SIAM J. NUMER. ANAL.
Vol. 20, No. 2, April 1983

0036-1429/83/2002-0011 $01.25/0

POLYNOMIAL PRECONDITIONERS FOR CONJUGATE GRADIENT CALCULATIONS*

OLIN G. JOHNSON,† CHARLES A. MICCHELLI‡ AND GEORGE PAUL§

Abstract. Dubois, Greenbaum and Rodrigue proposed using a truncated Neumann series as an approximation to the inverse of a matrix A for the purpose of preconditioning conjugate gradient iterative approximations to $Ax = b$. If we assume that A has been symmetrically scaled to have unit diagonal and is thus of the form $(I - G)$, then the Neumann series is a power series in G with unit coefficients. The incomplete inverse was thought of as a replacement of the incomplete Cholesky decomposition suggested by Meijerink and van der Vorst in the family of methods ICCG (n). The motivation for the replacement was the desire to have a preconditioned conjugate gradient method which only involved vector operations and which utilized long vectors.

We here suggest parameterizing the incomplete inverse to form a preconditioning matrix whose inverse is a polynomial in G. We then show how to select the parameters to minimize the condition number of the product of the polynomial and $(I - G)$. Theoretically the resulting algorithm is the best of the class involving polynomial preconditioners. We also show that polynomial preconditioners which minimize the mean square error with respect to a large class of weight functions are positive definite. We give recurrence relations for the computation of both classes of polynomial preconditioners.

1. Introduction.

The PCG, (Preconditioned Conjugate Gradient) algorithm for the solution of the system of linear equations

$$Ax = b, \tag{1.1}$$

where A is an $n \times n$ symmetric positive matrix, is given as follows [1].

PCG ALGORITHM. *Let x_0 be an initial approximation to x. Let M be an arbitrary $n \times n$ positive definite matrix. Compute the residual,*

$$r_0 = b - Ax_0.$$

Then solve the linear system

$$Md_0 = r_0$$

for d_0, and set $p_0 = d_0$.

Thereafter, for $i = 0, 1, 2, \cdots$ iteratively compute steps (a) *thru* (f) *until x_{i+1} is sufficiently close to x_i, then terminate.*

a) $a_i = (r_i, d_i)/(p_i, Ap_i)$,
b) $x_{i+1} = x_i + a_i p_i$,
c) $r_{i+1} = r_i - a_i A p_i$,
d) *Solve the linear system* $Md_{i+1} = r_{i+1}$ *for* d_{i+1},
e) $b_i = (r_{i+1}, d_{i+1})/(r_i, d_i)$,
f) $p_{i+1} = d_{i+1} + b_i p_i$.

This form of the PCG algorithm is called the computational form as opposed to the recursive form which presents the algorithm as the three term nonstationary recurrence

$$x_{i+1} = x_{i-1} + \omega_{i+1}(\alpha_i d_i + x_i - x_{i-1}), \tag{1.2}$$

* Received by the editors January 15, 1981, and in final revised form May 13, 1982.

† Computer Science Department, University of Houston, Houston, Texas 77004.

‡ Mathematical Sciences Department, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 10598.

§ Computer Sciences Department, IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.

where

$$d_i = M^{-1} r_i = M^{-1}(b - Ax_i) \tag{1.3}$$

and the parameters ω_{i+1} and α_i are chosen at each step so that the inner product

$$(r_i, d_j) = 0, \qquad i > j.$$

The matrix M is called the preconditioning matrix and the speed with which the algorithm converges depends strongly on the choice of M. It is required that M approximate A, that M retain any desirable features of A such as sparsity and that the system of equations

$$Md_i = r_i$$

be easily solvable.

Several papers in the past few years have dealt with the choice of M for matrices A which arise from discrete approximations to boundary value problems. Some natural choices for M are well known. For instance, if the given differential operator contains the Laplace operator, then M could be its discrete counterpart and (1.3) could be solved by a fast Poisson solver. Alternatively, the highly successful family of methods ICCG(n) suggested in [6] uses an incomplete Cholesky factorization of A for M. This particular device is not ideally suited to vector computers due to the nature of the forward–backward substitutions involved. In [3] the incomplete factorization was replaced with the incomplete inverse. We can assume without loss of generality that A is symmetrically scaled to have unit diagonal entries. Hence A is of the form $(I - G)$, where G is symmetric with zero entries on its diagonal. If A is a discrete approximation to a differential operator, then it is often the case that the spectral radius of G is less than one. Hence,

$$(I - G)^{-1} = I + G + G^2 + G^3 + \cdots.$$

The incomplete inverse of A is simply a truncated form of this power series. If A is sparse then so is G, and the computed incomplete inverse fills in as more and more terms of the series are used. Hence the incomplete inverse has many of the same computational advantages as the incomplete factorization. Experiments by the authors of [3], however, indicate that the incomplete inverse is slightly less effective than the incomplete factorization.

We now suggest parameterizing the incomplete inverse in the form:

$$M_m^{-1} \equiv \sum_{j=0}^{m} \gamma_{j,m} G^j. \tag{1.4}$$

In other words, we suggest preconditioning the method with a preconditioner whose inverse is a polynomial in G (or equivalently, a polynomial in A). The parameters are to be chosen to optimize the convergence of the algorithm. This will be done by minimizing the known constant in the usual contraction mapping theorem applied to the preconditioned conjugate gradient method.

In the following section, the influence of the spectral variation of $M^{-1}A$ on the convergence rate is reviewed. This allows us to formulate the problem of minimal spectral variance. In § 3, we examine the implications of polynomial preconditioning in light of the fact that PCG is itself a polynomial accelerator. This leads us to view polynomial preconditioning of the PCG method as an outer–inner iteration scheme in which the outer and inner iterations have different but coupled criteria for optimality.

In § 4, we solve the minimal spectral variation problem and present a theoretical comparison of the resulting bounds and work to the standard (unconditioned) conjugate gradient method, and in § 5 we develop the basic theory of generalized optimal polynomial preconditioners. We derive not only the min–max and least squares solutions but also a large class of weight functions for which the optimal polynomial is guaranteed to be positive definite (a necessary requirement for its use as a preconditioner in the PCG algorithm). In § 6 we derive recurrence relations for computing the optimal polynomials. In [4] we present numerical results comparing the optimal min–max and least-squares techniques with other parameter choices and with other methods, including ICCG(3) and point SOR.

2. Minimization of spectral variance by parameter selection. The rate of convergence of the preconditioned conjugate gradient algorithm is bounded by the following theorem.

THEOREM 1. *The error in the kth iterate of the* PCG *algorithm specified above is bounded by the relation*

$$\|x - x_k\|_A \leqq 2\left(\frac{\sqrt{\delta}-1}{\sqrt{\delta}+1}\right)^k \|x - x_0\|_A, \tag{2.1}$$

where

$$\delta = \kappa(M^{-1}A) \tag{2.2}$$

and

$$\kappa(Z) = \|Z\|_2 \, \|Z^{-1}\|_2.$$

Proof. Refer to [2, Chap. 5].

If X is a symmetric and positive definite matrix, then

$$\kappa(X) = \frac{\lambda_{\max}}{\lambda_{\min}},$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalues of the matrix X.

For the preconditioner M given in the previous section, we have

$$M_m^{-1}A = \sum_{j=0}^{m} \gamma_{j,m} G^j (I - G).$$

Equivalently as a polynomial in A

$$M_m^{-1}A = \delta_{0,m}A + \delta_{1,m}A^2 + \delta_{2,m}A^3 + \cdots + \delta_{m,m}A^{m+1} = p_m(A)A.$$

Hence $M_m^{-1}A$ is symmetric and its spectrum is given by $\lambda_i p_m(\lambda_i)$, $i = 1, 2, \cdots, n$, where each λ_i is an eigenvalue of A, ordered so that

$$\lambda_{\min} = \lambda_1 \leqq \lambda_2 \leqq \cdots \leqq \lambda_n = \lambda_{\max}.$$

Thus $M_m^{-1} = p_m(A)$ and the polynomial $q_{m+1}(y) = y p_m(y)$ vanishes at $y = 0$.

Ideally, we would like for the spectrum of $M_m^{-1}A$ to be the set consisting of the single value one, i.e., $\{1\}$. Since $M_m^{-1}A = q_{m+1}(A)$, suppressing the dependence of $\delta_{j,m}$ on m we have

$$q_{m+1}(y) = \delta_0 y + \delta_1 y^2 + \delta_2 y^3 + \cdots + \delta_{m-1} y^m + \delta_m y^{m+1}. \tag{2.3}$$

Then $q_{m+1}(y)$, for $\lambda_1 \leqq y \leqq \lambda_n$, is a continuous approximation to the spectrum of $M_m^{-1}A$.

Since the eigenvalues of A, $(\lambda_i, i = 1, 2, \cdots, n)$, are not known, the idea of picking the parameters $\delta_0, \delta_1, \cdots, \delta_m$ to force $q_{m+1}(y)$ to be as close to unity as possible for all y in the eigenvalue range presents itself. Unfortunately, $q_{m+1}(y)$ vanishes at $y = 0$, hence, we require a polynomial which is somewhat similar to a step function.

Let Q_{m+1} be the set of polynomials of degree $m+1$ or less which are positive on $[\lambda_1, \lambda_n]$ and which vanish at zero. All polynomials in Q_{m+1} can be written in the form (2.3). Conversely, any polynomial of the form (2.3) vanishes at zero but may not be positive on $[\lambda_1, \lambda_n]$. Hence, our problem is to find the minimum value for $q \in Q_{m+1}$ of the expression

$$\frac{\max_{\lambda_1 \leqq y \leqq \lambda_n} q(y)}{\min_{\lambda_1 \leqq y \leqq \lambda_n} q(y)} \tag{2.4}$$

which is an upper bound on the condition number for $M_m^{-1}A$. For any polynomial $q \in Q_{m+1}$ we call (2.4) cond (q). We solve this problem in § 4 and show that the result corresponds to a Chebyshev accelerator. Before doing so however, we discuss some of the advantageous features of the PCG algorithm with a polynomial preconditioner.

3. Inner and outer iterations. For any preconditioner M there is a splitting

$$A = M + (A - M)$$

and a corresponding stationary iterative method

$$z_{i+1} = M^{-1}(M - A)z_i + M^{-1}b$$

which can be accelerated by polynomial averaging

$$x_i = \sum_{j=0}^{i} \beta_{j,i} z_j,$$

where

$$\sum_{j=0}^{i} \beta_{j,i} = 1.$$

Let

$$e_i = x_i - x.$$

Since

$$z_{i+1} - x = M^{-1}(M - A)(z_i - x)$$

it follows that

$$e_i = R_i(M^{-1}A)e_0, \tag{3.1}$$

where

$$R_i(y) = \sum_{j=0}^{i} \beta_{j,i}(1 - y)^j.$$

Moreover, it can be shown that

$$x_i = x_0 + P_{i-1}(M^{-1}A)d_0, \tag{3.2}$$

where P_{i-1} is defined by

$$1 - xP_{i-1}(x) = R_i(x) \tag{3.3}$$

and d_0 is given by (1.3), i.e., $Md_0 = r_0$.

Let $K_i(d_0)$ be the Krylov space spanned by $\{d_0, M^{-1}Ad_0, \cdots, (M^{-1}A)^{i-1}d_0\}$. Since $R_i(0) = 1$, it follows from (3.1) that $(x_i - x_0) \in K_i(d_0)$.

When the polynomial $R_i(x)$ corresponds to the PCG method we have the following well-known optimality property of this acceleration method.

THEOREM 2. *The* PCG *algorithm computes the unique polynomial average which minimizes $x_i - x$ in the sense that*

$$\|x_i - x\|_A \leqq \|x_0 + u - x\|_A$$

for all $u \in K_i(d_0)$.

Proof. Refer to [2, Chap. 5].

Recalling that our choice for M_m^{-1} is

$$M_m^{-1} = p_m(A) = \sum_{k=0}^{m} \delta_{k,m} A^k,$$

we see from (3.1) that

$$e_i = R_i(Ap_m(A))e_0. \tag{3.4}$$

This equation shows that the choice of a polynomial preconditioner in the PCG method leads to a "composite" acceleration scheme determined by the residual polynomial

$$R_{i,m}(x) = R_i(xp_m(x)),$$

since clearly $R_{i,m}(0) = 1$. Note that the ith iterate in the PCG method corresponds to the $i(m+1)$th iterate of the composite method.

In this composite scheme, we will distinguish between the inner iteration whose residual polynomial is given by $r_{m+1}(x) = 1 - xp_m(x)$ and the outer iteration determined by $R_i(x)$.

Generally, the advantage of an inner–outer acceleration is that two iteration schemes which have a simple structure, for instance, given in the form of a three term nonstationary recurrence (1.2) can conveniently be combined to yield a composite scheme with greater computational power.

Moreover, when the outer iteration is the PCG method the effect of a polynomial preconditioner is a composite scheme which picks the best vector in the Krylov space $\{d_0, M^{-1}Ad_0, \cdots, (M^{-1}A)^{i-1}d_0\}$ for each i, rather than doing the same for the Krylov space for the matrix A, $\{r_0, Ar_0, \cdots, A^{i-1}r_0\}$ corresponding to the CG scheme. The use of this inner iteration can dramatically speed up the convergence (see [4] for numerical evidence of this fact).

The specific inner–outer acceleration schemes which we discuss will always be composed of two schemes of the form (1.2). Thus for the outer iteration an initial x_0 is specified and

$$x_{i+1} = x_{i-1} + \omega_{i+1}(\alpha_i d_i + x_i - x_{i-1}),$$

where the vector d_i is computed by means of an inner iteration

$$d_{i,j+1} = d_{i,j-1} + \omega'_{j+1}(\alpha'_j f_{i,j} + d_{i,j} - d_{i,j-1}) \tag{3.5}$$

with $d_{i,0} = 0$, $f_{ij} = r_i - Ad_{i,j}$, $r_i = b - Ax_i$ and terminating at

$$d_{i,m+1} = d_i.$$

Alternatively, $d_{i,m+1} = p_m(A)r_i$ and the residual polynomials

$$r_m(x) = 1 - xp_{m-1}(x)$$

for the inner iteration satisfy the recurrence

$$r_{j+1}(\lambda) = r_{j-1}(\lambda) + \omega'_{j+1}((1-\alpha'_j\lambda)r_j(\lambda) - r_{j-1}(\lambda)),$$

where $r_0(\lambda) = 1$ and $r_{-1}(\lambda) = 0$. In the following sections we discuss various optimal choices for inner iterations for use with the PCG algorithm.

4. Min–max preconditioners. Let

$$\mu(\lambda) = -1 + 2\left(\frac{\lambda - \lambda_1}{\lambda_n - \lambda_1}\right)$$

be the linear mapping taking $[\lambda_1, \lambda_n]$ into $[-1, +1]$ and define

$$q_{m+1}(\lambda) = 1 - \frac{T_{m+1}(\mu(\lambda))}{T_{m+1}(\mu(0))}. \tag{4.1}$$

Since $0 < \lambda_1 < \lambda_n$ it follows that $|T_{m+1}(\mu(0))| > 1$ and $|T_{m+1}(\mu(\lambda))| \leqq 1$ for $\lambda \in [\lambda_1, \lambda_n]$. Thus $q_{m+1}(\lambda) \in Q_{m+1}$.

THEOREM 3. *cond* $(q_{m+1}) \leqq$ *cond* (s) *for all* $s \in Q_{m+1}$ *with equality if and only if* s *is a scalar multiple of* q_{m+1}.

Proof. From (4.1) we have

$$\text{cond}\,(q_{m+1}) = \frac{\theta+1}{\theta-1},$$

where

$$\theta = |T_{m+1}(\mu(0))|.$$

Let s be any polynomial in Q_{m+1} with

$$\max_{\lambda_1 \leqq \lambda \leqq \lambda_n} s(\lambda) = V, \qquad \min_{\lambda_1 \leqq \lambda \leqq \lambda_n} s(\lambda) = v.$$

We must show

$$\frac{\theta+1}{\theta-1} \leqq \frac{V}{v}$$

or equivalently

$$\frac{V+v}{V-v} \leqq \theta$$

with equality if and only if $s = cq_{m+1}$. Note that $V > v$ since $s \in Q_{m+1}$. The last inequality can be seen from the fact that, for any polynomial u of degree $m+1$ or less and for $|\mu| \geqq 1$,

$$|u(\mu)| \leqq |T_{m+1}(\mu)| \max_{|t| \leqq 1} |u(t)| \tag{4.2}$$

with equality if and only if $u = cT_{m+1}$. The theorem is proved by applying this inequality to

$$u(\mu(\lambda)) = 1 - \frac{2s(\lambda)}{V+v}$$

at $\mu(0)$, since $u(\mu(0)) = 1$ and $\max_{|t| \leqq 1} |u(t)| = (V-v)/(V+v)$.

The optimal choice of Chebyshev iteration for the inner iteration suggested by Theorem 3 was already mentioned in [8] and referred to as cgT. However, its optimality property as stated in Theorem 3 was not observed. The recurrence relation for this method will be discussed in § 6.

Theorem 3 allows us to compare the convergence bounds of Theorem 1 for PCG preconditioned as above with the standard conjugate gradient technique in certain cases. Suppose, for instance, that

$$\lambda_1 = 1-\beta, \qquad \lambda_n = 1+\beta, \qquad \beta < 1$$

as is the case in the usual finite difference approximation to Laplace's equation. Then β is the spectral radius of G and

$$\kappa(A) = \frac{1+\beta}{1-\beta}.$$

Hence, for Theorem 1, the rate of convergence of the unconditioned conjugate gradient method is governed by

$$\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right) = \nu,$$

where

$$\nu = \beta^{-1} - \sqrt{\beta^{-2}-1}.$$

Note that $\nu < 1$. Also note that $\mu(0) = \beta^{-1}$ and

$$\text{cond}\,(q_{m+1}) = \frac{T_{m+1}(\beta^{-1})+1}{T_{m+1}(\beta^{-1})-1}.$$

Since

$$T_{m+1}(\beta^{-1}) = \tfrac{1}{2}(\nu^{m+1} + \nu^{-m-1}),$$

we have

$$\text{cond}\,(q_{m+1}) = \left(\frac{1+\nu^{m+1}}{1-\nu^{m+1}}\right)^2.$$

Finally, the convergence factor for PCG with this preconditioning is

$$\left(\frac{\sqrt{\text{cond}\,(q_{m+1})}-1}{\sqrt{\text{cond}\,(q_{m+1})}+1}\right) = \nu^{m+1}.$$

In other words, one step of PCG preconditioned as above has the same convergence bound as $(m+1)$ steps of conjugate gradient. On the other hand, the amount of work is less. Each step of the inner iteration involves only a multiplication by G and a coefficient whereas each outer iteration involves a multiplication by A in step (a) as well as the inner products, vector adds and scalar times vector multiples of steps (b), (c), (e) and (f).

Comparisons on the basis of bounds is not the whole story as is well known. In practice, preconditioning will often perform better than the above theoretical considerations indicate. Also, our experience indicates that other polynomial preconditioners will sometimes out perform the min–max preconditioner above [4]. This leads us to consider a more general class of optimal polynomial preconditioners.

5. Generalized optimal polynomial preconditioners. While the above preconditioners are best in the min–max sense, they, nonetheless, have a serious disadvantage. The optimal polynomial preconditioner M may map small eigenvalues of A into large ones of $M^{-1}A$. This fact seems to degrade the convergence rate of the iteration scheme. As an alternative choice we consider minimizing some quadratic norm of the residual polynomial

$$\int_{\lambda_1}^{\lambda_n} (1-q(\lambda))^2 w(\lambda)\, d\lambda. \tag{5.1}$$

The weight function $w(\lambda)$ can be chosen to emphasize the portion of the spectrum which is most important. For instance, the weight function

$$w(\lambda) = \lambda^{-1}((\lambda_n - \lambda)(\lambda - \lambda_1))^{-1/2}$$

will lead us back to the minimal polynomial of § 4. Although, in general, these polynomials are not the best, in the sense of § 4, their performance on a given problem can be better than the min–max optimum. We now ask the general question: What class of weight functions $w(\lambda) > 0$, $\lambda \in [\lambda_1, \lambda_n]$ produce an optimal solution which is contained in Q_{m+1}? This is answered in part by Theorem 4.

Theorem 4 is widely applicable. It applies to all the Jacobi weights

$$w(\lambda) = (\lambda_n - \lambda)^{\alpha}(\lambda - \lambda_1)^{\beta},$$

where

$$\alpha \geqq \beta \geqq -\tfrac{1}{2},$$

since it is known that the Jacobi polynomials in this set achieve their maximum at one [9, p. 166]. The set includes the Chebyshev weight function as well as the uniform (Legendre or least squares) weight function.

Moreover, whenever we have a sequence of orthogonal polynomials $\{s_j\}$ with corresponding weight function w satisfying the hypothesis of Theorem 4, then the orthogonal polynomials $\{p_j\}$ relative to $w_1 w$ also satisfy this condition if $w_1'(t)$ is completely monotonic. This fact follows from a result in [7] which implies that

$$p_i(x) = \sum_{j=0}^{i} a_{i,j} s_j(x),$$

where $a_{i,j} \geqq 0$. The nonnegativity of the coefficients in the above representation immediately implies that $p_j(\lambda) \leqq p_j(\lambda_1)$, $\lambda \in [\lambda_1, \lambda_n]$.

THEOREM 4. *Let $s_i(\lambda)$, $i = 1, 2, \cdots, m+1$ be orthonormal with respect to the arbitrary weight $w(\lambda) > 0$, $\lambda \in [\lambda_1, \lambda_n]$ normalized so that $s_i(0) > 0$. The solution q^* to* (5.1) *is positive on $[\lambda_1, \lambda_n]$ whenever each s_i, $i = 1, 2, \cdots, m+1$ obtains its maximum on $[\lambda_1, \lambda_n]$ at $\lambda = \lambda_1$.*

Proof. Let

$$J_{m+1}(\sigma, \lambda) = \sum_{j=0}^{m+1} s_j(\sigma) s_j(\lambda).$$

Then, by spectral decomposition, we have, for any polynomial q of degree $m+1$ or less,

$$1 - q(\sigma) = \int_{\lambda_1}^{\lambda_n} J_{m+1}(\sigma, \lambda)(1 - q(\lambda)) w(\lambda)\, d\lambda,$$

and by the Cauchy–Schwarz inequality, since $q(0)=0$ we have

$$1=(1-q(0))^2 \leqq J_{m+1}(0,0)\int_{\lambda_1}^{\lambda_n}(1-q(\lambda))^2 w(\lambda)\,d\lambda.$$

Now let

$$q^*_{m+1}(\lambda)=1-\frac{J_{m+1}(0,\lambda)}{J_{m+1}(0,0)}.$$

Since $1-q^*_{m+1}(\lambda)$ is colinear with $J_{m+1}(0,\lambda)$ and is equal to one at zero, Cauchy–Schwarz gives strict equality for this function and $q^*_{m+1}(\lambda)$ vanishes at zero. Hence the minimum is achieved by $q^*_{m+1}(\lambda)$. But for $\lambda\in[0,\lambda_n]$

$$\frac{J_{m+1}(0,\lambda)}{J_{m+1}(0,0)}<1$$

since each s_i achieves its maximum at zero. Thus $q^*_{m+1}(\lambda)$ is positive on $[\lambda_1,\lambda_n]$.

The class of polynomial preconditioners suggested by Theorem 4 has a theoretical convergence rate which is essentially as good as the method discussed in § 4. Our analysis below suggests more general results but we content ourselves here with the special case of the Legendre weight. First, we require the following simple lemma.

LEMMA 1. *For any polynomial $p(x)$ of degree less than or equal to m and $x\in[-1,+1]$ we have*

$$|p(x)|^2 \leqq c_m\int_{-1}^{+1}|p(t)|^2\,dt,$$

where $c_m=(m^2+2m)/2$ is the best constant in this inequality.

Proof. Just as in the proof of Theorem 4 it follows that

$$|p(x)|\leqq K(x,x)\int_{-1}^{+1}|p(t)|^2\,dt,$$

where $K(x,x)=\sum_{i=1}^{m}(i+1/2)P_i^2(x)$ and P_i is the ith Legendre polynomial. Hence

$$K(x,x)\leqq\sum_{i=1}^{m}(i+1/2)=(m^2+2m)/2$$

and the lemma is proved.

Now, as before we suppose that $\lambda_1=1-\beta$ and $\lambda_n=1+\beta$. Then it easily follows from Lemma 1 that the condition number κ for the polynomial preconditioner corresponding to the Legendre weight can be estimated from above by

$$\frac{1+\rho_{m+1}}{1-\rho_{m+1}},$$

for some ρ_{m+1} satisfying

$$\rho_{m+1}\leqq\frac{(m+2)/\sqrt{2}}{T_{m+1}(\beta^{-1})}$$

assuming m is large enough such that $\rho_{m+1}<1$. Thus the convergence factor for the corresponding PCG algorithm is bounded above by

$$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\leqq\frac{\kappa-1}{\kappa+1}=\rho_{m+1}\leqq\sqrt{2}(m+2)\nu^{m+1},$$

where ν was defined earlier as

$$\nu = \beta^{-1} - \sqrt{\beta^{-2} - 1}.$$

This bound compares quite favorably with the bound derived in § 4 for the PCG algorithm with a Chebyshev preconditioner.

In Fig. 1, we compare the optimal least-squares Legendre weight and minimum condition number polynomial preconditioners for the case $\lambda_1 = 1 - \beta$, $\lambda_n = 1 + \beta$, and for the values $\beta = 0.75$ and $\beta = 0.90$. For β near one, the spectral variance of the optimal least-squares Legendre weight preconditioner is only slightly worse than the minimum condition number polynomial and the optimal least-squares Legendre weight preconditioner flattens the spectrum much better as a whole, while emphasing the larger range of the spectrum. Although it is not shown here, the same is true of the optimal least-squares for the Chebyshev weight. There is little difference in the results for the two weights.

In summary then we see that minimizing the quadratic norm (5.1) leads to a large collection of suitable polynomial accelerators for the PCG algorithm as given by Theorem 4. It is interesting to note that although the unconditioned conjugate gradient algorithm may be considered as a polynomial preconditioner, when used in an inner–outer iteration scheme, it does not yield a preconditioner that is positive definite. Simple examples of this fact are easy to construct.

6. Polynomial preconditioning recurrence formulae. It is convenient to compute $d_{i,j}$ recursively for $j = 1, 2, \cdots, m$ as in (3.5) for a variety of reasons:

i) One can conveniently write a general program in which the degree of the preconditioning polynomial is a variable specified by the user.

ii) The vectors $d_{i,j}$ can be used for further research on inner loop convergence or on extrapolation techniques.

iii) Recursive calculations of this type are computationally efficient.

In this section we provide these recurrence formulas in some generality for use with the PCG method.

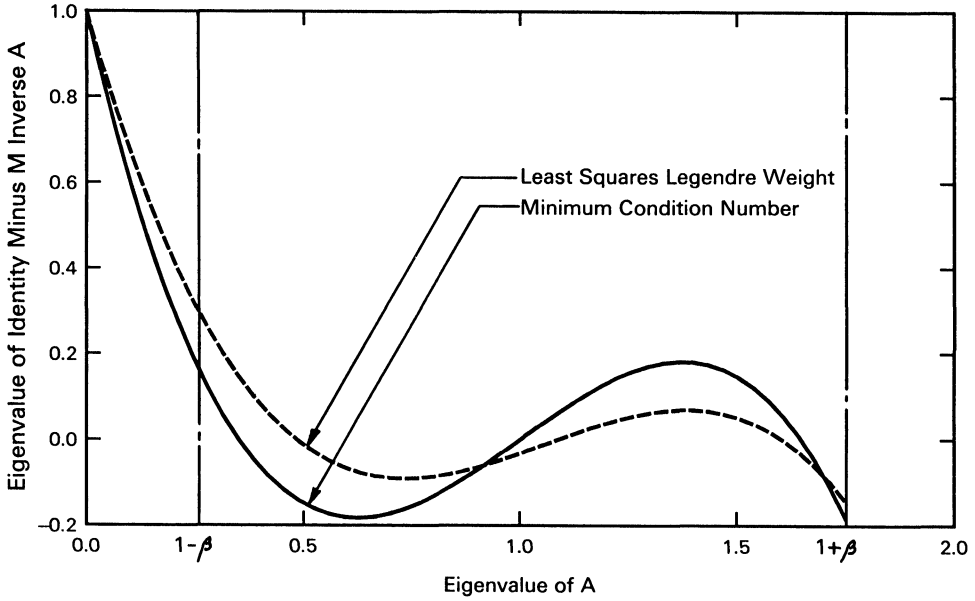

FIG. 1a. *Residual polynomials for minimum condition number and least-squares Legendre weight; beta* = 0.75.

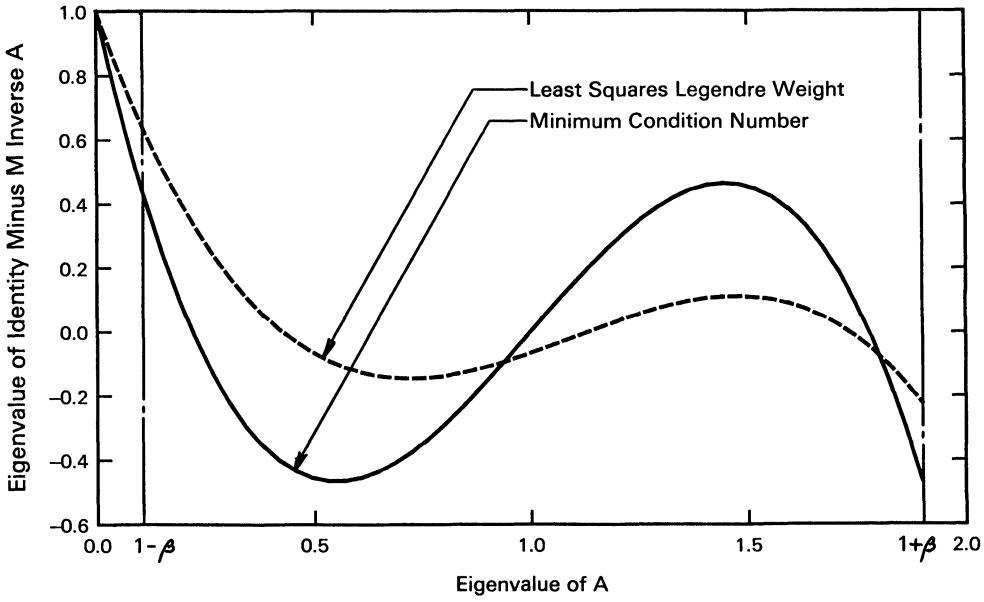

FIG. 1b. *Residual polynomials for minimum condition number and least-squares Legendre weight; beta* = 0.90.

Using the Christoffel–Darboux formula we can express the optimal polynomial of Theorem 4 in the form

$$1-q^*_{m+1}(\lambda)=\frac{v_{m+1}(\lambda)}{v_{m+1}(0)},$$

where

$$v_m(\lambda)=\frac{s_{m+1}(\lambda)-s_m(\lambda)}{\lambda}$$

and $\{s_m\}$ are the orthogonal polynomials relative to $w(\lambda)$ on $[\lambda_1, \lambda_n]$, normalized so that $s_m(0)=1$.

If we suppose that the recurrence relation for $s_m(\lambda)$ is available (which is certainly the case for the Jacobi weight function), it is possible to derive a recurrence relation for $1-q^*_{m+1}(x)$. We suppose

$$-\lambda s_j(\lambda)=r_j s_{j+1}(\lambda)-(r_j+t_j)s_j(\lambda)+t_j s_{j-1}(\lambda),$$

where $t_0=0$, $t_{j+1}>0$, $r_j>0$ for $j\geqq 0$, then Lemma 2 follows.

LEMMA 2.

$$-\lambda v_j(\lambda)=r_{j+1}v_{j+1}(\lambda)-(t_{j+1}+r_j)v_j(\lambda)+t_j v_{j-1}(\lambda).$$

Proof.

$$\begin{aligned}-\lambda(s_{j+1}(\lambda)-s_j(\lambda)) &= r_{j+1}s_{j+2}(\lambda)-(r_{j+1}+t_{j+1})s_{j+1}(\lambda)+t_{j+1}s_j(\lambda)\\ &\quad -(r_j s_{j+1}(\lambda)-(r_j+t_j)s_j(\lambda)+t_j s_{j-1}(\lambda))\\ &= r_{j+1}s_{j+2}(\lambda)-(r_{j+1}+t_{j+1}+r_j)s_{j+1}(\lambda)\\ &\quad +(t_{j+1}+r_j+t_j)s_j(\lambda)-t_j s_{j-1}(\lambda)\\ &= r_{j+1}(s_{j+2}(\lambda)-s_{j+1}(\lambda))-(t_{j+1}+r_j)(s_{j+1}(\lambda)-s_j(\lambda))\\ &\quad +t_j(s_j(\lambda)-s_{j-1}(\lambda)).\end{aligned}$$

Substituting the definition of $v_j(\lambda)$ from above and simplifying completes the proof.

The recurrence relation for $v_j(\lambda)$ yields a similar relation for the residual $e_{m+1}(x) = 1 - q^*_{m+1}(x)$

$$e_{j+1}(\lambda) = -\frac{t_j v_{j-1}(0)}{r_{j+1} v_{j+1}(0)} e_{j-1}(\lambda) + \frac{(t_{j+1}+r_j)}{r_{j+1}} \frac{v_j(0)}{v_{j+1}(0)} e_j(\lambda) - \lambda \frac{v_j(0)}{r_{j+1} v_{j+1}(0)} e_j(\lambda).$$

A further straightforward computation allows us to identify the parameters of the inner iteration (3.5) as

$$\omega'_{j+1} = \frac{t_{j+1}+r_j}{r_{j+1}} \frac{v_j(0)}{v_{j+1}(0)}, \qquad \alpha'_j = \frac{1}{t_{j+1}+r_j}.$$

Lemma 2 allows for the recursive computation of ω'_j by means of the formula

$$\omega'_{j+1} = \frac{1}{1 - \dfrac{t_j r_j}{\alpha'_j \alpha'_{j-1}} \omega'_j}.$$

In particular for the weight function $w(\lambda) = 1$, where $\lambda \in [\lambda_1, \lambda_n]$ we have,

$$s_j(\lambda) = \frac{P_j(\mu)}{P_j(\mu_0)},$$

where P_j is the jth Legendre polynomial and

$$\mu = -1 + \frac{2(\lambda - \lambda_1)}{(\lambda_n - \lambda_1)}, \qquad \mu_0 = -\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}.$$

Setting $\tau = 2 - \mu_0$ and using the recurrence relation for the Legendre polynomials

$$\lambda P_j(\lambda) = \frac{j+1}{2j+1} P_{j+1}(\lambda) + \frac{j}{2j+1} P_{j-1}(\lambda)$$

yields in this case

$$r_j = \frac{(\lambda_n - \lambda_1)(j+1)}{2(2j+1)} \delta_j, \qquad t_j = \frac{j(\lambda_n - \lambda_1)}{2(2j-1)} \delta_{j-1}^{-1},$$

where δ_j is defined recursively by

$$\delta_j = \frac{2j+1}{j+1} \tau - \frac{j}{j+1} \delta_{j-1}^{-1}$$

with $\delta_0 = \tau$.

7. Summary. We now summarize some of the advantages of the new family of parameterized algorithms which we call PPCG(m), conjugate gradient with mth degree polynomial preconditioner. PPCG(m) has the following advantageous properties:

i) Given an estimate of the spectral radius or of the maximum and minimum eigenvalues of the matrix G the optimal parameters $(\gamma_0, \gamma_1, \cdots, \gamma_m)$ may easily be computed for any of the optimization techniques above.

ii) Given the maximum and minimum eigenvalues of the matrix G or appropriate estimates, the condition number of the approximation to $M^{-1}A$ may be computed and an a priori bound on the convergence of the algorithm readily obtained from Theorem 1.

iii) The algorithm is fully vectorizable using the matrix multiplication via diagonals schema presented in [5]. Furthermore, the algorithm is oriented to long vector operations extending over the total number of grid points in the discretization.

iv) The algorithm may be easily programmed to conserve storage space since the fully expanded approximation to $M^{-1}A$ need not be computed and stored. The multiplication by the approximation to $M^{-1}A$ at each iteration may be realized as a sequence of Jacobi iterations.

Appendix. The parameters $\gamma_0, \cdots, \gamma_m$ can be computed in a variety of ways. For the min–max case, the recursive formulae of § 6 can of course be used. In the case $\lambda_1 = 1-\beta$, $\lambda_n = 1+\beta$ the solution for the case $m = 1$ is

$$\gamma_0 = \frac{2}{2-\beta^2} \quad \text{and} \quad \gamma_1 = \frac{2}{2-\beta^2}$$

which for $\beta = 1$ reduces to $\gamma_0 = 1$ and $\gamma_1 = 1$.

The solution for the case $m = 2$ is

$$\gamma_0 = 1, \quad \gamma_1 = \frac{4}{4-3\beta^2}, \quad \gamma_2 = \frac{4}{4-3\beta^2},$$

which for $\beta = 1$ reduces to $\gamma_0 = 1$, $\gamma_1 = 4$ and $\gamma_2 = 4$.

The solution for the case $m = 3$, is

$$\gamma_0 = \frac{8-8\beta^2}{8-8\beta^2+\beta^4}, \qquad \gamma_1 = \frac{8-8\beta^2}{8-8\beta^2+\beta^4},$$

$$\gamma_2 = \frac{8}{8-8\beta^2+\beta^4}, \qquad \gamma_3 = \frac{8}{8-8\beta^2+\beta^4},$$

which for $\beta = 1$ reduces to $\gamma_0 = 0$, $\gamma_1 = 0$, $\gamma_2 = 8$ and $\gamma_3 = 8$.

The parameters for the optimal least squares polynomial can also be computed directly. The continuous least squares formulation of (5.1) requires minimization of the definite integral

$$\int_{\lambda_1}^{\lambda_n} [\gamma_0 + (\gamma_1 - \gamma_0)y + (\gamma_2 - \gamma_1)y^2 + \cdots + (\gamma_m - \gamma_{m-1})y^m - \gamma_m y^{m+1} - 1]^2 \, dy.$$

This reduces to the Toeplitz system of linear equations

$$T\bar{\gamma} = 1,$$

where for $i = 0, 1, 2, \cdots, m$ and $j = 0, 1, 2, \cdots, m$

$$T_{i,j} = \int_{\lambda_{\min}}^{\lambda_{\max}} y^k (1-y)^2 \, dy \quad \text{where } k = m + i - j,$$

$$t_i = \int_{\lambda_{\min}}^{\lambda_{\max}} y^i (1-y) \, dy, \qquad \bar{\gamma} = (\gamma_m, \gamma_{m-1}, \cdots, \gamma_0)^T.$$

This system of Toeplitz equations was solved for the cases $m = 1$, 2 and 3, in which, as above

$$\lambda_1 = 1-\beta, \qquad \lambda_n = 1+\beta.$$

The solution for the case $m = 1$ is

$$\gamma_0 = -\frac{\beta^2 - 15}{3\beta^4 - 6\beta^2 + 15} \quad \text{and} \quad \gamma_1 = -\frac{5\beta^2 - 15}{3\beta^4 - 6\beta^2 + 15},$$

which for $\beta = 1$ reduces to $\gamma_0 = 7/6$ and $\gamma_1 = 5/6$.

The solution for the case $m = 2$ is

$$\gamma_0 = +\frac{60\beta^4 - 165\beta^2 + 175}{9\beta^6 + 45\beta^4 - 165\beta^2 + 175},$$

$$\gamma_1 = -\frac{15\beta^4 + 60\beta^2 - 175}{9\beta^6 + 45\beta^4 - 165\beta^2 + 175} \quad \text{and}$$

$$\gamma_2 = -\frac{105\beta^2 - 175}{9\beta^6 + 45\beta^4 - 165\beta^2 + 175}$$

which for $\beta = 1$ reduces to: $\gamma_0 = 35/32$, $\gamma_1 = 50/32$ and $\gamma_2 = 35/32$.

The solution for the case $m = 3$ is

$$\gamma_0 = +\frac{30\beta^6 + 1281\beta^4 + 3220\beta^2 + 2205}{45\beta^8 - 180\beta^6 + 1470\beta^4 - 3220\beta^2 + 2205},$$

$$\gamma_1 = -\frac{210\beta^6 - 1617\beta^4 + 3220\beta^2 - 2205}{45\beta^8 - 180\beta^6 + 1470\beta^4 - 3220\beta^2 + 2205},$$

$$\gamma_2 = -\frac{147\beta^4 + 1330\beta^2 - 2205}{45\beta^8 - 180\beta^6 + 1470\beta^4 - 3220\beta^2 + 2205} \quad \text{and}$$

$$\gamma_3 = +\frac{189\beta^4 - 1890\beta^2 + 2205}{45\beta^8 - 180\beta^6 + 1470\beta^4 - 3220\beta^2 + 2205}$$

which for $\beta = 1$ reduces to $\gamma_0 = 37/40$, $\gamma_1 = 49/40$, $\gamma_2 = 91/40$ and $\gamma_3 = 63/40$.

Acknowledgment. We wish to acknowledge that the idea of using Lemma 1 arose from a conversation with K. Scherer of the University of Bonn.

REFERENCES

[1] P. CONCUS, G. H. GOLUB AND D. P. O'LEARY, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in Sparse Matrix Computations, J. R. Bunch and D. J. Rose, eds., Academic Press, New York, 1976.

[2] J. W. DANIEL, *The Approximate Minimization of Functionals*, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[3] P. F. DUBOIS, A. GREENBAUM AND G. H. RODRIGUE, *Approximating the inverse of a matrix for use in iterative algorithms on vector processors*, Computing, 22 (1979), pp. 257–268.

[4] O. G. JOHNSON, C. A. MICCHELLI AND G. PAUL, *Polynomial preconditioners for conjugate gradient calculations*, Res. Rep. RC-9208, IBM T. J. Watson Research Center, Yorktown Heights, NY, January 1982.

[5] N. K. MADSEN, G. H. RODRIGUE AND J. I. KARUSH, *Matrix multiplication by diagonals on a vector/parallel processor*, Inform. Proc. Lett., 5, (1976), pp. 41–45.

[6] J. A. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comput. 31 (1977), pp. 148–162.

[7] C. A. MICCHELLI, *A characterization of M. W. Wilson's criterion on non-negative expansions of orthogonal polynomials*, Proc. Amer. Math. Soc., 71 (1978), pp. 69–72.

[8] H. RUTISHAUSER, *Theory of gradient methods*, Mitteilungen aus dem Institut für Angewandte Mathematik No. 8 (1979), pp. 24–49.

[9] G. SZËGO, *Orthogonal Polynomials*, Colloquium Publications 23, Revised Edition, American Mathematical Society, Providence, RI, 1959.