Journal of Computational Physics **168,** 155–183 (2001)
doi:10.1006/jcph.2000.6688, available online at http://www.idealibrary.com on IDEAL®

Accurate and Efficient Algorithms for Frequency Domain Scattering from a Thin Wire

Penny J. Davies,* Dugald B. Duncan,† and Stefan A. Funken‡

*Department of Mathematics, University of Strathclyde, 26 Richmond Street, Glasgow G1 1XH, United Kingdom; †Department of Mathematics, Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, United Kingdom; and ‡Mathematisches Seminar, Christian-Albrechts-Universität zu Kiel, Ludewig-Meyn-Strasse 4, D-24098 Kiel, Germany

E-mail: penny@maths.strath.ac.uk, D.B.Duncan@ma.hw.ac.uk, saf@numerik.uni-kiel.de

Received March 22, 2000; revised November 21, 2000

This paper contains a systematic study of numerical approximations for solving the exact kernel form of Pocklington's integro-differential equation for the current induced on a thin wire by an incident time-harmonic electromagnetic field. We consider various Galerkin (h, p, hp, and adaptive h) and collocation schemes and show that a sensible hp refinement strategy provides a very efficient way to solve the problem. We also describe how the kernel (itself a difficult singular integral) can be evaluated reliably and efficiently.

Key Words: thin wire; electromagnetic scattering; Pocklington; exact kernel; numerical approximation.

1. INTRODUCTION

Calculating the current induced on a thin wire by an incident electromagnetic field is an important problem with a long history (see for example [12, Chapter 6] or the introduction of [22]), dating back to Pocklington's paper of 1897 [17]. The computational problem has been widely studied (see [2, 5, 12, 16] and the references therein), although few authors have carried out systematic comparisons of various different approximation techniques. The 1975 paper [5] does this to some extent, but its scope is restricted by the limited computing power and analytical techniques available at that time and by the fact that it deals with an ill-posed version of the problem. Here we carry out a thorough study of various different approximation schemes for the difficult exact kernel form of the problem, describing how they can be implemented efficiently and reliably enough to be useful in practice.

It is vital that any numerical scheme used to compute the induced current is both accurate and efficient. Efficiency is significant because the problem (an integro–differential equation)

0021-9991/01 $35.00

is necessarily computationally intensive and because it often appears as a module in a much more complicated code (e.g., the computational electromagnetics code NEC [4]). This is also one reason why the current needs to be computed accurately (i.e., reliably found to within a given tolerance). Another reason is that it is important to be able to evaluate the errors introduced in the derivation of the thin wire model to determine when it is valid. Modeling errors can only be identified when numerical errors are essentially eliminated from the computed solution.

Here we consider the canonical problem of computing the current induced on a straight thin wire by the time-harmonic electric field $\boldsymbol{E}(\boldsymbol{x})e^{i\omega t}$. However, the solution techniques we develop and use could be extended to deal with curved wires, although the implementation will be significantly more complicated and costly in computer time. We note also that it is relatively straightforward to calculate the scattered field once the current flowing along the wire is known (this is described for example in [10]).

Suppose that the wire has unit length and radius $a \ll 1$ and lies along the z-axis, where $x = (x, y, z)$; i.e., it occupies the region $\mathcal{W} = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 \le a^2, z \in [0, 1]\}$. One of the most common formulations of the problem is Pocklington's equation, whose derviation is sketched in Section 1.1. This is the integro–differential equation (IDE) for the axial component u of the current

$$-(\omega^2 + d^2/dz^2) \int_0^1 K(z - z')u(z')\,dz' = f(z), \tag{1.1}$$

where the right-hand side f and kernel K are both known functions. Appropriate boundary conditions are

$$u(0) = u(1) = 0. \tag{1.2}$$

Inverting the Helmholtz operator in (1.1) leads to the Hallen version of the problem

$$\int_0^1 K(z - z')u(z')\,dz' = A\cos\omega z + B\sin\omega z - \frac{1}{\omega}\int_0^z f(z')\sin\omega(z - z')\,dz'$$

in terms of two constants A and B, which need to be chosen to satisfy the boundary conditions (1.2). Both formulations of the problem are equivalent [11, 20], but as pointed out in these two papers, solving Hallen's equation numerically can give rise to unpredictable results. The Hallen formulation of the problem is also much more restrictive in that it cannot easily be extended to cover the case of curved wires (althouth this is possible in certain situations, see [15]), whereas Pocklington's equation can be extended. From now on, we therefore restrict attention to consideration of numerical schemes for (1.1), (1.2).

Many authors use the *reduced kernel* form of the IDE (1.1) in which $K = K_R$ given by

$$K_R(\rho) = \frac{e^{-i\omega\sqrt{\rho^2+a^2}}}{4\pi\sqrt{\rho^2 + a^2}} \tag{1.3}$$

(see, e.g., [4, 5] and references in [2, 22]). However, since K_R and its derivatives are bounded for all $\rho \in [0, 1]$, the mapping $\mathbb{L}_R : L^2(0, 1) \to L^2(0, 1)$ (where $L^2(0, 1)$ is the space of square integrable functions on $(0, 1)$) defined by

$$\mathbb{L}_R u(z) \equiv -(\omega^2 + d^2/dz^2) \int_0^1 K_R(z - z')u(z')\,dz'$$

is a compact operator and hence has an unbounded inverse [13]. This means that the IDE (1.1) with this kernel is ill-posed (the solution u does not depend continuously on the data f). The ill-posedness of the equation has significant implications for anyone trying to solve it numerically. Basically it means that (1.1) with kernel K_R cannot give a "sensible" numerical solution unless the smallest space mesh-size h used in its discretization is large compared to the radius a of the wire. Taking h to be too small results in the solution being polluted by an oscillating error near the ends of the wire that grows as h is reduced (this is illustrated for the reduced kernel Hallen problem in [2, Figs. 3.8 and 3.9]: similar results are obtained for Pocklington's equation). A consequence of this is that it is not possible to increase the accuracy of the approximation by refining the mesh, and indeed it is not clear what the computed "solution" actually represents. Although this has been known since the early 1950s [21], the reduced kernel model still appears to be used in many situations.

For the reasons given above it is important to be able to compute the induced current accurately, and this is not possible using the reduced kernel. Hence we consider numerical schemes for the more complicated *exact kernel* formulation of (1.1), in which $K = K_E$ given by

$$K_E(\rho) = \frac{1}{4\pi^2} \int_0^\pi \frac{e^{-i\omega\sqrt{\rho^2+4a^2\sin^2\theta}}}{\rho^2 + 4a^2\sin^2\theta}\, d\theta. \tag{1.4}$$

This is a singular kernel, satisfying $K_E(\rho) \approx \frac{-1}{4\pi^2 a} \ln|\rho|$ for small ρ [16, 11]. Problem (1.1), (1.2) with kernel K_E has been shown to have a unique solution by Jones [11], and Rynne [20] has proved that it is well-posed. Rynne shows that if $f \in L^P(0, 1)$ (spaces are defined in Section 1.2 below), then the solution u of the exact kernel form of (1.1) is continuous and so the boundary conditions (1.2) are meaningful. He also proves that the solution has square root behavior at the ends of the wire (i.e., $u(z) = O(\sqrt{z(1-z)})$ for $z \approx 0, 1$).

In recent papers [18, 19], Rynne has derived a Gårding type inequality for the integro–differential operator and used it to obtain rigorous convergence results for Galerkin approximations of (1.1). Implementing Rynne's scheme efficiently is nontrivial since it involves computing double integrals involving the singular kernel K_E, which is itself an integral. The difficulties involved in evaluating K_E are well documented [16, 23, 24], and in Section 3 we describe a new algorithm for K_E that is both reliable and efficient. This section also contains other implementation details. We describe several different numerical schemes for the exact kernel form of (1.1) in Section 2, some of which (piecewise linear Galerkin and various collocation methods) have been used previously for the reduced kernel problem, and others (the p, hp, and adaptive h methods) which have not. Numerical results are presented in Section 4 and discussed in Section 5.

We conclude this section with a brief derivation of the thin wire model (1.1), (1.2) and a description of notation and function spaces used in the rest of the paper.

1.1. *Derivation of the IDE (1.1)*

We now sketch the derivation of the model (1.1), (1.2), showing the assumptions that result in both the reduced and exact kernel forms. Full details are given in [10]. An alternative derivation is given by Tijhuis *et al.* in [22].

The current induced on the wire by the time-harmonic incident field $\boldsymbol{E}(\boldsymbol{x})e^{i\omega t}$ is $\boldsymbol{u}e^{i\omega t}$, and we assume that the current flows along the wire so that $\boldsymbol{u}(\boldsymbol{x}') = u(\boldsymbol{x}')\boldsymbol{e}_z$ where $\boldsymbol{e}_z = (0, 0, 1)$.

This means that the vector potential at $\boldsymbol{x} \in \mathbb{R}^3$ is $A(\boldsymbol{x})e^{i\omega t}\boldsymbol{e}_z$ where A is given by the integral

$$A(\boldsymbol{x}) = \frac{\mu}{4\pi} \int_{\partial\mathcal{W}} \frac{u(\boldsymbol{x}')e^{-i\omega|\boldsymbol{x}-\boldsymbol{x}'|/c}}{|\boldsymbol{x}-\boldsymbol{x}'|}\, dS'$$

over the surface $\partial\mathcal{W}$ of the wire, where μ is the permeability and c the speed of light. The z-component E_s of the scattered electric field is given by

$$(c^2\partial^2/\partial z^2 + \omega^2)A(\boldsymbol{x}) = i\omega E_s(\boldsymbol{x}).$$

Assuming the wire to be a perfect conductor implies that the tangential component of the total (incident and scattered) electric field is zero on the wire's surface and hence

$$-(c^2\partial^2/\partial z^2 + \omega^2)A(\boldsymbol{x}) = i\omega E(\boldsymbol{x}),$$

for $\boldsymbol{x} \in \partial\mathcal{W}$, where E is the z-component of $\boldsymbol{E}$. Making the change of variables $E_1 = E/c$, $u_1 = \mu u$, and $\omega_1 = \omega/c$ (so ω_1 is the wavenumber k), and then dropping the subscripts from E, u, and ω results in the nondimensionalized equations

$$A(\boldsymbol{x}) = \frac{1}{4\pi} \int_{\partial\mathcal{W}} \frac{u(\boldsymbol{x}')e^{-i\omega|\boldsymbol{x}-\boldsymbol{x}'|}}{|\boldsymbol{x}-\boldsymbol{x}'|}\, dS' \tag{1.5}$$

and

$$-(\partial^2/\partial z^2 + \omega^2)A(\boldsymbol{x}) = i\omega E(\boldsymbol{x}). \tag{1.6}$$

Note that if the wire has length $L \neq 1$, then the nondimensionalized frequency ω corresponds to kL and the nondimensionalized radius a to the true wire radius divided by L.
The model (1.1), (1.2) is derived after making some further simplifying assumptions.

1. The wire is regarded as an open tube and hence $u(x, y, z) = 0$ for $(x, y, z) \in \partial\mathcal{W}$ and $z \in \{0, 1\}$.
2. Because the wire is thin, the (x, y)-dependence of the electric field on the surface of the wire is ignored: i.e., set $E(\boldsymbol{x}) = E(z)$ for $\boldsymbol{x} \in \partial\mathcal{W}$.
3. The final simplification concerns the current $u(\boldsymbol{x})$.

(a) The surface current is also assumed to be azimuthally symmetric; i.e., if $\boldsymbol{x} \in \partial\mathcal{W}$ then

$$u(x, y, z) = \frac{u(z)}{2\pi a}.$$

This implies that the term $|\boldsymbol{x} - \boldsymbol{x}'|$ in the integrand of (1.5) can be written as

$$\begin{aligned} |\boldsymbol{x}-\boldsymbol{x}'|^2 &= |(a\cos\phi, a\sin\phi, z) - (a\cos\phi', a\sin\phi', z')|^2 \\ &= 2a^2[1 - \cos(\phi - \phi')] + |z - z'|^2 \end{aligned}$$

for some ϕ, ϕ'. After some manipulation this yields $A(\boldsymbol{x}) = A_E(z)$ for $\boldsymbol{x} \in \partial\mathcal{W}$, where

$$A_E(z) = \int_0^1 K_E(z - z')u(z')\, dz'$$

is given in terms of the exact kernel K_E defined by (1.4).

(b) The reduced kernel formulation follows from regarding the current as acting on the wire's midline, instead of on the surface (but still applying the electric field boundary condition on its surface). In this case the integrand term in (1.5) is

$$|\boldsymbol{x} - \boldsymbol{x}'|^2 = a^2 + |z - z'|^2$$

when $\boldsymbol{x} \in \partial\mathcal{W}$, and it follows that $A(\boldsymbol{x}) = A_R(z)$ where

$$A_R(z) = \int_0^1 K_R(z - z')u(z')\,dz'$$

for K_R defined in (1.3).

Substituting A_E or A_R for A into (1.6) results in either the exact or reduced kernel forms of the IDE (1.1) respectively, where the right-hand side is $f(z) = i\omega E(z)$. The boundary conditions (1.2) follow immediately from assumption 1.

Note that the model used in the code NEC-4 [4] is for a closed tube and so includes an approximation of the wire end caps. In [4] the current is regarded as flowing along the wire's surface and the electric field boundary condition is applied on the axis, again leading to $A = A_R$ given in 3(b) above.

1.2. *Notation and Spaces*

We conclude this section with definitions of the function spaces and norms needed in the rest of the paper.

Suppose that $U \subset \mathbb{R}^m$ for $m = 1$ or 2 (in most of the text $U = (0, 1) \in \mathbb{R}$). For any $1 \le p < \infty$ let $L^p(U)$ be the set of complex valued functions g for which the integral of $|g|^p$ over U is finite, and define the L^2 norm

$$\|g\|_{L^2(U)} \equiv \left(\int_U |g(x)|^2\,dx\right)^{1/2}$$

in the usual way. The inner product of two functions f and $g \in L^2(U)$ is written as $\langle f, g\rangle$ and defined by

$$\langle f, g\rangle = \int_U f(x)\,g(x)\,dx.$$

Additionally let $H^1(U)$ be the Sobolev space of functions whose generalized derivatives up to order 1 belong to $L^2(U)$.

We also require fractional order Hilbert spaces based on $(0, 1)$. To this end, let $\Omega \in \mathbb{R}^2$ be a simply connected bounded domain whose boundary $\partial\Omega$ is smooth curve, and let $H^{1/2}(\partial\Omega)$ denote the (Sobolev) space [14]

$$H^{1/2}(\partial\Omega) \equiv \{u|_{\partial\Omega} : u \in H^1(\mathbb{R}^2)\}.$$

Let Γ with $\overline{\Gamma} \subset \partial\Omega$ be a connected piece of $\partial\Omega$ and set

$$\tilde{H}^{1/2}(\Gamma) \equiv \left\{u \in H^{1/2}(\partial\Omega) : \mathrm{supp}(u) \subset \Gamma\right\}$$

with associated norm $\|u\|_{\tilde{H}^{1/2}(\Gamma)}$ (see [14]). This space is used in Section 2.6 with $\Gamma = (0, 1)$. A different but equivalent definition of $\tilde{H}^{1/2}(0, 1)$ is given in [18].

2. NUMERICAL METHODS

This section contains a description of several different numerical methods for the problem (1.1), (1.2) with $K = K_E$. We shall only consider the exact kernel form of the problem from now on, and consequently drop the subscript $_{\mathrm{E}}$ and use K to denote the exact kernel (1.4).

We begin Section 2.1 with a description of the uniform mesh piecewise linear Galerkin approximation that Rynne [18] has shown to be convergent. When implemented naively this method is extremely slow, and we identify ways in which it can be made efficient enough to use in practice. However, it is still slower than similar collocation schemes and two of these are detailed in Sections 2.2 and 2.3. We also consider some more sophisticated finite element approximations (the p, hp, and adaptive h methods) in Sections 2.4–2.6, and we present and discuss numerical results for all these schemes in Section 4. Although using an adaptive method may appear superficially attractive, the work involved in computing the residual-based error estimate is such that the scheme cannot compete with a sensible nonadaptive hp method like that described in Section 2.5. This and other specific implementation details (how to evaluate and integrate the kernel (1.4) and how to perform the linear algebra) are described in Section 3. In all the methods that follow we use n to represent the number of subintervals of the wire (0, 1) used in the approximation, and N for the total number of unknowns.

2.1. *Piecewise Linear Galerkin*

Multiplying the IDE (1.1) by a test function $\psi(z) \in H^1(0,1)$ that satisfies $\psi(0) = \psi(1) = 0$ and integrating by parts gives

$$\langle f, \psi \rangle = \langle \mathbb{K}Du, D\psi \rangle - \omega^2 \langle \mathbb{K}u, \psi \rangle, \tag{2.1}$$

where $D = d/dz$, $\langle .,. \rangle$ denotes the L^2 inner product (described in Section 1.2), and the operator $\mathbb{K}$ is defined by

$$(\mathbb{K}\psi)(z) \equiv \int_0^1 K(z - z')\psi(z')\, dz'.$$

We use piecewise linear trial and test functions on a uniform mesh of size $h = 1/n$, defining $\phi_j(z) \equiv \phi(z/h - j)$ for

$$\phi(t) \equiv \begin{cases} 1 - |t| & \text{if } |t| < 1 \\ 0 & \text{otherwise.} \end{cases}$$

Expanding u in terms of these basis functions as

$$u(z') \approx U^{(h)}(z') \equiv \sum_{j=1}^{n-1} U_j \phi_j(z') \tag{2.2}$$

yields the linear system

$$(B - \omega^2 C)\boldsymbol{U} = \boldsymbol{f}, \tag{2.3}$$

where $\boldsymbol{U} = (U_1, \ldots, U_{n-1})^T$, $\boldsymbol{f} = (\langle f, \phi_1 \rangle, \ldots, \langle f, \phi_{n-1} \rangle)^T$, and the components of the matrices B and C are given by

$$B_{jk} = \langle \mathbb{K} D\phi_j, D\phi_K \rangle, \quad C_{jk} = \langle \mathbb{K}\phi_j, \phi_k \rangle.$$

The unknowns for the problem are the coefficients of $\boldsymbol{U}$, and so $N = n - 1$ for this approximation scheme.

Rynne [18] has shown that the scheme (2.3) converges to the solution of the exact problem as the mesh-size $h \to 0$, when exact integration is used to evaluate the components of B and C. In practice the component integrals cannot be found analytically and have to be approximated. It is important for this to be done efficiently since they are all three-dimensional (3D) integrals (because they involve K, which is itself an integral). Fortunately it is possible to use a simple 2D change of variables so that each component can be written as a 2D integral. We illustrate how this can be done by looking at the matrix B.

It is straightforward to show that $B_{j,k} = 2V_{j-k} - V_{j-k-1} - V_{j-k+1}$ for $j, k = 1 : n-1$, where

$$V_m \equiv \int_0^1 \int_0^1 K(h[x - x' + m])\, dx' dx,$$

and hence B is a symmetric Toeplitz matrix whose entries are given in terms of triple integrals. Using the change of variables $s = x - x'$, $s' = x + x'$ means that the V_m can be written as

$$V_m = \int_0^1 (1 - s)\{K(h[m + s]) + K(h[m - s])\}\, ds. \tag{2.4}$$

Therefore in order to compute B, it is necessary just to evaluate the n double integrals V_m, $m = 0 : n - 1$ in (2.4). The same procedure can be used to evaluate the components of C, which reduce to $C_{j,k} = W_{j-k}$ where

$$W_m = \int_0^2 \gamma(s)\{K(h[m + s]) + K(h[m - s])\}\, ds,$$

and γ is the twice continuously differentiable function defined by

$$\gamma(s) = \begin{cases} \left(\frac{s^3}{2} - s^2 + \frac{2}{3}\right) & \text{for } 0 \le s \le 1 \\ \frac{(2-s)^3}{6} & \text{for } 1 < s \le 2. \end{cases}$$

The integrals V_m and W_m can either be evaluated by a 1D quadrature formula that uses values of K (this involves first approximating the kernel K), or by using 2D quadrature for the double integral. Both approaches have been compared and the most efficient approach that we found is to evaluate K explicitly and use 1D adaptive quadrature (details are given in Section 3).

Thus the linear system (2.3) can be written as

$$A\boldsymbol{U} = \boldsymbol{f},$$

where $A = B - \omega^2 C$ is a symmetric Toeplitz $N \times N$ matrix. As discussed in Section 3.4, the fact that A is Toeplitz means that Levinson's algorithm can be used to solve the linear system efficiently for large N (see also [22]).

The method is applied by successively halving the mesh size (i.e., doubling n) until two approximate solutions are sufficiently close together in the L^2 norm. The algorithm for the method is summarized below. Other Galerkin schemes have been considered for the reduced kernel form of (1.1) in [5, 12].

ALGORITHM 1

1. Choose an initial uniform grid with n elements (e.g., use $n = 4$) and a tolerance τ.
2. Set the mesh spacing to be $h = 1/n$ and compute the approximate solution $U_{\text{old}} = U^{(h)}$ on this mesh.
3. Set $n := 2n$, $h = 1/n$ and compute the approximate solution $U_{\text{new}} = U^{(h)}$ on this finer mesh.
4. If $\|U_{\text{new}} - U_{\text{old}}\|_{L^2(0,1)} < \tau$ then terminate the calculation. Otherwise set $U_{\text{old}} := U_{\text{new}}$ and return to 3.

2.2. *Piecewise Linear Collocation*

Collocation is another popular way of solving (1.1) and often has the advantage that the matrix elements are faster to evaluate than for the corresponding Galerkin scheme [5, 12]. However, there are no convergence results for collocation schemes of the type given for Galerkin schemes in [18, 19].

The first collocation scheme we consider also uses the expansion (2.2) for u in terms of the piecewise linear basis functions ϕ_j given in the previous subsection (and so again $N = n - 1$, where the mesh spacing is $h = 1/n$). This is substituted into the IDE (1.1), and a second central difference is used to approximate the second derivative. The resulting equation is forced to be satisfied at the node points $z_j = jh$ to yield

$$\sum_{k=1}^{n-1} \left\{ \frac{1}{h^2} \int_0^1 [K(z_{j+1} - z') - (2 - h^2\omega^2) K(z_j - z') + K(z_{j-1} - z')]\, \phi_k(z')\, dz' \right\} U_k = f(z_j)$$

for $j = 1 : n - 1$. Once the integrals are approximated (see Section 3.2 for details) this scheme also reduces to a symmetric Toeplitz linear system $A\boldsymbol{U} = \boldsymbol{f}$ (for a different matrix A and where here $\boldsymbol{f} = (f(z_1), \ldots, f(z_n)^T)$. The method is again applied with a uniform (halving) mesh refinement strategy, and is identical to Algorithm 1.

2.3. *Piecewise Trigonometric Collocation*

Other basis functions have often been used to solve (the reduced kernel form of) (1.1) [5, 4, 12], and one of the most popular of these is the set of piecewise trigonometric basis functions ψ_j defined on a uniform mesh with spacing $h = 1/n$ by

$$\psi_j(z) = \begin{cases} a_j + b_j \sin \omega(z - z_{j-1/2}) + c_j \cos \omega(z - z_{j-1/2}) & \text{if } z \in [z_{j-1}, z_j) \\ 0 & \text{otherwise,} \end{cases} \tag{2.5}$$

where $z_k = kh$ and the constants a_j, b_j and c_j, $j = 1 : n$ are the unknowns for the problem (so $N = 3n$ for this type of approximation).

The current u is expanded in terms of these basis functions as

$$u(z') \approx \sum_{j=1}^{N-1} \psi_j(z').$$

The two boundary conditions (1.2) are enforced by

$$\left.\begin{aligned} a_1 - b_1 \sin(\omega h/2) + c_1 \cos(\omega h/2) &= 0 \\ a_n + b_n \sin(\omega h/2) + c_n \cos(\omega h/2) &= 0 \end{aligned}\right\} \tag{2.6}$$

and both the approximate current and its first derivative are forced to be continuous for all $z' \in (0, 1)$ by assuming the unknowns to satisfy

$$\left.\begin{aligned} a_j + b_j \sin(\omega h/2) + c_j \cos(\omega h/2) &= a_{j+1} - b_{j+1} \sin(\omega h/2) + c_{j+1} \cos(\omega h/2) \\ b_j \sin(\omega h/2) - c_j \cos(\omega h/2) &= -b_{j+1} \sin(\omega h/2) - c_{j+1} \cos(\omega h/2) \end{aligned}\right\} \tag{2.7}$$

for $j = 1 : n - 1$. Together (2.6) and (2.7) comprise $2n$ equations, and the remaining n equations are obtained from the IDE (1.1). The approximate current is substituted into the IDE to give

$$\begin{aligned} -\sum_{j=1}^{n} (\omega^2 + D^2) \int_{z_{j-1}}^{z_j} K(z - z') \big(a_j + b_j \sin \omega\big(z' - z_{j-1/2}\big) \\ + c_j \cos \omega\big(z' - z_{j-1/2}\big)\big)\, dz' = f(z), \end{aligned}$$

where $D = d/dz$. Integrating by parts separately on each subinterval (using the continuity of u and Du on $(0, 1)$) then gives

$$K(z - 1) Du(1) - K(z) Du(0) - \omega^2 \sum_{j=1}^{n} a_j \int_{z_{j-1}}^{z_j} K(z - z')\, dz' = f(z), \tag{2.8}$$

where note that the integrand is particularly simple because the sine and cosine terms in the basis function are annihilated by the Helmholtz operator in (1.1). This equation is assumed to hold at the patch midpoints $z = z_{j-1/2}$, $j = 1 : n$ to give a system of n equations involving the integrals

$$\int_{-1/2}^{1/2} K(h[m - s])\, ds, \quad m = 0 : n - 1.$$

The full linear system comprises (2.6), (2.7), and (2.8) at the patch midpoints, and ways in which it can be solved most efficiently are discussed in Section 3.4. This collocation scheme is also applied with a uniform (halving) mesh refinement strategy and hence is also that given in Algorithm 1. Numerical results are presented in Section 4.

We note that the alternative representation of the basis functions in the piecewise trigonometric scheme as

$$\psi_j(z) = a'_j + b'_j \sin \omega(z - z_{j-1/2}) + c'_j(\cos \omega(z - z_{j-1/2}) - 1) \quad \text{if } z \in [z_{j-1}, z_j) \tag{2.9}$$

is used in NEC-4 [4] to deal with rounding error problems in low-frequency simulations. This is because $\cos\omega(z - z_{j-1/2}) \approx 1$ at low frequency and the independence of the a_j and $c_j \cos\omega(z - z_{j-1/2})$ terms in (2.5) is destroyed by rounding errors. In contrast, the independence of the a'_j and $c'_j(\cos\omega(z - z_{j-1/2}) - 1)$ terms in (2.9) is maintained for small ω if the identity $\cos x - 1 = -2\sin^2(x/2)$ is used in numerical evaluation to avoid rounding errors.

2.4. *Uniform p-Method*

The three methods discussed above have all been refined by reducing the mesh size according to Algorithm 1, keeping the basis functions unchanged apart from a scaling. The p-method instead involves using a fixed grid throughout the calculation and is refined by increasing the degree of the piecewise polynomial basis functions used.

The initial approximation ($p = 1$) is the piecewise linear Galerkin solution of Section 2.1 on a uniform grid with mesh-size $h = 1/n$ (where e.g., $n = 8$). Higher order approximations are obtained by augmenting the piecewise linear test and basis functions ϕ_j, $j = 1 : n$ by higher degree polynomial bubble functions. These are taken to be anti-derivatives of Legendre polynomials defined on the interval $[-1, 1]$ and then mapped onto each element $[z_{j-1}, z_j]$ (where again $z_k = kh$). The $p - 1$ bubble functions for the method of degree $p > 1$ are defined (on the canonical interval $[-1, 1]$) by

$$\varphi_k(s) = \sqrt{k + 1/2} \int_{-1}^{s} P_k(\xi)\, d\xi, \quad \text{for } k = 1 : p - 1,$$

where P_k is the Legendre polynomial of degree k. Note that each of these are zero at the endpoints $s = \pm 1$. Other nice properties of these functions are described in [9].

The $p - 1$ bubble functions associated with the jth interval are $\varphi_{j,k}(z)$ for $k = 1 : p - 1$, where

$$\varphi_{j,k}(z) = \begin{cases} \varphi_k(2z/h + 1 - 2j) & \text{if } z \in [z_{j-1}, z_j] \\ 0 & \text{otherwise.} \end{cases}$$

For ease of notation we relabel the "hat" functions of Section 2.1 as $\phi_j = \varphi_{j,0}$, and we expand the current u in terms of the hat and bubble basis functions as

$$u(z') \approx U^{(p)}(z') \equiv \sum_{j=1}^{n} \sum_{k=0}^{p-1} U_{j,k} \varphi_{j,k}(z').$$

Note that each of the basis functions is continuous on the whole interval $[0, 1]$ and that this approximation automatically satisfies the boundary conditions (1.2). The unknowns are the $N = np$ constants U_{jk} for $j = 1 : n, k = 0 : p - 1$. Taking the test function ψ in the inner product (2.1) with each of the basis functions $\varphi_{j,k}$ then yields an $N \times N$ linear system of the form

$$A\boldsymbol{U} = \boldsymbol{f}, \tag{2.10}$$

where $\boldsymbol{U} = (U_{1,0}, \ldots, U_{1,p-1}, U_{2,0}, \ldots, U_{2,p-1}, \ldots, U_{n,0}, \ldots, U_{n,p-1})^T$, and $\boldsymbol{f} = (f_{1,0}, \ldots, f_{n,p-1})^T$ for $f_{j,k} = \langle f, \varphi_{j,k} \rangle$.

The algorithm is summarized below. It uses a stopping criterion based on the L^2 norm of the difference between two successive approximations but this is not the only possibility.

ALGORITHM 2

1. Choose a fixed uniform grid with mesh-spacing h (e.g., $h = 1/8$), and a tolerance τ, and set $p = 1$.
2. Compute the approximate solution $U_{\text{old}} = U^{(1)}$ using the piecewise linear Galerkin approximation on this grid.
3. Set $p := p + 1$, and compute the approximate solution $U_{\text{new}} = U^{(p)}$ using the hat and bubble basis functions of degree up to p as described above.
4. If $\|U_{\text{new}} - U_{\text{old}}\|_{L^2(0,1)} < \tau$, then terminate the calculation. Otherwise set $U_{\text{old}} := U_{\text{new}}$ and return to 3.

2.5. *Nonadaptive hp-Method*

The hp-method involves refining both the size of (some) grid elements and the degree of the piecewise polynomial basis functions according to a predetermined strategy. The initial approximation ($p = 1$) is again the piecewise linear Galerkin solution of Section 2.1 on a uniform grid with mesh size $h = 1/n$ (where e.g., $n = 4$ or 8). The strategy used to generate successive approximations of increasing accuracy should be based on knowledge of properties of the exact solution u. Because u behaves like a square root near the ends of the wire but is otherwise smoother [20], a sensible strategy is to subdivide the elements at the two ends of the mesh (i.e., those that contain the points 0 and 1), and increase the degree of the bubble basis functions in all other elements by 1.

Suppose that the current mesh is level $k-1$ and has $n^{(k-1)}$ (not necessarily uniform) intervals with mesh points

$$z_j^{(k-1)}, \quad j = 0 : n^{(k-1)}$$

where

$$z_0^{(k-1)} = 0, \quad z_{n^{(k-1)}}^{(k-1)} = 1.$$

Denote the jth interval by $\Gamma_j^{(k-1)} = [z_{j-1}^{(k-1)}, z_j^{(k-1)}]$ and suppose that the degree of basis functions used on the interval $\Gamma_j^{(k-1)}$ is $p_j^{(k-1)}$. We always use degree 1 polynomial basis functions at the two ends of the wire, so that $p_1^{(k-1)} = p_{n^{(k-1)}}^{(k-1)} = 1$ for any k. The refinement strategy to obtain the level k mesh-points and basis functions is to choose $\lambda \in (0, 1)$ and then

- set $n^{(k)} = n^{(k-1)} + 2$;
- if $j = 1$,

$$\Gamma_1^{(k)} = \left[0, \lambda z_1^{(k-1)}\right], \quad p_1^{(k)} = 1$$
$$\Gamma_2^{(k)} = \left[\lambda z_1^{(k-1)}, z_1^{(k-1)}\right], \quad p_2^{(k)} = p_1^{(k-1)} + 1 = 2;$$

- if $1 < j < n^{(k-1)}$,

$$\Gamma_{j+1}^{(k)} = \Gamma_j^{(k-1)}, \quad p_{j+1}^{(k)} = p_j^{(k-1)} + 1;$$

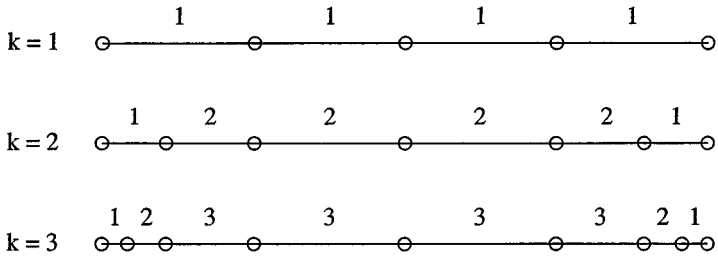

FIG. 2.1. The first three levels of mesh and basis refinement for the (nonadaptive) *hp*-method starting with a uniform mesh with four intervals when $k = 1$. The circles represent mesh points and the degree of polynomial basis function used in each interval is written above it.

- if $j = n^{(k-1)}$,

$$\Gamma^{(k)}_{n^{(k)}-1} = \left[z^{(k-1)}_{n^{(k-1)}}, \mu^{(k)}\right] \quad p^{(k)}_{n^{(k)}-1} = p^{(k-1)}_{n^{(k-1)}} + 1 = 2$$
$$\Gamma^{(k)}_{n^{(k)}} = [\mu^{(k)}, 1] \quad p^{(k)}_{n^{(k)}} = 1,$$

where $1 - \mu_{(k)} = \lambda[1 - z^{(k-1)}_{n^{(k-1)}}]$.

In calculations we used $\lambda = 0.17$ motivated by [7] and numerical experiments to obtain nearly optimal (exponential) convergence. The first three levels starting from a uniform mesh with $n^{(1)} = 4$ are shown in Fig. 2.1.
The method is also applied repeatedly until two successive approximate solutions are sufficiently close in the $L^2(0, 1)$ norm.

2.6. *Adaptive h-Method*

The final scheme that we consider is adaptive—it uses the piecewise linear Galerkin algorithm of Section 2.1 on a nonuniform mesh that is chosen in order to minimize an estimate of the error in the computed solution. As in the previous subsection we denote the mesh intervals by Γ_j, $j = 1 : n$, and set h_j to be the length of the interval Γ_j. This adaptive strategy needs an a posteriori error estimate η, and a reliable estimate based on the residual is given by Rynne in [18],

$$\eta(\Gamma_j, U) = h_j^{1/2} \|\mathbb{L}U - f\|_{L^2(\Gamma_j)}, \tag{2.11}$$

where $\mathbb{L}$ is the operator corresponding to the IDE (1.1), i.e.,

$$\mathbb{L}u(z) \equiv -\frac{d}{dz}\int_0^1 K(z - z')\frac{du}{dz'}(z')\,dz' - \omega^2 \int_0^1 K(z - z')u(z')\,dz'.$$

Rynne shows that the difference between the exact solution $u(z)$ and its Galerkin approximation $U(z)$ on a particular mesh satisfies

$$\|u - U\|^2_{\tilde{H}^{1/2}(0,1)} \le C \sum_{j=1}^{n} \eta(\Gamma_j, U)^2,$$

where C is a constant independent of the mesh.

The mesh and solution are computed according to the following algorithm, which again uses a stopping criterion based on the L^2 norm.

ALGORITHM 3

1. Choose an initial coarse mesh $\Gamma_j^{(1)}$, $j = 1 : n^{(1)}$ (e.g., set $n^{(1)} = 4$ and let each $h_j = 1/n^{(1)}$), and a tolerance τ and initialize the level to $k = 1$.
2. Compute the approximate solution $U_{\text{old}} = U^{(1)}$ using the piecewise linear Galerkin approximation on the coarse mesh.
3. For each $j = 1 : n^{(k)}$ compute $\eta_j^{(k)} \equiv \eta(\Gamma_j^{(k)}, U^{(k)})$ and set $\eta^{(k)} = \max_{1 \le l \le n^{(k)}} \eta_l^{(k)}$.
4. Refine (halve) the element $\Gamma_j^{(k)}$ if $\eta_j^{(k)} \ge 0.5\eta^{(k)}$ to obtain a new mesh $\Gamma_j^{(k+1)}$, $j = 1 : n^{(k+1)}$.
5. Set $k := k + 1$, and compute the Galerkin solution $U_{\text{new}} = U^{(k)}$ on the current mesh $\{\Gamma_j^{(k)}\}$.
6. If $\|U_{\text{new}} - U_{\text{old}}\|_{L^2(0,1)} < \tau$, then to terminate the calculation. Otherwise set $U_{\text{old}} := U_{\text{new}}$ and return to 3.

Note that the estimate η in (2.11) must be computed on any given mesh. This is computationally intensive since it involves the evaluation of 3D integrals. Ways in which η can be approximated efficiently are discussed in Section 3.5.

3. IMPLEMENTATION

This section contains a description of some of the specific implementation details for the algorithms of Section 2. One of the most important and difficult parts of any of these algorithms is the evaluation of the exact kernel (1.4) and its integrals weighted by polynomials. We begin with two sections describing how this can be done both accurately and efficiently; the first considers the efficient evaluation of (1.4) and the second deals with the integrals that are needed to construct the coefficient matrix for each algorithm. We next look at efficient techniques for solving the various linear systems that arise, and in the final section we examine ways in which the computational effort involved in approximating the error estimator η used in the h-adaptive algorithm of Section 2.6 can be reduced. We note that [4] contains a discussion of implementation issues for integrals involving the reduced kernel (1.3).

3.1. *Evaluation of the Exact Kernel*

We first rescale the exact kernel (1.4) with respect to the wire radius, rewriting it as

$$K(\rho) = \frac{1}{4a\pi^2} F(\rho/(2a), 2a\omega),$$

where

$$F(\lambda, \nu) \equiv \int_0^{\pi/2} \frac{e^{-i\nu R}}{R}\, d\theta, \tag{3.1}$$

$R = R(\lambda, \theta) \equiv \sqrt{\lambda^2 + \sin^2\theta}$, the scaled distance variable is

$$\lambda = \rho/(2a),$$

and the scaled frequency is

$$\nu = 2a\omega.$$

To establish some properties of the scaled kernel F we follow [16, 23, 24] and split it into two parts,

$$F(z,\nu) = \int_0^{\pi/2} \frac{1}{R}\, d\theta + \int_0^{\pi/2} \frac{(e^{-i\nu R}-1)}{R}\, d\theta,$$

noting that the second term is bounded (its integrand is bounded) and that the first term can be written as

$$\int_0^{\pi/2} \frac{1}{R}\, d\theta = \frac{1}{\sqrt{\lambda^2+1}}\,\mathrm{EllipticK}\left(\frac{1}{\sqrt{\lambda^2+1}}\right),$$

where EllipticK is the complete elliptic integral of the first kind (there is no simple expression for the second term). This yields the well-known result (see, e.g., [16, 11])

$$K(\rho) = O(\ln|\rho|) \quad \text{as } \rho \to 0,$$

since EllipticK has a logarithmic singularity as its argument approaches 1. More details of the properties of K can be found in [18, 20].

This singularity is the main source of difficulty in evaluating the scaled kernel F. The degree of difficulty in calculating F at a value of λ (for fixed ν) is roughly correlated with the ease or difficulty in computing EllipticK $(1/\sqrt{\lambda^2+1})$ (i.e., hard when λ is small and relatively easy for large λ). There are also problems when ν is large, since then the kernel is highly oscillatory. We have found that the most efficient strategy for evaluating F is to treat the small and large λ cases separately, combining them in a single kernel evaluation subroutine. We begin with a description of the easier case.

Large λ. We first rewrite the scaled kernel (3.1) as

$$F(\lambda,\nu) = e^{-i\nu R_*} \int_0^{\pi/2} \frac{e^{-i\nu(R-R_*)}}{R}\, d\theta, \tag{3.2}$$

where R_* is the midpoint value $\sqrt{\lambda^2+1/2}$. An accurate estimate of F for large λ can be obtained by applying the composite trapezoidal rule with M intervals to the integral (3.2), repeatedly doubling M until the relative difference between two successive approximations is smaller than some predefined tolerance τ (typically $\tau = 10^{-12}$). Very accurate results can be obtained by using moderately low values of M, as shown in Fig. 3.1 (the accuracy depends on the scaled frequency ν). This is presumably because away from $\lambda = 0$ the integrand is a smooth, periodic (in θ) function. The $e^{-i\nu R_*}$ term is extracted to improve the accuracy of computing the complex exponential term in the integrand when the scaled frequency ν is large. We also use the identity

$$R - R_* = -\frac{\cos(\theta/2)}{2(R+R_*)}$$

to reduce rounding error problems when λ is very large, since the direct calculation of $R - R_*$ in standard 64-bit arithmetic gives only about $16 - 2\log_{10}\lambda$ significant digits of accuracy for $\lambda \in [1, 10^8]$. All precision is lost for $\lambda > 10^8$.

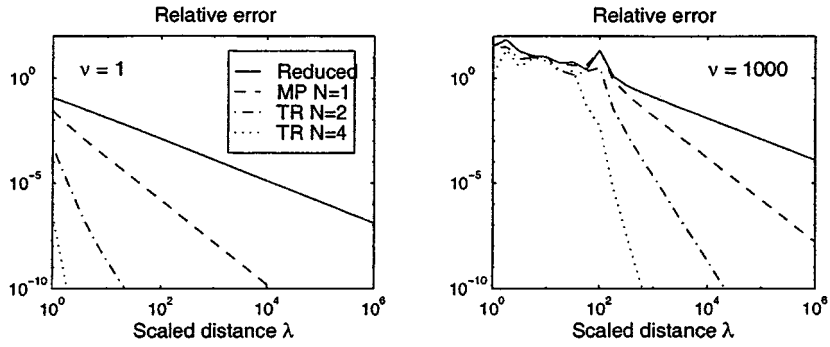

FIG. 3.1. Relative error against scaled distance λ for various simple approximations of the exact kernel (1.4). The approximations are: reduced kernel (1.3) (-), single interval midpoint rule (3.3) (- -), trapezoidal rule with 2(-.), and 4 (:) subintervals. Results are shown for a small- and large-scaled frequency ν.

Various authors [5, 16] use the reduced kernel (1.3) to approximate the exact kernel (1.4), and Fig. 3.1 shows that it is a good match as $\lambda \to \infty$. However, it is interesting to note from the figure that the approximation

$$K(\rho) \approx \frac{1}{4\pi} \frac{e^{-i\omega\sqrt{\rho^2+2a^2}}}{\sqrt{\rho^2+2a^2}} \tag{3.3}$$

obtained by a single interval midpoint rule approximation of the integral (3.2) gives even better results for the same computational effort at large λ.

Small λ. When the scaled distance argument λ is small, the logarithmic singularity in the kernel dominates its computation. The most efficient way we have found to evaluate it is to treat the real and imaginary parts separately, writing the scaled kernel as

$$F(\lambda, \nu) = \underbrace{\int_0^{\pi/2} \frac{\cos(\nu R)}{R}\, d\theta}_{=F_1(\lambda,\nu)} - i \underbrace{\int_0^{\pi/2} \frac{\sin(\nu R)}{R}\, d\theta}_{=F_2(\lambda,\nu)}. \tag{3.4}$$

The term $F_2(z, \nu)$ can be efficiently evaluated to any required accuracy by the composite trapezoidal rule, which again exhibits superconvergence as the number of subdivisions is increased.

We needed to use a more sophisticated approximation strategy for F_1. As noted above,

$$F_1(\lambda, 0) = \frac{1}{\sqrt{\lambda^2+1}} \,\mathrm{EllipticK}\left(\frac{1}{\sqrt{\lambda^2+1}}\right).$$

A standard form for the elliptic integral is

$$\mathrm{EllipticK}(l) = \int_0^1 \frac{dt}{\sqrt{(1-t^2)(1-l^2t^2)}} \quad \text{for } 0 \le l \le 1, \tag{3.5}$$

and hence $\mathrm{EllipticK}(0) = \pi/2$. Care is required in evaluating this integral when the argument $l \approx 1$, and a standard method used in mathematical software libraries for evaluating

EllipticK uses the Landen transform

$$k = \frac{1-\sqrt{1-l^2}}{1+\sqrt{1-l^2}}, \quad t := \frac{(1+k)t}{1+kt^2}$$

(see [8, p. 250] for details) in (3.5) to obtain the identity

$$\text{EllipticK}(l) = (1+k)\text{EllipticK}(k).$$

Note that $0 < k < l < 1$, and so the transform converts the elliptic integral to one with a smaller argument. This identity can be used repeatedly to obtain

$$\text{EllipticK}(k_0) = (1+k_1)(1+k_2)\dots(1+k_n)\text{EllipticK}(k_n), \quad k_{j+1} = \frac{1-\sqrt{1-k_j^2}}{1+\sqrt{1-k_j^2}}$$

and, since $k_n \to 0$ as $n \to \infty$, the process is terminated in floating point arithmetic when k_n is small enough to ensure that $fl(1+k_n) = 1$, giving

$$fl(\text{EllipticK}(k_0)) = (1+k_1)(1+k_2)\dots(1+k_n)\frac{\pi}{2}$$

to machine precision. This approach is also called the method of arithmetic-geometric means [1, Section 17.6].

We have found that the Landen transform is also an efficient method for calculating the real part F_1 of the kernel in (3.4). Using the change of variable $t = \cos\theta$ gives

$$F_1(\lambda, \nu) = \frac{1}{k_0}\int_0^1 \frac{g_0(t)\,dt}{\sqrt{(1-k_0^2t^2)(1-t^2)}},$$

where

$$k_0 = \frac{1}{1+z^2} \quad \text{and} \quad g_0(t) = \cos\left(\frac{\omega}{k_0}\sqrt{1-k_0^2t^2}\right).$$

We then use the Landen transform

$$k_{j+1} = \frac{1-\sqrt{1-k_j^2}}{1+\sqrt{1-k_j^2}}, \quad g_{j+1}(t) = g_j\left(\frac{1+k_{j+1}t}{1+k_{j+1}t^2}\right), \quad t := \frac{1+k_{j+1}t}{1+k_{j+1}t^2}$$

for $j = 0, 1, \dots$ to obtain

$$\begin{aligned} F_1(\lambda, \nu) &= k_0\int_0^1 \frac{g_0(t)\,dt}{\sqrt{(1-k_0^2t^2)(1-t^2)}} \\ &= k_0(1+k_1)\int_0^1 \frac{g_1(t)\,dt}{\sqrt{(1-k_1^2t^2)(1-t^2)}} \\ &= k_0(1+k_1)\dots(1+k_n)\int_0^1 \frac{g_n(t)\,dt}{\sqrt{(1-k_n^2t^2)(1-t^2)}}, \end{aligned}$$

where the sequence k_j is the same as for EllipticK. The process again terminates when $fl(1+k_n)=1$ giving

$$fl(F_1(\lambda,\nu)) = k_0\prod_{j=1}^{n}(1+k_j)\int_0^1 \frac{g_n(t)\,dt}{\sqrt{1-t^2}} = k_0\prod_{j=1}^{n}(1+k_j)\int_0^{\pi/2} g_n(\sin\theta)\,d\theta$$

to machine precision, and the final integral can be approximated using the composite trapezoidal rule.

Switching between small and large λ methods. We carried out numerical tests to measure flop counts for both the small and large λ evaluation methods described above, over a large range of values of λ and for a wide range of scaled frequencies ν. In general both schemes are roughly equally efficient for the middle values of λ, and for larger λ the direct (large λ) method is more reliable (because it takes care of rounding error problems) and is about five times faster than the Landen transform (small λ) method. For small λ the direct method is extremely inefficient compared to the Landen transform method. Taking into account the effects of the scaled frequency ν on the flop count for the calculations, we use the Landen transform method when

$$\lambda \le \lambda_{\text{switch}}(\nu) \equiv \frac{10}{10+|\nu|}$$

and the large λ method otherwise. This is simply a rule of thumb, safeguarded by favoring the Landen transform (small λ) method for the reasons give above.

3.2. *Efficient Evaluation of the Integrals in the Coefficient Matrices*

The coefficient matrices generated by the numerical methods described in Section 2 require the evaluation of integrals of the form

$$I_P = \int_0^1 K(sh+\sigma)P(s)\,ds, \tag{3.6}$$

where $P(s)$ is a low-order polynomial, $\sigma = \pm z_j$ (node point) or $\pm z_{j+1/2}$ (element midpoint), and h is the length of the mesh interval. The exact details depend on the scheme, and we note that none of the integrals required for any of the schemes gives a simple closed form expression. We must therefore use numerical approximation.

The main problem in approximating (3.6) is again the singular behavior of the kernel function K. There are many ways to approximate (3.6), ranging from a direct approximation by a 1D adaptive quadrature package, such as D01AJF from NAG, to substitution of the kernel definition (1.4) into (3.6) followed by a direct approximating of the resulting 2D integral by a 2D adaptive quadrature method, such as the NAG routine D01FCF. An intermediate strategy is described in [23].

After much experimentation with these various approaches, we conclude that an efficient way to approximate (3.6) for all the cases required is to split it into a singular and a nonsingular integral and use adaptive 1D quadrature on each part separately. We first write

$$I_P = \underbrace{\int_0^1 (K(sh+\sigma) - K_0(sh+\sigma))P(s)\,ds}_{=I_{P1}} + \underbrace{\int_0^1 K_0(sh+\sigma)P(s)\,ds}_{=I_{P2}},$$

where

$$K_0(\rho) = \begin{cases} \frac{1}{2\pi^2 a}\left(-\log\left|\frac{\rho}{2a}\right| + \left|\frac{\rho}{2a}\right| - 1\right), & |\rho| \le 2a \\ 0, & \text{otherwise} \end{cases}$$

is continuously differentiable for all $|\rho| \neq 0$ (including $|\rho| = 2a$) and has the same leading order behavior as $K(\rho)$ as $\rho \to 0$. The integral I_{P2} is simple enough to evaluate explicitly, and the integrand of I_{P1} is (following [18, 20]) continuously differentiable, and so I_{P1} can be tackled directly by an adaptive 1D quadrature routine, such as the NAG routine D01AJF.

3.3. *Matrix Assembly in the p-, hp-, and Adaptive h-Schemes*

In the p-method of Section 2.4 the mesh is fixed and enhancing the approximation from degree p to $p+1$ only requires the additional calculation of

$$(K\phi_{j,p}, \phi_{l,m}) \quad \text{for } j, l = 1, \ldots, n, m = 0, \ldots, p$$

rather than the recalculation of the complete matrix. Thus the coefficient matrix is simply extended as the degree of the approximation in each element increases.

The mesh in the adaptive h-scheme is refined at the ends as the scheme progresses, but most of the elements remain the same from step to step and so their contribution to the coefficient matrix does not change. To avoid duplicated effort we store the element submatrices, modify them to cope with the local changes in the mesh, and assemble the full coefficient matrix from them at each step.

The hp-scheme uses both of these strategies.

3.4. *Linear Algebra*

The end result of each of the methods described in Section 2 is a linear system of equations

$$A\boldsymbol{U} = \boldsymbol{f},$$

where the vector $\boldsymbol{U}$ contains the unknown coefficients used in the approximate solution, and vector $\boldsymbol{f}$ is generated from the known incident field function in (1.1). The details depend on the scheme. In general there is no special structure to be exploited in the linear algebra required for the nonuniform grid schemes, and we use a standard Gaussian elimination solver (the MATLAB backslash (\) operator) in this case. This requires $O(N^3)$ flops and storage for the N^2 (complex) entries in the matrix. However, the uniform grid schemes do have a special structure which can be used both to speed up the solution of this linear system from $O(N^3)$ to $O(N^2)$ flops, and to reduce the storage required from $O(N^2)$ to $O(N)$. We examine this below.

The uniform grid piecewise linear Galerkin and collocation methods of Sections 2.1 and 2.2 result in a dense system of equations with symmetric Toeplitz structure. That is, the $N \times N$ coefficient matrix has entries

$$A_{j,k} = \alpha_{|j-k|}.$$

The complex constants $\alpha_s, s = 0, \ldots, N-1$ depend on the method used. Symmetric Toeplitz systems can be solved using Levinson's algorithm [2, 6, Section 4.7] in $O(N^2)$

flops, compared to $O(N^3)$ flops by standard Gaussian elimination for dense matrices. Also, the storage required is only $O(N)$. The Levinson algorithm requires fewer flops than standard Gaussian elimination for all nontrivial systems ($N > 1$), but in our experiments using MATLAB on a standard Sun Ultra 5 workstation, the Levinson algorithm was slower (that is in execution time) than the standard MATLAB dense linear system solve command for systems of size less than about $N = 150$. The difference is due to the way MATLAB's built-in linear algebra routines make use of the hardware and vary from system to system. The choice of which algorithm to use at a given system size is thus likely to be both software and hardware dependent, but the Levinson algorithm will always require fewer floating point operations.

Note that if these schemes are applied on a *curved* wire, then the resulting linear system is not Toeplitz in general, even on a uniform mesh, and so the solution times and storage will be $O(N^3)$ and $O(N^2)$, respectively.

In the uniform grid piecewise trigonometric collocation scheme of Section 2.3 we order the unknowns a_j, b_j, c_j into the solution vector as $\boldsymbol{U} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$. The $3n \times 3n$ coefficient matrix A then has a dense $n \times n$ block multiplying the a_j coefficients, while the rest of the matrix is sparse. The system can be condensed by eliminating $\boldsymbol{b}$ and $\boldsymbol{c}$, but the dense $n \times n$ matrix that results does not have a nice Toeplitz structure. Our experiments in MATLAB indicate that it is faster to feed the full $3n \times 3n$ mixed dense-sparse system to the MATLAB sparse system solver than to eliminate $\boldsymbol{b}$ and $\boldsymbol{c}$. However, the flop counts for both approaches are essentially the same, with a computational cost of $O(N^3)$ (where the number of unknowns $N = 3n$).

3.5. *Evaluating the Error Estimator for the Adaptive h-Method*

The adaptive h-method of Section 2.6 is based on an error estimator which requires calculation of the norm of the residual over each mesh element Γ_j to obtain the quantities

$$\eta(\Gamma_j, U) = h_j^{1/2} \|\mathbb{L}U - f\|_{L^2(\Gamma_j)},$$

for $j = 1 : n$, where $\mathbb{L}$ is the operator corresponding to the IDE (1.1), i.e.,

$$\mathbb{L}u(z) \equiv -\frac{d}{dz}\int_0^1 K(z - z')\frac{du}{dz'}(z')\,dz' - \omega^2 \int_0^1 K(z - z')u(z')\,dz'.$$

The approximate solution $U(z)$ for this scheme is a piecewise linear function on the (nonuniform) grid with nodes z_j, and hence the first part of $\mathbb{L}U(z)$, is

$$\begin{aligned}\frac{d}{dz}\int_0^1 K(z - z')\frac{dU}{dz'}(z')\,dz' &= \sum_{j=1}^{n}\left(\frac{U_j - U_{j-1}}{z_j - z_{j-1}}\right)[K(z - z_j) - K(z - z_{j-1})] \\ &= \sum_{j=0}^{n} W_j K(z - z_j), \qquad (3.7)\end{aligned}$$

where $U_j = U(z_j)$ and the weights are

$$W_0 = -\left(\frac{U_1 - U_0}{z_1 - z_0}\right), \quad W_n = \left(\frac{U_n - U_{n-1}}{z_n - z_{n-1}}\right), \quad W_j = \left(\frac{U_j - U_{j-1}}{z_j - z_{j-1}}\right) - \left(\frac{U_{j+1} - U_j}{z_{j+1} - z_j}\right)$$

for $j = 1 : n-1$. The integral in the second part of $\mathbb{L}U(z)$ can be split into contributions over each mesh element, but this is not necessary for our discussion and so we write

$$\mathbb{L}U(z) = -\sum_{j=0}^{n} W_j K(z - z_j) - \omega^2 \int_0^1 K(z - z')U(z')\,dz', \tag{3.8}$$

and note that this function has logarithmic singularities at each node point $z = z_j$ because $K(\rho) = O(\log|\rho|)$ as $\rho \to 0$.

If we work to the same accuracy as in computing the coefficient matrix, the computational effort required to obtain the value of the residual at a single point z can be large. From (3.8) we see that this requires $n+1$ evaluations of the kernel, and an integral of the kernel weighted by the approximate solution. This could take approximately the same time as evaluating a complete row of the coefficient matrix. This calculation has to be repeated for many different values of z within each element to obtain an estimate of $\|\mathbb{L}U - \boldsymbol{f}\|_{L^2(\Gamma_j)}$ by quadrature, resulting in a computational cost for the error indicator function η which is far in excess of the cost of finding the approximate solution U on a given mesh. Clearly it is too expensive and unnecessary to work to high precision when evaluating the error estimate. An accuracy of 1% (or even 10%) in evaluating η is sufficient for the mesh adaptation algorithm. We outline our approach below.

First we replace direct (and expensive) evaluation of the kernel function K by the approximation

$$\tilde{K}(\rho) \equiv \frac{1}{4\pi} \frac{e^{-i\omega\sqrt{\rho^2 + 2a^2}}}{\sqrt{\rho^2 + 2a^2}} + I_1(\rho),$$

where $I_1(\rho)$ is a piecewise linear function designed so that $|K(\rho) - \tilde{K}(\rho)|/|K(\rho)| < 10^{-3}$. The choice of the first term in $\tilde{K}$ comes from the results of Section 3.1, where it is seen to be a good approximation in its own right for large values of $\rho/(2a)$. Secondly we approximate

$$\int_0^1 K(z - z')U(z')\,dz' \approx I_2(z),$$

where I_2 is a piecewise linear function satisfying

$$I_2(z_j) = \int_0^1 \tilde{K}(z_j - z')U(z')\,dz'$$

at each of the space node points z_j. The integrals are carried out by adaptive quadrature as in the calculation of the coefficient matrix, but working to a tolerance of 1%. Finally the norm calculation is approximated by

$$\|\mathbb{L}U - \boldsymbol{f}\|^2_{L^2(\Gamma_j)} \approx \int_{\Gamma_j} \left| \sum_{k=0}^{n} W_k \tilde{K}(z_k - z') + w^2 I_2(z') + \boldsymbol{f}(z') \right|^2 dz'$$

and adaptive quadrature is used with a relative tolerance of 1%.

4. NUMERICAL RESULTS

4.1. *Uniformly Refined Galerkin and Collocation Schemes*

We begin by presenting results for the three uniform grid methods described in Sections 2.1–2.3 that use the uniform mesh refinement strategy of Algorithm 1. All of our numerical tests have been run using a constant right-hand side function $\boldsymbol{f} \equiv 1$ in the IDE (1.1). Results are summarized in Figs. 4.1–4.4: these figures are typical in that they illustrate the behavior of the schemes both for large and small a (10^{-2} and 10^{-6}) and for low, high, and very low frequencies ($\omega = 1$, 100, and 0.01). They show the dependence of the relative L^2 solution error on both n (the number of mesh intervals) and the total flops and the dependence of the flop counts for setting up the coefficient matrices and solving the linear systems on n for each scheme. Each figure contains data computed with $n = 2^k$ for $k = 2:12$ (and so the finest grid has $n = 4096$ in each case). The numbers appearing as a legend in each graph are the slopes of the respective plots, calculated as a least squares fit of the data (to four significant digits) with $n = 2^k$ for $k = 7:10$ in the top right-hand graph and $k = 8:12$ for the bottom two graphs of each figure. The absolute value of the computed solution is shown for a range of radius sizes and frequencies in Fig. 4.5.

The relative L^2 errors for each approximation are straightforward to calculate from the available data. Suppose that $U^{(n)}(z)$ is an approximate solution computed with mesh spacing $h = 1/n$ (the coarse solution), and $U^{(n^*)}(z)$ is the approximate solution computed on the

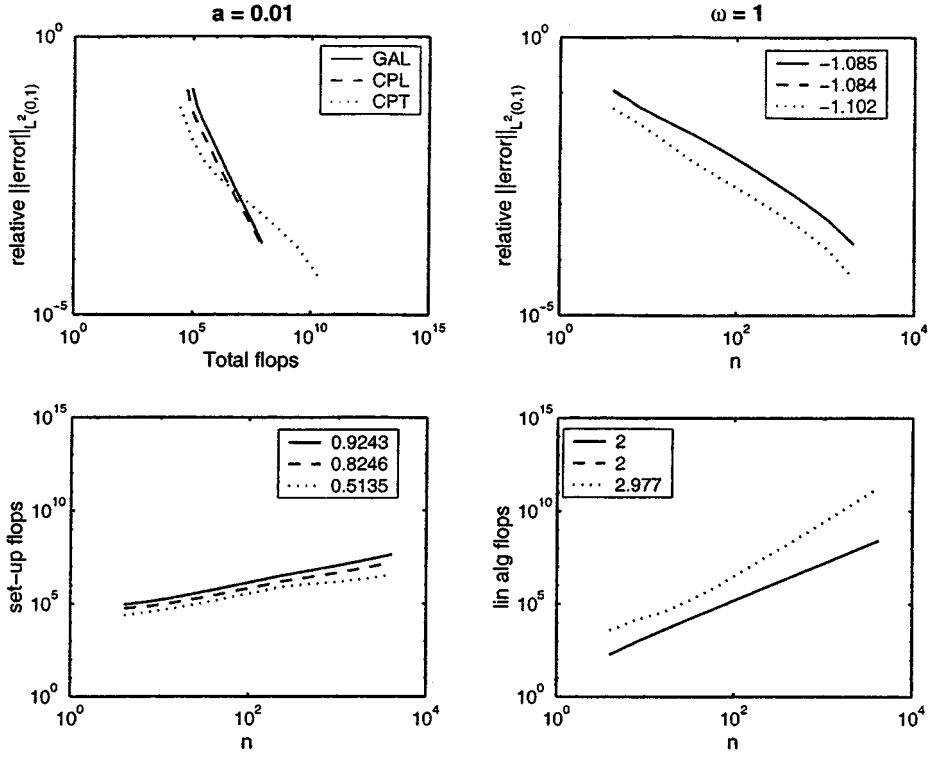

FIG. 4.1. Results for piecewise linear Galerkin (—), piecewise linear collocation (- -), and piecewise trigonometric ($\cdots$) schemes when $a = 0.01$ and $\omega = 1$.

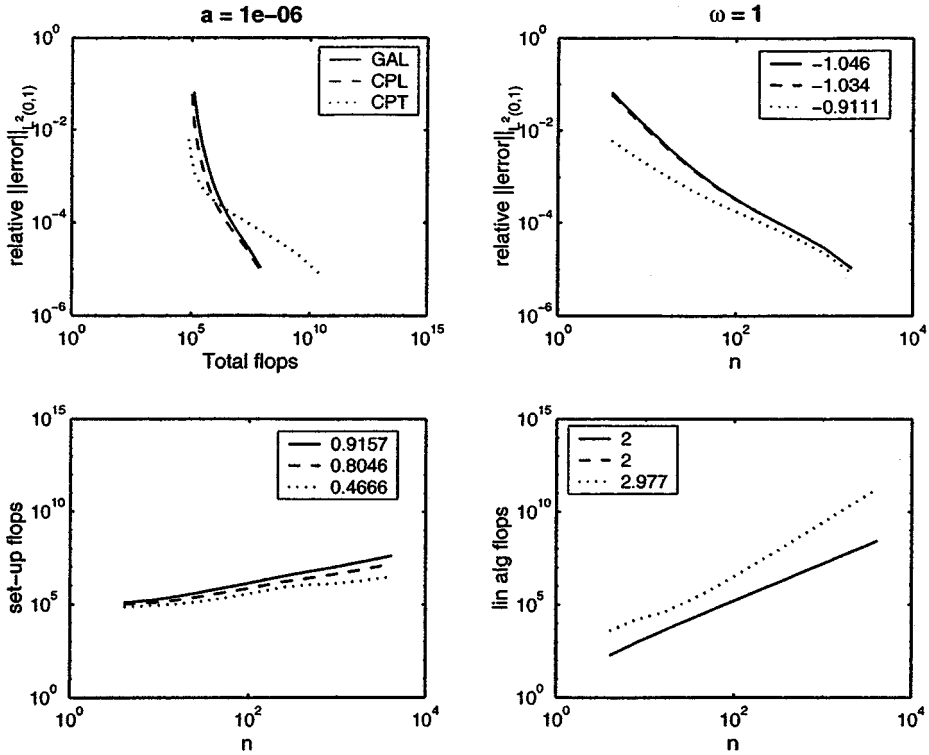

FIG. 4.2. Results for piecewise linear Galerkin (—), piecewise linear collocation (- -), and piecewise trigonometric ($\cdots$) schemes when $a = 10^{-6}$ and $\omega = 1$.

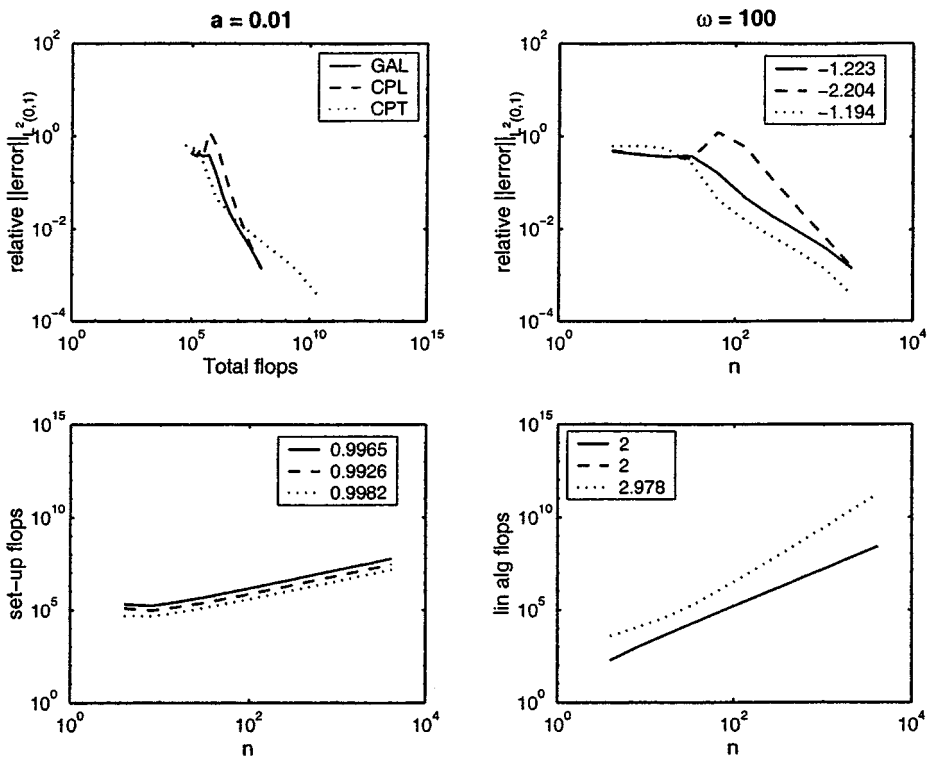

FIG. 4.3. Results for piecewise linear Galerkin (—), piecewise linear collocation (- -), and piecewise trigonometric ($\cdots$) schemes when $a = 0.01$ and $\omega = 100$.

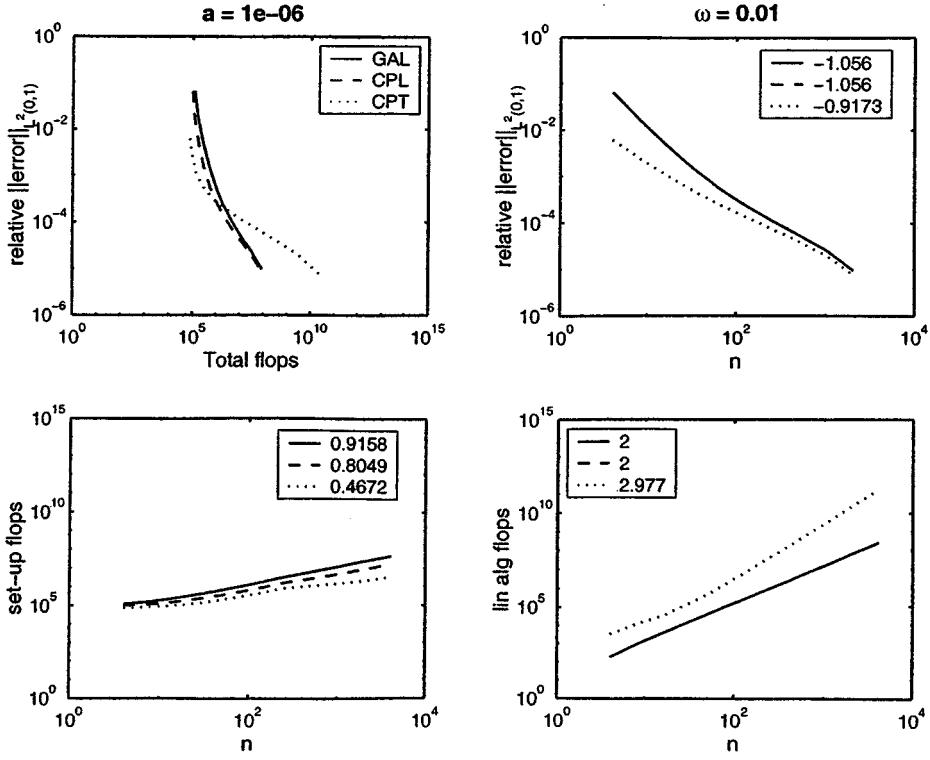

FIG. 4.4. Results for piecewise linear Galerkin (—), piecewise linear collocation (- -), and piecewise trigonometric ($\cdots$) schemes when $a = 10^{-6}$ and $\omega = 0.01$.

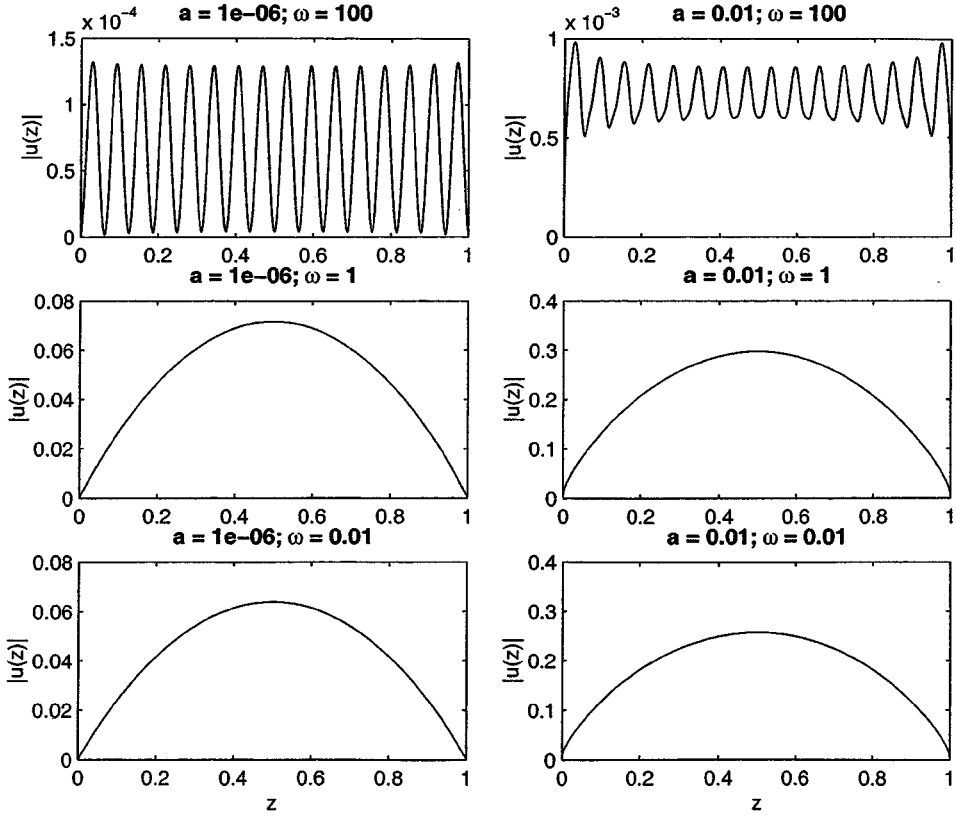

FIG. 4.5. Plot of absolute value of the solution $|u(z)|$ against z for various values of a and ω.

finest available mesh with spacing $h^* = 1/n^*$ (i.e., $n^* = 4096$ for each scheme). Let ε denote the difference between the coarse and fine solutions, and so $\varepsilon(z) = U^{(n)}(z) - U^{(n^*)}(z)$. For the piecewise linear Galerkin and collocation scheme, $\varepsilon(z)$ is a piecewise linear function on the fine mesh, and so $\|\varepsilon\|_{L^2(0,1)}$ can be calculated exactly in terms of the fine mesh nodal values of $U^{(n)}(z)$ and $U^{(n^*)}(z)$. The function $\varepsilon(z)$ is piecewise trigonometric on the fine mesh for the piecewise trigonometric collocation scheme, and its norm can be either computed exactly using this representation or approximated by the norm of its piecewise linear interpolant. For simplicity we used the piecewise linear approximation. The relative solution error plotted in each of Figs. 4.1–4.4 is then taken to be

$$\frac{\|\varepsilon\|_{L^2(0,1)}}{\|U_G^{(n^*)}\|_{L^2(0,1)}},$$

where $U_G^{(n^*)}$ is the piecewise linear Galerkin solution on the fine mesh (the same solution is used to normalize each error).

As predicted by Rynne [18, 19], the convergence rate of the piecewise linear Galerkin scheme appears to be $O(1/n)$. The two collocation schemes also appear to exhibit first-order convergence, as shown in the top right plots of Figs. 4.1–4.4. Note that the apparent superiority of the piecewise trigonometric scheme in these graphs is somewhat misleading since it uses three times as many unknowns as the other two schemes. The top left-hand plots (flops vs. relative error) provide a fairer comparison.

As can be predicted from the number and complexity of the kernel integrals required for the entries in the coefficient matrices (described in Section 3.2), the set-up flop count is roughly proportional to n^s for some $s \leq 1$, and for fixed n it is always lowest for piecewise trigonometric collocation and highest for the Galerkin scheme. As discussed in Section 3.4, the two piecewise linear methods use Levinson's algorithm to solve the linear system, and this is reflected in the quadratic growth of their linear algebra flop count with n. The piecewise trigonometric scheme uses $O(n^3)$ flops for the linear algebra, which means that if n is sufficiently large then this scheme will be less efficient than the other two. For small n, however, the piecewise trigonometric approximation generally performs well, giving a lower error than the other two schemes for a fixed flop count. The system size at which the error and flop count for this scheme are roughly comparable to that for the Galerkin approximation (i.e., where their respective curves in the top left graphs overlap) appears to depend on both ω and a. At small a the piecewise trigonometric scheme is uncompetitive at all other than very coarse meshes. The piecewise linear collocation scheme does not work well for low n when ω is moderately large, taking a long time to start converging.

The tests were repeated at very low frequency to compare the two versions of the piecewise trigonometric collocation scheme described in Section 2.3. The representation (2.9) gives more reliable results for small ωh than (2.5), which is corrupted by rounding errors. As a rough guide, when $a \approx 0.01$ the relative size of the rounding error using the representation (2.5) is very close to the relative difference between the floating point evaluations of $(\cos \omega h - 1)$ and $-2\sin^2(\omega h/2)$ (see the end of Section 2.3). There are about $16 + 2\log_{10} wh$ significant digits of accuracy using (2.5) in standard 64-bit arithmetic and rounding errors of $\geq$1% when $wh \leq 10^{-7}$. These rounding errors appear to be reduced slightly when a is decreased.

4.2. *The p-, hp- and Adaptive h-Schemes*

We now describe numerical results for the schemes of Sections 2.4–2.6 and compare them with those for the uniform mesh Galerkin approximation obtained in the previous section. Again we take $\boldsymbol{f} \equiv 1$ in the IDE (1.1). Starting from an initial uniform mesh with $h = 1/4$ (resp. $h = 1/32$ for the p- and hp-schemes when $\omega = 100$) we used Algorithm 2 to compute a uniform p-method solution, used Algorithm 3 for the adaptive h-method solution, and calculated solutions for the hp-scheme on a sequence of six meshes using the (end) element refinement parameter $\lambda = 0.17$.

We use the approximate solution $U_G^{(n^*)}(z)$ obtained from the uniform h-scheme with $h = 2^{-17}$ (i.e., using $n^* = 131072$ elements) to calculate the relative L^2 error for each scheme; i.e., the error is $\|U^{(N)} - U_G^{(n^*)}\|_{L^2(0,1)} / \|U_G^{(n^*)}\|_{L^2(0,1)}$. Figures 4.6–4.8 show plots of the computed error against both the flops used and number N of degrees of freedom for ($a = 0.01, \omega = 1$), ($a = 0.01, \omega = 100$), and ($a = 1e-6, \omega = 0.01$). The numbers appearing as a legend in each graph are again the slopes of the respective plots, calculated as a least-squares fit to the four most accurate solutions.

The upper two graphs of each figure show that in common with the schemes described in the previous subsection, the p- and adaptive h-schemes also converge algebraically for large n. The numerically obtained convergence rate for the p-method agrees with the theoretically expected rate of 2. The adaptive mesh-refining strategy of Algorithm 3 increases the convergence rate for the piecewise linear Galerkin scheme from 1 (for the uniform mesh

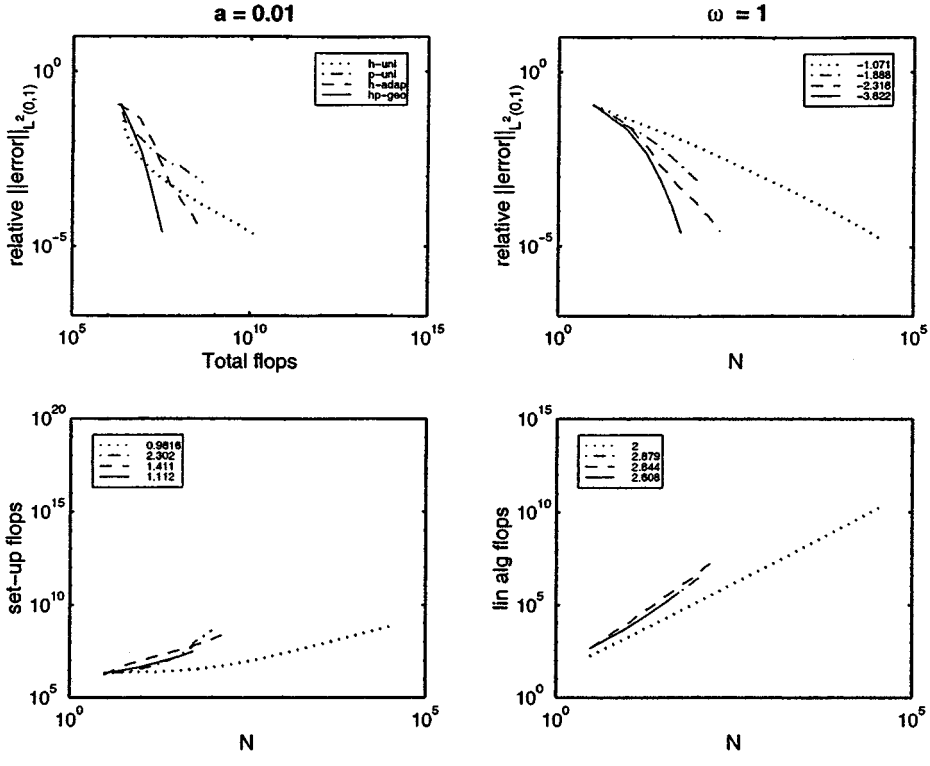

FIG. 4.6. Results for h-uniform ($\cdots$), p-uniform (– · –), h-adaptive (- -), and hp-geometric (—) Galerkin schemes when $a = 0.01$ and $\omega = 1$.

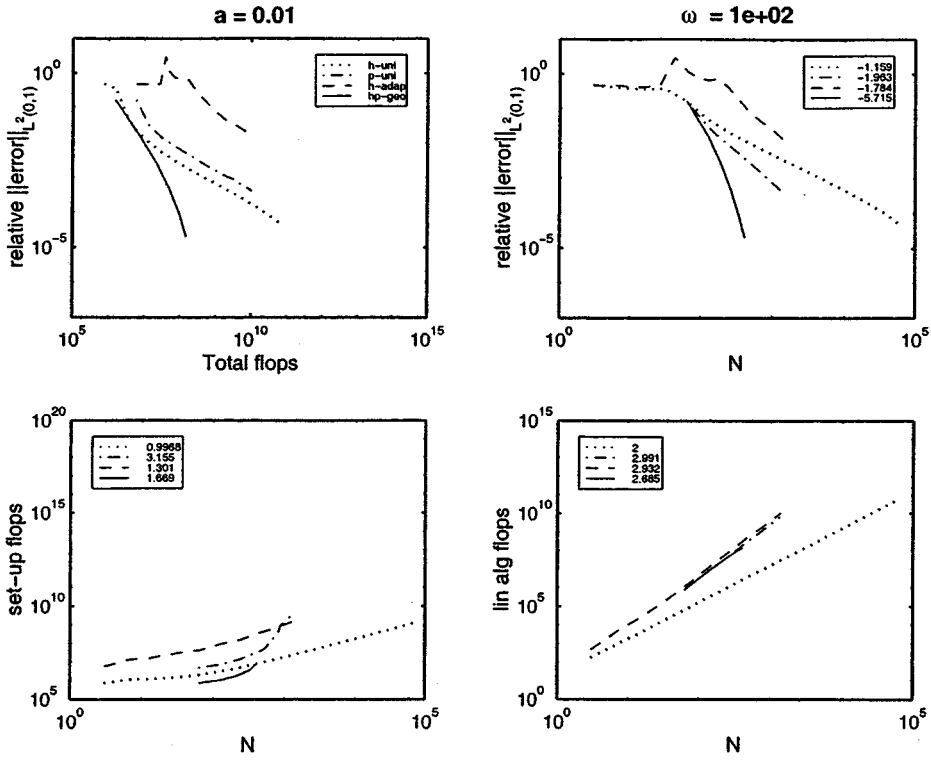

FIG. 4.7. Results for h-uniform ($\cdots$), p-uniform ($-\cdot-$), h-adaptive (- -), and hp-geometric (—) Galerkin schemes when $a = 0.01$ and $\omega = 100$.

approximation) to roughly 2, which is likely to be optimal for this finite element space. The slope of the graph for the hp-scheme increases for increasing N, illustrating the exponential convergence of this method.

It is interesting to note that if low accuracy (relative error $\geq 10^{-3}$) is required, then the h-uniform scheme is more efficient than the h-adaptive scheme in the tests. This is partly due to the overhead in computing the error estimate and the additional costs of matrix assembly and linear algebra on a nonuniform grid. Also the h-adaptive scheme seems to perform badly at high frequency, perhaps because the grid adaptation strategy does not work well until the highly oscillatory solution is well-resolved everywhere. However, the plots clearly show that the hp-scheme is far more efficient than the others at moderate to high accuracies and is at least as efficient at lower accuracy.

The figures also show the dependence of the set-up and linear algebra flop counts on N. Note that the adaptive h-scheme and uniform p-scheme are in general considerably more expensive to set up than the other schemes. For the p-scheme this is presumably because computing the matrix entries corresponding to high-degree polynomial basis functions takes a long time. The reason for the adaptive h-scheme is slightly more subtle: because the "current" mesh for this scheme depends on all the previous meshes and solutions we take the set-up flop count to be the sum of all the flops used to set up the linear system and calculate the a posteriori error estimator on all previous meshes. Similarly, the linear algebra flop count is defined to be the sum of all flops used in solving the linear systems to obtain the solution on all previous meshes.

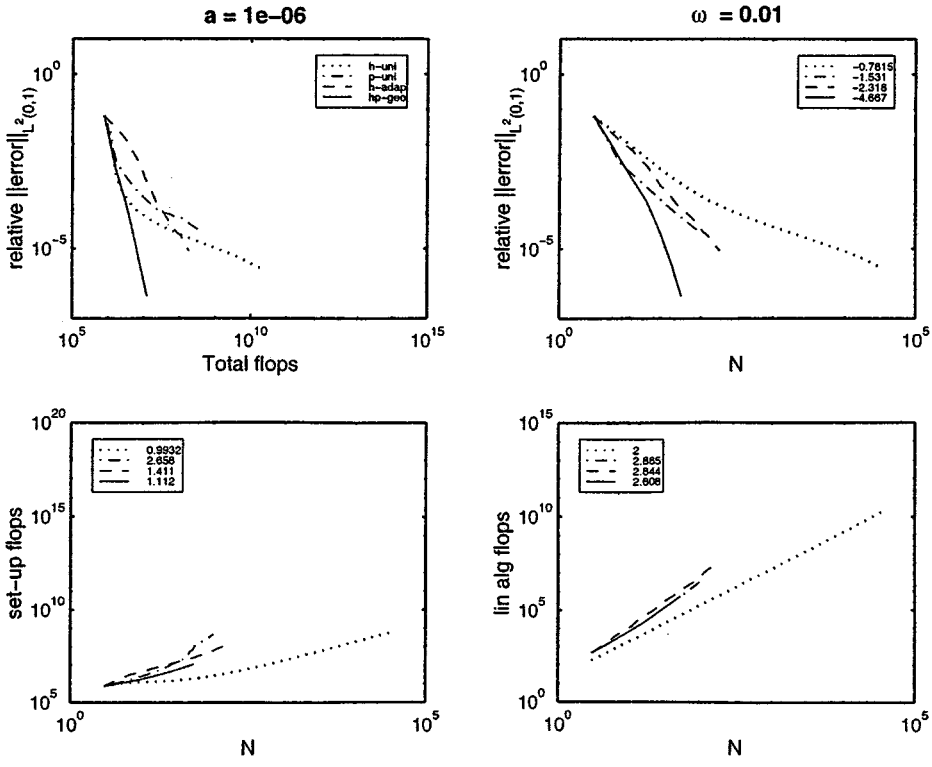

FIG. 4.8. Results for h-uniform ($\cdots$), p-uniform ($- \cdot -$), h-adaptive (- -) and hp-geometric (—) Galerkin schemes when $a = 10^{-6}$ and $\omega = 0.01$.

5. CONCLUSIONS

This paper provides a systematic study and comparison of numerical approximation methods for the exact kernel form of Pocklingon's thin wire equation. This is a difficult computational problem, and we have described ways in which it can be tackled efficiently.

Our numerical results show that the theoretically predicted first-order convergence rate [18, 19] for the uniform piecewise linear Galerkin scheme is achieved in practice. The two collocation schemes described in Sections 2.2 and 2.3 also exhibit first-order convergence. A second-order convergence rate can be obtained by using either an adaptive h- or a uniform p-refinement strategy. The geometric (nonadaptive) hp-method whose predetermined refinement strategy is designed to take known solution properties (the square root behavior at the ends of the wire [18, 19]) into account is by far the most efficient of the algorithms we have considered. As shown in Figs. 4.6–4.8 this scheme appears to converge exponentially, its convergence rate increasing with the number of degrees of freedom. It has long been known that grading the mesh appropriately to take account of singularities in the solution can be used to obtain good convergence rates for numerical approximations of integral equations (see for example [3] for a comprehensive survey).

One might be tempted to try an adaptive hp-algorithm to achieve even better results. However, the h-adaptive algorithm of Section 2.6 requires repeated evaluation of the error estimator η which, although this can be speeded up tremendously as described in Section 3.5, still imposes a considerable overhead on the calculations. An hp-adaptive scheme would suffer at least as badly, and so would probably be uncompetitive.

Another way of tackling the end singularities is to augment the approximation space by adding the two new basis functions $\varphi_0(x) = \sqrt{x}$ and $\varphi_1(x) = \sqrt{1-x}$. Rynne proves in [18] that the convergence rate for the uniform piecewise linear Galerkin approximation can be improved to second order if this is done correctly. Even though this type of approximation may well be more efficient than either of the second-order schemes considered here, its algebraic convergence means that it cannot compete with the hp-scheme.

We conclude with a short list of recommendations for anyone who wants to compute the current induced on a perfectly conducting thin wire.

- Use the exact kernel form of Pocklington's equation to model the problem (the Hallen form has some undesirable properties [11, 20], and the reduced kernel problem is ill-posed [21]).
- Evaluate and integrate the kernel as efficiently as possible (Sections 3.1 and 3.2).
- Use the nonadaptive (end-refined) hp-method of Section 2.5 to obtain high-accuracy solutions efficiently.
- If low accuracy (relative error $\geq 1\%$) is sufficient, then the uniform grid piecewise trigonometric collocation scheme of Section 2.3 is an efficient method.

ACKNOWLEDGMENTS

We are grateful to the British Council and the German Academic Exchange Service (DAAD) for supporting this project through the ARC programme.

REFERENCES

1. M. Abramowitz and I. A. Stegun, *Pocketbook of Mathematical Functions* (Verlag Harri Deutsch, 1984)
2. R. Bancroft, *Understanding Electromagnetic Scattering Using the Moment Method: A Practical Approach* (Artech House, Norwood, MA, 1996).
3. H. Brunner, A. Pedas, and G. Vainikko, The piecewise polynomial collocation method for nonlinear weakly singular Volterra equations, *Math. Comput.* **68**, 1079 (1999).
4. G. J. Burke, *Numerical Electromagnetics Code—NEC-4 Method of Moments part II: Program Descriptor—theory* Lawrence Livermore National Laboratory report UCRL-MA-109338 Pt. II (1992).
5. C. M. Butler and D. R. Wilton, Analysis of various numerical techniques applied to thin-wire scatterers, *IEEE Trans. Antennas Propag.* **AP-23**(4), 534 (1975).
6. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. (John Hopkins Press, Baltimore, 1989).
7. W. Gui and I. Babuška, The h, p and h–p versions of the finite element method in 1 dimension, *Numerische Math.* **49**, 613 (1986).
8. H. Hancock, *Theory of Elliptic Functions* (Dover, New York, 1958).
9. N. Heuer, hp-Versionen der Randelementmethode, Ph.D. thesis (Universität Hannover, 1992).
10. D. S. Jones, *Theory of Electromagnetism* (Oxford Univ. Press, London, 1964).
11. D. S. Jones, Note on the integral equation for a straight wire antenna, *IEE Proc.* **128**, 114 (1981).
12. D. S. Jones, *Methods in Electromagnetic Wave Propagation*, 2nd ed. (Oxford Univ. Press, London, 1995).
13. R. Kress, *Linear Integral Equations* (Springer-Verlag, New York, 1989).
14. J. L. Lions and E. Magenes, *Non-Homogeneous Boundary Value Problems and Applications: Vol. I* (Springer-Verlag, New York, 1972).
15. K. K. Mei, On the integral equations of thin wire antennas, *IEEE Trans. Antennas Propag.* **AP-13**, 374 (1965).
16. L. W. Pearson, Analysis of various numerical techniques applied to thin-wire scatterers, *IEEE Trans. Antennas Propag.* **AP-23**, 256 (1975).
17. H. C. Pocklington, Electrical oscillations in wires, *Proc. Cambridge Philos. Soc.* **9**, 324 (1897).

18. B. P. Rynne, Convergence of Galerkin method solutions of the integral equation for thin wire antennas, *Adv. Comp. Math.* **12**, 251 (2000).
19. B. P. Rynne, *On the Well-Posedness of Pocklington's Equation for a Straight Wire Antenna and Convergence of Numerical Solutions*, *Journal of Electromagnetic Waves and Applications* **14**, 1489 (2000).
20. B. P. Rynne, The well-posedness of the integral equations for thin wire antennas, *IMA J Applied Math.* **49**, 35 (1992).
21. S. A. Schelkunoff, *Advanced Antenna Theory* (Wiley, New York, 1952).
22. A. G. Tijhuis, P. Zhongqui, and A. R. Bretones, Transient excitation of a straight thin wire segment: A new look at an old problem, *IEEE Trans. Antennas Propag.* **40**, 1132 (1992).
23. D. H. Werner, J. A. Huffman, and P. L. Werner, Techniques for evaluating the uniform current vector potential at the isolated singularity of the cylindrical wire kernel, *IEEE Trans. Antennas Propag.* **42**(11), (1994).
24. D. R. Wilton and C. M. Butler, Effective methods for solving integral and integro-differential equations, *Electromagnetics* **1**, 289 (1981).