$$\sigma^{(\rho_k)}(t) = \int_{-1}^{t} \frac{\sqrt{1-\tau^2}}{\rho_k(\tau)} d\tau, \quad -1 \leq t \leq 1. \tag{7}$$

Here, ρ_k is an arbitrary real polynomial of degree $k \geq 0$ with $\rho_k(t) > 0$ for $-1 < t < 1$ that may have simple zeros at ± 1. The corresponding orthogonal polynomials $\varphi_n^{(\rho_k)}$ (and their three-term recurrence) are known explicitly for $n \geq \lfloor (k+1)/2 \rfloor$.

Our algorithm proceeds as follows. First, we run a small number, denoted by l in the sequel, of CG steps. Exploiting the connection between CG and the Lanczos process, we then compute the $l \times l$ tridiagonal Lanczos matrix T_l, its eigenvalues (the so-called Ritz values) $\theta_1 < \theta_2 < \cdots < \theta_l$, and the first components τ_j of the corresponding eigenvectors. Based on these quantities, we define interpolation points

$$(\mu_j, \vartheta_j) := \begin{cases} (\theta_1/2, 0) & \text{for } j = 0, \\ (\theta_j, \tau_j^2/2 + \sum_{i=1}^{j-1} \tau_i^2) & \text{for } j = 1, 2, \ldots, l-1, \\ (\theta_l, 1) & \text{for } j = l. \end{cases}$$

Furthermore, let $t(\lambda) = (2\lambda - \mu_l - \mu_0)/(\mu_l - \mu_0)$ be the linear function that maps $[\mu_0, \mu_l]$ onto the unit interval $[-1, 1]$. Next, we determine a polynomial ρ_{l-1} of degree $l-1$ with $\rho_{l-1}(t) > 0$ for $-1 < t < 1$ as the solution of the following nonlinear interpolation problem:

$$\vartheta_j = \int_{-1}^{t(\mu_j)} \frac{\sqrt{1-\tau^2}}{\rho_{l-1}(\tau)}\, d\tau, \quad j = 0, 1, \ldots, l. \tag{8}$$

An actual numerical procedure for solving (8) is described in [1]. With ρ_{l-1} denoting the solution of (8) and using (7) (with $k = l-1$), we now set

$$\widetilde{\sigma}(\lambda) = \sigma^{(\rho_{l-1})}(t(\lambda)) = \int_{-1}^{t(\lambda)} \frac{\sqrt{1-\tau^2}}{\rho_{l-1}(\tau)}\, d\tau \quad \text{and} \quad \widetilde{\varphi}_n(\lambda) = \frac{\varphi_n^{(\rho_{l-1})}(t(\lambda))}{\varphi_n^{(\rho_{l-1})}(t(0))}.$$

Note that, for $n \geq \lfloor l/2 \rfloor$, the polynomial $\widetilde{\varphi}_n$ is an nth orthogonal polynomial with respect to $\widetilde{\sigma}$. Finally, we use the $\widetilde{\varphi}_n$'s as the residual polynomials of our algorithm, and we set $r_n = \widetilde{\varphi}_n(A) r_0$ for all $n \geq \lfloor l/2 \rfloor$. We remark that the associated iterates of this algorithm can be conveniently generated by recurrences of the form (3), where the recurrence coefficients α_n and β_n are known explicitly. Thus, unlike CG, this algorithm based on Bernstein-Szegő polynomials does not involve inner products.

We illustrate the convergence behavior of our algorithm with one typical example. It is a linear system (1) of order $N = 2500$ resulting from a discretization of $-\Delta u = f$ on the unit square. In Figure 1, we show the relative residual norms $\|r_n\|_2 / \|r_0\|_2$ for our Bernstein-Szegő algorithm (with $l = 20$), for standard Chebyshev iteration (based on the exact extreme eigenvalues) [4], and for the CG algorithm.

References

[1] B. Fischer and R. W. Freund, *On adaptive weighted polynomial preconditioning for Hermitian positive definite matrices*, SIAM J. Sci. Comput., 15 (1994), pp. 408–426.

[2] R. W. Freund, *On some approximation problems for complex polynomials*, Constr. Approx., 4 (1988), pp. 111–121.

[3] ——, *Polynomial preconditioners for Hermitian and certain non-Hermitian matrices*, Minisymposium Presentation, 1989 SIAM Annual Meeting, San Diego, July 1989.

[4] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods*, Numer. Math., 3 (1961), pp. 147–168.

[5] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–436.

[6] G. Szegő, *Orthogonal Polynomials*, 4th ed., Amer. Math. Soc., Providence, RI, 1975.

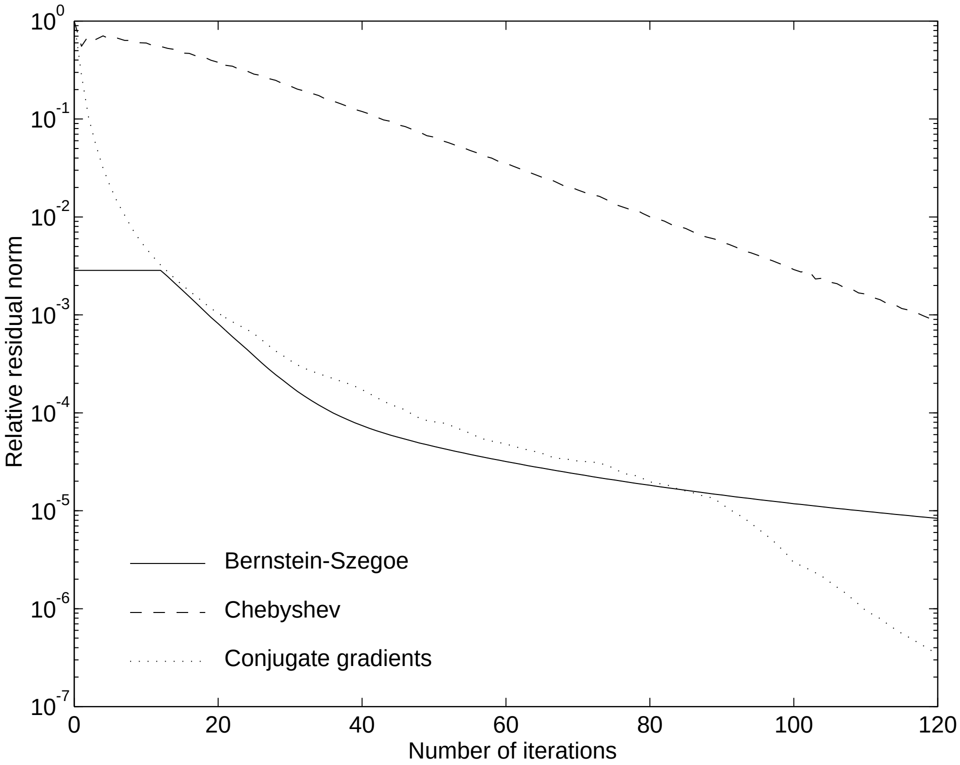

FIG. 1. Convergence behavior for a typical example