JOURNAL OF COMPUTATIONAL PHYSICS 97, 127–143 (1991)

# Conjugate Gradient Methods for the Solution of Boundary Integral Equations on a Piecewise Smooth Boundary

K. CHEN*

*Department of Mathematics, University of Reading, Whiteknights, P.O. Box 220, Reading RG6 2AX, Berkshire, England*

Received November 30, 1989; revised August 14, 1990

The conjugate gradient methods (CGMs) have been successfully applied to solve the complex matrix equations arising from discretization of boundary integral equations. If the underlying integral operator is compact, its eigenvalue clustering property ensures the fast convergence of these methods. Such an integral operator is usually compact if the integral boundary is globally smooth. In this paper however, we consider the numerical solution of the boundary integral equation with a non-compact operator where the non-compactness is due to the non-smoothness of a pieceiwe smooth boundary. Two particular algorithms are presented and tested for a model problem. We show that such non-compact integral equations can be solved efficiently by the preconditioned conjugate gradient method and that the algorithm using the normal equation appears to be particularly efficient. 

CONTENTS



## 1. INTRODUCTION

In this paper we study the problem of the numerical solution of boundary integral equations on piecewise smooth boundaries by the preconditioned conjugate gradient methods. As is well known, many boundary value problems of partial differential equations may be advantageously reformulated as boundary integral equations over the finite boundary of the domain of interest once a fundamental solution of the differential equation is available. Refer to [14, 21, 24, 26]. These

* Permanent address: Department of Applied Mathematics, Dalian University of Technology, Dalian 116024, China.



0021-9991/91 $3.00

boundary integral equations are subsequently solved by boundary element methods. The numerical theory on solving boundary integral equations, usually of the second kind, is quite satisfactory provided that the underlying boundary is globally smooth. Refer to [5, 9, 11] and the references therein.

The discretization of a second kind integral equation by numerical methods usually produces a large linear system with a full and complex coefficient matrix. Suppose that the size of the system is $N$. Then the solution by direct methods such as Gaussian elimination requires $O(N^3)$ floating point operations. Recently iterative methods have been developed and successfully implemented for fast solution of the linear system, reducing the cost to $O(N^2)$ operations. In most of these iterative methods, including the multigrid type methods and the conjugate gradient methods, the underlying integral operator is assumed to be compact in some Banach space. The assumption can be in general justified if the boundary is smooth. Refer to [1, 2, 4, 16, 18, 26]. In particular, the simple two-grid methods of Atkinson [4] have been found extremely efficient for problems with a smooth boundary (see [1]).

For the reformulation of the Helmholtz equation, we can establish the smoothness and the compactness of all relevant integral operators if the boundary is sufficiently smooth. See [23]. These two properties are essential in determining the efficiency of numerical methods and developing efficient iterative methods.

However, when the boundary of interest admits corners and edges (i.e., has geometric singularities), the underlying integral operator is non-compact and the solution of the integral equation will usually possess similar non-smooth behaviour near such singularities. See [6, 12, 24]. For the numerical solution of such integral equations, [3, 10] have recently studied collocation methods and established a convergence analysis.

But the generalization of iterative methods for solving non-compact integral operator equations is not yet complete. Multigrid methods have been devised for such equations in the recent works of [17, 27, 28], ensuring fast convergence. However, their methods essentially require an inversion of a full matrix of smaller size $r \times r$, where $r$ ($< N$) is an integer depending on $N$. Such methods are therefore expected to be faster than direct methods. But the efficiency of these methods cannot be maintained either theoretically or pracitcally at the level of $O(N^2)$ operations, the ideal multigrid efficiency (refer to [1, 16, 18]). Improvements upon existing multigrid methods have been recently made in [8, 13]. As far as we are aware, the application of the conjugate gradient type methods for such equations has not been reported.

In general conjugate gradient methods (CGMs) as surveyed in [25] can be applied to any linear system with complex non-Hermitian coefficient matrix. But the overall efficiency is generally at the same level of $O(N^3)$ as for direct methods. When the linear system is from discretization of the second kind boundary integral equation with a compact operator, we have established in [2] that the fast convergence of conjugate gradient methods has the property of "fixed step" termination given a required tolerance. By "fixed step" we mean that the number of iterations required to achieve a desirable accuracy is fixed and independent of $N$—the dimen-

sion of the underlying discrete problem. There we use the eigenvalue clustering property of compact operators.

It is not a trivial matter to generalize the results on CGMs to the case of solving non-compact integral operator equations. However, since most discrete operators approximating the underlying non-compact operator are in general compact, we should be able to observe the eigenvalue clustering pattern of a given matrix of the linear system and hence the good performance of the CGMs as applied to the linear system. Refer to [11, Chap. 6] and Section 6 later, for an example. But unlike the case with compact operators, such clustering patterns will vary as the size of the linear system changes and they do not asymptotically converge to a simple fixed pattern. This implies that we may not observe the "fixed step" termination property.

Nevertheless, we shall show that for solving such non-compact integral operator equations, it is always possible to find a good preconditioner which will help to cluster the eigenvalues of discrete operators to some "fixed" pattern. With such a preconditioner for the linear system, the CGMs can achieve a practical efficiency of $O(N^2)$ operations.

In Section 2, we discuss the reformulation of the Helmholtz equation by considering it with the Dirichlet boundary condition in an exterior infinite domain with a non-smooth boundary. In Section 3, we solve the non-compact integral operator equation arising from the reformulation of Section 2 using the collocation method. We present in Section 4 two conjugate gradient algorithms for solving the subsequent complex matrix equations. In Section 5, we discuss the eigenvalue redistributions by preconditioning in order to accelerate the convergence of the CGMs. Numerical experiments on the algorithms are carried out in Section 6 and conclusions are given in Section 7.

## 2. Reformulation of the Helmholtz Equation

To describe briefly the boundary integral equation reformulations, we consider the two-dimensional Helmholtz equation in a domain $E$ exterior to the closed boundary $S$

$$(\nabla^2 + k^2)\,\phi(p) = 0, \qquad p \in E, \tag{1}$$

with the boundary conditions

$$\begin{aligned} \phi(p) &= f(p), && p \in S, \\ \lim_{r \to \infty} r^{1/2} \left\{ \frac{\partial \phi}{\partial r} - ik\phi(p) \right\} &= 0, && p \in E, \end{aligned} \tag{2}$$

where $r = |p|$. The solution $\phi(p)$ of Eq. (1) may be represented by $\mathscr{H}_k \sigma$, the hybrid layer potential of the single and double layers, where

$$(\mathscr{H}_k \sigma)(p) = \int_S \left\{ \frac{\partial G_k}{\partial n_q} - \eta G_k \right\} \sigma(q)\, d_{S_q}, \qquad p \in E, \tag{3}$$

where $\sigma(p)$ is an unknown density function, $n_q$ is the unit normal directed into $E$, $G_k(p, q)$ is the fundamental solution for the Helmholtz equation, and $\eta$ is a non-zero coupling parameter required to ensure the uniqueness of $\sigma(p)$ for any $k$. By imposing the boundary conditions (2), this yields the second kind integral equation

$$\frac{\chi(p)}{2}\sigma(p)+\int_S\left\{\frac{\partial G_k}{\partial n_q}(p, q)-\eta G_k(p, q)\right\}\sigma(q)\,dS_q=f(p), \qquad p\in S, \tag{4}$$

where $\chi(p)\pi$ is the internal angle at point $p$ between the two tangents (note that $\chi(p)=1$ at all smooth points). Here for moderately large $k$, we choose $\eta=k\mathbf{i}/2$ as this choice can be shown to improve the conditioning of Eq. (4) (see [2, 22]). The fundamental solution in 2D is given by the zero-order Hankel function of the first kind

$$G_k(p, q)=\frac{i}{4}\mathbf{H}_0^{(1)}(kr), \qquad r=|p-q|, \tag{5}$$

where a $j$th-order Hankel function of the first kind is defined by

$$\mathbf{H}_j^{(1)}(x)=\mathbf{J}_j(x)+i\mathbf{N}_j(x),$$

with $j$ an integer or zero, $\mathbf{J}_j$ and $\mathbf{N}_j$ the $j$th-order Bessel functions of the first kind and the second kind, respectively.

We are especially interested in the case where the boundary $S$ is only piecewise smooth. Without essential loss of generality, we now consider the boundary $S$ as shown in Fig. 1, having a single corner point at $o$ and comprising of two straight

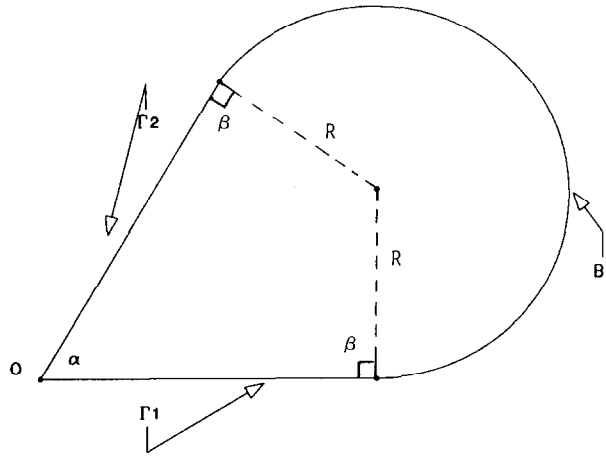


FIG. 1. 2D boundary curve $S$ with a corner at point $o$: (1) $B$ is smooth everywhere; (2) $\beta=\pi/2=90°$.

lines $\Gamma_1$ and $\Gamma_2$ and a smooth curve $B$. Here the angle between $\Gamma_1$ and $\Gamma_2$ is denoted by $\alpha$ (refer to [6, 12, 24] for a similar example). Equation (4) can be written as

$$\frac{\sigma(p)}{2} - \frac{i}{4}\int_S \{\eta \mathbf{H}_0^{(1)}(kr) + k\mathbf{H}_1^{(1)}(kr)\,\mathrm{Grad}(r)\cdot n_q\}\,\sigma(q)\,d_{S_q} = f(p), \qquad p \in S/\{0\}. \tag{6}$$

Then with such a specific choice of $S = \Gamma_1 \cup B \cup \Gamma_2$, we may expand Eq. (6) using the expansions of the Hankel functions to separate the most singular integral; refer to [11, Chap. 5; 12; 24].

Further it can be shown that Eq. (6) may be characterized by the following prototype equation

$$(\mathscr{I} - \mathscr{K} - \mathscr{L})u = g, \qquad 0 \leqslant s \leqslant 1, \tag{7}$$

where $\mathscr{K}$: $C^0[0, 1] \to C^0[0, 1]$ is non-compact, defined by

$$(\mathscr{K}\psi)(s) = \begin{cases} \dfrac{\pi - \alpha}{\pi}\psi(0), & s = 0, \\ \displaystyle\int_0^1 K\left(\frac{s}{t}\right)\psi(t)\frac{dt}{t}, & 0 < s \leqslant 1, \end{cases} \tag{8}$$

with the kernel $K(t) = (\sin\alpha)(t/(1 + t^2 - 2t\cos\alpha))/\pi$, and the operator $\mathscr{L}$ defined by $(\mathscr{L}v)(s) = \int_0^1 L(s, t)\,v(t)\,dt$ is compact with its kernel $L(s, t)$ obviously less singular than the kernel $K(s/t)/t$ of $\mathscr{K}$. Refer to [10; 11, Chap. 5].

Using the local Mellin transforms, we have shown in [12] that the solution to Eq. (7) (or Eq. (6)) near the corner $o$ of the boundary has the asymptotic form

$$u(s) = u(0) + As^{\beta} + Y(s) \tag{9}$$

where $s$ is the arc length measured from the corner and $As^{\beta}$ denotes the leading singular form in the expansion with $\beta = \min\{\pi/\alpha, \pi/(2\pi - \alpha)\}$. Therefore one natural space setting for the solution of (7) with the form (9) may be defined by

$$C_{\beta}^{r} = \{x: \|x(s) - x(0)\|_{r,\beta} < \infty\}$$

with the norm

$$\|y\|_{r,\beta} = \max_{0 \leqslant \nu \leqslant r} \sup_{0 < s \leqslant 1} |s^{\nu - \beta} D^{\nu} y(s)|,$$

where $D = d/ds$ is the 1D differential symbol.

## 3. Numerical Solution of Non-smooth Operator Equations

We now discuss the numerical solution of Eq. (7) by collocation methods based on piecewise polynomials, following the work of [10]. This type of methods is

related to engineering applications (see [9]). However, other numerical approaches such as the Nyström method may also be considered (refer to [7]).

To introduce a piecewise polynomial space, let us define a mesh

$$\Pi_n: 0=\eta_0<\eta_1<\cdots<\eta_n=1 \tag{10}$$

over the interval $[0, 1]$. For $1\leqslant i\leqslant n$, we set $I_i=(\eta_{i-1}, \eta_i]$, $h_{(i)}=\eta_i-\eta_{i-1}$, and $h_n=\max_{1\leqslant j\leqslant n} h_{(j)}$. Then denote by $S_i^{n,r}$ the space of piecewise polynomials of order 1 (i.e., degree 0) on each of the first $i$ subintervals $I_{i'}$, $i'=1, ..., i$, and order $r$ (i.e., degree $r-1$) on each of the last $n-i$ subintervals $I_{i'}$, $i'=i+1, ..., n$, where $0\leqslant i<n$. Denote the mid-points of subintervals $I_{i'}$ $(i'=1, ..., i)$ by

$$s_{i'1}^n=(\eta_{i'-1}+\eta_{i'})/2$$

and $r$ distinct points in subintervals $I_{i'}$ $(i'=i+1, ..., n)$ by

$$s_{i'j}^n=\eta_{i'-1}+\xi_j h_{(i')}, \qquad 1\leqslant j\leqslant r,$$

where $\{\xi_j\}_1^r$ are the nodes of some quadrature rule on $[0, 1]$ with $0<\xi_1<\cdots<\xi_r<1$. Then for any function $v\in C^0[0, 1]$, we may approximate it by an interpolatory piecewise polynomial $v_n\in S_i^{n,r}$. In detail we write

$$v_n(s)=\begin{cases} X_{i'}(s)\, v(s_{i'1}^n), & s\in I_{i'}(i'\leqslant i),\ s\in[0,1] \\ \sum_{j=1}^r l_{i'j}(s)\, v(s_{i'j}^n), & s\in I_{i'}(i'>i),\ s\in[0,1], \end{cases} \tag{11}$$

where for $s\in[0, 1]$,

$$X_{i'}(s)=\begin{cases} 1, & s\in I_{i'} \\ 0, & s\notin I_{i'} \end{cases}$$

$$l_{i'j}(s)=\begin{cases} \displaystyle\prod_{\substack{m=1\\ m\neq j}}^{r} \frac{s-s_{i'm}^n}{s_{i'j}^n-s_{i'm}^n}, & s\in I_{i'}, \\ 0, & s\notin I_{i'}. \end{cases}$$

For convenience, we shall write $\{s_j\}_1^{N_n}$ for the collection of nodes $\{s_{i'1}^n\}_1^i$ and $\{s_j^{ni'}\}_{i'=i+1}^n$, and $\{\psi_j(s)\}_1^{N_n}$ for the collection of basis functions $\{X_{i'}(s)\}_1^i$ and $\{l_{i'j}(s)\}_{i'=i+1}^n$ (after renumbering in sequential order). It is easy to see that $N_n=\dim(S_i^{n,r})=(n-i)r+i$. Therefore the space $S_i^{n,r}$ can be viewed as the span of $N_n$ independent basis functions $\{\psi_j\}$.

Using these nodes $\{s_j\}_1^{N_n}$, we may define a projection operator $\mathscr{P}_n$: $C^0[0, 1]+S_i^{n,r}\to S_i^{n,r}$ by

$$(\mathscr{P}_n v)(s)=\sum_{j=1}^{N_n} v(s_j)\,\psi_j(s), \qquad s\in[0,1], \tag{12}$$

where $v_n(s) = (\mathscr{P}_n v)(s)$ interpolates $v(s)$ at these nodes, i.e.,

$$(\mathscr{P}_n v)(s_j) = v(s_j), \qquad j = 1, ..., N_n.$$

Using operator notation, the collocation approximation $u_n \in S_i^{n,r}$ to solution $u$ of Eq. (7) is defined by

$$(\mathscr{I} - \mathscr{P}_n \mathscr{K} - \mathscr{P}_n \mathscr{L})\, u_n = \mathscr{P}_n g. \tag{13}$$

Further, the product integration approximation $u_n^* \in C^0[0, 1]$ may be defined by

$$(\mathscr{I} - \mathscr{K}\mathscr{P}_n - \mathscr{L}\mathscr{P}_n)\, u_n^* = g, \tag{14}$$

with the relation to $u_n$ given by

$$u_n = \mathscr{P}_n u_n^*$$

and having the equivalent definition

$$u_n^* = g + (\mathscr{K} + \mathscr{L}) u_n.$$

For an error analysis on $u_n$, we obtain from Eqs. (7) and (13)

$$u - u_n = (\mathscr{I} - \mathscr{P}_n(\mathscr{K} + \mathscr{L}))^{-1} (\mathscr{I} - \mathscr{P}_n) u$$

and then

$$\|u - u_n\| \leqslant \|(\mathscr{I} - \mathscr{P}_n \mathscr{K})^{-1}\| \cdot \|(\mathscr{I} - \mathscr{P}_n) u\|, \tag{15}$$

where the supremum norm is used. While for the error analysis of $u_n^*$, we have similarly from Eqs. (7) and (14)

$$u - u_n^* = (\mathscr{I} - (\mathscr{K} + \mathscr{L})\mathscr{P}_n)^{-1} (\mathscr{K} + \mathscr{L})(\mathscr{I} - \mathscr{P}_n) u$$

and, further, from the identity

$$(\mathscr{I} - (\mathscr{K} + \mathscr{L})\mathscr{P}_n)^{-1} = \mathscr{I} + (\mathscr{K} + \mathscr{L})(\mathscr{I} - \mathscr{P}_n(\mathscr{K} + \mathscr{L}))^{-1} \mathscr{P}_n,$$

we obtain

$$\|u - u_n^*\| \leqslant (c_1 + c_2 \|(\mathscr{I} - \mathscr{P}_n(\mathscr{K} + \mathscr{L}))^{-1}\|) \cdot \|(\mathscr{K} + \mathscr{L})(\mathscr{I} - \mathscr{P}_n) u\|, \tag{16}$$

where $c_j$'s (from now on) are generic constants independent of $n$ and the solution $u$.

The mesh $\Pi_n$ in (10) is generally non-uniform, the choice of which depends on the smoothness of the solution $u$. In our case, we define the graded mesh as

$$\eta_j = \left(\frac{j}{n}\right)^q, \qquad j = 0, 1, ..., n, \tag{17}$$

with the integer $q \geqslant 1$. Here $q = 1$ defines the uniform mesh.

THEOREM 1. *Let a mesh $\Pi_n$ be as set up in* (17).

(1) *If $u \in C_\beta^r$ and the exponent $q \geqslant r/\beta$, then there exists an $i = i_1$ such that $u_n \in S_{i_1}^{n,r}$ satisfies*

$$\|u - u_n\| = O(n^{-r});$$

(2) *If $u \in C_\beta^{2r}$, the exponent $q \geqslant 2r/\beta$ and the quadrature points $\{\xi_j\}$ are chosen to be the Gauss–Legendre nodes shifted to $[0, 1]$, then there exists an $i = i_2$ such that based on the space $S_{i_2}^{n,r}$, $u_n^*$ satisfies*

$$\|u - u_n^*\| = O(n^{-2r}).$$

*Proof.* The proof follows from [10]; there exists an $i = i^*$ such that $\|(\mathscr{I} - \mathscr{P}_n\mathscr{K})^{-1}\| \leqslant c_3$. It is then easy to prove the stability result in (15) and (16) (i.e., $\|(\mathscr{I} - \mathscr{P}_n(\mathscr{K} + \mathscr{L}))^{-1}\| \leqslant c_4$). Refer to [11, Chap. 5]. ∎

To solve Eq. (13), we collocate at the nodes $s_j$, $j = 1, ..., N_n$, giving a linear system of size $N_n$,

$$A_n \mathbf{u}_n = \mathbf{g} \qquad \text{with} \quad A_n = I - B_n, \tag{18}$$

where

$$\left.\begin{aligned} (B_n)_{ij} &= \int_0^1 [K(s_i, t) + L(s_i, t)]\, \psi_j(t)\, dt, \qquad j = 1, 2, ..., N_n, \\ (\mathbf{u}_n)_i &= u_n(s_i), \\ (\mathbf{g})_i &= g(s_i), \end{aligned}\right\} \; i = 1, ..., N_n.$$

Then the collocation solution is given by $u_n(s) = \mathscr{P}_n u = \sum_{j=1}^{N_n} (\mathbf{u}_n)_j \psi_j(s)$ and the product integration approximation by $u_n^*(s) = g(s) + (\mathscr{K} + \mathscr{L})\, u_n(s)$.

Now consider the important problem of accurately evaluating quantities $(B_n)_{ij}$ by numerical integration. Since basis functions $\psi_j$'s are piecewise, we can rewrite $(B_n)_{ij}$ in the form (assume that the piecewise polynomial space $S_{i^*}^{n,r}$ with $i^* \geqslant 0$ is used)

$$\begin{aligned} (B_n)_{ij} = &\sum_{m=1}^{i^*} \int_{\eta_{m-1}}^{\eta_m} [K(s_i, t) + L(s_i, t)]\, dt \\ &+ \sum_{m=i^*+1}^{n} \int_{\eta_{m-1}}^{\eta_m} [K(s_i, t) + L(s_i, t)]\, \psi_j(t)\, dt, \end{aligned} \tag{19}$$

where $\psi_j$ is the Lagrange interpolation basis function; refer to (11). Further, making use of the specific form of kernel $K$, it is possible to evaluate anaytically the integrals in (19) involving $K$; refer to [10]. For the remaining integrals in (19)

involving the smooth kernel $L$, standard integration rules such as the Gauss–Legendre rule may be applied to obtain numerical approximations; refer to [11, Chap. 1] and the references therein for a general discussion.

*Remark.* The inclusion of a parameter $i$ in defining the piecewise polynomial space is for the sole purpose of proving the stability theoretically. In practice, however, the choice of $i=0$ is often acceptable unless numerical solvers detect any instability of the problem. In Section 5, we shall use the choice of $i>1$ even if the problem is stable, as this does not affect the overall accuracy.

## 4. Conjugate Gradient Methods

To solve Eq. (18) iteratively, we shall now discuss the application of the conjugate gradient methods (CGMs). Let us start with the basic version of the CGM. Suppose that a linear system of size $N$,

$$A\mathbf{x}=\mathbf{b}, \tag{20}$$

possesses a real symmetric positive definite matrix $A$. Then the CGM for solving Eq. (20) is based on the equivalence of solving the linear system and minimizing the quadratic functional

$$\phi(\mathbf{x})=\tfrac{1}{2}\mathbf{x}^{\mathrm{T}}A\mathbf{x}-\mathbf{x}^{\mathrm{T}}\mathbf{b}.$$

That is, the minimum value of $\phi$ is $-\frac{1}{2}\mathbf{b}^{\mathrm{T}}A^{-1}\mathbf{b}$, achieved by setting $\mathbf{x}\equiv\mathbf{x}^{*}=A^{-1}\mathbf{b}$. See [15, Chap. 10] and the references therein for more discussions.

The basic CGM algorithm of solving Eq. (20) may be given below, where the initial guess $\mathbf{x}=\mathbf{x}^{(0)}=0$.

Algorithm 0.

(0) Set $\mathbf{x}:=\mathbf{p}:=0$, $c_1:=1$, $\mathbf{r}:=\mathbf{b}$ and input TOL;
(1) Compute $c_2:=\|\mathbf{r}\|_2^2$, $E:=\sqrt{c_2}$;
(2) If $E/\|\mathbf{b}\|_2\leqslant \mathrm{TOL}$, then terminate with solution in $\mathbf{x}$;
(3) Compute $\beta:=c_2/c_1$ and set $c_1:=c_2$;
(4) Modify $\mathbf{p}:=\mathbf{r}+\beta\mathbf{p}$;
(5) Compute $\mathbf{q}:=A\mathbf{p}$;
(6) Compute $\alpha:=c_2/\mathbf{p}^{\mathrm{T}}\mathbf{q}$;
(7) Modify $\mathbf{x}:=\mathbf{x}+\alpha\mathbf{p}$;
(8) Modify $\mathbf{r}:=\mathbf{r}-\alpha\mathbf{q}$;
(9) Return to step (1).

An important property of the CGM is the analogy between CG iterations and a polynomial curve fitting, which is usually exploited in deriving error bounds for the CGM. See [19]. The classic result for the convergence of the CGM states

$$E(\mathbf{x}^{(k)}) \leqslant 4\left[\frac{\sqrt{C}-1}{\sqrt{C}+1}\right]^{2k} \cdot E(\mathbf{x}^{(0)}), \qquad k = 1, 2, ..., \tag{21}$$

where $\mathbf{x}^{(k)}$ $(k \geqslant 0)$ denotes the $k$th iterate, $C$ is the 2-norm condition number of $A$, and $E(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^*\|_A = (\mathbf{x} - \mathbf{x}^*)^{\mathrm{T}} A(\mathbf{x} - \mathbf{x}^*)$ is the $A$-norm measure of the error. The error bound in (21) is generally quite pessimistic and has been improved in [2] in special cases, as stated below.

THEOREM 2. *Let $A$ be a $N \times N$ real symmetric positive definite matrix, where most of its eigenvalues cluster in region $[u, v]$ with $C_0 = v/u \ll C$ and only a few eigenvalues lie outside the region. Then applying the CGM to solve $A\mathbf{x} = \mathbf{b}$, we have*

$$E(\mathbf{x}^{(k_0+k)}) \leqslant 4\left[\frac{\sqrt{C_0}-1}{\sqrt{C_0}+1}\right]^{2k} \cdot E(\mathbf{x}^{(0)}), \qquad k = 1, 2, ..., \tag{22}$$

*where $k_0$ is a small positive integer, only depending on the number of eigenvalues of $A$ outside $[u, v]$.*

Results that have been discussed so far in the section, for real symmetric positive definite matrices $A$, are also valid for Hermitian positive definite matrices $A$, provided that the transpose of a vector $\mathbf{y}^{\mathrm{T}}$ is replaced by a complex conjugate transpose $\mathbf{y}^*$ and that the new $A$-norm for a complex vector $\mathbf{y}$ is defined by $\|\mathbf{y}\|_A = \mathbf{y}^* A\mathbf{y}$. But in practice, matrices $A$ such as in Eq. (18) are generally non-Hermitian. We now present two treatments of such matrices, leading to the application of the CGMs. Suppose that our linear system is again as given in (20), where $A$ is a non-Hermitian matrix.

The first approach, quite well known, is to use the normal equation, i.e., to solve instead of Eq. (20),

$$A^*A\mathbf{x} = A^*\mathbf{b} \tag{23}$$

or

$$AA^*\mathbf{y} = \mathbf{b} \tag{24}$$

with $\mathbf{x} = A^*\mathbf{y}$, where $A^*$ is the conjugate transpose of $A$. Both matrices $A^*A$ and $AA^*$ are Hermitian positive definite. So the basic CGM is immediately applicable. An algorithm using Eq. (24) for solving $A\mathbf{x} = \mathbf{b}$ may be presented as follows, where the initial guess $\mathbf{x} = \mathbf{x}^{(0)} = 0$.

ALGORITHM 1.

(0) Set $\mathbf{x} := \mathbf{p} := 0$, $c_1 := 1$, $\mathbf{r} := \mathbf{b}$ and input TOL;
(1) Compute $c_2 := \|\mathbf{r}\|_2^2$, $E := \sqrt{c_2}$;
(2) If $E/\|\mathbf{b}\|_2 \leqslant \text{TOL}$, then terminate with solution in $\mathbf{x}$;
(3) Compute $\beta := c_2/c_1$ and set $c_1 := c_2$;
(4) Modify $\mathbf{p} := \mathbf{r} + \beta\mathbf{p}$;
(5) Compute $\mathbf{q} := A^*\mathbf{p}$;
(6) Compute $\alpha := c_2/\|\mathbf{q}\|_2^2$;
(7) Modify $\mathbf{x} := \mathbf{x} + \alpha\mathbf{q}$;
(8) Modify $\mathbf{r} := \mathbf{r} - \alpha A\mathbf{q}$;
(9) Return to step (1).

The disadvantage of the algorithm is that the condition number of the original linear system is squared since $\text{cond}(A^*A) = \text{cond}(AA^*) = [\text{cond}(A)]^2$. In our application area, however, the condition number of $A$ and hence that of $AA^*$ are often not very large.

The second approach, called the augmented conjugate gradient method, is to augment the original linear system in order to form a new system with a Hermitian matrix. This is studied in [25]. For a linear system such as (20) with $A$ non-Hermitian, the augmented system takes the form

$$B\mathbf{T} = \mathbf{Z} \tag{25}$$

with $B = \begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix}$, $\mathbf{T} = \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix}$, $\mathbf{Z} = \begin{bmatrix} \mathbf{b} \\ \bar{\mathbf{b}} \end{bmatrix}$. Here $B$ is of order $2N \times 2N$, both $\mathbf{T}$ and $\mathbf{Z}$ are of order $2N \times 1$ and $\bar{\mathbf{b}}$ denotes the conjugate of $\mathbf{b}$. Now since $B$ in Eq. (25) is only Hermitian, not positive definite, the direct application of the basic CGM may not work. In [25], a step of minimizing residuals is proposed in each step of CG iterations. Hence we find a least-squares solution. The modified CGM may be presented in the following algorithm, where complex vectors $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$, $\mathbf{T}$, $\tilde{\mathbf{T}}$ are of order $2N \times 1$, only the second half of $\tilde{\mathbf{T}}$ is required and the initial guess $\mathbf{T} = \mathbf{T}^{(0)} = 0$.

ALGORITHM 2.

(0) Set $\tilde{\mathbf{T}} := \mathbf{T} := 0$, $\mathbf{p} := \mathbf{r} := \begin{bmatrix} \mathbf{b} \\ \bar{\mathbf{b}} \end{bmatrix}$ and input TOL;
(1) Compute $c_2 := \|\mathbf{p}\|_2^2$, $E := \sqrt{c_2}$;
(2) If $E/(2\|\mathbf{b}\|_2) \leqslant \text{TOL}$, then terminate with solution in the second half of $\tilde{\mathbf{T}}$;
(3) Compute $\mathbf{q} := B\mathbf{p}$;
(4) Compute $\alpha := c_2/\mathbf{q}^*\mathbf{p}$;
(5) Modify $\mathbf{T} := \mathbf{T} + \alpha\mathbf{p}$;
(6) Modify $\mathbf{r} := \mathbf{r} - \alpha\mathbf{q}$;
(7) Modify $\tilde{\mathbf{T}} := (\mathbf{T} + \beta\tilde{\mathbf{T}})/(1 + \beta)$;
(8) Modify $\mathbf{p} := (\mathbf{r} + \beta\mathbf{p})/(1 + \beta)$;
(9) Return to step (1).

As discussed in [25], this algorithm (coded there as GMCG) is more reliable than the biconjugate gradient method (which is a method specially designed for solving non-Hermitian matrix equations) and it also has the advantage over Algorithm 1 in that the condition number of the original system is not squared. The above introduced Algorithms 1 and 2 will be illustrated in Section 6.

## 5. Eigenvalue Redistribution by Preconditioning

As our prime purpose is to study the solution of Eq. (18) by the CGMs, we shall now show how the convergence may be increased by preconditioning. From the last section, we know that the convergence of a CGM is fast if *either* the underlying condition number of the coefficient matrix is small *or* the eigenvalues of the coefficient matrix cluster together.

In the literature, many existing preconditioners are designed for improving alone the conditioning of a sparse matrix (see [15, Chap. 10; 20, 29] and the references therein). In [2], we note that the CGMs can be very efficient without preconditioning when the eigenvalues of a linear system approximate some fixed clustering pattern. There such a pattern is ensured by the presence of compact integral operators. Here for solving a class of non-compact integral operator equations, we propose to precondition the discrete linear systems so as to find a fixed clustering pattern and hence speed up the convergence of the CGMs.

Recall that the discrete operator equation (13) originating from Eq. (7) yields the linear system Eq. (18). The presence of non-compact $\mathscr{K}$ causes the asymptotic irregularity in clustering patterns of eigenvalues of matrices $A_n$'s. Now let us separate a compact operator $\mathscr{K}_1$ from $\mathscr{K}$ by the splitting

$$\mathscr{K} = \mathscr{K}_\delta + \mathscr{K}_1, \tag{26}$$

where

$$(\mathscr{K}_\delta v)(s) = \begin{cases} \displaystyle\int_0^\delta K\left(\frac{s}{t}\right) v(t)\frac{dt}{t}, & 0 \leqslant s \leqslant \delta \\ 0, & \delta < s \leqslant 1, \end{cases} \tag{27}$$

$$\mathscr{K}_1 = \mathscr{K} - \mathscr{K}_\delta$$

with $\delta = (i_0/n_0)^q > 0$ for integers $i_0$, $n_0 \geqslant 1$ (to be specified). Suppose that $N^*$ is the largest order of linear systems one particular computer can handle. Then we choose

$$n^* = \frac{N^*}{r} \qquad \text{and} \qquad n_0 \geqslant n^*, \tag{28}$$

where $r$ is the order of a chosen quadrature rule (see Section 3). In view of Eq. (27), we set up the mesh on $[0, 1]$,

$$\Pi_n: 0 = \eta_0 < \eta_1 < \cdots < \eta_{i_1 - 1} = \eta_{i_1} < \cdots < \eta_n = 1$$

with $\eta_j = (j/n)^q$ for $j = 1, ..., n$, where

$$i_1 = \mathrm{INT}\left[\frac{i_0}{n_0} \cdot n\right] + 1 \qquad \text{and} \qquad \eta_{i_1 - 1} \leqslant \delta < \eta_{i_1}.$$

Hence, a piecewise polynomial space $S_{i_1}^{n,r}$ may be defined based on the mesh $\Pi_n$ (as in Section 3). On substituting Eq. (26) into Eq. (7) and then applying the collocation method, we obtain a discrete operator equation (similar to Eq. (13))

$$(\mathscr{I} - \mathscr{P}_n \mathscr{K}_\delta - \mathscr{P}_n \mathscr{K}_1 - \mathscr{P}_n \mathscr{L}) u_n = \mathscr{P}_n g. \tag{29}$$

Further collocating Eq. (29) at nodes $\{s_j\}_1^{N_n}$ yields a linear system, equivalent to Eq. (18),

$$(I_n - K_{\delta,n} - K_{1,n} - L_n)\, \mathbf{u}_n = \mathbf{g}, \tag{30}$$

where $I_n - K_{\delta,n} - K_{1,n} - L_n = A_n$. Now let us denote

$$D_n = I_n - K_{\delta,n}, \qquad C_n = K_{1,n} + L_n.$$

Then Eq. (30) may be written as

$$(D_n - C_n)\, \mathbf{u}_n = \mathbf{g}. \tag{31}$$

In order to solve Eq. (18), i.e., Eq. (31) by the CGMs, we propose to use $D_n^{-1}$ as a preconditioner and transform Eq. (31) into

$$\tilde{A}_n \mathbf{u}_n = \tilde{\mathbf{g}}, \tag{32}$$

where $\tilde{A}_n = I_n - D_n^{-1} C_n$ and $\tilde{\mathbf{g}} = D_n^{-1} \mathbf{g}$. It can be shown that the eigenvalues of $\tilde{A}_n$'s approximate a fixed clustering pattern asymptotically. This is because $\tilde{A}_n$ may be viewed as the collocation matrix corresponding to the discrete operator

$$\mathscr{I} - (\mathscr{I} - \mathscr{P}_n \mathscr{K}_\delta)^{-1} (\mathscr{P}_n \mathscr{K}_1 + \mathscr{P}_n \mathscr{L})$$

which approximates the operator $\mathscr{I} - (\mathscr{I} - \mathscr{K}_\delta)^{-1} (\mathscr{K}_1 + \mathscr{L})$ with $(\mathscr{I} - \mathscr{K}_\delta)^{-1} (\mathscr{K}_1 + \mathscr{L})$ compact for $\delta > 0$.

Note that in Eq. (31), $D_n$ is a $N_n \times N_n$ matrix of sparse structure

$$D_n = \begin{pmatrix} d_{i_1} & 0 \\ 0 & I_{N_n - i_1} \end{pmatrix},$$

where $d_{i_1}$ is an $i_1 \times i_1$ full matrix and $I_{N_n - i_1}$ is the $(N_n - i_1) \times (N_n - i_1)$ unit matrix. Therefore the complexity of computing $D_n^{-1}$ is $O(i_1^3)$, increasing as $i_1$ becomes larger. The choice of $i_1$ depends on that of $i_0$, i.e., $\delta$. We may use $i_1$ for fine tuning

Eq. (32). With the extreme choice of $i_1 = 1$, the matrix $D_n$ is a diagonal matrix, satisfying $(i, j = 1, ..., N_n)$

$$(D_n)_{ij} = 0 \quad \text{if} \quad i \neq j \quad \text{and} \quad (D_n)_{ii} = 1 \quad \text{if} \quad i > 1,$$

the inverse of which is given by

$$(D_n^{-1})_{11} = \frac{1}{(D_n)_{11}}, \qquad (D_n)_{ii} = 1 \quad \text{if} \quad i > 1 \text{ and } (D_n^{-1})_{ij} = 0 \quad \text{if} \quad i \neq j.$$

## 6. Numerical Experiments

In this section, we shall experiment on the CGMs of Section 4 and show how the preconditioning in Eq. (32) of Section 5 improves the performance of the CGMs. Following Eq. (6), we choose our test problem to be the boundary integral equation

$$\sigma(s) - \frac{i}{2}\int_0^1 \left\{\eta \mathbf{H}_0^{(1)}(kr) + \frac{ks \sin \alpha}{r}\mathbf{H}_1^{(1)}(kr)\right\}\sigma(s')\,ds' = g(s), \qquad 0 < s \leqslant 1, \quad (33)$$

with $\alpha = \pi/10$, $k = 5.0$, $\eta = 2.5i$, $r = \sqrt{s'^2 + s^2 - 2s's\cos\alpha}$, and $g(s)$ is found accurately so the solution is $\sigma(s) = s^{1/2}e^{2\pi i} = s^{1/2}$.

We shall solve Eq. (33) using the product integration collocation method (Section 3) with piecewise constant and piecewise linear basis functions. Suppose that the interval $[0, 1]$ is subdivided into $n$ small intervals with $0 = \eta_0 < \eta_1 < \cdots < \eta_{n-1} < \eta_n = 1$ as in (10) with the graded meshes in (17). Then in the case of piecewise constants ($r = 1$), we collocate at the midpoints of each interval, while in the case of piecewise linears ($r = 2$) we collocate at the two Gaussian points on each of the last $(n - i)$ intervals. The computing error with $n$ is defined by

$$E_n = \|\sigma_n - \sigma\| = \max\{\|\sigma_n^*(s_j) - \sigma(s_j)\|\}$$

TABLE I

Algorithm 1 for Piecewise Constant Approximations ($r = 1$)

| No preconditioning | | | | With preconditioning | | | |
|---|---|---|---|---|---|---|---|
| $q$ | Size $N$ | Steps | CPU | $q$ | Size $N$ | Steps | CPU |
| 1 | 36 | 7 | 0.3 | 1 | 36 | 5 | 0.3 |
| | 108 | 6 | 2.1 | | 108 | 5 | 1.9 |
| | 324 | 6 | 18.7 | | 324 | 5 | 16.3 |
| 4 | 36 | 21 | 0.9 | 4 | 36 | 16 | 0.8 |
| | 108 | 34 | 11.9 | | 108 | 19 | 7.1 |
| | 324 | 46 | 143.6 | | 324 | 20 | 65.8 |

TABLE II

Algorithm 2 for Piecewise Constant Approximations ($r = 1$)

| No preconditioning | | | | With preconditioning | | | |
|---|---|---|---|---|---|---|---|
| $q$ | Size $N$ | Steps | CPU | $q$ | Size $N$ | Steps | CPU |
| 1 | 36 | 12 | 0.5 | 1 | 36 | 9 | 0.4 |
| | 108 | 13 | 4.5 | | 108 | 9 | 3.0 |
| | 324 | 13 | 38.7 | | 324 | 9 | 26.0 |
| 4 | 36 | 49 | 2.2 | 4 | 36 | 33 | 1.5 |
| | 108 | 77 | 27.0 | | 108 | 37 | 12.4 |
| | 324 | 97 | 288.4 | | 324 | 41 | 119.0 |

over all collocation points $\{s_j\}_1^{N_n}$, where $\sigma_n^*$ represents the product integration solution (refer to (14)) and $N_n = (n-i)r + i$. Then the expected order for the error $E_n$ is $O(n^{-1/2})$ with $q = 1$ for both approximations, $O(n^{-2})$ with $q = 4$ for the piecewise constants, and $O(n^{-4})$ with $q = 8$ for the piecewise linears.

The number of iterations required to achieve a specified tolerance is denoted by "Steps" and the corresponding cpu seconds used by Prime-750 by "CPU." A direct solver requires CPU = 0.2, 5.0, and 130.0 respectively for solving the $36 \times 36$, $108 \times 108$, and $324 \times 324$ linear systems. To solve linear systems iteratively up to an accuracy comparable to the level of the discretization error, we choose the tolerance TOL as follows: (a) with $q = 1$ for both approximations, TOL $= 10^{-2}$ for any $N_n \leqslant 324$; (b) with $q = 4$ for the piecewise constants, TOL $= 10^{-3}$ for $N_n = 36$, TOL $= 10^{-4}$ for $N_n = 108$, and TOL $= 10^{-5}$ for $N_n = 324$; (c) with $q = 8$, TOL $= 10^{-4}$ for $N_n = 36$, TOL $= 10^{-5}$ for $N_n = 108$, and TOL $= 10^{-6}$ for $N_n = 324$.

In Tables I–IV, we present our numerical results from solving the Eq. (33) using Algorithms 1 and 2 as introduced in Section 4. Throughout the tables we choose $i \equiv 0$ in using the unpreconditioned CGMs. Here with the PRIME-750, we choose

TABLE III

Algorithm 1 for Piecewise Linear Approximations ($r = 2$)

| No preconditioning | | | | With preconditioning | | | |
|---|---|---|---|---|---|---|---|
| $q$ | Size $N$ | Steps | CPU | $q$ | Size $N$ | Steps | CPU |
| 1 | 36 | 6 | 0.3 | 1 | 36 | 5 | 0.1 |
| | 108 | 6 | 2.2 | | 108 | 5 | 1.0 |
| | 324 | 6 | 19.5 | | 324 | 5 | 8.8 |
| 8 | 36 | 36 | 1.6 | 8 | 36 | 19 | 0.5 |
| | 108 | 58 | 21.0 | | 108 | 22 | 4.4 |
| | 324 | 91 | 295.1 | | 324 | 23 | 40.1 |

TABLE IV

Algorithm 2 for Piecewise Linear Approximations ($r = 2$)

| No preconditioning | | | | With preconditioning | | | |
|---|---|---|---|---|---|---|---|
| $q$ | Size $N$ | Steps | CPU | $q$ | Size $N$ | Steps | CPU |
| 1 | 36 | 13 | 0.5 | 1 | 36 | 9 | 0.3 |
| | 108 | 13 | 4.3 | | 108 | 9 | 1.8 |
| | 324 | 13 | 37.4 | | 324 | 9 | 15.2 |
| 8 | 36 | 75 | 3.1 | 8 | 36 | 39 | 1.1 |
| | 108 | 123 | 40.6 | | 108 | 43 | 8.6 |
| | 324 | 181 | 519.6 | | 324 | 46 | 77.4 |

$N^* = 400$. For Tables I–II, we set $i_0 = 100$, $n_0 = 400$ (corresponding to $\delta = 0.25^q$) in using the preconditioned CGMs. While for Tables III–IV, we take $i_0 = 200$, $n_0 = 400$ (corresponding to $\delta = 0.5^q$) in using the preconditioned CGMs. Refer to (27) and (28).

These tables clearly demonstrate that the preconditioning as discussed in Section 5 speeds up the convergence of the CGMs and that Algorithm 1 using the normal equation is quite efficient. In other experiments on the choice of $\delta$, we note that increasing $\delta$ results in faster convergence of the preconditioned CGMs and that the overall efficiency may not improve if $\delta$ is too large (due to the increased work in computing the inverse matrix $D_n^{-1}$). The appropriate choice of $\delta$ is the subject of further investigations.

## 7. Conclusions

In this paper we have investigated the problem of the numerical solution of the boundary integral equation, from reformulation of the Helmholtz equation, by the conjugate gradient methods (CGMs). When the integral boundary is only piecewise smooth, the integral operator is non-compact and the solution is non-smooth. The non-compactness of the integral operator hinders the fast convergence of the CGMs for the solution of discrete linear systems. Such a linear system has a full and complex non-Hermitian matrix coefficient. For a piecewise smooth boundary with one typical corner, we have developed an efficient preconditioner which improves the clustering pattern of eigenvalues of the discrete linear system and hence established the fast convergence of the CGMs. Numerical experiments are carried out for a model problem using two general algorithms. It appears that the algorithm using the normal equation with our preconditioning is particularly efficient.

## Acknowledgment

The author thanks the anonymous referee for making useful comments and for improving the readability of the paper.

## REFERENCES


1. S. AMINI AND K. CHEN, "Iterative Techniques for Boundary Element Methods—Application to the Exterior Acoustic Problem," in *Boundary Element X*, edited by C. A. Brebbia (CMP, Springer-Verlag, New York, 1988), Chap. 4, p. 317.
2. S. AMINI AND K. CHEN, *Eng. Anal. Bound. Elem.* **6**, No. 2, 72 (1989).
3. S. AMINI AND I. H. SLOAN, *J. Integr. Equations Appl.* **2**, No. 1, 1 (1989).
4. K. E. ATKINSON, *Numer. Math.* **22**, 17 (1973).
5. K. E. ATKINSON, *A survey of boundary integral equation methods for the numerical solution of Laplace's equation in three dimensions*, Applied Mathematics Preprint AM88/8, School of Mathematics, University of New South Wales, 1988.
6. K. E. ATKINSON AND F. DE HOOG, *IMA J. Numer. Anal.* **4**, 19 (1984).
7. K. E. ATKINSON AND I. G. GRAHAM, "An Iterative Variant of the Nyström Method for Boundary Integral Equations on Non-smooth Boundaries," in *The Mathematics of Finite Elements and Applications*, edited by J. R. Whiteman (Academic Press, New York, 1988), p. 297.
8. K. E. ATKINSON AND I. G. GRAHAM, Iterative solution of linear systems arising from the boundary integral method, to appear (1990).
9. C. A. BREBBIA, *The Boundary Element Methods for Engineers*, (Pentech, Plymouth, England, 1980).
10. C. A. CHANDLER AND I. G. GRAHAM, *Math. Comput.* **50**, 125 (1988).
11. K. CHEN, Ph.D. thesis, Polytechnic South West, Plymouth PL4 8AA, England, 1989 (unpublished).
12. K. CHEN AND S. AMINI, Analytical behaviour of solutions of boundary integral equations for the Helmholtz problem on a non-smooth region, *IMA J. Appl. Math.* **45**, 147 (1990).
13. K. CHEN AND S. AMINI, Numerical analysis of boundary integral solution of the Helmholtz equation in domains with non-smooth boundaries, submitted for publication (1990).
14. D. COLTON AND R. KRESS, *Integral Equation Methods in Scattering Theory* (Wiley, New York, 1983).
15. G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations* (The Johns Hopkins Univ. Press, Baltimore, 1983).
16. W. HACKBUSCH, *Multigrid Methods and Applications*, Monograph (Springer-Verlag, Berlin, 1985).
17. F. K. HEBEKER, *Numer. Math.* **52**, 81 (1988).
18. P. W. HEMKER AND H. SCHIPPERS, *Math. Comput.* **36** (153), 215 (1981).
19. A. JENNINGS, *J. Inst. Math. Appl.* **20**, 61 (1977).
20. A. JENNINGS AND G. M. MALIK, *Int. J. Numer. Methods Eng.* **12**, 141 (1978).
21. R. E. KLEINMAN AND G. F. ROACH, *SIAM Rev.* **16**, 214 (1974).
22. R. KRESS AND W. I. SPASSOV, *Numer. Math.* **42**, 77 (1983).
23. Z. C. LIN, *J. Integr. Equations Appl.* **1**, No. 1, 83 (1988).
24. R. RIDDELL, *J. Comput. Phys.* **31**, 21 (1979).
25. T. K. SARKAR, X. P. YANG, AND E. ARVAS, *Wave Motion* **10**, 527 (1988).
26. H. SCHIPPERS, *J. Comput. Phys.* **48**, 441 (1982).
27. H. SCHIPPERS, *Numer. Math.* **46**, 351 (1985).
28. H. SCHIPPERS, "Multi-grid Methods in Boundary Element Calculations," in *Boundary Elements IX*, edited by C. A. Brebbia *et al.* (CMP, Springer-Verlag, Berlin, 1987), Vol. 1, p. 475.
29. H. A. VAN DER VORST, *J. Comput. Phys.* **44**, 1 (1981).