JOURNAL OF COMPUTATIONAL PHYSICS 36, 154–169 (1980)

On the Construction of Well-Conditioned Systems for Fredholm I Problems by Mesh Adapting*

FRANK HAGIN

Department of Mathematics, University of Denver, Denver, Colorado 80208

Received March 1, 1979; revised August 27, 1979

The paper addresses the numerical solution of linear Fredholm I integral equations of the type arising from various one-dimensional inverse scattering problems. Typically when these equations are discretized they lead to very ill-conditioned algebraic systems. It is shown that for these types of kernels a relatively simple mesh adapting scheme leads to very well-conditioned systems. Moreover, when the kernel is not known explicitly but must be generated by numerically solving a boundary-value problem, it is shown that an asymptotic analysis can extract the information necessary to successfully adapt this mesh.

1. INTRODUCTION

We are interested in Fredholm I integral equations of the type arising in various "inverse" problems; for example, the problem of determining velocity or density profiles as in Bleistein and Cohen [1] or Chen and Tsien [2]. In such problems one is led to boundary-value problems of the type

$$\begin{aligned} y''(x) + (\omega^2/c(x)^2)(1 + \alpha(x))\, y(x) &= 0, \\ a_0 y'(0) + b_0 y(0) &= d_0, \\ a_1 y'(1) + b_1 y(1) &= d_1, \end{aligned} \tag{1}$$

where we assume the parameter ω can be assigned arbitrary values as needed. The constants a_j, b_j, d_j sometimes vary with ω and could be complex valued. The term $c(x) > 0$ here is viewed as a known "reference velocity" and $\alpha(x)$ is considered a perturbation on $c(x)$. The main feature of the problem is that α, along with y, is *unknown*; hence we have an "inverse" problem. (Chen and Tsien [2] use an alternative form of differential equations, say, $((\rho + \alpha)\, y')' + \omega^2 y = 0$, but the problems are essentially the same.)

The approach to this difficult problem used in [1, 2] is, very briefly, as follows. Suppose y solves (1) above and that u solves (1) with $\alpha(x) \equiv 0$. If L denotes the

* This work was supported in part by the Department of Energy, under Contract EY-76-S-02-2482.

0021-9991/80/080154–16$02.00/0

original differential operator above, we define $v = y - u$ and make use of Green's formula (see [3]) as follows:

$$\int_0^1 (yLu - uLy)\,dx = \int_0^1 yLu\,dx = \omega^2 \int_0^1 yu \frac{\alpha}{c^2}\,dx$$

$$= \omega^2 \int_0^1 [u + v]u \frac{\alpha}{c^2}\,dx = [yu' - y'u]_0^1 = \omega^2 g(\omega). \qquad (2)$$

The argument is made that if α is small, v is of order α and dropping the term involving v gives the linear integral equation

$$\int_0^1 k(x, \omega)\,\alpha(x)\,dx = \int_0^1 (u/c)^2\,\alpha\,dx = g(\omega), \qquad (3)$$

where $u = u(x; \omega)$ comes from solving (1) with $\alpha \equiv 0$ (denote this by $L_0 u = 0$). In these applications u is called the "incident field"; and g is thought of as observed data at the boundaries.

We now turn our attention to the integral equation (3) and point out its distinctions. First, the kernel is not known explicitly and must be generated at the desired (x, ω)-values by numerically solving $L_0 u = 0$. More importantly, it is not "compact" in the sense of having as the (x, ω) domain a fixed finite rectangle. For example, in the constant reference speed case, $c(x) \equiv c_0$, the problem can be attacked with Fourier transform methods as in [1]. In this instance, the ω-values represent the Fourier transform variable which of course is unbounded. In our approach ω will be unbounded in the sense that as more accuracy is needed larger ω-values are required (e.g., ω may take on values $\frac{1}{2}, \frac{3}{2}, \ldots, n - \frac{1}{2}$, where n is the rank of the discrete system).

The point just made is critical in what follows since our approach is to discretize the problem in such a way that the linear algebraic system remains well conditioned. This is in happy contrast to the usual compact setting in which the discrete problem is typically ill conditioned if not singular. We are able to consistently solve our discrete systems via standard Gaussian elimination, thus avoiding more elaborate methods like singular-value decomposition or regularization (e.g., see [4]) or the Backus–Gilbert method as in [2].

In Section 2 we study two model integral equations. The first of these involves the cosine transform kernel $\cos(\pi\omega x)$. Such kernels arise naturally in the constant reference speed, $c(x) \equiv c_0$, problem for which Fourier methods are available. Such integral equations can easily be discretized in a manner to produce orthogonal algebraic systems (the advantages of which will be discussed shortly). Our original hope was that the same mesh used for $c(x) \equiv c_0$ would also be appropriate for perturbation problems, i.e., $c(x) = (1 + p(x))\,c_0$, where $p(x)$ is relatively small. However, we found that for p as small as 0.1 to 0.2 the discrete problem became ill conditioned. This lead us to attempt to "adapt' the mesh for x, the variable of integration, in an attempt to compensate for the variation in $c(x)$. (Mesh adapting, or "variable stepsize," is effectively used, for example, in modern differential equation solvers; e.g., see [9].) The mesh-adapting procedure is illustrated by a second model kernel cos $(\pi\omega\phi(x))$. Using the new mesh in discretizing the integral equation with this kernel again leads to an orthogonal discrete system. Numerical examples illustrate the necessity for, and effectiveness of, adapting for this model kernel.

In Section 3 the integral equation (3) is attacked using the mesh-adapting procedure described in Section 2. However, there is one additional complication; since the kernel is not known explicitly, how does one adapt the mesh and thus assure the resulting discrete system is well suited for computation? Our procedure is to do an asymptotic analysis, for large ω, of the differential equation in (1) in order to approximate the oscillatory behavior of the kernel. This turns out to be surprisingly effective and the resulting linear systems are very well conditioned. We illustrate this by numerically solving a problem that could be considered a typical variable reference speed problem.

Section 4 contains some comments on why the adapting procedure is effective for these problems and the possibility of extending these ideas to other types of integral equations.

The following few facts from computational linear algebra will be very useful in the sequel.

A complex square matrix U is *unitary* iff $U^{-1} = U^*$ (the conjugate transpose). A real matrix B is *orthogonal* iff $B^{-1} = B^T$, and this is true if and only if the rows (columns) form an orthonormal set. We will use the term "nearly orthogonal" nonrigorously to mean that $B^TB \approx I$, the identity.

Suppose for matrix A, $n \times n$ and real, the exact problem of interest is $A\mathbf{f} = \mathbf{g}$. If the available right side has some error, say $\boldsymbol{\delta}$, then one really solves the problem $A(\mathbf{f} + \boldsymbol{\epsilon}) = \mathbf{g} + \boldsymbol{\delta}$. If this problem is solved exactly the following (relative) error bound is available:

$$\frac{\|\boldsymbol{\epsilon}\|}{\|\mathbf{f}\|} \leqslant \|A\| \cdot \|A^{-1}\| \frac{\|\boldsymbol{\delta}\|}{\|\mathbf{g}\|} = \kappa(A) \frac{\|\boldsymbol{\delta}\|}{\|\mathbf{g}\|}$$

where $\kappa = \kappa(A) = \|A\| \cdot \|A^{-1}\|$ is called the *condition number* of A (relative to the particular norm). More generally, if A also has some error E (or E could represent the round-off error during computation), one has the following error bound:

$$\frac{\|\boldsymbol{\epsilon}\|}{\|\mathbf{f}\|} \leqslant \kappa(A) \left(\frac{\|E\|}{\|A\|} + \frac{\|\boldsymbol{\delta}\|}{\|\mathbf{g}\|}\right) \Big/ (1 - \|E\| \cdot \|A^{-1}\|).$$

It is easily shown that, for nonsingular A, $1 \leqslant \kappa(A) < \infty$ (see Ortega [5]). A useful interpretation for κ is that it measure (i.e., bounds) the amount that the error in the data is magnified in solving for $\mathbf{f}$. Clearly, given the choice, one would prefer to have κ as close to one as possible.

If the Euclidean norm is used, the following results hold:

1. $\kappa(A) = 1$ iff $A = kV$ for k scalar and $V^{-1} = V^T$. (This reaffirms the simple geometric idea that matrices are ill conditioned if some of the rows (columns) are nearly parallel, and well conditioned if they are all nearly orthogonal.)

2. $\kappa(A) = \sigma_1/\sigma_n \geqslant |\lambda_1|/|\lambda_n|$, where $|\lambda_1| \geqslant |\lambda_j|$ and $|\lambda_n| \leqslant |\lambda_j|$; λ_j are the eigenvalues of A. Also $\sigma_1 \geqslant \sigma_j$ and $\sigma_n \leqslant \sigma_j$ for the *singular values* of A (σ_j are the square roots of the eigenvalues of $A^T A$). It is a well-established myth that the ratio λ_1/λ_n is a good measure of condition, "stiffness," etc. of A. This is true for symmetric matrices but not in general (see E 2.1.11 of [5] for an example where $\lambda_1/\lambda_n = 1$ and $\kappa(A)$ is huge). However, the ratio σ_1/σ_n is valid for such measure and is used below.

2. Some Motivation; Two Model Problems

We now discuss two model problems for which we can make the discrete version "perfectly conditioned" in the sense of paragraph 1 above. The first example is a cosine transform kernel. Fourier kernels arose in our early investigations of constant reference speed problems, and they provide some valuable insight regarding discretization of the variables. Consider

$$\int_0^1 \cos(\pi\omega x) f(x)\, dx = g(\omega), \tag{4}$$

where we can choose the ω values at our discretion.

We want to discretize both ω and x in such a way as to make the resulting matrix orthogonal, if possible. Fourier series (or trigonometric interpolation) theory suggests that we choose $\omega = 0, 1, 2,..., n - 1$ and $x_j = (j - 1)/n$ for $1 \leqslant j \leqslant n$; for example, see [6] or [7]. Although this is in some ways a very natural mesh, we modify it a bit to

$$\omega_k = k - 0.5, \qquad x_j = (j - 0.5)/n \qquad \text{for } 1 \leqslant j, k \leqslant n. \tag{5}$$

If discretization (5) is applied to (4) in the simplest way we get

$$\frac{1}{n} \sum_{j=1}^{n} \cos(\pi\omega_k x_j) f(x_j) \approx g(\omega_k), \qquad k = 1,..., n. \tag{6}$$

It is not difficult to show that the matrix

$$A = ((2/n)^{1/2} \cos \pi\omega_k x_j)$$

is orthogonal. There are two reasons we prefer the mesh in (5) as opposed to the more natural one suggested above. One can view (6) as a trigonometric interpolation, but also as approximating the integral in (4) by the "midpoint rule" quadrature. Viewed the latter way, the midpoint rule is more accurate than the "rectangle rule" which results from the originally suggested mesh. Second, the suggested mesh is more "robust." For example, if the kernel in question were sin $(\pi\omega x)$ the same mesh would generate an orthogonal discrete system. Other meshes do not have this flexibility, which is especially important in the applications in Section 3 where the kernels are only approximated by the model kernels of this section.

In summary, we view (4) as our first model problem and (5)–(6) as its discretization. Since $((2/n)^{1/2} \cos \pi\omega_k x_j) = A$ is an orthogonal matrix, (6) can be solved by

$$\mathbf{f} = (2n)^{1/2} A^T \mathbf{g}.$$

Alternatively, if Gaussian elimination is used the system is extremely well conditioned.

We now consider our second model problem.

$$\int_0^1 \cos(\pi\omega\phi(x)) f(x)\, dx = g(\omega). \tag{7}$$

In the applications of the next section $\phi(x)$ will relate to the variable reference speed cases, whereas $\phi(x) = x$ relates to constant reference speed. However, for now we *simply assume that $\phi(x)$ is nondecreasing on* [0, 1] *and that it has been normalized so that* $\phi(0) = 0$ and $\phi(1) = 1$. (We note in passing that if $\phi'(x) > 0$ and the inverse ϕ^{-1} is available, one could solve (7) by cosine transform methods. That is, by using the substitution $u = \phi(x)$ to get

$$\int_0^1 \cos(\pi\omega u) \left[f(\phi^{-1}(u)) \frac{d}{du} \phi^{-1}(u) \right] du = g(\omega);$$

and solving for the function in brackets, etc. However, this approach does not generalize adequately for our purpose.)

The purpose in studying (7) is to learn how to "adapt" the x-mesh so that we retain our orthogonality (or nearly so) when the problem is discretized. The same adapting procedure is then used in the harder problems of Section 3. The idea is quite simple if we ask the question: How do we select the x_j values so that the discretization of (7) is analogous to (5)–(6)? The answer then is rather clear; simply pick x_j so that

$$\phi(x_j) = (j - 0.5)/n \tag{8}$$

or symbolically, $x_j = \phi^{-1}((j - 0.5)/n)$. If we do this and again use $\omega_k = k - 0.5$ for $1 \leqslant k \leqslant n$ our discrete problem is

$$\sum_{j=1}^{n} h_j \cos\left(\pi\omega_k \frac{j - 0.5}{n}\right) f_j = g(\omega_k), \qquad k = 1,\ldots, n. \tag{9}$$

The only difference between (9) and (6) is the fact that the steps h_j are no longer uniform due to the nonuniform selection of the x-mesh. Postponing for the moment the matter of how best to choose the h_j, note that the matrix resulting from (9) is $(h_j \cos(\pi\omega_k(j - 0.5)/n))$ compared to the matrix in (6), $(h \cos(\pi\omega_k(j - 0.5)/n))$. In particular, the h_j will, unless constant, destroy the orthogonality of the matrix. A solution to this is to absorb the h_j into the f_j temporarilly, i.e., define $\hat{f} = h_jf_j$ and solve the orthogonal system

$$\sum_{j=1}^{n} \cos\left(\pi\omega_k \frac{j-0.5}{n}\right)\hat{f}_j = g(\omega_k).$$

Once the $\hat{f}_j$ are found one solves $f_j = \hat{f}_j/h_j$. This process is equivalent to "column scaling" of the matrix in an attempt to make it better for computation. Fortunately, we know the scale factors h_j in advance and we know that the resulting matrix is "perfectly" conditioned.

Figure 1 below provides some graphical motivation for the adapted grid given by (8). Note that for arbitrary $\phi(x)$ and any integer $n > 0$, the function $\cos(\pi n\phi(x)) = 0$ precisely at the mesh points x_j (just as in the simple case $\phi(x) = x$). The construction of the mesh and this phenomenon are illustrated for $n = 6$ and $\phi(x) = x^2$ in Fig. 1.

Let us now return to the question of determining the proper values for h_j in (9). If $\phi'(x)$ is available and strictly positive a good choice is

$$h_j = 1/(n\phi'(x_j)). \tag{10}$$

This choice seems reasonable by a simple graphical argument, but can be put on a mathematical foundation as follows. If one made the independent variable change

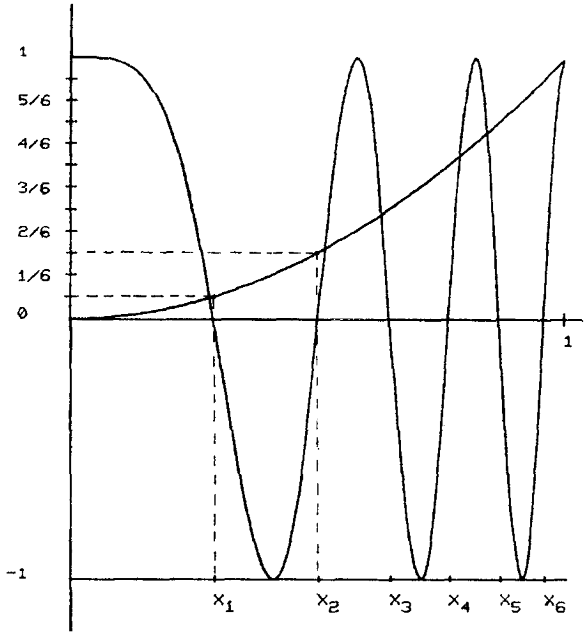

FIG. 1. The adapted mesh $(x_1, \ldots, x_6)$ for $\phi(x) = x^2$ and the function $\cos(6\pi\phi(x))$, which vanishes at the mesh points.

$u = \phi(x)$ in (7), the new kernel is simply cos $\pi\omega u$ and hence a uniform mesh is in order. If one uses the midpoint rule on the new integral the result is precisely the same as using h_j as in (10) in the discrete system (9). We leave the verification of this to the reader.

Now if ϕ' is not available or if it vanishes sometimes, one may have to derive a discrete version of (10). This turns out to be less trivial than one would expect. We suggest a slightly different way of constructing the x-mesh which, although it sacrifices a bit of orthogonality, works very well in practical problems. Define ξ_j and mesh points x_j by

$$\begin{aligned} \phi(\xi_m) &= m/n \qquad \text{for} \quad 0 \leqslant m \leqslant n, \\ x_j &= \frac{\xi_j + \xi_{j-1}}{2}, \\ h_j &= \xi_j - \xi_{j-1} \qquad \text{for} \quad 1 \leqslant j \leqslant n. \end{aligned} \tag{11}$$

Here we have "adapted" the endpoints 0, $1/n$, $2/n$,..., 1 and then taken as x_j the midpoints of the new subintervals. It is no longer true that $\phi(x_j) = (j - 0.5)/n$ in general. The resulting discretization of (7) is

$$\sum_{j=1}^{n} \cos(\pi\omega_k\phi(x_j))(h_jf_j) = g(\omega_k), \tag{12}$$

where $\omega_k = (k - 0.5)$, $h_j = \xi_j - \xi_{j-1}$, and x_j is given by (11). The matrix in (12) is "close to" but not precisely orthogonal, consequently the system would have to be solved by, say, Gaussian elimination (as opposed to multiplication by the matrix transpose). However, the matrix is very well conditioned as is illustrated below.

In summary, for the second model problem (7), if the mesh scheme (8) is used, orthogonality of the discrete system is achieved. In this case the h_j is computed by $h_j = (n\phi'(x_j))^{-1}$ or some good discrete version of this. Alternatively, the mesh scheme (11) may be used, in which case the discrete system is no longer orthogonal, but in practice it is nearly so in the sense that the condition number of the matrix is close to one. This assumes of course that the h_j are scaled out as discussed in the paragraph following (9).

We conclude this section with some numerical results which illustrate the necessity and the effectiveness of adapting the mesh in the second model problem. First we attempt to solve

$$\int_0^1 f(x)\cos(\pi\omega\phi(x))\,dx = \sin \pi\omega/(2\pi\omega), \tag{13}$$

where $\phi(x) = x^2$. The solution in this case is $f(x) = x$. In all cases we use the same ω-mesh, $\omega_k = k - 0.5$ for $1 \leqslant k \leqslant n$. In Table I (left side) are shown the results when no adapting is done and the midpoint rule $x_j = (j - 0.5)/n$ is used. The results are much like those typically experienced in compact Fredholm I problems; as n increases the matrix becomes more ill conditioned and meaningful results are not

TABLE I

Results from Solving (13) with $\phi(x) = x^2$; Solution is $f(x) = x$

Unadapted Midpoint rule				Adapting using (8), (10)			
$\kappa(A)$	$E_{\max}$	E_{med}	n	$\kappa(A)$	$E_{\max}$	E_{ess}	E_{med}
3×10^3	2×10^5	65	10	1	0.124	0.0218	0.00160
1×10^{12}	9×10^{11}	7964	20	1	0.126	0.0089	0.00039
1×10^{12}	3×10^{11}	460	30	1	0.126	0.0047	0.00017
2×10^{12}	2×10^{14}	9999+	40	1	0.126	0.0029	0.00009

forthcoming. In the table $E_{\max}$ is the maximum error in the f_j. E_{ess} (the "essential" error) is the maximum error in the f_j after excluding the $n/10$ largest errors (we consistently observed several points, about $n/10$, near the endpoints at which the largest errors occurred). E_{med} is the median error.

Also in Table I are shown the results for problem (13) when adapting is done using the original mesh (8) and $h_j = (n\phi'(x_j))^{-1} = (2nx_j)^{-1}$. Note that $E_{\max}$ does not get small as n increases, E_{ess} approaches zero as $1/n$, and E_{med} approaches zero as $1/n^2$. In the cases we have tried, the largest errors typically occur at one of the endpoints (e.g., see Fig. 2). Because of the kinship of all problems discussed herein with Fourier series, we speculate that the lack of convergence at the endpoints is related to the loss of smoothness in the function when it is extended via periodicity beyond $0 \leqslant x \leqslant 1$.

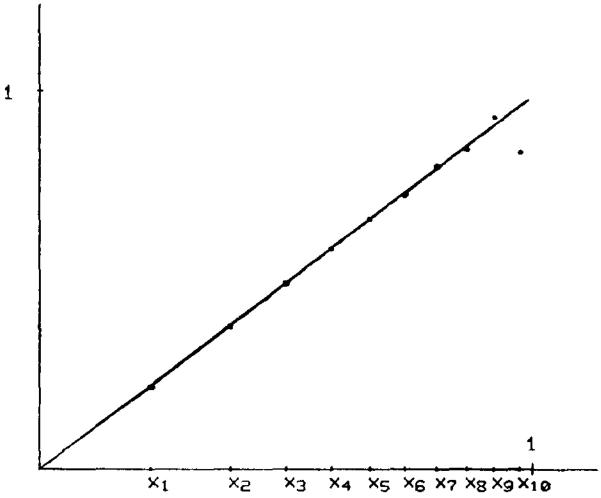

FIG. 2. Graph of the exact solution $f(x) = x$ to (13) and the discrete solution using mesh adapting (8), (10) with $n = 10$.

Let us now consider a $\phi(x)$ which is related to the applications to be discussed in Section 3. Suppose $c(x)$ represents a reference velocity and is assumed to be

$$\begin{aligned} c(x) &= 1 + \beta x \quad &&\text{for} \quad 0 \leqslant x \leqslant 0.5, \\ &= 1 + \beta/2 \quad &&\text{for} \quad x > 0.5, \end{aligned} \tag{14}$$

where $\beta > 0$ will be set below. Now define $\phi(x)$ to be the (normalized) integral of $1/c$, i.e.,

$$\begin{aligned} \phi(x) &= \frac{1}{K}\int_0^x \frac{ds}{c(s)} = \frac{1}{\beta K}\ln(1 + \beta x), \qquad x \leqslant 0.5, \\ &= \frac{1}{K}\left[\frac{1}{\beta}\ln(1 + \beta/2) + \frac{(x - 0.5)}{1 + \beta/2}\right], \qquad x > 0.5, \end{aligned} \tag{15}$$

where $K = \int_0^1 ds/c(s)$. Clearly ϕ is of the form assumed, $\phi(0) = 0$, $\phi(1) = 1$; moreover, $\phi'(x) = 1/c(x)\,K > 0$ is available if needed. Consider

$$\int_0^1 f(x)\cos(\pi\omega\phi(x))\,dx = \sin \pi\omega/\pi\omega, \tag{16}$$

where $\phi(x)$ is given in (15) with $\beta = 1.5$. The solution is $f(x) = 1/c(x)$.

Table II (left side) shows the results when no adapting is used. Once again the situation is poor for small n and deteriorates as n becomes large. Figure 3 illustrates the difficulty for the case $n = 10$, when no adapting is done; it also illustrates the effectiveness of adapting.

Table II also shows the results when the adapting scheme (8), (10) is used. Note again that the "essential" error approaches zero as $h = 1/n$ and the median error goes to zero as $1/n^2$.

Finally, we solve (16) using the alternative adapting scheme given by (11) and show these results in Table III. Note the similarity between the results in Tables II and III. Also note that while the latter scheme does not produce a genuinely orthogonal matrix, the condition number is very close to one and seems to approach one as n gets large.

TABLE II

Results from Solving (16) with $\phi(x)$ Given by (15), $\beta = 1.5$

Unadapted Midpoint rule				Adapting using (8), (10)			
$\kappa(A)$	$E_{\max}$	E_{med}	n	$\kappa(A)$	$E_{\max}$	E_{ess}	E_{med}
18	0.45	0.108	10	1	0.073	0.0135	0.00140
1180	9	1.53	20	1	0.073	0.0054	0.00034
1×10^5	370	18	30	1	0.073	0.0029	0.00015
8×10^6	2×10^4	353	40	1	0.073	0.0018	0.00008

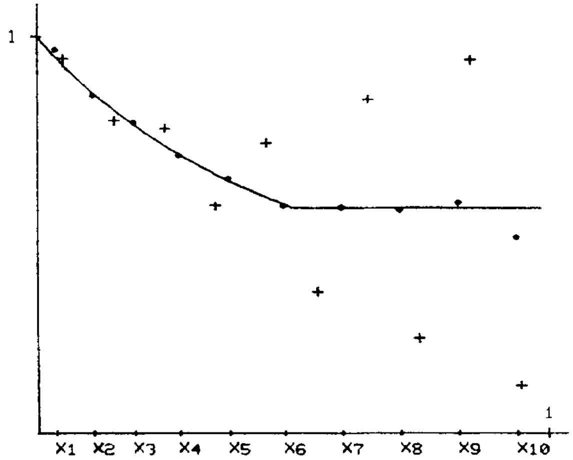

FIG. 3. Graph of the exact solution $f(x) = 1/c(x)$ to (16) and the discrete solution ($n = 10$) with ($\cdot$) adapting (8), (10) and without (+).

TABLE III

Results from Solving (16) Using Alternate Adapting Scheme (11)

n	$\kappa(A)$	$E_{\max}$	E_{ess}	E_{med}
10	1.036	0.073	0.0174	0.00524
20	1.019	0.073	0.0086	0.00172
30	1.014	0.073	0.0057	0.00096
40	1.010	0.073	0.0030	0.00046

In conclusion, the adapted midpoint rule works quite well for the model problem, particularly if very high accuracy is not required.

3. Back to the Inverse Problem

In the Introduction we discussed an inverse problem described by the boundary value problem (1). We indicated how a linearization of the problem leads to the integral equation

$$\int_0^1 k(x, \omega)\, \alpha(x)\, dx = \int_0^1 \left(\frac{u}{c}\right)^2 \alpha(x)\, dx = g(\omega), \tag{17}$$

where $u = u(x; \omega)$ solves the differential equation

$$u'' + (\omega^2/c^2)\, u = 0 \tag{18}$$

plus appropriate boundary conditions. In order to take advantage of the adapting procedure discussed in Section 2, we need to know something about the nature of the kernel in (17). In particular, we must find a function to play the role of ϕ of the last section.

When $c = c(x)$ is variable (18) cannot generally be solved analytically. However, it can be analyzed rather well, particularly for large values of ω. Of the several approaches available we believe that the so-called WKB method (e.g., see [8]) is probably the most fruitful for our purposes. We now give a brief description of how this works.

Physical arguments, and experience, suggest that one seek solutions to (18), for large ω, of the form

$$u(x; \omega) \sim \exp \int_0^x \left[\omega\psi_0 + \psi_1 + \frac{1}{\omega}\psi_2 + \cdots\right] ds. \tag{19}$$

Substituting this form into (18) and equating powers of ω lead to recursive formulas for the ψ_k:

$$\psi_0{}^2 + 1/c^2 = 0,$$

$$2\psi_0\psi_1 + \psi_0' = 0$$

and so on for terms of lower order in ω. Solving for ψ_0 and ψ_1 gives

$$\psi_0 = \pm i/c, \qquad \psi_1 = \tfrac{1}{2}c'/c.$$

Putting these into (19), and ignoring lower-order terms, gives

$$u(x; \omega) \sim (c(x))^{1/2} \exp\left(\pm i\omega \int_0^x \frac{ds}{c(s)}\right). \tag{20}$$

That is, there are two solutions (one for each sign) with this asymptotic behavior. Let us now get an asymptotic expression for the *real part* of u^2,

$$u^2(x; \omega) \sim c(x) \exp\left(\pm i2\omega \int_0^x \frac{ds}{c(s)}\right),$$

$$\mathrm{Re}[u^2(x; \omega)] \sim c(x) \cos\left(2\omega \int_0^x \frac{ds}{c(s)}\right).$$

More precisely, this expression is valid for one of the solutions to $u'' + \omega^2 u/c^2 = 0$. For simplicity let us assume the boundary conditions give this particular solution. We are now close to having our kernel in the form seen in Section 2, at least asymptotically. To this end define

$$\phi(x) = \frac{1}{K}\int_0^x \frac{ds}{c(s)},$$
$$\tilde{\omega} = \frac{2\omega}{\pi} K, \qquad \text{where} \qquad K = \int_0^1 \frac{ds}{c(s)}. \tag{21}$$

Note that $\tilde{\omega}$ now denotes our scaled ω values; they will take on values $\tilde{\omega}_k = k - 0.5$ for $1 \leqslant k \leqslant n$. In this notation we have

$$\operatorname{Re} k(x, \omega) = \operatorname{Re} \frac{u^2(x; \omega)}{c^2(x)} \sim \frac{1}{c(x)} \cos(\pi\tilde{\omega}\phi(x)),$$

i.e., the *Re* $k(x, \omega)$, for large ω, is like the model kernel of Section 2. This suggests that the mesh adapting scheme of Section 2 could be useful in making the discrete version of (17) well conditioned.

Suppose the adapted mesh (8), (10) is applied to (17). Assuming, for simplicity, that both α and g are real valued, we have

$$\int_0^1 \operatorname{Re} k(x, \omega)\, \alpha(x)\, dx = \int_0^1 \operatorname{Re} \frac{[u^2(x; \omega)]}{c^2(x)}\, \alpha(x)\, dx = g(\omega).$$

The usual discretization of this gives:

$$\sum_{j=1}^{n} \left[\frac{\operatorname{Re}[u^2(x_j, \omega_k)]}{c(x)^2} h_j\right] \alpha_j = g(\omega_k), \qquad k = 1, \ldots, n. \tag{22}$$

Now a remarkable thing happens. Using (10) to define the stepsize gives

$$h_j = \frac{1}{n\phi'(x_j)} = \frac{Kc(x_j)}{n}.$$

Hence, for the matrix, call it $A = (a_{kj})$, in (22):

$$a_{kj} \sim \frac{1}{c(x_j)} \cos(\pi\tilde{\omega}_k\phi(x_j))\, h_j = \frac{K}{n} \cos(\pi\tilde{\omega}_k\phi(x_j))$$
$$= \frac{K}{n} \cos\left(\pi(k - 0.5)\frac{j - 0.5}{n}\right).$$

Consequently, with the h_j taken as part of the matrix, system (22) appears to be "nearly orthogonal." The catch is of course that the asymptotic behavior noted is for "large" ω. However, often this behavior is surprisingly meaningful for small and moderate ω as well. In our applications we have experienced this and as a result have found system (22) to be extremely well conditioned. (Note: we do not want to overemphasize the fact that our kernel had precisely the right power of c to make the discrete system nearly orthogonal. Had we found an unwelcome c factor in the system, we would simply scale it out as we did the h_j in Section 2.)

We now study a numerical example of problem (17). Let us use the same reference speed as in the example of Section 2:

$$\begin{aligned} c(x) &= 1 + 1.5x \qquad &&\text{for } 0 \leqslant x \leqslant 0.5, \\ &= 1.75 &&\text{for } x > 0.5. \end{aligned}$$

The appropriate ϕ is given by (15). The (x, ω)-mesh is computed as usual so that

$$\phi(x_j) = \frac{j - 0.5}{n}, \qquad \tilde{\omega}_k = k - 0.5 \qquad \text{for} \quad 1 \leqslant j, k \leqslant n.$$

In applications the $g(\omega)$ in (17) is observed data. In these test problems, however, g must be computed for a particular α. Let us put aside until the close of this section the discussion on how g is computed and proceed under the assumption it is known.

Suppose g is available at the prescribed ω values and that the exact solution to (17) is

$$\begin{aligned} \alpha(x) &= 0.1 \cos^2(2\pi x - \pi) \qquad &&\text{for} \quad | x - 0.5 | \leqslant 0.25, \\ &= 0 &&\text{elsewhere.} \end{aligned} \tag{23}$$

A graph of α is shown in Fig. 4. The kernel in (17) is $\text{Re}(u^2)/c^2$, where u solves (18) with complex boundary conditions:

$$\begin{aligned} u'(0; \omega) + i\omega u(0; \omega) &= 2i\omega, \\ u'(1; \omega) + i\frac{\omega}{c(1)} u(1; \omega) &= 0. \end{aligned} \tag{24}$$

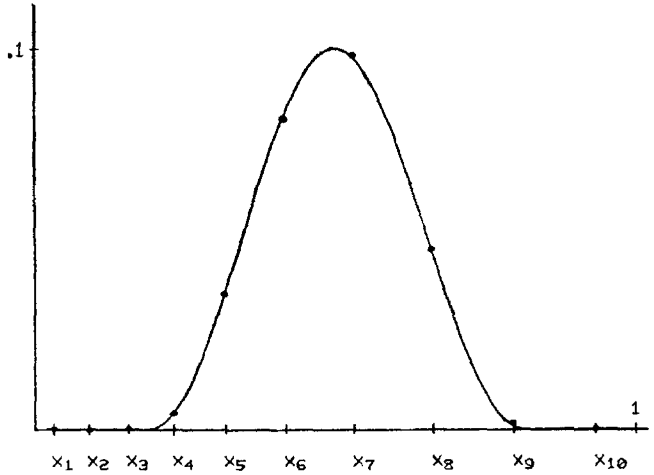

FIG. 4. Graph of the exact solution to (17), $\alpha(x)$ as in (23), and the discrete solution using (22) with $n = 10$ adapting by (8), (10).

(These boundary conditions are suggested by physical arguments associated with the wave equation; e.g., see [1].) Note that if $c(x) \equiv 1$ the solution of (18) and (24) would be $u(x; \omega) = e^{i\omega x}$ and we would have kernel $k(x, \omega) = \cos 2\omega x$. Hence we can view our test problem as a (large) perturbation of this simple problem.

To numerically solve for α in (17) we proceed as follows. The boundary-value problem (18), (24) is solved numerically for the n ω-values

$$\omega = \frac{\pi}{2K}\tilde{\omega} = \frac{\pi}{2K}(k - 0.5), \qquad 1 \leqslant k \leqslant n;$$

see (21). This was done by first solving two real initial-value problems numerically, one with $u(0) = 1$, $u'(0) = 0$, and one with $u(0) = 0$, $u'(0) = 1$. Then superposition was applied to satisfy the boundary conditions. We used a Runge–Kutta–Fehlberg variable step-size method (see [10]) to generate our numerical solutions to four or five significant figures. These values were saved for the mesh points x_j, where $\phi(x_j) = (j - 0.5)/n$ as usual. We then formed the matrix in (22) and solved for $\alpha_j \approx \alpha(x_j)$ by a standard Gaussian elimination (with partial pivoting) routine. Of course the matrices will not be exactly orthogonal; however, they do turn out to be extremely well conditioned. The results are shown in Table IV. Note the similarity in the accuracy achieved here compared to the model problem of Section 2; see Table II.

TABLE IV

Results from Solving (17) via the Discrete System (22)[a]

n	$\kappa(A)$	$10 \cdot E_{\max}$	$10 \cdot E_{\text{ess}}$	$10 \cdot E_{\text{med}}$
10	1.818	0.009	0.0051	0.00224
20	1.821	0.005	0.0034	0.00036
30	1.824	0.005	0.0022	0.00030

[a] Exact solution $\alpha(x)$ is defined by (23). See Fig. 4.

Finally, let us return to the matter of computing the "observed" data $g(\omega)$. This discussion is not essential to the purpose of this paper, but is included for the curious reader. In Section 1 we sketched how the inverse problem (1) led to a nonlinear integral equation which in turn was linearized to give (3) or (17). In the above test problem we computed $g(\omega)$ from (2), i.e., $\omega^2 g(\omega) = [yu' - y'u]_0^1$, where y solves (1), (24). Then we solved (22) using this g to obtain, say, $\alpha_j^{(0)}$, $1 \leqslant j \leqslant n$. However, this g is really not appropriate for the linear problem (17) since it includes the small nonlinear term involving $v = y - u$ in (2). Therefore, to compute a g appropriate for (17) and our test purposes, we iterated as follows. We put $\alpha^{(0)}$ in (1) and resolved (1) for y, and then $v = y - u$. A new g was computed by numerically approximating

$$g_{\text{new}} = g_{\text{old}} - \int_0^1 uv \frac{\alpha^{(0)}}{c^2}\, dx; \tag{25}$$

see (2). Using the new g we resolved (22) for, say, $\alpha_j^{(1)}$. We repeated this process until the α_j converged (this usually took three iterations). The final α_j are effectively what one would have if the $g(\omega)$ appropriate for (17) were known and (22) solved once. These are the α_j summarized in Figure 4 and Table IV. (Note: the same iteration is sometimes desired in real applications. For example, if $g(\omega)$ were observed data for the original inverse problem (1), one could solve the linear system (22), modify g as in (25), resolve (22), and iterate to find the α_j approximating the solution to (1). In this regard see [2].)

4. Summary

For the class of applied problems discussed herein we have found this special mesh adapting and the mid point rule very effective. We have consistently been able to get very well-conditioned linear systems and the accuracy of the solutions has been adequate for our purposes using systems of order 10–20. The well-conditioned matrices are particularly important to us for two reasons. The right sides $g(\omega)$ represent observed data which in practice may have only modest accuracy; hence we cannot afford to lose much. Moreover, the matrices are developed by numerically solving numerous boundary-value problems, a relatively expensive operation. Hence it is important that we do not need to compute these coefficients to, say, 8 or 10 places; this might very well be the case if the matrices were poorly conditioned.

A few words about the "accuracy" of our scheme is in order. All numerical evidence points to the fact that the resulting solutions are far more accurate than one would expect from an analysis of the crude (mid-point rule) quadrature scheme employed. The numerical results consistently look like what one would obtain from a Fourier series expansion of $\alpha(x)$ through n terms. Moreover, preliminary analysis indicates that this is just what is taking place. Hopefully, this analysis can be completed and presented in a subsequent article. This Fourier series-type convergence is probably the most salient feature of the procedure. We are convinced that it will ultimately prove extremely competive when carefully compared with alternative methods.

Perhaps a few words about "why" our particular matrices turn out to be nearly orthogonal will help others find broader applications of these ideas. No doubt the orthogonality we have enjoyed starts with the fact that our kernel comes from the solution of a symmetric differential equation. Recall from classical Sturm–Liouville theory that if the boundary conditions were homogeneous and if we found the eigenvalues, say ω_k^*, then the resulting solutions $u(x; \omega_k^*)$ would be orthogonal on $[0, 1]$. In this case one would expect that an intelligent mesh on $[0, 1]$ would lead to an orthogonal discrete system, $u(x_j, \omega_k^*)$, $1 \leqslant j, k \leqslant n$. Of course we have not suggested the (very expensive) process of finding the ω_k^*. However, our asymptotic analysis and choice of ω_k have led to a reasonable approximation to the ω_k^*, at least in the sense of the proper oscillation of the functions $u(x; \omega_k)$. In this regard, recall that the kth eigenfunction of a Sturm–Liouville problem has exactly $k - 1$ zeros on $(0, 1)$. A look at our model kernel functions of Section 2, e.g., $\cos(\pi\omega_k\phi(x))$, shows that they have this same property. Because of the strength of the WKB analysis in this regard, our kernels then also show this behavior, or nearly so. In short, the near orthogonality of our kernels originates with the defining boundary-value problem.

More generally, if the kernel in question arises naturally from an orthogonal setting, there is probably a good chance that the approach suggested herein could have some value. Of course, (nearly) orthogonal matrices are not the only ones which are well conditioned. Suppose, for example, that a kernel is oscillatory in x and the rate of oscillations increase as the second variable (e.g., ω) increases. We conjecture that a discretization scheme analogous to that used here might be effective on such integral equations. Such generalizations are being investigated. We hope that the ideas presented here will be a modest step toward an effective method for solving a large class of integral equations.

Acknowledgments

The author is grateful to Professors N. Bleistein and J. Cohen for several very helpful discussions prior to and during this investigation. Also helpful were several discussions with Dr. Sam Gray whose doctoral thesis work on the constant reference speed problem helped provide important insights. Recognition is also due Mourad Lahlou for his expert help in the computational aspects of this work.

References

1. N. Bleistein and J. K. Cohen, *SIAM J. Appl. Math.* **32**, No. 4 (1977), 784.
2. Y. M. Chen and D. S. Tsien, *J. Comput. Phys.* **25**, No. 4 (1977), 366.
3. E. A. Coddington and N. Levinson, "Theory of Differential Equations," McGraw-Hill, New York, 1955.
4. C. T. H. Baker, "The Numerical Treatment of Integral Equations," Chap. 5, Oxford Univ. Press (Clarendon), 1977.
5. J. M. Ortega, "Numerical Analysis, A Second Course," Academic Press, New York, 1972.
6. A. Zygmund, "Trigonometric Series," Vol. II, Cambridge Univ. Press, Cambridge, 1959.
7. G. Dahlquist and A. Björck, "Numerical Methods," Prentice-Hall, Englewood Cliffs, N. J., 1974.
8. G. F. Carrier, M. Krook, and C. E. Pearson, "Functions of a Complex Variable," McGraw-Hill, New York, 1966.
9. L. F. Shampine and M. K. Gordon, "Computer Solution of Ordinary Differential Equations," Freeman, San Francisco, 1975.
10. W. H. Enright and T. E. Hull, *SIAM J. Numer. Anal.* **13**, No. 6, (1976), 944.