JOURNAL OF COMPUTATIONAL PHYSICS **146,** 627–663 (1998)
ARTICLE NO. CP986077

Numerical Solution of the Helmholtz Equation in 2D and 3D Using a High-Order Nyström Discretization[1]

Lawrence F. Canino, John J. Ottusch, Mark A. Stalzer, John L. Visher, and Stephen M. Wandzura

Computational Physics Department of the Communications and Photonics Laboratory, HRL Laboratories, M/S RL65, 3011 Malibu Canyon Road, Malibu, CA 90265-4799
E-mail: ottusch@hrl.com

Received December 16, 1997; revised August 11, 1998

We show how to solve time-harmonic scattering problems by means of a high-order Nyström discretization of the boundary integral equations of wave scattering in 2D and 3D. The novel aspect of our new method is its use of local corrections to the discretized kernel in the vicinity of the kernel singularity. Enhanced by local corrections, the new algorithm has the simplicity and speed advantages of the traditional Nyström method, but also enjoys the advantages of high-order convergence for controlling solution error. We explain the practical details of implementing a scattering code based on a high-order Nyström discretization and demonstrate by numerical example that a scattering code based on this algorithm can achieve high-order convergence to the correct answer. We also demonstrate its performance advantages over a high-order Galerkin code.

Key Words: high-order numerical method; Nyström method; boundary integral equation; Nyström discretization; local corrections; acoustic scattering; electromagnetic scattering.

I. INTRODUCTION

High-order methods are numerical methods characterized by their ability to obtain extra digits of precision with comparatively small additional effort. Scattering codes that employ high-order methods have a distinct advantage over scattering codes that use low-order methods when it comes to computing results accurately. We demonstrated this advantage with a Galerkin method of moments scattering code called FastScat™ [1, 2], which employs

[1] This research was supported by the Defense Advanced Research Projects Agency of the U.S. Department of Defense under Contract MDA972-95-C-0021 and by the Hughes Electronics Corporation.

0021-9991/98 $25.00

high-order methods in its geometry description, current basis functions, and quadratures. In terms of memory efficiency, the advantage of using a high-order code such as FastScat was clear. For a given number of unknowns, results obtained with FastScat were generally more accurate than those obtainable by low-order codes, with the accuracy gap widening rapidly as the number of unknowns applied to the problem was increased. In terms of CPU time efficiency, however, the advantage of using a high-order code such as FastScat was not so clear. The precomputation phase of the calculation often accounted for an undesirably large fraction of the total solution time. Although we were able to significantly accelerate the part of the precomputation phase devoted to computing near-interaction matrix elements by using high-order regulated kernels [3], the overall matrix fill procedure was still considered too slow.

The precomputation phase of a Galerkin scattering calculation is time consuming because it requires numerical evaluation of the convolution of the kernel with basis functions on every pair of source and field patches. This amounts to N^2 numerical double integrations over patches, where N is the number of unknowns. By contrast, when a point-based (Nyström) discretization is used, the impedance matrix fill step consists of nothing more than a kernel *evaluation* to fill most matrix elements and $\mathcal{O}(N)$ single integrations and some low-rank linear algebra to fill the others (specifically, the near interactions). As a result, use of a point-based discretization dramatically reduces precomputation time.

Despite its simplicity and speed advantages, the Nyström method has not been widely used for discretizing the integral equations that arise in 2D and 3D scattering problems. In fact, we know of only a few reported instances, of which [4, 5] are examples. The problem is that the conventional Nyström method [6] is designed to handle regular kernels, whereas the Helmholtz kernel for wave scattering is singular wherever the source point coincides with the field point. The standard way [6] to try to overcome this problem is to use so-called "singularity extraction," which, in practice, removes the infinity in the kernel but not the singularities in the kernel's derivatives. While singularity extraction avoids the dilemma caused by numerical evaluation of the kernel at infinities, it does not generalize easily to arbitrary surface patch geometries and it is a low-order method. In this paper, we introduce "local corrections" as a means to overcome the problems associated with kernel singularities. This enhanced Nyström discretization method has all the advantages of the standard Nyström method combined with the high-order convergence capability required to achieve error control.

This paper provides a detailed explanation for using the Nyström method to solve scattering problems in the 2D and 3D scalar cases and the 3D vector case (by which we mean electromagnetic scattering based on the Maxwell equations), as well as numerical evidence, demonstrating the method's utility. The first section reviews the traditional Nyström method for discretizing integral equations and explains how it can be adapted to handle singular kernels by incorporating local corrections. The second section discusses practical aspects of implementing a high-order Nyström code, such as appropriate surface models and meshes, choice of testing functions for computing local corrections, and how to compute scattering results. In the fourth section, we show numerical results for some 2D and 3D canonical scatterers to demonstrate that our implementation of the Nyström method achieves high-order convergence to the correct answer. We also demonstrate the run-time performance benefits of a using high-order Nyström code, compared to high- and low-order Galerkin codes, in this section. Finally, the Appendix describes how the local correction integrals for 2D scalar, 3D scalar, and 3D electromagnetic scattering can be formulated for efficient and accurate numerical evaluation.

II. NYSTRÖM METHOD

A. *Conventional Nyström Method*

The conventional Nyström method is a simple and efficient mechanism for discretization of integral equations with nonsingular kernels. Consider the integral equation

$$\phi(\mathbf{x}) = \int_S ds' G(\mathbf{x} - \mathbf{x}')\psi(\mathbf{x}') \tag{1}$$

and a quadrature rule for integrating a function $f(\mathbf{x})$ over the region S

$$\int_S ds\, f(\mathbf{x}) \cong \sum_{n=1}^{N} \omega_n f(\mathbf{x}_n). \tag{2}$$

Such a quadrature rule will be provided by Gauss–Legendre or Gauss–Jacobi rules on a parameterization of S, so that the weights ω_n will be the products of the elementary weights w_n with the Jacobian of the parameterization:

$$\omega_n = \sqrt{g(u_n)}w_n, \tag{3}$$

$$\mathbf{x}_n = \mathbf{x}(u_n), \tag{4}$$

where u_n are the abscissae of the elementary rule, $\mathbf{x}(u)$ is the mapping function of the surface S, and $g(u)$ is the determinant of the mapping metric. The extension to patched parameterizations is straightforward.

The Nyström discretization of a function on S is simply the tabulation of the function at the quadrature points $\mathbf{x}_n$:

$$\psi_n = \psi(\mathbf{x}_n). \tag{5}$$

To discretize integral Eq. (1), we simply form a matrix from the kernel:

$$\phi_m = \sum_{n=1}^{N} \omega_n G(\mathbf{x}_m - \mathbf{x}_n)\psi_n. \tag{6}$$

This discretization has an error of the same order as the underlying quadrature rule [7]. In other words, if the surface S is smooth, ϕ and $G(\mathbf{x} - \mathbf{x}')$ are regular functions, and if a high-order quadrature rule is used, then the solution to Eq. (6) represents a high-order approximation to the exact solution.

Unfortunately, the kernels $G(\mathbf{x} - \mathbf{x}')$ for wave scattering are not regular. Instead, they have singularities (or even hypersingularities) at short distances. With such kernels it is often not even possible to make a matrix out of the kernel because its value is undefined when $\mathbf{x} = \mathbf{x}'$. Even if the kernel were finite at vanishing separation, a kernel singular in its higher derivatives would spoil the high-order properties of the above prescription.

B. *High-Order Nyström Method for Singular Kernels*

We have adapted the Nyström method to handle singular kernels, without sacrificing high-order convergence, by incorporating Strain's method [8] for obtaining high-order quadrature

rules for singular functions. The essence of the method is that by computing convolutions of the kernel with a suitable set of testing functions, it is possible to determine how to adjust the quadrature rule so that it is just as accurate near the singularity as far from it. The beauty of the method is that these quadrature rule modifications are required only in the vicinity of the singularity, hence the name *local corrections*.

Conceptually, local corrections may be viewed as adjustments to the quadrature weights (at the original set of sample points) that are required to make the quadrature rule high-order accurate when the (singular) function $G(\mathbf{x}-\mathbf{x}')$ is included in the integrand. In practice, since quadrature weights and discretized kernel terms always enter into the quadrature rule as product pairs, one can equally well "locally correct" the discretized representation of kernel and keep the original quadrature weights. This is the preferred approach because the modified representation of the kernel has no infinities. We can write the "corrected" matrix representation of the kernel as

$$\tilde{G}_{mn} \equiv \begin{cases} L_{mn}, & \text{when } \mathbf{x}_n \in D_m, \\ G(\mathbf{x}_m - \mathbf{x}_n), & \text{otherwise}, \end{cases} \tag{7}$$

where L_{mn} is a (sparse) matrix of local corrections whose entries are nonzero only for source points $\mathbf{x}_n$ within a small domain D_m centered on the field point $\mathbf{x}_m$. For $|\mathbf{x}_m - \mathbf{x}'|$ sufficiently large (i.e., outside the local correction domain D_m), $G(\mathbf{x}_m - \mathbf{x}')$ is a smoothly varying function of position and the underlying quadrature rule provides a high-order approximation to the desired integral. Close to the singularity, on the other hand, the singular nature of the kernel spoils the high-order behavior of the underlying quadrature rule, and it becomes necessary to use locally corrected values for the kernel instead of $G(\mathbf{x}_m - \mathbf{x}_n)$ in order to achieve high-order convergence. The mechanism for computing the local corrections for a given set of source points is explained below. The size of the local correction domain is discussed in Section III.D.

The underlying quadrature rule is exact for integration of a certain class of functions (typically polynomials). We choose the local corrections to make convolution of the singular kernel with the same class of functions exact. They are obtained by solving the linear system

$$\sum_n \omega_n L_{mn} f^{(k)}(\mathbf{x}_m - \mathbf{x}_n) = \int_{D_m} ds' G(\mathbf{x}_m - \mathbf{x}') f^{(k)}(\mathbf{x}_m - \mathbf{x}'), \tag{8}$$

which represents K constraints (one for each testing function $f^{(k)}$) on J local correction coefficients (one for each of J source points in the vicinity of the mth field point). The integral over D_m can be obtained by oversampling the region of integration until the result has converged to the desired accuracy. The nonzero components of the mth row of the local correction matrix are obtained by inverting the (small) system of equations above, either by factorization (via LU decomposition) if $J = K$ or by singular value decomposition (SVD) if $J \neq K$. Computing local corrections is the most time consuming step of the precomputation phase. Fortunately, it needs to be done only once at every sample point.

C. *High-Order Nyström Method Advantages*

There are several reasons for using the Nyström method to achieve a high-order discretization:

• *Faster precomputation*. Unlike the Galerkin method, which requires N^2 numerical double integrations to fill the impedance matrix, the Nyström method requires less than N^2 kernel evaluations and $\mathcal{O}(N)$ calculations of local correction coefficients (each of which involves a small number of adaptive integrations and a low-rank matrix inversion). An additional acceleration is possible when multiple solutions are desired at different frequencies. This comes about because a frequency-dependent Helmholtz kernel can be written as the product of a smoothly varying, frequency-dependent function and a frequency-independent Laplace kernel. Once the local corrections for the Laplace kernel have been computed, they can be used with minor modification at any frequency.

• *Elimination of multipatch, parametric basis functions*. Conventional method of moments scattering codes require basis functions with a certain level of continuity (in the surface parameterization) across patch boundaries to facilitate differentiation. For example, an important property of the popular RWG [9] basis functions for electromagnetic scattering is that their normal components are continuous across patch boundaries. One can also use high-order extensions to the RWG basis functions [10], although we have found that implementing these basis functions in a scattering code can be both complicated and inconvenient, especially for arbitrary, curved surfaces. Fortunately, for high-order codes the requirement to use elemental sources with guaranteed continuity between patches disappears because continuity of the source distribution is achieved as a natural consequence of accurately solving the integral equation. (The reason this is so has to do with the fact that the error caused by not enforcing continuity of the elemental sources is comparable to the error of the underlying discretization. With a low-order discretization (e.g., RWG basis functions on flat patches), continuity enforcement has a significant payoff because the error in the underlying discretization is also significant. With a high-order discretization, where the error due to the underlying discretization can more easily be made insignificant, the situation is reversed. Thus, for high-order codes, whether Galerkin or Nyström, the benefits of enforcing source continuity between patches do not outweigh the inconveniences.)

• *More amenable to fast solution algorithms*. Implementation of a fast method that requires segregation of the discretized scatterer into groups (such as the fast multipole method (FMM) [11] or adaptive integral method (AIM) [12]) is simpler and more natural with a point-based discretization. When a Galerkin implementation with overlapping basis function domains is employed, the fast algorithm is either more complicated (because multipatch basis functions must be split apart) or less efficient (because the groups are larger). A Galerkin implementation that uses high-order basis functions (even those confined to single patches) cannot achieve optimum efficiency from the FMM because high-order basis functions are used to their greatest advantage on patches larger than a wavelength, whereas optimum use of the FMM favors groups smaller than a wavelength. In a Nyström discretization, the groups consist of individual sample points on the surface, so no such grouping restrictions apply.

• *Iterative solver memory reduction*. With the Nyström method, the memory requirement for an iterative solver using the full impedance matrix can be reduced from $\mathcal{O}(N^2)$ (storing the full impedance matrix) to $\mathcal{O}(N)$ (storing only the sparse local correction matrix). This is practical because reconstruction of the unsaved portions of the impedance matrix only requires evaluations of the kernel, which are fast. If the FMM is used to represent the far interactions, the storage requirement goes from $\mathcal{O}(N^{5/4})$ in the single-stage case [13] to $\mathcal{O}(N \log(N))$ in the multilevel case [14].

• *Symmetry exploitation*. When basis functions are used, it is more complicated to reflect geometrical symmetries in the matrix representation. It may be necessary to explicitly consider basis function transformation properties and to provide special treatment for some variables (e.g., the coefficients of basis functions whose domains intersect reflection planes). In the Nyström case, the representation of symmetries is much simpler.

III. PRACTICAL CONSIDERATIONS

A. *Surface Description*

Without a high-order surface description, a high-order Nyström discretization is of little benefit. For example, representing a curved surface by means of flat facets limits the rate of solution convergence to low order whether or not the rest of the discretization method is high order. Ideally, the internal representation of the surface exactly matches the physical surface. Such a representation is possible for idealized curved shapes such as circles, ellipses, ogives, etc. in 2D, and spheres, ellipsoids, etc. in 3D. For curved objects of more practical interest, a high-order description of the physical surface may be given by high-order parametric representations such as bicubic splines or NURBS (nonuniform rational B-splines). As these are often the representations used by a CAD program to describe the object as it is being designed and built, it is appropriate that we should also use them for electromagnetic or acoustic modelling purposes.

Use of a high-order surface description is distinguished from that of a faceted description in that the subdivision of the surface into patches is typically done once and refining the discretization to improve accuracy is accomplished by increasing the order of the quadrature rule (which increases the number of sample points per patch).

B. *Meshing*

The essence of a point-based discretization is the tabulation of functions at a set of points lying on the surface. This need not have anything to do with subdividing a surface into patches. Indeed, in the 2D case, patches can be done away with entirely on closed surfaces (i.e., closed curves) parameterized by arc length, because the trapezoidal rule is a high-order quadrature rule for periodic functions. In 3D, however, global parameterizations with natural, high-order quadrature rules are much harder to come by, so subdivision of a surface into patches, each of which comes with its own high-order quadrature rule, becomes a practical necessity.

Since patches are introduced solely for the purpose of providing ready-made, high-order quadrature rules on the surface, the job of meshing a surface is simpler and less restrictive. Specifically, whereas a mesh designed for use with RWG-type basis functions is not allowed to have a vertex in the middle of an edge, there is no such restriction on a mesh designed for a point-based discretization. The only practical restrictions are that the mesh cover the surface and that the patches not be so distorted or curved that the supposedly high-order quadrature rules are not actually high order.

C. *Testing Functions*

The choice of testing functions goes together with the choice of quadrature rule. If the quadrature rule is designed to efficiently integrate regular functions, the testing functions

should be regular functions of increasing order. In locations where singular behavior of the source function is expected, such as near geometric singularities (e.g., edges and corners), it may be desirable to apply a different quadrature rule and use appropriately singular testing functions [15]. For purposes of this discussion, we will assume the scattering surface and the sources are smooth functions of position. Any departures from regularity can be accommodated reasonably efficiently by tapering the size of the patches in the direction of the singularity.

Testing functions may be global or local. Examples of global testing functions are monomials in the surface parameter u in the 2D case, and powers of x, y, and z in the 3D case. The advantage of using global testing functions to compute local corrections on smooth surfaces is that such testing functions are manifestly continuous across patch boundaries, just like the sources. Sometimes enforcing continuity is a mistake, however, such as when the field point and source patch are near each other but on separate, unconnected surfaces. Global testing functions can also perform badly near geometric singularities such as a right-angle bend. Local testing functions (i.e., testing functions confined to individual patches) do not take full advantage of the guaranteed continuity of the sources on touching patches but are the preferred choice because they are simpler to implement and more robust.

With local testing functions, the local corrections for a given field point can be computed on a patch by patch basis. Thus, the number of points whose quadrature weights are being corrected always equals the number of sample points on the patch. Doing this has the side benefit of keeping down the size of the local correction linear systems that must be solved when it becomes necessary to compute local corrections for points on several patches.

The number of local testing functions to use is still a free parameter. In 2D, where use of a Gauss–Legendre rule of order M allows exact integration of polynomials up to order $2M$ (i.e., degree $2M-1$), it makes sense to use as many testing functions as there are points to locally correct. In effect, the singular kernel and the unknown source function are both being approximated to order M, which means the order of approximation for the product is $2M$. This results in an exactly determined system of equations for computing local corrections.

In 3D, if a Gauss–Legendre product rule of order $M_x M_y$ is used on quadrilateral patches, the natural number of local testing functions to use is $4M_x M_y$. This leads to an exactly determined system. If the patches are triangles, one can use the quadrature rules of Lyness and Jespersen [16] and their higher-order extensions. For these triangle rules, a natural correspondence between the number of sample points and the maximum testing function degree is less obvious. When the number of sample points and the number of testing functions are not the same, they can at least be made close, in which case the nonsquare linear system of equations for the local corrections can be solved by computing a pseudoinverse using SVD. In our experience, local correction systems that are square or nearly square perform best.

C.1. Two-dimensional scalar testing functions. Monomials of increasing degree in the parameterization, i.e., $f^{(k)}(u) = u^k$, are the simplest testing functions, but they can also be troublesome when using high-order rules because they produce linear systems for computing local corrections whose condition number grows exponentially with degree. The alternative we favor is orthogonal polynomials such as Legendre or Lagrange polynomials. With either of these polynomials as testing functions, it takes a little longer to compute the integral on the right-hand side of Eq. (8), but the linear system is well conditioned for all polynomial degrees. In addition, if the number of testing functions K equals the number of source

points whose quadrature weights are being corrected J, then the system is orthogonal and the matrix consisting of the K testing functions evaluated at the J different source points can be inverted simply by transposition.

C.2. Three-dimensional scalar testing functions. The trade-off between the simplicity of monomials and the better conditioning behavior associated with orthogonal polynomials exists also in the 3D cases. In 3D, however, our experience have been confined to testing functions of a low enough degree that use of monomial functions generally does not pose any serious trouble. On triangular patches, we use testing functions of the form

$$f^{(k)}(\mathbf{u}) = (u^1)^m (u^2)^n, \tag{9}$$

where u^1 and u^2 are the parameters of the surface description and the exponents obey $0 \le m, n \le M$ and $0 \le m + n \le M$ for some maximum testing function degree M.

C.3. Three-dimensional vector testing functions. In this case, vector testing functions locally tangent to the surface are required; continuity of the testing functions between adjacent patches is not. A natural set of basis vectors is given by the derivatives of the surface with respect to the two surface parameters u^1 and u^2. We use testing functions of the form

$$\mathbf{t}_\nu^{(k)}(\mathbf{u}) = \frac{\partial_\nu \mathbf{x}(\mathbf{u})}{\sqrt{g(\mathbf{u})}} f^{(k)}(\mathbf{u}), \tag{10}$$

where $\nu = 1, 2$ and the scalar functions $f^{(k)}(\mathbf{u})$ are the same as those used in the 3D scalar case. This form for the testing functions has the property that the surface divergence of $\mathbf{t}_\nu^{(k)}$ is

$$\nabla \cdot \mathbf{t}_\nu^{(k)}(\mathbf{u}) = \frac{\partial_\nu \mathbf{x}(\mathbf{u})}{\sqrt{g(\mathbf{u})}} \cdot \left(\nabla f^{(k)}(\mathbf{u})\right) = \frac{\partial_\nu \mathbf{x}(\mathbf{u})}{\sqrt{g(\mathbf{u})}} \cdot \left(\sum_{\alpha\beta} g^{\alpha\beta} \partial_\alpha f^{(k)}(\mathbf{u})\, \partial_\beta \mathbf{x} \right) \tag{11}$$

since $\partial_\nu \mathbf{x}(\mathbf{u})/\sqrt{g(\mathbf{u})}$ is divergenceless (see Appendix C). This form for the divergence of $\mathbf{t}_\nu^{(k)}(\mathbf{u})$ (which enters into the computation of local corrections for the hypersingular kernel) has the especially desirable property that it avoids the need to compute second or higher order derivatives of the surface.

D. *Extent of Local Correction Domain*

When local testing functions are used, the region over which local corrections should be computed always includes the patch containing the field point, and it extends out to include other patches until the underlying quadrature rule is accurate enough to replicate the exact answer to within a desired tolerance. Since the testing functions have local support, the problem of computing local corrections for a region containing several patches decouples naturally into several smaller local correction problems, one for each patch. The tolerance should be based on an estimate of the optimum accuracy that the particular discretization could achieve; there is, after all, little to be gained by trying to evaluate the impedance matrix more accurately than what is warranted by the discretization. The integrals on the right-hand side of Eq. (8) can be computed by adaptive integration to comparable accuracy.

E. *Local Corrections for "Regular" Parts of the Kernel*

In principle, it is unnecessary to compute local corrections for regular components of the kernel because they will be efficiently integrated by a quadrature rule of sufficiently high order. If such components are strongly peaked, however, the required order may be so high that it is computationally more efficient to treat them as if they were singular and compute local corrections for them. For example, the scalar kernel $\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}')$ in 2D or 3D is a strongly peaked function of $\mathbf{x}'$ when the field point $\mathbf{x}$ is close to, but not on, the source patch. This situation arises in the analysis of scattering from thin layers, for example. One way to handle this problem is to put a fine discretization on each layer, in effect subdividing the strongly peaked kernel function into small parts, each of which is relatively smooth. This procedure is inefficient, however, because it uses many more sample points than are warranted by the expected spatial structure of the source. A better approach would be to discretize each layer densely enough to adequately represent the sources and compute local corrections for the strongly peaked kernel. Computing such local corrections can be a nontrivial task by itself, but one might expect that the extra time spent in precomputation would be compensated by a less time-consuming solution phase.

F. *Using the Results*

F.1. Computing scattered fields. The amplitude of a scattered wave can be computed by convolving the scattered wave with the source distribution. Even though a Nyström discretization specifies the source only at a finite set of points, these points are ideally suited for evaluating integrals in a high-order fashion by virtue of Eq. (2). For example, the amplitude $F(\mathbf{k})$ for 3D scalar scattering of a source distribution $\psi(\mathbf{x})$ on a surface S with Neumann boundary conditions (i.e., $\hat{\mathbf{n}} \cdot \nabla \psi(\mathbf{x}) = 0$ for $\mathbf{x}$ on S) into the plane wave given by $\phi(\mathbf{x}) = e^{i\mathbf{k}\cdot\mathbf{x}}$ is

$$F(\mathbf{k}) = \frac{1}{4\pi} \oint_S ds (\hat{\mathbf{n}} \cdot \nabla \phi^*(\mathbf{x}))\, \psi(\mathbf{x}) \tag{12}$$

$$\cong \frac{1}{4\pi} \sum_i \omega_i (\hat{\mathbf{n}}(\mathbf{x}_i) \cdot \nabla \phi^*(\mathbf{x}_i))\, \psi(\mathbf{x}_i), \tag{13}$$

where the sum is over all quadrature points and $*$ indicates complex conjugation. The extensions to other forms of scattering, whether near- or far-field, are straightforward.

F.2. Source interpolation. When a scattering problem is solved using a Galerkin scattering code, it is obvious how to compute the value of the source distribution at any point on the surface because the solved-for coefficients multiply basis functions that are uniquely defined at every point on the surface. The Nyström discretization, on the other hand, returns values of the sources only at a finite set of discrete sample points, so that determining the value of the source distribution at a point that is not part of this set requires interpolation.

When the scattering computation is performed using a second kind integral formulation, one can use the original Nyström interpolation formula, augmented by local corrections, to interpolate the source distribution. As an example, if the magnetic field integral equation (MFIE) is used to solve for the electric current distribution $\mathbf{J}(\mathbf{x})$ induced on a perfectly electrically conducting (PEC) scatterer by an incident magnetic field $\mathbf{H}^{\text{inc}}(\mathbf{x})$, one can write

the current at any point $\mathbf{x}$ on the surface S as [17]

$$\mathbf{J}(\mathbf{x}) = 2\hat{\mathbf{n}}(\mathbf{x}) \times \left[\mathbf{H}^{inc}(\mathbf{x}) - \oint_S ds' \, \nabla' G(\mathbf{x}, \mathbf{x}') \times \mathbf{J}(\mathbf{x}') \right]. \tag{14}$$

We obtain an interpolation formula from this continuous equation by using Eq. (2) to approximate the integral, i.e.,

$$\mathbf{J}(\mathbf{x}) = 2\hat{\mathbf{n}}(\mathbf{x}) \times \left[\mathbf{H}^{inc}(\mathbf{x}) - \sum_i \omega_i \nabla' G(\mathbf{x}, \mathbf{x}_i) \times \mathbf{J}(\mathbf{x}_i) \right], \tag{15}$$

where the sum over i extends over all sample points on S. Of course, to make this a high-order interpolation formula, it may be necessary to compute local corrections to the quadrature rule at source points in the vicinity of the field point $\mathbf{x}$.

Another interpolating function, which does not require computing new local corrections and is usable with first or second kind integral formulations, takes the form of a linear combination of the functions that are integrated exactly by the underlying quadrature rule. The coefficients may be determined by convolving the source with the projection operator

$$I(\mathbf{x}, \mathbf{x}') = \sum_{m,n} f_m(\mathbf{x}) (N^{-1})_{mn} \, f_n(\mathbf{x}'), \tag{16}$$

where the summation extends over all functions $f_i(\mathbf{x})$ for which the quadrature rule is exact, and N is a normalization matrix whose components are given by

$$N_{mn} = \int_S ds \, f_m(\mathbf{x}) \, f_n(\mathbf{x}). \tag{17}$$

If the $f_i(\mathbf{x})$'s are orthonormal over S, then N is simply the identity matrix. Convolution with $I(\mathbf{x}, \mathbf{x}')$ eliminates the part of a function that is orthogonal to all the $f_i(\mathbf{x})$'s. If we evaluate the convolution of $I(\mathbf{x}, \mathbf{x}')$ with the source function by means of the underlying quadrature rule, we arrive at the following source interpolation function $s(\mathbf{x})$, which only requires knowledge of the source at the discrete set of sample points $s(\mathbf{x}_i)$:

$$s(\mathbf{x}) = \sum_{m,n} f_m(\mathbf{x}) (N^{-1})_{mn} \sum_i \omega_i \, f_n(\mathbf{x}_i) \, s(\mathbf{x}_i). \tag{18}$$

The summation over i in the above equation extends over all sample points.

IV. RESULTS

This section is composed of two parts. The objective of the first part is to show that our most recent version of FastScat, which uses a Nyström discretization, achieves high-order convergence to the correct answer for a few small, benchmark problems from 2D scalar and 3D vector scattering. In the second part, we benchmark the performance of this code against two Galerkin codes, comparing them on the basis of CPU time and solution accuracy.

A. *Validation*

The most common practice seen in the literature for demonstrating the validity of a scattering code is to show that the results obtained from the code with a *particular* discretization compare favorably to a reference solution obtained from a series solution, another scattering code, or measurements. Individual results such as this, while useful and necessary, say nothing about the convergence properties of the algorithm on which the code is based. To show how an algorithm converges, one must compute results with a sequence of increasingly fine discretizations and observe whether and how the results converge to the correct answer.

This is especially important when validating a (purportedly) high-order code. One cannot expect to enjoy the benefits of a high-order code (more accurate solutions, solution error control, etc.) on large scattering problems without first verifying that the code achieves high-order convergence on small scattering problems (where it is easier to generate solutions with very small errors). The order of convergence of a numerical method relates to the rate at which the error in the computed solution decreases as the discretization scale decreases. For small enough discretization scales h, the error in the solution computed by a pth-order method scales as h^p. The results presented in this section will be shown to follow this scaling law.

The benchmark problems include a circle and an ellipse in 2D, and a sphere and an ellipsoid in 3D. In the 2D scalar scattering cases, results for both Dirichlet and Neumann boundary conditions on the surface will be presented; in the 3D vector (electromagnetic) scattering cases, it will be assumed that the surfaces are perfect conductors. The surface boundary conditions are chosen mainly for simplicity; similar convergence behavior has been shown for other types of boundary conditions (such as impedance boundary conditions and dielectric interfaces) as well.

A.1. Two-dimensional scalar. We solved four different integral equations to obtain 2D scalar scattering results. For Dirichlet boundary conditions (which correspond to the TM polarization case of electromagnetic scattering from an object with cylindrical symmetry) the first-kind integral equation is

$$\phi^{\text{inc}}(\mathbf{x}) = -\oint_C dl'\, G(\mathbf{x}, \mathbf{x}')\, \sigma(\mathbf{x}'), \tag{19}$$

and the second-kind equation is

$$-\hat{\mathbf{n}} \cdot \nabla \phi^{\text{inc}}(\mathbf{x}) = \frac{1}{2}\sigma(\mathbf{x}) + \oint_C dl'(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}'))\, \sigma(\mathbf{x}'). \tag{20}$$

In these equations $\phi^{\text{inc}}(\mathbf{x})$ is the incident scalar field, $G(\mathbf{x}, \mathbf{x}')$ is the 2D scalar kernel, and $\hat{\mathbf{n}}$ and $\hat{\mathbf{n}}'$ are the unit normals to the contour C at the field and source points, respectively. For this polarization case, the 2D scalar source σ is proportional to the z component of the electric current $\mathbf{J}$ in the corresponding 3D vector problem, assuming $\mathbf{z}$ is the axis of translational symmetry.

For Neumann boundary conditions (which correspond to the TE polarization case of electromagnetic scattering) the first-kind integral equation is

$$\hat{\mathbf{n}} \cdot \nabla \phi^{\text{inc}}(\mathbf{x}) = \oint_C dl'(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}'))\, \psi(\mathbf{x}') \tag{21}$$

and the second-kind equation is

$$\phi^{\text{inc}}(\mathbf{x}) = \frac{1}{2}\psi(\mathbf{x}) - \oint_C dl'(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}'))\, \psi(\mathbf{x}'). \tag{22}$$

For this polarization case, the electric current **J** in the corresponding 3D vector problem, assuming **z** is the axis of translational symmetry, is related to the 2D scalar source ψ by

$$\mathbf{J} = \psi \hat{\mathbf{n}} \times \hat{\mathbf{z}}. \tag{23}$$

A combined field equation can be obtained in either case by adding the first and second kind equations together using an appropriate combination coefficient [18]. Although no combined field equation results are reported here, it should be noted that use of a combined field formulation is often recommended because, by being insensitive to internal resonances, it can improve the condition number of the impedance matrix.

A.1.a. 1 λ-*radius circle*. A circle is the ideal problem for benchmarking a high-order scattering code because its surface is smooth and easy to define exactly, and its cross section can be determined, for purposes of comparison, to arbitrary accuracy using the Mie series [19]. We used FastScat to compute the bistatic cross section of a 1 λ-radius circle whose surface obeys either Dirichlet or Neumann boundary conditions, which correspond to TM and TE polarizations, respectively. Meshing the circle consisted of dividing it into circular segments of equal arc length. Nyström sample points were distributed on each patch (parameterized by arc length) according to a Gauss–Legendre integration rule of a given order and Legendre polynomial testing functions up to half this order were used for computing local corrections. The resultant local correction linear systems are square.

We performed a series of calculations with different discretizations (i.e., different numbers of patches and different Nyström quadrature orders) and compared the results to the Mie series results (shown in Fig. 1). For a given Nyström quadrature order (which we henceforth

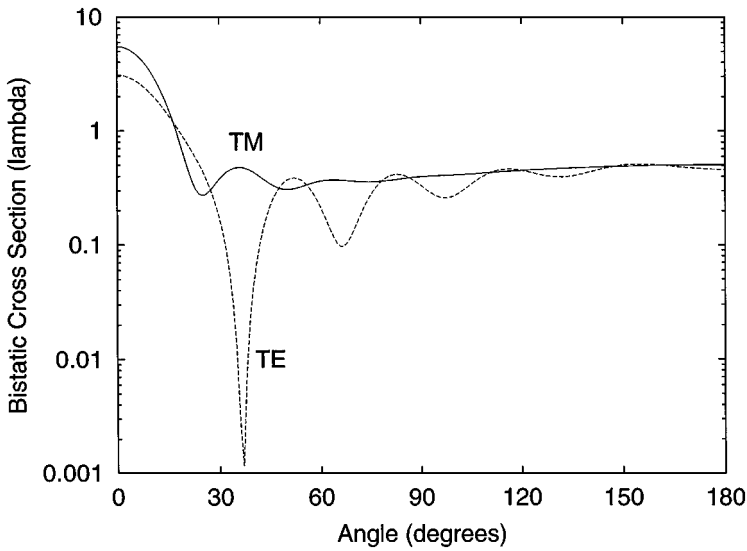

FIG. 1. Bistatic cross section of a 1 λ-radius circle for TM and TE polarizations computed by the Mie series.

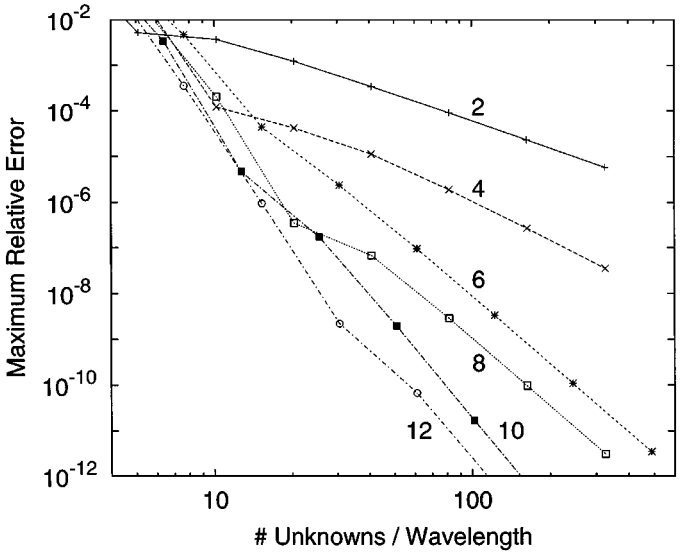

FIG. 2. Log–log plot of maximum relative error vs unknown density for 1 λ-radius circle and TM polarization. Each set of points is labeled by Nyström order.

abbreviate to Nyström order), as the size of the patches decreases, the difference between the exact result and the FastScat calculation also decreases.

A more quantitative measure of convergence behavior is given in Fig. 2, where we have plotted maximum relative error (defined as $\max[|\sigma(\theta)/\sigma_{\text{ref}}(\theta) - 1|]$, where $\sigma(\theta)$ and $\sigma_{\text{ref}}(\theta)$ are the calculated and exact cross sections, respectively, for $\theta = 0$ to 180° in 1° increments) versus the density of unknowns for a first-kind integral formulation of the TM polarization case. The number of patches spanning the circle ranged from 4 to 2048 and the Nyström order ranged from 2 to 12. One of the important features to note is that, with enough unknowns, the data fit a linear trend line whose slope increases as the Nyström order increases. Since the discretization scale h is inversely proportional to the density of unknowns, a linear fit on a log–log plot of error versus unknown density reflects the fact that the error scales asymptotically as h^p, where p (the order of convergence) increases with Nyström order. Large values of p signify a high-order algorithm. For the lower Nyström orders, the slopes of the lines connecting points of a given order are observed to be close to integers, namely 2 for order 2; 3 for order 4; and 5 for orders 6 and 8. The slopes for orders 10 and 12 are still higher, although even at the highest sampling densities used, the discretization error has not yet reached the asymptotic regime where each would be expected to have a slope of 7.

The results for the second-kind integral formulation of the TM polarization case are very similar. This should not be too surprising, since, despite the additional derivative, the singularity of the kernel is no worse than $\log(r)$.

The corresponding plot for the TE polarization case, also using a first-kind integral formulation, is shown in Fig. 3. In the TE case, however, the first-kind integral equation involves the 2D hypersingular kernel. The effect of using a more singular kernel is that the source must be represented more accurately in order to achieve the same accuracy in the cross section, or equivalently, that an equally well represented source (i.e., one employing the same collection of unknowns) produces a less accurate value for the cross section. This

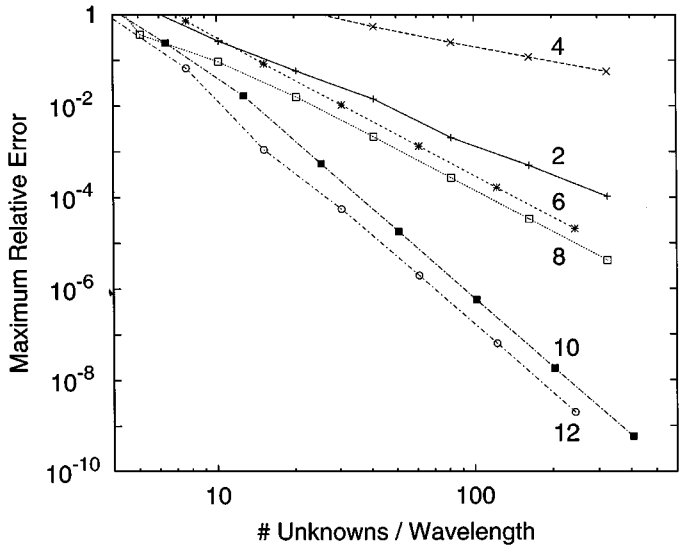

FIG. 3. Log–log plot of maximum relative error vs unknown density for 1 λ-radius circle and TE polarization. Each set of points is labeled by Nyström order.

is easily seen by comparing Figs. 2 and 3. For a given discretization, the calculated cross section for the TE case is two or more orders of magnitude less accurate than that for the TM polarization. Nonetheless, the TE polarization data also fit linear trend lines with integer slopes when the discretization is fine enough. In order from lowest (2) to highest (12) Nyström orders, the observed slopes are 2, 1, 3, 3, 5, and 5.

Cross section calculations resulting from the second-kind formulation of the TE polarization scattering problem are generally more accurate than those of the first-kind formulation. In fact, as the Nyström order increases, they become nearly as accurate as those for the TM polarization case. Again, the reason is that the singularity of the kernel for the second-kind TE case is no worse than $\log(r)$, which is also the singularity of the kernels in the first and second-kind TM polarization cases.

The process of improving a discretization by reducing the size of the patches is called "*h*-refinement." This is what has been exhibited in the previous two figures. Keeping the number of patches fixed and increasing the number of parameters used to describe the source distribution on each patch, on the other hand, is known as "*p*-refinement." With a high-order Nyström code such as FastScat, *p*-refinement is accomplished by increasing the Nyström order for a given meshing. In general, this is the preferred method for improving a discretization for two reasons: one can avoid the usually tedious process of remeshing the scatterer, and the accuracy of the answer usually improves faster this way. The data in the next plot demonstrate this feature.

Figure 4 presents the TM and TE polarization data given in Figs. 2 and 3 in a different way. The behavior of the calculation for each polarization under *p*-refinement is illustrated by connecting points corresponding to a fixed number of patches instead of a fixed Nyström order. In some cases, data points corresponding to Nyström orders higher than 12 have been added. The fact that the data points on a semilog plot can be connected by nearly straight lines indicates that *p*-refinement can achieve *exponential* convergence, as opposed to the *geometric* convergence that was observed for *h*-refinement. The convergence rate gets higher the larger the patch size.

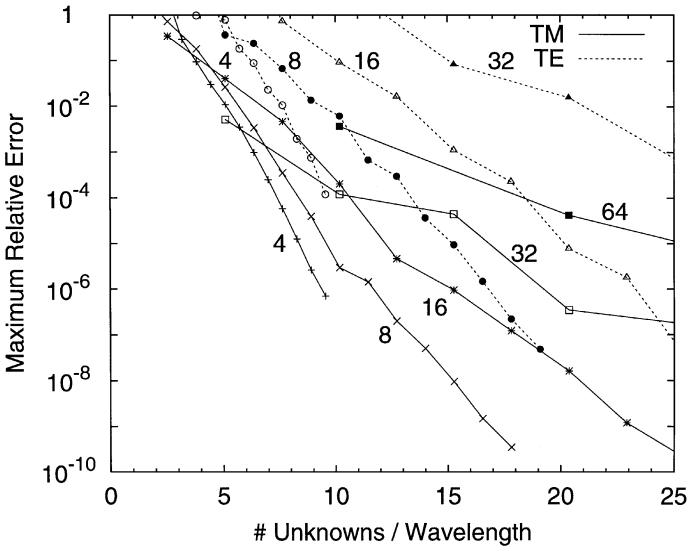

FIG. 4. Semilog plot of maximum relative error vs unknown density for scattering from a 1 λ-radius circle. Points corresponding to different Nyström quadrature orders for a fixed patch size are connected by lines (*solid* for TM polarization and *dashed* for TE polarization) and labeled by the number of patches.

With regard to numbers of unknowns, the most efficient way to achieve high accuracy is to use a high-order method on large patches. For example, with only four patches and a 30th-order quadrature rule, it was possible to achieve an accuracy of 10^{-6} for the TM polarization case and 10^{-4} in the TE case. With this discretization, the unknown density is about 10 unknowns/wavelength and the arc length of each patch is about $1\frac{1}{2}$ wavelengths. For lower accuracies, the advantage of using large patches and high-order methods on the circle is less clear. As a general rule, the optimum discretization is one that uses large patches and high-order methods over smooth regions of the scatterer and smaller patches over more highly curved regions.

A.1.b. 20 λ × 2 λ *ellipse*. A 20 λ × 2 λ ellipse is a 2D scatterer that is less symmetric than a circle, but is still smooth. It is a more challenging scattering problem than a 1 λ-radius circle for several reasons, not least of which is the fact that it extends much more than a wavelength in at least one dimension. In addition, it is a good candidate problem for applying the discretization rule described above.

In our code, the ellipse is described by the pair of parametric equations,

$$\begin{aligned} x &= a\cos u, \\ y &= b\sin u, \end{aligned} \tag{24}$$

where $a = 10\ \lambda$ and $b = 1\ \lambda$. A sensible patching, which puts the highest density of patches in the most highly curved regions and vice versa for the flatter regions, is obtained if the patches cover equal increments in the parameter u. The circumference of a 20 λ × 2 λ ellipse is about 40.64 λ.

We used FastScat to compute the monostatic cross section of a 20 λ × 2 λ ellipse discretized using several different combinations of patch number and Nyström order. The boundary conditions on the surface were either Dirichlet or Neumann, corresponding to TM and TE polarizations, respectively.

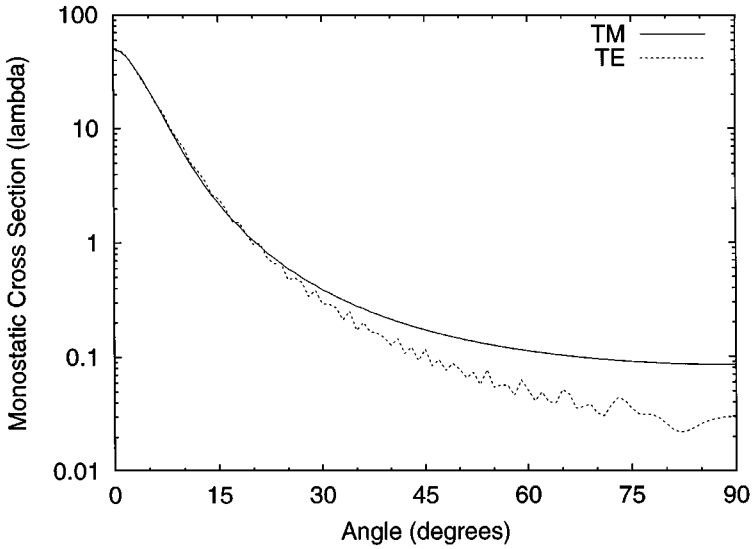

FIG. 5. Monostatic cross section of a 20 $\lambda \times 2$ λ ellipse for TM and TE polarizations. One quadrant of observation angles is shown; the others may be obtained by considering the fourfold symmetry of the scatterer.

We do not have at our disposal a series solution for the cross section of an ellipse (which we might otherwise use to compute an arbitrarily accurate reference solution). However, we can still estimate the accuracy of the computed solutions by comparing them to the most finely discretized solution, which we designate the "reference solution." We computed reference solutions for the TM and TE polarization cases by meshing the ellipse into 128 patches and putting a 20th-order Gauss-Legendre rule (i.e., 10 sample points) on each patch. We deduce that these reference solutions are accurate to at least six decimal places, given the high-order manner in which all the more coarsely discretized solutions are observed to converge to them. Plots of the monostatic cross section versus incident angle for the reference solutions are given in Fig. 5. As seen in the figure, the monostatic cross section for TM polarization ranges from about 50 λ looking at the broadside to less than 0.1 λ looking at the tip. The TE cross section is similar, although it is not as smooth a function of angle. In both cases, the dynamic range of the cross section is more than 500.

The *p*-refinement behavior of the calculations on the ellipse using first-kind integral equation formulations for both TM and TE polarization is shown in Fig. 6. Like the circle, exponential convergence is observed and accurate solutions are most efficiently obtained when the mesh consists of patches larger than a wavelength.

A.2. Three-dimensional vector. As in the 2D scalar case, first-kind and second-kind integral formulations were explored. For 3D vector scattering off a PEC scatterer, the first-kind formulation is the electric field integral equation (EFIE) [17]

$$\mathbf{E}^{\text{inc}}_{\tan}(\mathbf{x}) = i\omega \oint_S ds' \left[-G(\mathbf{x}, \mathbf{x}')\, \mathbf{J}(\mathbf{x}') + \frac{1}{k^2} \nabla(\nabla' G(\mathbf{x}, \mathbf{x}') \cdot \mathbf{J}(\mathbf{x}')) \right]_{\tan}, \tag{25}$$

and the second-kind formulation is the magnetic field integral equation (MFIE)

$$\mathbf{H}^{\text{inc}}_{\tan}(\mathbf{x}) = -\frac{1}{2} \hat{\mathbf{n}} \times \mathbf{J}(\mathbf{x}) + \oint_S ds' [\nabla' G(\mathbf{x}, \mathbf{x}') \times \mathbf{J}(\mathbf{x}')]_{\tan}, \tag{26}$$

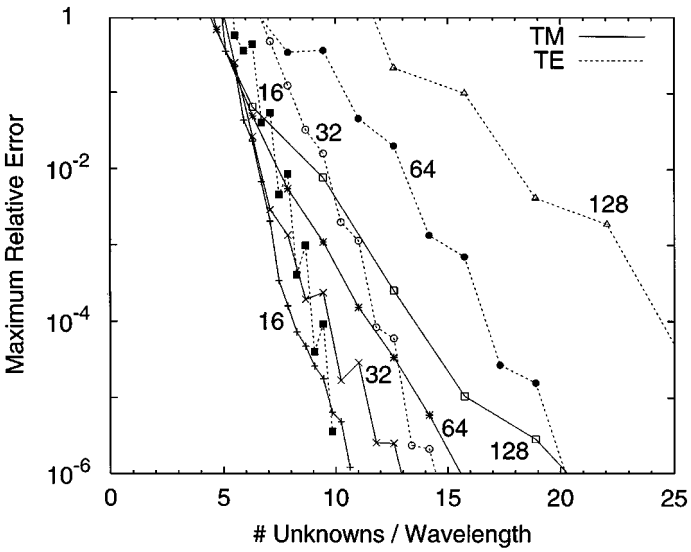

FIG. 6. Semilog plot of maximum relative error vs unknown density for scattering from a 20 $\lambda \times 2\ \lambda$ ellipse. Points corresponding to different Nyström orders for a fixed patch size are connected by lines (*solid* for TM polarization and *dashed* for TE polarization) and labeled by the number of patches.

where $G(\mathbf{x}, \mathbf{x}') \equiv \exp(ik|\mathbf{x}-\mathbf{x}'|)/|\mathbf{x}-\mathbf{x}'|$ is the Helmholtz kernel in 3D, $k = |\mathbf{k}| = \omega/c$ is the radiation wavenumber, $\mathbf{J}$ refers to the electric surface current, $\mathbf{E}^{\text{inc}}$ and $\mathbf{H}^{\text{inc}}$ are the incident electric and magnetic fields, and the subscript *tan* means that only the vector components tangent to surface at the field point are being used.

The EFIE and MFIE can be summed to form a combined field integral equation (CFIE) having some of the same desirable properties as the CFIE in the 2D scalar case. Although no CFIE results are reported in this paper, the same techniques apply.

Note also, that, while the results presented here are restricted to PEC scatterers, it is trivial to generalize the method to the more general scattering problem of homogeneous regions with smooth boundaries.

A.2.a. *One-fourth λ-radius sphere*. Writing a code that correctly calculates 3D vector scattering results is more difficult than writing a correct 2D scalar code. This is doubly true if the code is designed to be high order. Therefore, it is particularly important to verify that the output of a purportedly high-order 3D vector code actually converges to the correct answer under both *h*- and *p*-refinement and that it does so in a high-order fashion. In this subsection, we present results demonstrating that our 3D vector Nyström code achieves high-order convergence to the correct answer on a sphere.

A sphere is the ideal surface to use for benchmarking a high-order 3D vector code for the same reasons that a circle is ideal for a high-order 2D scalar code—it is uniformly smooth and the accuracy of computed results can be determined by comparison to the Mie series solution. Since the size of the surface, and therefore the number of unknowns, grows in proportion to r^2 for a sphere, as opposed to just r for a circle, memory limitations prevented us from pushing the unknown density on a 1 λ-radius sphere to the same extremes as were possible on a 1 λ-radius circle. Nonetheless, when we did run FastScat on a 1 λ-radius sphere with a wide selection of discretizations, we found that the results converged to the correct answer just as one would expect for a high-order scattering code. To reach the asymptotic regime, where the convergence behavior is more obvious, however, we chose the radius

TABLE I
3D Quadrature Rule and Testing Function Parameters

Nyström quadrature order	Number sample points	Maximum testing function degree	Number testing functions
2	1	0	1
3	3	1	3
5	6	2	6
7	12	3	10
8	15	4	15

of the sphere to be $\frac{1}{4}\lambda$, which allows us to increase the unknown density fourfold before running out of primary memory (for storing the full impedance matrix). For this reason alone we present the data for the $\frac{1}{4}\lambda$-radius sphere.

The internal surface representation of the sphere corresponds to an ideal sphere and its surface is assumed to be perfectly conducting. The coarsest patching of the sphere consists of 20 identical triangular patches, formed by mapping the triangles of an inscribed icosahedron onto the surface of the sphere. Finer meshes were generated by dividing each of the 20 triangles into n^2 nearly identical subtriangles, where n ranged from 2 up to 10. The distribution of Nyström quadrature points on each patch was determined by a high-order triangle rule [16]. The triangle rule orders that we used and corresponding numbers of sample points are given in Table I. The number of testing functions (products of monomials in the two surface parameters) and the maximum degree of the testing functions used with each triangle rule are also listed in the table.

In all cases except Nyström order 7, the number of sample points equals the number of testing functions, resulting in an exactly-determined local correction linear system. In the seventh-order case, the maximum testing function degree was chosen to make an under-determined linear system.

Solutions for the bistatic cross section of the $\frac{1}{4}\lambda$-radius sphere were computed with the various discretizations and compared against the Mie series solution (shown in Fig. 7). For a sphere this small, the cross sections for the two polarizations are similar (in terms of smoothness and dynamic range), so we present the discretization refinement results only for the $\theta\theta$ case. Cross polarization results are also not presented at all, although it may be noted that such computed cross sections were extremely small (i.e., always less than the co-polarized results by at least eight orders of magnitude).

The convergence behavior of the scattering results under h-refinement is shown in Fig. 8. Refining the mesh for a given Nyström order always improves the accuracy of the solution. It is apparent for the lower Nyström orders that the data approach linear trend lines with integer slopes as the patches get smaller, just as they did in 2D. In the case of the EFIE, the slopes of the trend lines for Nyström orders 2 and 3 are both unity and in the case of the MFIE, they are 2 and 3, respectively. For the higher orders, the slopes appear to be increasing, but it is not as clear what their asymptotic values will be. For Nyström order 5, the last pair of points produce slopes close to 3 and 5 for the EFIE and MFIE solutions, respectively. In all cases, the solution at a particular discretization obtained by using the less singular kernel (i.e., the MFIE) is more accurate.

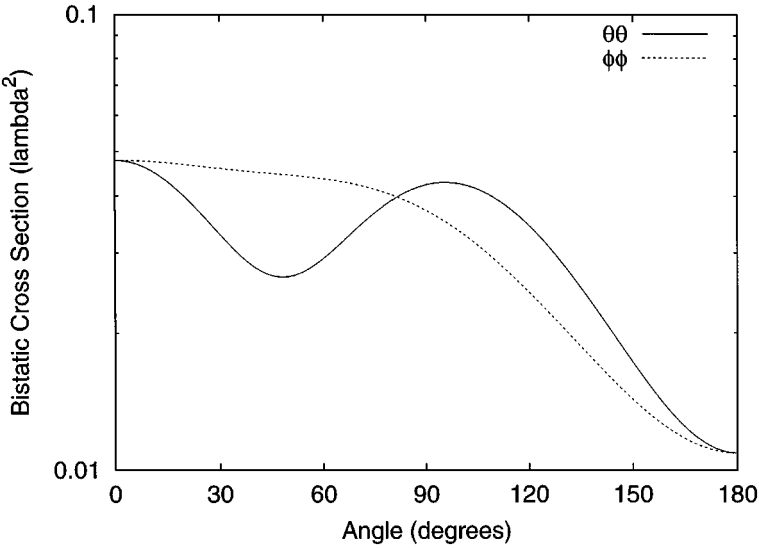

FIG. 7. Bistatic cross section of a $\frac{1}{4}\lambda$-radius PEC sphere for $\theta\theta$ and $\phi\phi$ polarizations computed by the Mie series.

The behavior of the sphere results under p-refinement are shown in Fig. 9. The observed p-refinement behavior is similar to that in the 2D scalar case. The fastest convergence is usually achieved by applying a high-order quadrature to a coarse meshing. One notable difference from the 2D scalar case is that the 3D vector calculation requires a higher density of unknowns to achieve a comparable maximum relative error in the bistatic cross section. The jaggedness of the p-refinement curves for the EFIE data may be explained by reference to the h-refinement plot, which shows that the 2nd- and 3rd-order results have nearly the same accuracy, and that the 7th-order results are actually less accurate than those for 5th-order.

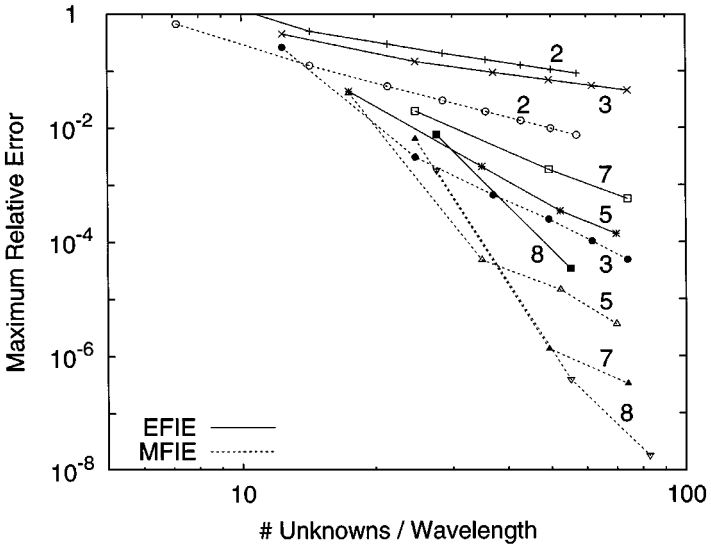

FIG. 8. Log–log plot of maximum relative error vs unknown density for $\frac{1}{4}\lambda$-radius PEC sphere in $\theta\theta$ polarization. Points obtained with different meshings but the same Nyström order are connected by lines. A solid (dashed) line indicates use of the EFIE (MFIE) integral formulation.

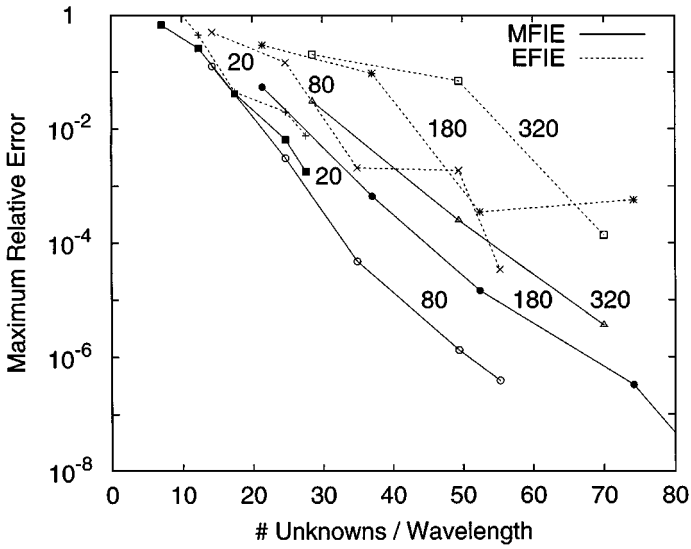

FIG. 9. Semilog plot of maximum relative error vs unknown density for scattering from a $\frac{1}{4}\lambda$-radius PEC sphere. Points corresponding to different Nyström quadrature orders for a fixed patch size are connected by lines (*solid* for MFIE and *dashed* for EFIE) and labeled by the number of patches.

For Nyström orders higher than about 8, problems related to ill-conditioning arise in the EFIE formulation. Although the increasingly ill-conditioned nature of the local correction linear system is a contributing factor, the more important contribution probably comes from the fact that the EFIE is especially susceptible to conditioning problems when the Nyström sample points get too close together. Unfortunately, this is exactly what happens for the higher-order triangle rules. As the order increases, the quadrature points tend to bunch up near the edges and corners of the triangle. It may be possible to overcome this problem by inventing different high-order triangle rules with better sample point spacing and by using a better conditioned integral equation formulation such as the MFIE or CFIE (combined field integral equation).

A.2.b. $2\ \lambda \times 2\ \lambda \times 0.2\ \lambda$ *ellipsoid*. As an example of a smooth, but less symmetric 3D scatterer, we next consider a PEC ellipsoid with principal axis diameters $2\ \lambda$, $2\ \lambda$, and $0.2\ \lambda$. We computed the monostatic cross sections of this discus-shaped scatterer in $\theta\theta$ and $\phi\phi$ polarizations using a MFIE formulation and an eighth-order quadrature rule, which put 15 points on each patch. Four different meshings, comprising 20, 80, 180, and 320 patches, were tried. Each meshing was tailored to put smaller patches in the vicinity of the $r = 1\ \lambda$ equator, where the one of the radii of curvature is small, and larger patches everywhere else, where the surface is relatively flat. The number of unknowns distributed over the $6.47\ \lambda^2$ surface of the ellipsoid in the four cases ranged from 600 with the coarsest meshing to 9600 with the finest.

As we did with the ellipse in 2D, we can designate the solution computed with the finest discretization to be the reference solution and obtain accuracy estimates of the other solutions by comparing them to this reference solution. Figure 10 shows the reference solutions for the $\theta\theta$ and $\phi\phi$ polarization cases.

Differences between the reference solution and the other, less finely discretized solutions are shown in Fig. 11. As expected, the accuracy of the solution improves as one refines the

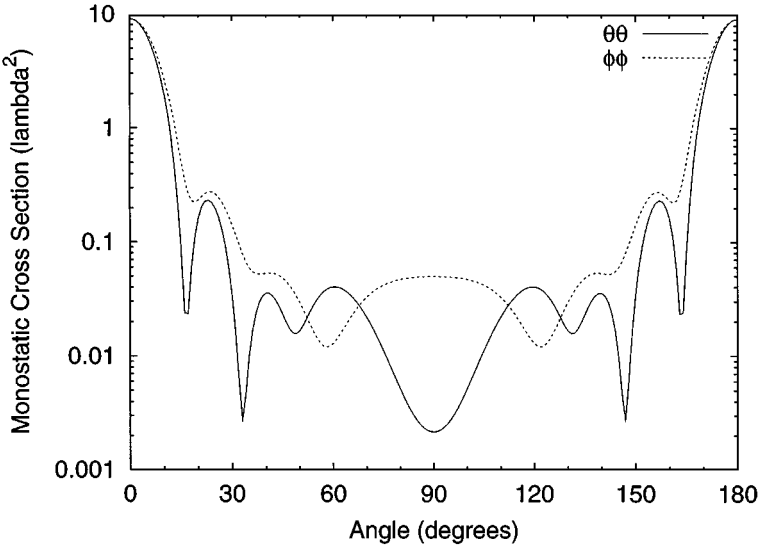

FIG. 10. Reference solutions for the monostatic cross section of a $2\ \lambda \times 2\ \lambda \times 0.2\ \lambda$ PEC ellipsoid in $\theta\theta$ and $\phi\phi$ polarizations. At 0° the observer is looking at the flattest part of the ellipsoid; at 90° he is looking edge on.

discretization. It should also come as no surprise that the solutions are also most accurate near 0° and 180°, where the cross section is highest. What is particularly notable about this plot, however, is the fact that the error in the cross section decreases by orders of magnitude when one reduces the (linear) size of each patch by factors of 2 or 3. Such large reductions in the error are a direct consequence of our using an exact surface description and a high-order rule (8th-order, in this case) on each patch.

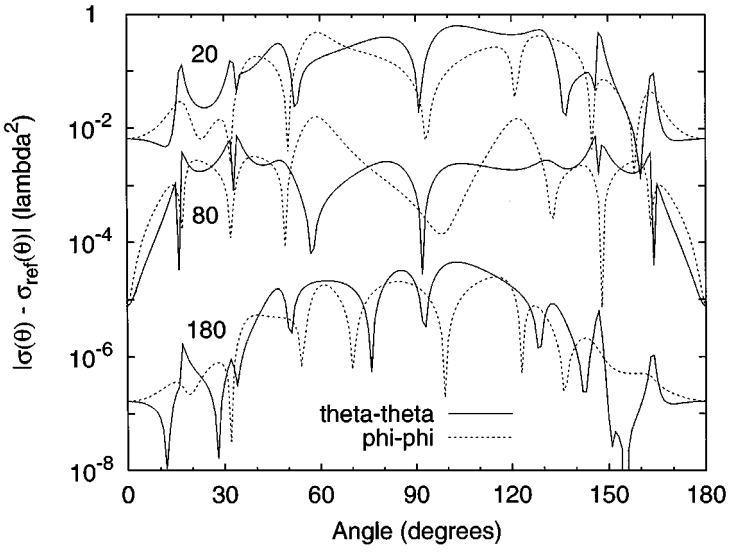

FIG. 11. Semilog plot of the differences between cross sections computed using meshings consisting of 20, 80, and 180 patches, and a reference cross section computed using a meshing consisting of 320 patches. The asymmetry of each curve reflects the fact that the meshings did not possess reflection symmetry.

B. *Run-Time Performance Comparisons*

In this section we compare the run-time performance of our high-order Nyström implementation of FastScat to that of two method of moments scattering codes. The first comparison code is an earlier, high-order Galerkin implementation of FastScat [20]. The second is a low-order code (RWG basis and testing functions on flat facets) called FISC [21]. We ran each code under comparable conditions to obtain solutions for the bistatic cross section in the $\theta\theta$ polarization of three different size PEC spheres. The high-order Nyström discretizations were constructed using an eighth-order quadrature rule (15 sample points per patch) and fourth-degree testing functions for computing local corrections. The high-order Galerkin discretizations were constructed from the same surface mesh using patch-based, polynomial (in the parameterization) basis functions up to degree 4 to give the same number of unknowns per patch, namely 30. The surface mesh used by FISC was necessarily different from that used by both versions of FastScat because, with an RWG discretization, one unknown is associated with each edge rather than multiple unknowns being associated with each patch. Nonetheless, its surface meshes were constructed to maintain the density of unknowns at about 7.7 unknowns/wavelength, the same as for the both FastScat discretizations. All computations were performed using a dense matrix fill, an LUD solver, and a MFIE formulation.

Table II gives a summary of the results. The reported times are run times on a SPARC-10 workstation with 512 MB primary memory. The total run time is broken into setup time (which includes the time spent setting up the problem and filling the impedance matrix) and solve time (which includes the time spent performing the LUD and solving for the bistatic cross section at 181 angles).

In comparing the results from the two high-order implementations of FastScat, two features are especially noteworthy. The first is that the high-order Galerkin result is more accurate by about a factor of 5 than the high-order Nyström result. The second is that use of the Nyström discretization can speed up the setup phase of the computation enormously, with the speedup factor increasing as the number of unknowns increases. The observation that the high-order Galerkin code computes results somewhat more accurately than the Nyström code is consistent with our experience computing cross sections for other scatterers, both in 2D and 3D. It is compensated, however, by the fact that the setup phase (and to a lesser extent the solve phase) runs much faster using the Nyström code. Furthermore, the factor of 5 difference in accuracy is actually less significant in this case than it would

TABLE II
Nyström vs Galerkin Performance on PEC Spheres

Scattering code	Radius (λ)	No. of unknowns	Setup time (s)	Solve time (s)	RMS error (dB)
FastScat (Nyström)	0.9	600	74	36	0.35
FastScat (Galerkin)	0.9	600	972	88	0.07
FISC (Galerkin)	0.9	600	83	42	1.28
FastScat (Nyström)	1.8	2400	539	2742	0.26
FastScat (Galerkin)	1.8	2400	8177	3395	0.05
FISC (Galerkin)	1.8	2430	873	2255	0.61
FastScat (Nyström)	2.7	5400	1953	31735	0.097
FastScat (Galerkin)	2.7	5400	38803	36152	0.021
FISC (Galerkin)	2.7	5880	8230	28795	0.723

be if we were comparing low-order codes. Given the $\mathcal{O}(h^9)$ convergence rate expected of an eighth-order quadrature rule, it should be possible to recover the factor of 5 in accuracy with further h-refinement by a modest 20%.

The high-order Nyström code computes more accurate answers than the low-order Galerkin code (FISC) in all cases. For the spheres considered here, this is largely due to the fact that FISC uses a low-order surface representation. The high-order Nyström code also requires less setup time, an advantage that grows as the problems get bigger. Even a comparison based on total solution time shows the high-order Nyström implementation of FastScat to be more efficient for computing accurate answers.

Finally, it is useful to note that an equivalent Nyström discretization exists for every method of moments discretization and vice versa [22], so it is possible, at least in principle, to eliminate the observed accuracy discrepancy between the two versions of FastScat by implementing a Nyström code whose discretization error precisely matches that obtained by the Galerkin code. We have not attempted to do this, but suspect that to do so would entail additional complications and computations that would negate the substantial simplicity and efficiency of the present implementation. On balance, we find the high-order Nyström method in its present form preferable to the high-order Galerkin method for solving integral equations, especially when one adds in its other benefits such as reduced implementation complexity and potential for significantly improved FMM performance.

V. SUMMARY

The standard Nyström method is a simple and efficient mechanism for discretizing integral equations. We have shown how it can be adapted to provide a high-order discretization of the boundary integral equations of wave scattering in 2D and 3D, which have singular kernels. Numerical results obtained with a software implementation of this method show that the algorithm can achieve high-order convergence to the correct answer for scattering cross sections in 2D and 3D. We also demonstrated that a high-order Nyström code considerably reduces the CPU time cost of a scattering calculation by comparison to a high-order Galerkin code, especially the precomputation time cost. The high-order Nyström code also outperformed a well-tuned, low-order Galerkin code (FISC) in terms of solution accuracy and total run time. Demonstrations of how a high-order Nyström code can be used in conjunction with the FMM to reduce the memory and CPU time requirements of solving large scattering problems will be the subject of a future publication.

APPENDIX

A. *Local Corrections*

Eleven different kernels arise in boundary integral equation formulations of 2D scalar, 3D scalar, and 3D electromagnetic scattering:

2D & 3D Scalar	3D Electromagnetic
$G(r)$	$G(r)(\mathbf{t}(\mathbf{x}) \cdot \mathbf{t}'(\mathbf{x}'))$
$\hat{\mathbf{n}}' \cdot \nabla' G(r)$	$\mathbf{t}(\mathbf{x}) \cdot (\nabla' G(r) \times \mathbf{t}'(\mathbf{x}'))$
$\hat{\mathbf{n}} \cdot \nabla G(r)$	$(\mathbf{t}(\mathbf{x}) \cdot \nabla)(\nabla' G(r) \cdot \mathbf{t}'(\mathbf{x}'))$
$(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G(r))$	

where

$$G(r) \equiv \begin{cases} \frac{i}{4} H_0^{(1)}(kr) & \text{in 2D,} \\ \frac{e^{ikr}}{r} & \text{in 3D,} \end{cases} \tag{27}$$

r is the magnitude of the vector $\mathbf{r} \equiv \mathbf{x}' - \mathbf{x}$ from the field point at $\mathbf{x}$ to the source point at $\mathbf{x}'$; k is the wavenumber of the waves; $\hat{\mathbf{n}}$ and $\hat{\mathbf{n}}'$ are the unit normals to the surface at the field and source points, respectively; ∇ and ∇' are gradient operators for the field and source coordinates, respectively; and $H_0^{(1)}$ refers to the zeroth order Hankel function of the first kind, defined by $H_0^{(1)}(x) \equiv J_0(x) + iY_0(x)$, where $J_n(x)$ and $Y_n(x)$ represent nth-order Bessel functions of the first and second kinds, respectively.

For the 3D electromagnetic case, the source and excitation are surface tangent vectors so it becomes necessary to compute local corrections for four scalar kernels, one for each of the four combinations of (two) independent surface tangent vectors at the field point and (two) independent surface tangent vectors at the source point. These surface tangent vectors at the field and source points, represented by $\mathbf{t}(\mathbf{x})$ and $\mathbf{t}'(\mathbf{x}')$, respectively, are included as part of the 3D electromagnetic kernel in recognition of this fact and for clarity of presentation.

In this section, we show how to compute local corrections for each of these kernels. We will make use of the vector calculus identity [23]

$$(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' g(r)) = (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}')(\nabla \cdot \nabla' g(r)) - (\hat{\mathbf{n}} \times \nabla) \cdot (\hat{\mathbf{n}}' \times \nabla' g(r)) \tag{28}$$

$$= (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}')\, k^2 g(r) - (\hat{\mathbf{n}} \times \nabla) \cdot (\hat{\mathbf{n}}' \times \nabla' g(r)), \tag{29}$$

where the second line follows if $g(r)$ obeys the homogeneous Helmholtz equation

$$(\nabla^2 + k^2) g(r) = 0. \tag{30}$$

This identity allows one to convert between double normal derivative and double tangential derivative operators on the Green function.

A.1. Two-dimensional scalar.

A.1.a. $G(r)$,

$$G(r) = \frac{i}{4} H_0^{(1)}(kr) = \underbrace{\frac{i}{4} J_0(kr)}_{\text{regular}} - \underbrace{\frac{1}{4} Y_0(kr)}_{\text{singular}}. \tag{31}$$

This kernel may be written as the sum of a regular part and a singular part. It is necessary to compute local corrections only for the singular part because the regular part will be efficiently integrated by the underlying high-order quadrature rule. The function $Y_0(kr)$ contains a $\log(r)$ singularity. Therefore, one can use "lin–log" quadrature rules [24] to efficiently compute local correction integrals when the region of integration contains the field point, and Gauss–Legendre rules otherwise.

A.1.b. $\hat{\mathbf{n}}' \cdot \nabla' G(r)$,

$$\hat{\mathbf{n}}' \cdot \nabla' G(r) = \frac{\hat{\mathbf{n}}' \cdot \mathbf{r}}{r} \frac{d}{dr} G(r) = \underbrace{-\frac{i}{4} k^2 \overbrace{(\hat{\mathbf{n}}' \cdot \mathbf{r})}^{\text{regular}} \overbrace{\frac{J_1(kr)}{kr}}^{\text{regular}}}_{\text{regular}} + \underbrace{\frac{1}{4} \overbrace{\frac{\hat{\mathbf{n}}' \cdot \mathbf{r}}{r^2}}^{\text{regular}} \overbrace{kr Y_1(kr)}^{\text{singular}}}_{\text{singular}}. \tag{32}$$

The first term is regular; the second is singular. The second term is singular not because its value diverges at the origin (in fact, $\lim_{r\to 0}(\hat{\mathbf{n}}'\cdot\mathbf{r}/r^2)krY_1(kr)=1/\pi R$, where R is the radius of curvature of the surface at the field point), but because its higher derivatives do. The singularity is still a $\log(r)$ singularity, so local correction integrals can be computed in the same manner as for the previous kernel.

A.1.c. $\hat{\mathbf{n}}\cdot\nabla G(r)$,

$$\hat{\mathbf{n}}\cdot\nabla G(r) = -\frac{\hat{\mathbf{n}}\cdot\mathbf{r}}{r}\frac{d}{dr}G(r) = \underbrace{\frac{i}{4}k^2\overbrace{(\hat{\mathbf{n}}\cdot\mathbf{r})}^{\text{regular}}\overbrace{\frac{J_1(kr)}{kr}}^{\text{regular}}}_{\text{regular}} - \underbrace{\frac{1}{4}\overbrace{\frac{\hat{\mathbf{n}}\cdot\mathbf{r}}{r^2}}^{\text{regular}}\overbrace{krY_1(kr)}^{\text{singular}}}_{\text{singular}}. \tag{33}$$

This kernel is identical to that for $\hat{\mathbf{n}}'\cdot\nabla' G(r)$ with $\hat{\mathbf{n}}'$ replaced by $-\hat{\mathbf{n}}$ and it has similar properties.

A.1.d. $(\hat{\mathbf{n}}\cdot\nabla)(\hat{\mathbf{n}}'\cdot\nabla' G(r))$,

$$\begin{aligned}
&(\hat{\mathbf{n}}\cdot\nabla)(\hat{\mathbf{n}}'\cdot\nabla' G(r)) \\
&\quad= \frac{(\hat{\mathbf{n}}\cdot\mathbf{r})(\hat{\mathbf{n}}'\cdot\mathbf{r})}{r^2}\left(\frac{1}{r}\frac{dG(r)}{dr} - \frac{d^2G(r)}{dr^2}\right) - \frac{(\hat{\mathbf{n}}\cdot\hat{\mathbf{n}}')}{r}\frac{dG(r)}{dr} \qquad (34)\\
&\quad= \underbrace{\frac{ik^2}{4}\left(\overbrace{(\hat{\mathbf{n}}\cdot\hat{\mathbf{n}}')}^{\text{regular}}\overbrace{\frac{J_1(kr)}{kr}}^{\text{regular}} - \overbrace{\frac{(\hat{\mathbf{n}}\cdot\mathbf{r})(\hat{\mathbf{n}}'\cdot\mathbf{r})}{r^2}}^{\text{regular}}\overbrace{J_2(kr)}^{\text{regular}}\right)}_{\text{regular}} \\
&\qquad+ \underbrace{(\hat{\mathbf{n}}\cdot\nabla)(\hat{\mathbf{n}}'\cdot\nabla' G^R(r))}_{\text{hypersingular}}. \qquad (35)
\end{aligned}$$

Applying the derivatives to the real part of $G(r)$, namely $G^R(r)\equiv -\frac{1}{4}Y_0(kr)$, produces a term that is not merely singular but hypersingular. When convolved with a regular function, this term is not (in general) integrable because it diverges like $1/r^2$, relative to the field point. The following discussion shows how to manipulate it into a form that allows numerical evaluation when the region of integration contains the field point. When the region of integration does not include the field point, Gauss–Legendre rules may be used.

The convolution of $(\hat{\mathbf{n}}\cdot\nabla)(\hat{\mathbf{n}}'\cdot\nabla' G^R(r))$ with testing function $f(\mathbf{x}')$ is

$$\int_C dl'(\hat{\mathbf{n}}\cdot\nabla)(\hat{\mathbf{n}}'\cdot\nabla' G^R(r))\, f(\mathbf{x}'). \tag{36}$$

Strictly speaking this is not a proper integral unless it is assumed to represent the limiting value as the field point approaches the surface from off the surface. We implicitly make this assumption throughout. Using the vector identity (29) and the fact that $G^R(r)$ obeys the homogenous Helmholtz equation when $\mathbf{x}$ is not on S, we can convert the double normal derivative operator to a double tangential derivative operator:

$$\int_C dl'[k^2(\hat{\mathbf{n}}\cdot\hat{\mathbf{n}}')G^R(r) - (\hat{\mathbf{n}}\times\nabla)\cdot(\hat{\mathbf{n}}'\times\nabla' G^R(r))]\, f(\mathbf{x}'). \tag{37}$$

In 2D, we can rewrite the second term even more explicitly in terms of tangential derivatives, obtaining

$$\int_C dl'[k^2(\hat{\mathbf{n}}\cdot\hat{\mathbf{n}}')G^R(r) - (\hat{\mathbf{t}}\cdot\nabla)\,(\hat{\mathbf{t}}'\cdot\nabla' G^R(r))]\,f(\mathbf{x}'), \tag{38}$$

where $\hat{\mathbf{t}}$ and $\hat{\mathbf{t}}'$ are unit tangent vectors at the field and source points, respectively. The first term has a $\log(r)$ singularity, which we already know how to integrate numerically; the second term is hypersingular and requires further manipulation.

The gradient operators ∇ and ∇' commute with the unit tangent vectors $\hat{\mathbf{t}}'$ and $\hat{\mathbf{t}}$, respectively, so we can rearrange the factors of the second term and integrate it by parts as

$$\begin{aligned}
&-\int_C dl'(\hat{\mathbf{t}}\cdot\nabla)(\hat{\mathbf{t}}'\cdot\nabla' G^R(r))\,f(\mathbf{x}') \\
&\quad = -\int_C dl'\,f(\mathbf{x}')\,\hat{\mathbf{t}}'\cdot\nabla'(\hat{\mathbf{t}}\cdot\nabla G^R(r)) \qquad (39)\\
&\quad = -\int_C dl'\,\hat{\mathbf{t}}'\cdot\nabla'(f(\mathbf{x}')(\hat{\mathbf{t}}\cdot\nabla G^R(r))) \\
&\qquad + \int_C dl'(\hat{\mathbf{t}}'\cdot\nabla' f(\mathbf{x}'))(\hat{\mathbf{t}}\cdot\nabla G^R(r)). \qquad (40)
\end{aligned}$$

The first integral on the right-hand side of (40) is

$$\begin{aligned}
&-\int_C dl'\,\hat{\mathbf{t}}'\cdot\nabla'(f(\mathbf{x}')(\hat{\mathbf{t}}\cdot\nabla G^R(r))) \\
&\quad = -\int_C d\mathbf{l}'\cdot\nabla'(f(\mathbf{x}')(\hat{\mathbf{t}}\cdot\nabla G^R(r))) \qquad (41)\\
&\quad = -[f(\mathbf{x}')(\hat{\mathbf{t}}\cdot\nabla G^R(r))]_{C_1}^{C_2}; \qquad (42)
\end{aligned}$$

i.e., since the integrand is a total derivative, the value of the integral is a difference of values at the endpoints. Rearranging factors and using

$$\nabla G^R(r) = -\nabla' G^R(r), \tag{43}$$

we can rewrite the second integral as

$$-\int_C dl'\,\nabla' G^R(r)\cdot[\hat{\mathbf{t}}(\hat{\mathbf{t}}'\cdot\nabla' f(\mathbf{x}'))]. \tag{44}$$

In this form, the integral is not yet evaluable because $\nabla' G^R(r)$ diverges like $1/r$ relative to the field point. We can make it integrable by adding and subtracting a smooth function that matches the integrand at the field point. Specifically, let us write (44) as

$$-\int_C dl'\,\nabla' G^R(r)\cdot[\hat{\mathbf{t}}(\hat{\mathbf{t}}'\cdot\nabla' f(\mathbf{x}')) - \hat{\mathbf{t}}'(\hat{\mathbf{t}}\cdot\nabla' f(\mathbf{x}))] - \int_C dl'\,\nabla' G^R(r)\cdot[\hat{\mathbf{t}}'(\hat{\mathbf{t}}\cdot\nabla' f(\mathbf{x}))], \tag{45}$$

where $\hat{\mathbf{t}}' \cdot \nabla' f(\mathbf{x}')$ and $\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x})$ represent tangential derivatives of the testing function $f(\mathbf{x}')$ evaluated at the field and source points, respectively. The first integral in this expression is integrable because the zero of

$$[\hat{\mathbf{t}}(\hat{\mathbf{t}}' \cdot \nabla' f(\mathbf{x}')) - \hat{\mathbf{t}}'(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))] \tag{46}$$

at the field point cancels the pole from $\nabla' G^R(r)$ at the field point, leaving a singularity no worse than $\log(r)$ relative to the field point. By rearranging factors, the integrand of the second integral can be shown to be a total derivative, so that

$$\begin{aligned} &-\int_C dl' \, \nabla' G^R(r) \cdot [\hat{\mathbf{t}}'(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))] \\ &\quad = -\int_C dl' \, \hat{\mathbf{t}}' \cdot (\nabla' G^R(r)(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))) \qquad (47) \\ &\quad = -[G^R(r)(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))]_{C_1}^{C_2}. \qquad (48) \end{aligned}$$

Putting the various terms together, we arrive at the following numerically tractable expression for the integral needed to compute local corrections for the hypersingular component of the kernel

$$\begin{aligned} &\int_C dl' \{k^2(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}') G^R(r) f(\mathbf{x}') - \nabla' G^R(r) \cdot [\hat{\mathbf{t}}(\hat{\mathbf{t}}' \cdot \nabla' f(\mathbf{x}')) - \hat{\mathbf{t}}'(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))]\} \\ &\quad - [f(\mathbf{x}')(\hat{\mathbf{t}} \cdot \nabla G^R(r)) + G^R(r)(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))]_{C_1}^{C_2}, \end{aligned} \tag{49}$$

or, substituting for $G^R(r)$,

$$\begin{aligned} &-\frac{k^2}{4} \int_C dl' \left\{ (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}') Y_0(kr) f(\mathbf{x}') + \frac{Y_1(kr)}{kr} \hat{\mathbf{r}} \cdot \left[\hat{\mathbf{t}} \frac{df}{dl'}(\mathbf{x}') - \hat{\mathbf{t}}' \frac{df}{dl'}(\mathbf{x}) \right] \right\} \\ &\quad - \frac{1}{4} \left[k^2 \frac{Y_1(kr)}{kr} (\hat{\mathbf{t}} \cdot \hat{\mathbf{r}}) f(\mathbf{x}') - Y_0(kr) \frac{df}{dl'}(\mathbf{x}) \right]_{C_1}^{C_2}. \end{aligned} \tag{50}$$

A.2. Three-dimensional scalar.

A.2.a. $G(r)$,

$$G(r) = \frac{e^{ikr}}{r} = \underbrace{i \frac{\sin(kr)}{r}}_{\text{regular}} + \underbrace{\frac{\cos(kr)}{r}}_{\text{singular}}. \tag{51}$$

As in the 2D scalar case, this kernel may be written as the sum of a regular part and a singular part. It is necessary to compute local corrections only for the singular part because the regular part will be efficiently integrated by the underlying high-order quadrature rule. The singular term contains a $1/r$ singularity. Computing local corrections for the singular part requires evaluation of integrals of $\cos(kr)/r$ times polynomials in the parameters $\mathbf{u} = (u^1, u^2)$ used to describe the surface. When the region of integration contains the field point, it may be subdivided into triangles with the field point at one vertex, and the integration may be performed by using the Duffy transformation [25] and Gauss–Legendre product

rules on the subtriangles. Otherwise, one can apply efficient quadrature rules for smooth functions such as high-order triangle rules [16].

A.2.b. $\hat{\mathbf{n}}' \cdot \nabla' G(r)$,

$$\hat{\mathbf{n}}' \cdot \nabla' G(r) = \frac{\hat{\mathbf{n}}' \cdot \mathbf{r}}{r} \frac{d}{dr} G(r) = \frac{(ikr - 1)\, e^{ikr}}{r^2} \frac{\hat{\mathbf{n}}' \cdot \mathbf{r}}{r} \tag{52}$$

$$= \underbrace{ik^3 \frac{\overbrace{\left(\cos(kr) - \frac{\sin(kr)}{kr}\right)}^{\text{regular}}}{(kr)^2} \overbrace{(\hat{\mathbf{n}}' \cdot \mathbf{r})}^{\text{regular}}}_{\text{regular}} - \underbrace{\overbrace{(\cos(kr) + (kr)\sin(kr))}^{\text{regular}} \overbrace{\frac{(\hat{\mathbf{n}}' \cdot \mathbf{r})}{r^2} \frac{1}{r}}^{\text{singular}}}_{\text{singular}}. \tag{53}$$

In 2D, $(\hat{\mathbf{n}}' \cdot \mathbf{r})/r^2$ is a regular function with a removable singularity at the origin. In 3D, the singularity is removable only if the principal radii of curvature of the surface at the field point are the same. Otherwise its limiting value depends on the direction from which the origin is approached. Nonetheless, local correction integrals can be computed efficiently by means of triangle subdivision and the Duffy transformation.

A.2.c. $\hat{\mathbf{n}} \cdot \nabla G(r)$,

$$\hat{\mathbf{n}} \cdot \nabla G(r) = \underbrace{-ik^3 \frac{\overbrace{\left(\cos(kr) - \frac{\sin(kr)}{kr}\right)}^{\text{regular}}}{(kr)^2} \overbrace{(\hat{\mathbf{n}} \cdot \mathbf{r})}^{\text{regular}}}_{\text{regular}} + \underbrace{\overbrace{(\cos(kr) + (kr)\sin(kr))}^{\text{regular}} \overbrace{\frac{(\hat{\mathbf{n}} \cdot \mathbf{r})}{r^2} \frac{1}{r}}^{\text{singular}}}_{\text{singular}}. \tag{54}$$

This kernel is identical to that for $\hat{\mathbf{n}}' \cdot \nabla' G(r)$ with $\hat{\mathbf{n}}'$ replaced by $-\hat{\mathbf{n}}$ and has similar properties.

A.2.d. $(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G(r))$,

$$(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G(r))$$

$$= (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}') \left(\frac{1 - ikr}{r^3}\right) e^{ikr} + (\hat{\mathbf{n}} \cdot \mathbf{r})(\hat{\mathbf{n}}' \cdot \mathbf{r}) \left(\frac{k^2 r^2 + 3ikr - 3}{r^5}\right) e^{ikr} \tag{55}$$

$$= \underbrace{ik^3 \left(\frac{\overbrace{\left(\frac{\sin(kr)}{kr} - \cos(kr)\right)}^{\text{regular}}}{(kr)^2} \overbrace{(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}')}^{\text{regular}} + k^2 \frac{\overbrace{\frac{\sin(kr)}{kr} - 3\left(\frac{\sin(kr)}{kr} - \cos(kr)\right)}^{\text{regular}}}{(kr)^2} \overbrace{(\hat{\mathbf{n}} \cdot \mathbf{r})(\hat{\mathbf{n}}' \cdot \mathbf{r})}^{\text{regular}} \right)}_{\text{regular}}$$

$$+ \underbrace{(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G^R(r))}_{\text{hypersingular}}. \tag{56}$$

Applying the derivatives to the real part of $G(r)$, namely $G^R(r) \equiv \cos(kr)/r$, produces a term that is not merely singular but hypersingular. When convolved with a regular function, this term is not (in general) integrable because it diverges like $1/r^3$ relative to the field point. The following discussion shows how to manipulate it into a form that allows numerical evaluation when the region of integration contains the field point. When the region of

integration does not include the field point, standard, high-order rules for integrating regular, two-parameter functions may be used.

The convolution of $(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G^R(r))$ with testing function $f(\mathbf{x}')$ is

$$\int_S ds'(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G^R(\mathbf{x}, \mathbf{x}'))\, f(\mathbf{x}') \tag{57}$$

or

$$\int_S ds'[k^2(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}')G^R(\mathbf{x}, \mathbf{x}') - (\hat{\mathbf{n}} \times \nabla) \cdot (\hat{\mathbf{n}}' \times \nabla' G^R(\mathbf{x}, \mathbf{x}'))]\, f(\mathbf{x}'), \tag{58}$$

where the second form follows from Eq. (29). As in the 2D case, we implicitly assume a limiting procedure whereby the field point approaches its final destination on the surface from off the surface. The first term in brackets is only singular like $1/r$; we already know how to deal with such expressions. It is the second term that requires further attention. Write this term in component form using the Levi–Civita tensor ϵ_{ijk} and manipulate the expression as shown using the fact that $\mathbf{x}$ and $\mathbf{x}'$ are independent. Summation over repeated indices is implied.

$$\begin{aligned}
&-\int_S ds'((\hat{\mathbf{n}} \times \nabla) \cdot (\hat{\mathbf{n}}' \times \nabla' G^R(\mathbf{x}, \mathbf{x}')))\, f(\mathbf{x}') \\
&\quad = -(\hat{\mathbf{n}} \times \nabla) \cdot \int_S ds'(\hat{\mathbf{n}}' \times \nabla' G^R(\mathbf{x}, \mathbf{x}'))\, f(\mathbf{x}') && (59)\\
&\quad = -\epsilon_{ijk} n_j \partial_k \left[\int_S ds'(\hat{\mathbf{n}}' \times \nabla' G^R(\mathbf{x}, \mathbf{x}'))\, f(\mathbf{x}')\right]_i && (60)\\
&\quad = -\epsilon_{ijk} n_j \left[\int_S ds'\left(\hat{\mathbf{n}}' \times \nabla'\left(\partial_k G^R(\mathbf{x}, \mathbf{x}')\right)\right) f(\mathbf{x}')\right]_i && (61)\\
&\quad = -\epsilon_{ijk} n_j \left[\int_S ds'\, \hat{\mathbf{n}}' \times \nabla'\left(f(\mathbf{x}')\partial_k G^R(\mathbf{x}, \mathbf{x}')\right)\right]_i \\
&\qquad + \epsilon_{ijk} n_j \left[\int_S ds'\, \partial_k G^R(\mathbf{x}, \mathbf{x}')(\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}'))\right]_i && (62)
\end{aligned}$$

The last step shows the result of integrating by parts. Letting

$$\psi = f(\mathbf{x}')\partial_k G^R(\mathbf{x}, \mathbf{x}'), \tag{63}$$

we apply an adjunct to Stokes's theorem,

$$\int_S ds(\hat{\mathbf{n}} \times \nabla\psi) = \oint_{\partial S} d\mathbf{l}\psi \tag{64}$$

to the part of the first term inside the brackets, to get

$$\begin{aligned}
&-\epsilon_{ijk} n_j \left[\int_S ds'\, \hat{\mathbf{n}}' \times \nabla'\left(f(\mathbf{x}')\partial_k G^R(\mathbf{x}, \mathbf{x}')\right)\right]_i \\
&\quad = -\epsilon_{ijk} n_j \left[\oint_{\partial S} d\mathbf{l}' f(\mathbf{x}')\partial_k G^R(\mathbf{x}, \mathbf{x}')\right]_i && (65)
\end{aligned}$$

$$= -\epsilon_{ijk} n_j \oint_{\partial S} dl'_i \, f(\mathbf{x}') \partial_k G^R(\mathbf{x}, \mathbf{x}') \tag{66}$$

$$= -\oint_{\partial S} d\mathbf{l}' \cdot (\hat{\mathbf{n}} \times \nabla G^R(\mathbf{x}, \mathbf{x}')) \, f(\mathbf{x}'), \tag{67}$$

which is integrable. To evaluate the rest, use the fact that

$$\nabla G^R(\mathbf{x}, \mathbf{x}') = -\nabla' G^R(\mathbf{x}, \mathbf{x}') \tag{68}$$

to write

$$\epsilon_{ijk} n_j \left[\int_S ds' \, \partial_k G^R(\mathbf{x}, \mathbf{x}') (\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) \right]_i$$

$$= -\int_S ds' \, \partial'_k G^R(\mathbf{x}, \mathbf{x}') \epsilon_{kij} (\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}'))_i n_j \tag{69}$$

$$= -\int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot [(\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) \times \hat{\mathbf{n}}]. \tag{70}$$

At the field point, the vector in brackets becomes

$$(\hat{\mathbf{n}} \times \nabla' f(\mathbf{x}')) \times \hat{\mathbf{n}} = -\hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \nabla' f(\mathbf{x}')) = \nabla'_{\parallel} f(\mathbf{x}'). \tag{71}$$

Some notation from differential geometry is useful at this point: $\partial_\mu \mathbf{x} \equiv \partial \mathbf{x} / \partial u^\mu$ is the derivative of the surface with respect to surface parameter u^μ; $g_{\mu\nu}$ is the metric tensor given by $\partial_\mu \mathbf{x} \cdot \partial_\nu \mathbf{x}$; $g^{\mu\nu}$ is the inverse of $g_{\mu\nu}$; g is the determinant of $g_{\mu\nu}$; and $\partial'_\mu f$ represents the derivative of f with respect to u^μ, i.e., $\partial'_\mu f \equiv \partial f(\mathbf{x}'(\mathbf{u})) / \partial u^\mu$.

Thus, in the language of differential geometry, the vector in brackets becomes

$$\partial'^\mu f \partial'_\mu \mathbf{x}' = g^{\mu\nu} \partial'_\nu f \partial'_\mu \mathbf{x}' = \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \tag{72}$$

when α^μ is defined as

$$\sqrt{g(\mathbf{u})} g^{\mu\nu} \partial'_\nu f \tag{73}$$

evaluated at the field point. Therefore, we may write

$$-\int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot [(\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) \times \hat{\mathbf{n}}]$$

$$= \int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\hat{\mathbf{n}} \times (\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) + \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right]$$

$$- \int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right]. \tag{74}$$

The first term is integrable because the zero of

$$\left[\hat{\mathbf{n}} \times (\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) + \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right] \tag{75}$$

at the field point cancels one of the two poles from $\nabla' G^R(\mathbf{x}, \mathbf{x}')$ at the field point. The other term may be rewritten as

$$
\begin{aligned}
&-\int_S ds'\, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] \\
&\quad = -\int_S ds'\, \nabla'_\parallel G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] \qquad (76) \\
&\quad = -\int_S ds'\, \nabla'_\parallel \cdot \left(G^R(\mathbf{x}, \mathbf{x}') \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right) \\
&\qquad + \alpha^\mu \int_S ds'\, G^R(\mathbf{x}, \mathbf{x}') \nabla'_\parallel \cdot \left[\frac{\partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right], \qquad (77)
\end{aligned}
$$

where the last step shows the result of integrating by parts. The part of the first term in parentheses has no normal component so it can be converted to a boundary integral using the divergence theorem for open surfaces (see Appendix B):

$$
\begin{aligned}
&-\int_S ds'\, \nabla'_\parallel \cdot \left(G^R(\mathbf{x}, \mathbf{x}') \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right) \\
&\quad = -\oint_{\partial S} (d\mathbf{l}' \times \hat{\mathbf{n}}') \cdot \left(G^R(\mathbf{x}, \mathbf{x}') \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right) \qquad (78) \\
&\quad = -\oint_{\partial S} d\mathbf{l}' \cdot \left[\hat{\mathbf{n}}' \times \left(\alpha^\mu \partial'_\mu \mathbf{x}'\right)\right] \frac{G^R(\mathbf{x}, \mathbf{x}')}{\sqrt{g(\mathbf{u})}}. \qquad (79)
\end{aligned}
$$

The second term is zero since (see Appendix C)

$$
\nabla'_\parallel \cdot \left[\frac{\partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] = 0. \qquad (80)
$$

Putting the various terms together, we arrive at the numerically tractable expression for the integral needed to compute local corrections for the hypersingular component of the kernel,

$$
\begin{aligned}
&\int_S ds' \left(k^2 (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}') G^R(\mathbf{x}, \mathbf{x}')\, f(\mathbf{x}') + \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\hat{\mathbf{n}} \times (\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) + \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] \right) \\
&\quad - \oint_{\partial S} d\mathbf{l}' \cdot \left((\hat{\mathbf{n}} \times \nabla G^R(\mathbf{x}, \mathbf{x}'))\, f(\mathbf{x}') + \left(\hat{\mathbf{n}}' \times \left(\alpha^\mu \partial'_\mu \mathbf{x}'\right)\right) \frac{G^R(\mathbf{x}, \mathbf{x}')}{\sqrt{g(\mathbf{u})}} \right), \qquad (81)
\end{aligned}
$$

where

$$
\alpha^\mu \equiv \sqrt{g(\mathbf{u})} g^{\mu\nu} \partial'_\nu f(\mathbf{x}'(\mathbf{u})), \qquad (82)
$$

evaluated at the field point. The first integral is a surface integral whose integrand diverges no worse than $1/r$ near the field point; the second is a boundary integral of a regular function (so long as the field point is never situated on the boundary).

A.3. Three-dimensional vector.

A.3.a. $G(r)(\mathbf{t}(\mathbf{x}) \cdot \mathbf{t}'(\mathbf{x}'))$.

This kernel is identical to $G(r)$ in the 3D scalar case, except that the regular function with which it must be convolved is the inner product of a tangent vector $\mathbf{t}(\mathbf{x})$ at the field point and a tangent vector $\mathbf{t}'(\mathbf{x}')$ at the source point. Four sets of local corrections must be computed for each field point since there are two independent tangent vectors at each field point and two at each source point.

A.3.b. $\mathbf{t}(\mathbf{x}) \cdot (\nabla' G(r) \times \mathbf{t}'(\mathbf{x}'))$,

$$\begin{aligned}
&\mathbf{t} \cdot (\nabla' G(r) \times \mathbf{t}'(\mathbf{x}')) \\
&\quad = (ikr - 1)\, e^{ikr} \frac{(\mathbf{t}'(\mathbf{x}') \times \mathbf{t}(\mathbf{x})) \cdot \mathbf{r}}{r^2} \frac{1}{r} \qquad (83) \\
&\quad = \underbrace{ik^3 \frac{\overbrace{\left(\frac{\sin(kr)}{kr} - \cos(kr)\right)}^{\text{regular}}}{(kr)^2} \overbrace{((\mathbf{t}(\mathbf{x}) \times \mathbf{t}'(\mathbf{x}')) \cdot \mathbf{r})}^{\text{regular}}}_{\text{regular}} \\
&\qquad + \underbrace{\overbrace{(\cos(kr) + (kr)\sin(kr))}^{\text{regular}} \overbrace{\frac{((\mathbf{t}(\mathbf{x}) \times \mathbf{t}'(\mathbf{x}')) \cdot \mathbf{r})}{r^2} \frac{1}{r}}^{\text{singular}}}_{\text{singular}}. \qquad (84)
\end{aligned}$$

The analysis of the singular component is as follows. We can write $\mathbf{t}(\mathbf{x})$ in terms of surface derivatives at the field point

$$\mathbf{t}(\mathbf{x}) = \zeta^\mu \partial_\mu \mathbf{x} \qquad (85)$$

with some pair of coefficients ζ^μ, $\mu = 1, 2$. Letting $\mathbf{u}'$ denote the parameterization of the source point relative to the field point, we can write the expansions for $\mathbf{t}'(\mathbf{x}')$ and $\mathbf{r}(\mathbf{x}')$ about the field point,

$$\mathbf{t}'(\mathbf{x}') = \xi^\rho \partial'_\rho \mathbf{x}' = \xi^\rho \left(\partial_\rho \mathbf{x} + \partial_\rho \partial_\sigma \mathbf{x}\, u'^\sigma + \cdots\right), \qquad (86)$$

for some other pair of coefficients ξ^ρ with $\rho = 1, 2$ and

$$\mathbf{r}(\mathbf{x}') = \partial_\tau \mathbf{x}\, u'^\tau + \cdots. \qquad (87)$$

Then

$$\begin{aligned}
((\mathbf{t}(\mathbf{x}) \times \mathbf{t}'(\mathbf{x}')) \cdot \mathbf{r}) &= \zeta^\mu \xi^\rho \left(\partial_\mu \mathbf{x} \times \partial_\rho \mathbf{x} + \partial_\mu \mathbf{x} \times \partial_\rho \partial_\sigma \mathbf{x}\, u'^\sigma + \cdots\right) \cdot \left(\partial_\tau \mathbf{x}\, u'^\tau + \cdots\right) \qquad (88) \\
&= \zeta^\mu \xi^\rho ((\partial_\mu \mathbf{x} \times \partial_\rho \partial_\sigma \mathbf{x}) \cdot \partial_\tau \mathbf{x}) u'^\sigma u'^\tau + \cdots. \qquad (89)
\end{aligned}$$

Since the leading term in $1/r^2$ is also second order in $\mathbf{u}'$, the ratio $((\mathbf{t}(\mathbf{x}) \times \mathbf{t}'(\mathbf{x}')) \cdot \mathbf{r})/r^2$ does not diverge in the limit as $r \to 0$. However, like the factors $(\hat{\mathbf{n}}' \cdot \mathbf{r})/r^2$ and $(\hat{\mathbf{n}} \cdot \mathbf{r})/r^2$ from the 3D scalar case, this ratio is not a regular function unless the principal radii of curvature at the field point are identical. Computation of local correction integrals for each combination of tangent vectors at the field and source points proceeds as in the corresponding 3D scalar case.

A.3.c. $(\mathbf{t}(\mathbf{x})\cdot\nabla)(\nabla' G(r)\cdot\mathbf{t}'(\mathbf{x}'))$,

$$
\begin{aligned}
&(\mathbf{t}(\mathbf{x})\cdot\nabla)(\nabla' G(r)\cdot\mathbf{t}'(\mathbf{x}')) \\
&\quad = (\mathbf{t}\cdot\mathbf{t}')\left(\frac{1-ikr}{r^3}\right)e^{ikr} + (\mathbf{t}\cdot\mathbf{r})(\mathbf{t}'\cdot\mathbf{r})\left(\frac{k^2r^2+3ikr-3}{r^5}\right)e^{ikr} \qquad (90)\\
&\quad = \underbrace{ik^3\left(\overbrace{\frac{\left(\frac{\sin(kr)}{kr}-\cos(kr)\right)}{(kr)^2}}^{\text{regular}}\overbrace{(\mathbf{t}\cdot\mathbf{t}')}^{\text{regular}} + k^2\overbrace{\frac{\frac{\sin(kr)}{kr}-3\left(\frac{\frac{\sin(kr)}{kr}-\cos(kr)}{(kr)^2}\right)}{(kr)^2}}^{\text{regular}}\overbrace{(\mathbf{t}\cdot\mathbf{r})(\mathbf{t}'\cdot\mathbf{r})}^{\text{regular}}\right)}_{\text{regular}} \\
&\qquad + \underbrace{(\mathbf{t}\cdot\nabla)(\nabla' G^R(r)\cdot\mathbf{t}')}_{\text{hypersingular}}. \qquad (91)
\end{aligned}
$$

The result is very similar to that in the 3D scalar case. The real part of $G(r)$, namely $G^R(r)\equiv\cos(kr)/r$, produces a hypersingular term that is not (in general) integrable because it diverges like $1/r^3$ relative to the field point. We now show how to manipulate it into a form that can be evaluated numerically when the region of integration contains the field point.

Reformulating the integral of the hypersingular term begins with an integration by parts:

$$
\begin{aligned}
&\int_S ds'(\mathbf{t}(\mathbf{x})\cdot\nabla)(\nabla' G^R(\mathbf{x},\mathbf{x}')\cdot\mathbf{t}'(\mathbf{x}')) \\
&\quad = \int_S ds'\,\mathbf{t}'(\mathbf{x}')\cdot\nabla'_{\|}(\mathbf{t}(\mathbf{x})\cdot\nabla G^R(\mathbf{x},\mathbf{x}')) \qquad (92)\\
&\quad = \int_S ds'\,\nabla'_{\|}\cdot[\mathbf{t}'(\mathbf{x}')(\mathbf{t}(\mathbf{x})\cdot\nabla G^R(\mathbf{x},\mathbf{x}'))] \\
&\qquad - \int_S ds'(\mathbf{t}(\mathbf{x})\cdot\nabla G^R(\mathbf{x},\mathbf{x}'))(\nabla'_{\|}\cdot\mathbf{t}'(\mathbf{x}')). \qquad (93)
\end{aligned}
$$

The first term on the last line can be converted to a boundary integral using the divergence theorem for open surfaces (see Appendix B) and the fact that the argument of $\nabla'_{\|}\cdot$ is tangential to the surface:

$$
\int_S ds'\,\nabla'_{\|}\cdot[\mathbf{t}'(\mathbf{x}')(\mathbf{t}(\mathbf{x})\cdot\nabla G^R(\mathbf{x},\mathbf{x}'))] = \oint_{\partial S} dl(\hat{\mathbf{e}}'\cdot\mathbf{t}'(\mathbf{x}'))(\mathbf{t}(\mathbf{x})\cdot\nabla G^R(\mathbf{x},\mathbf{x}')). \qquad (94)
$$

The second term is

$$
-\int_S ds'(\mathbf{t}(\mathbf{x})\cdot\nabla G^R(\mathbf{x},\mathbf{x}'))(\nabla'_{\|}\cdot\mathbf{t}'(\mathbf{x}')) = \int_S ds'\,\nabla' G^R(\mathbf{x},\mathbf{x}')\cdot[\mathbf{t}(\mathbf{x})(\nabla'_{\|}\cdot\mathbf{t}'(\mathbf{x}'))]. \qquad (95)
$$

Write this as

$$
\int_S ds'\,\nabla' G^R(\mathbf{x},\mathbf{x}')\cdot\left[\mathbf{t}(\mathbf{x})(\nabla'_{\|}\cdot\mathbf{t}'(\mathbf{x}')) - \frac{\alpha^\mu\partial'_\mu\mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] + \int_S ds'\,\nabla' G^R(\mathbf{x},\mathbf{x}')\cdot\left[\frac{\alpha^\mu\partial'_\mu\mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right], \qquad (96)
$$

where the constant α^μ is chosen to make $\mathbf{t}(\mathbf{x})(\nabla'_\parallel \cdot \mathbf{t}'(\mathbf{x}'))$ and $\alpha^\mu \partial'_\mu \mathbf{x}'/\sqrt{g(\mathbf{u})}$ equal at the field point. In other words, α^μ is defined as

$$\sqrt{g(\mathbf{u})} g^{\mu\nu} (\mathbf{t}(\mathbf{x}) \cdot \partial'_\nu \mathbf{x}')(\nabla'_\parallel \cdot \mathbf{t}'(\mathbf{x}')) \tag{97}$$

evaluated at the field point. The first term is integrable because the zero of

$$\left[\mathbf{t}(\mathbf{x})(\nabla'_\parallel \cdot \mathbf{t}'(\mathbf{x}')) - \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right] \tag{98}$$

at the field point cancels one of the two poles from $\nabla' G^R(\mathbf{x}, \mathbf{x}')$ at the field point. As shown in the 3D scalar case, the second term reduces to the boundary integral:

$$\begin{aligned} &\int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right] \\ &\quad = \oint_{\partial S} d\mathbf{l}' \cdot \left[\hat{\mathbf{n}}' \times \left(\alpha^\mu \partial'_\mu \mathbf{x}' \right) \right] \frac{G^R(\mathbf{x}, \mathbf{x}')}{\sqrt{g(\mathbf{u})}} \qquad (99) \\ &\quad = \oint_{\partial S} dl' \, \hat{\mathbf{e}}' \cdot \left(G^R(\mathbf{x}, \mathbf{x}') \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right). \qquad (100) \end{aligned}$$

Putting the various terms together, we arrive at the numerically tractable expression for the integral needed to compute local corrections for the hypersingular component of the kernel,

$$\begin{aligned} &\int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\mathbf{t}(\mathbf{x})(\nabla'_\parallel \cdot \mathbf{t}'(\mathbf{x}')) - \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right] \\ &\quad + \oint_{\partial S} dl' \, \hat{\mathbf{e}}' \cdot \left((\mathbf{t}(\mathbf{x}) \cdot \nabla G^R(\mathbf{x}, \mathbf{x}')) \mathbf{t}'(\mathbf{x}') + G^R(\mathbf{x}, \mathbf{x}') \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right), \end{aligned} \tag{101}$$

where

$$\alpha^\mu \equiv \sqrt{g(\mathbf{u})} g^{\mu\nu} (\mathbf{t}(\mathbf{x}) \cdot \partial'_\nu \mathbf{x}')(\nabla'_\parallel \cdot \mathbf{t}'(\mathbf{x}')) = \sqrt{g(\mathbf{u})} g^{\mu\nu} (\mathbf{t}(\mathbf{x}) \cdot \partial'_\nu \mathbf{x}') \left(g^{\rho\sigma} \partial'_\rho \mathbf{t}' \cdot \partial'_\sigma \mathbf{x}' \right), \tag{102}$$

evaluated at the field point. The first integral is a surface integral whose integrand diverges no worse than $1/r$; the second is a boundary integral of a regular function (so long as the field point is never situated on the boundary).

If, as suggested in Section III.C.3, the μth tangent vector at the field point (with surface parameter $\mathbf{u}_0$) is given by

$$\mathbf{t}_\mu(\mathbf{u}) = \partial_\mu \mathbf{x}(\mathbf{u}) \tag{103}$$

and the νth vector testing function associated with scalar testing function $f^{(k)}(\mathbf{u})$ is given by

$$\mathbf{t}_\nu^{(k)}(\mathbf{u}) = \frac{\partial_\nu \mathbf{x}(\mathbf{u})}{\sqrt{g(\mathbf{u})}} f^{(k)}(\mathbf{u}), \tag{104}$$

then Eq. (101) simplifies to

$$\int_S ds'\, \nabla' G^R(\mathbf{x},\mathbf{x}') \cdot \left(\partial_\mu \mathbf{x} \partial'_\nu f^{(k)}(\mathbf{u}) - \partial'_\mu \mathbf{x}' \partial'_\nu f^{(k)}(\mathbf{u}_0)\right) \Big/ \sqrt{g(\mathbf{u})} + \oint_{\partial S} dl'\, \hat{\mathbf{e}}' \cdot \left(G^R(\mathbf{x},\mathbf{x}') \partial'_\nu f^{(k)}(\mathbf{u}_0) \partial'_\mu \mathbf{x}' + \left(\partial_\mu \mathbf{x} \cdot \nabla G^R(\mathbf{x},\mathbf{x}')\right) f^{(k)}(\mathbf{u}) \partial'_\nu \mathbf{x}'\right) \Big/ \sqrt{g(\mathbf{u})}. \tag{105}$$

B. *Divergence Theorem for Open Surfaces*

Substitute

$$\mathbf{B} = \hat{\mathbf{n}} \times \mathbf{A} \tag{106}$$

into Stokes's theorem

$$\int_S ds\, \hat{\mathbf{n}} \cdot (\nabla \times \mathbf{B}) = \oint_{\partial S} d\mathbf{l} \cdot \mathbf{B} \tag{107}$$

to get

$$\begin{aligned}
&\int_S ds\, \hat{\mathbf{n}} \cdot (\nabla \times (\hat{\mathbf{n}} \times \mathbf{A})) \\
&\quad = \int_S ds\, \hat{\mathbf{n}} \cdot [\hat{\mathbf{n}}(\nabla \cdot \mathbf{A}) - (\hat{\mathbf{n}} \cdot \nabla)\mathbf{A} - \mathbf{A}(\nabla \cdot \hat{\mathbf{n}}) + (\mathbf{A} \cdot \nabla)\hat{\mathbf{n}}] && (108)\\
&\quad = \int_S ds[(\nabla_\parallel \cdot \mathbf{A}) - (\hat{\mathbf{n}} \cdot \mathbf{A})(\nabla \cdot \hat{\mathbf{n}})] && (109)\\
&\quad = \oint_{\partial S} d\mathbf{l} \cdot (\hat{\mathbf{n}} \times \mathbf{A}) && (110)\\
&\quad = \oint_{\partial S} (d\mathbf{l} \times \hat{\mathbf{n}}) \cdot \mathbf{A} && (111)\\
&\quad = \oint_{\partial S} dl\, \hat{\mathbf{e}} \cdot \mathbf{A}, && (112)
\end{aligned}$$

where we have used the definition of tangential gradient

$$\nabla_\parallel \equiv \nabla - \hat{\mathbf{n}}(\hat{\mathbf{n}} \cdot \nabla) \tag{113}$$

and the following equation which relates the vector line element $d\mathbf{l}$ and the surface normal $\hat{\mathbf{n}}$ to the scalar line element dl and the unit edge vector $\hat{\mathbf{e}}$,

$$d\mathbf{l} \times \hat{\mathbf{n}} = dl\, \hat{\mathbf{e}}, \tag{114}$$

and the observation that

$$\hat{\mathbf{n}} \cdot [(\mathbf{A} \cdot \nabla)\hat{\mathbf{n}}] = [(\mathbf{A} \cdot \nabla)\hat{\mathbf{n}}] \cdot \hat{\mathbf{n}} = \frac{1}{2}(\mathbf{A} \cdot \nabla)(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}) = 0. \tag{115}$$

In other words, the divergence theorem for open surfaces is

$$\int_S ds[(\nabla_\parallel \cdot \mathbf{A}) - (\hat{\mathbf{n}} \cdot \mathbf{A})(\nabla \cdot \hat{\mathbf{n}})] = \oint_{\partial S} dl\, \hat{\mathbf{e}} \cdot \mathbf{A} = \oint_{\partial S} (d\mathbf{l} \times \hat{\mathbf{n}}) \cdot \mathbf{A}, \tag{116}$$

which simplifies to

$$\int_S ds(\nabla_\parallel \cdot \mathbf{A}) = \oint_{\partial S} dl\, \hat{\mathbf{e}} \cdot \mathbf{A} = \oint_{\partial S} (d\mathbf{l} \times \hat{\mathbf{n}}) \cdot \mathbf{A} \tag{117}$$

when $\mathbf{A}$ is everywhere tangential to S.

C. *Proof that* $\nabla'_\parallel \cdot [\partial'_\mu \mathbf{x}'/\sqrt{g(\mathbf{u})}] = 0$

Note. Summation over repeated indices is implied:

$$\begin{aligned}
\nabla'_\parallel \cdot \left[\frac{\partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] &= \partial'^\rho \left(\frac{\partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right) \cdot \partial'_\rho \mathbf{x}' \\
&= g^{\rho\sigma}\partial'_\sigma \left(\frac{\partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right) \cdot \partial'_\rho \mathbf{x}' \\
&= g^{\rho\sigma}\left(\frac{\partial'_\sigma \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} - \frac{\partial'_\mu \mathbf{x}'}{2\sqrt{g(\mathbf{u})^3}}\partial'_\sigma g(\mathbf{u})\right) \cdot \partial'_\rho \mathbf{x}' \\
&= \frac{g^{\rho\sigma}}{\sqrt{g(\mathbf{u})}}\left(\partial'_\sigma \partial'_\mu \mathbf{x}' \cdot \partial'_\rho \mathbf{x}' - \frac{\partial'_\mu \mathbf{x}' \cdot \partial'_\rho \mathbf{x}'}{2g(\mathbf{u})}\left(2g(\mathbf{u})g^{\alpha\beta}\partial'_\alpha \mathbf{x}' \cdot \partial'_\sigma \partial'_\beta \mathbf{x}'\right)\right) \\
&= \frac{g^{\rho\sigma}}{\sqrt{g(\mathbf{u})}}\left(\partial'_\rho \mathbf{x}' \cdot \partial'_\sigma \partial'_\mu \mathbf{x}' - g_{\mu\rho}g^{\alpha\beta}\partial'_\alpha \mathbf{x}' \cdot \partial'_\sigma \partial'_\beta \mathbf{x}'\right) \\
&= \frac{1}{\sqrt{g(\mathbf{u})}}\left(g^{\rho\sigma}\partial'_\rho \mathbf{x}' \cdot \partial'_\sigma \partial'_\mu \mathbf{x}' - \delta^\sigma_\mu g^{\alpha\beta}\partial'_\alpha \mathbf{x}' \cdot \partial'_\sigma \partial'_\beta \mathbf{x}'\right) \\
&= \frac{1}{\sqrt{g(\mathbf{u})}}\left(g^{\rho\sigma}\partial'_\rho \mathbf{x}' \cdot \partial'_\sigma \partial'_\mu \mathbf{x}' - g^{\alpha\beta}\partial'_\alpha \mathbf{x}' \cdot \partial'_\mu \partial'_\beta \mathbf{x}'\right) \\
&= \frac{1}{\sqrt{g(\mathbf{u})}}\left(g^{\rho\sigma}\partial'_\rho \mathbf{x}' \cdot \partial'_\sigma \partial'_\mu \mathbf{x}' - g^{\rho\sigma}\partial'_\rho \mathbf{x}' \cdot \partial'_\sigma \partial'_\mu \mathbf{x}'\right) = 0.
\end{aligned}$$

ACKNOWLEDGMENTS

We are grateful to Drs. Vladimir Rokhlin and Leslie Greengard for considerable guidance regarding the use of high-order Nyström discretizations in scattering calculations. We also thank Dr. George Valley for reviewing the manuscript and offering useful suggestions for improvement. The U.S. Government's right to retain a nonexclusive royalty-free license in and to the copyright covering this paper, for governmental purposes, is acknowledged.

REFERENCES

1. J. J. Ottusch, Performance comparison of FastScat(TM) and RAM2D, in *Presentations of Electromagnetic Code Consortium Annual Meeting, Albuquerque, NM, May 1994.*
2. L. R. Hamilton, J. J. Ottusch, M. A. Stalzer, R. S. Turley, J. L. Visher, and S. M. Wandzura, FastScat benchmark data, in *Proc. 1994 HAVE FORUM Symposium, Wright Patterson AFB, OH 454-7523*, Vol. I, p. 255 (Wright Laboratory, Feb. 1995). [WL-TR-95-6003]
3. S. Wandzura, High-order discretization of integral equations with singular kernels, in *IEEE Antennas Propag. Soc. Int. Sympos. Digest, Newport Beach, CA*, Vol. 1, p. 792 (IEEE, New York, June 1995).
4. J. S. Kot, Computer modelling of mm-wave integrated circuit antennas using the Nyström method, in *International Conference on Computation in Electromagnetics*, Vol. 3, p. 25 (IEEE Press, New York, Nov. 1991).

5. R. Kress, Numerical solution of boundary integral equations in time-harmonic electromagnetic scattering, *Electromagnetics* **10**, 1 (1990).
6. W. H. Press, B. P. Flannery, S. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C—The Art of Scientific Computing* (Cambridge Univ. Press, Cambridge, 1988).
7. L. M. Delves and J. L. Mohamed, *Computational Methods for Integral Equations* (Cambridge Univ. Press, New York, 1985).
8. J. Strain, Locally-corrected multidimensional quadrature rules for singular functions, *SIAM J. Sci. Comput*. **16**, 992 (1995).
9. S. M. Rao, D. R. Wilton, and A. W. Glisson, Electromagnetic scattering by surfaces of arbitrary shape, *IEEE Trans. Antennas Propag*. **AP-30**, 409 (1982).
10. S. M. Wandzura, Electric current basis functions for curved surfaces, *Electromagnetics* **12**, 77 (1992).
11. R. Coifman, V. Rokhlin, and S. Wandzura, The fast multipole method: A pedestrian prescription, *IEEE Antennas Propag. Soc. Mag*. **35**, 7 (1993).
12. E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, AIM: adaptive integral method for solving large-scale electromagnetic scattering and radiation problems, *Radio Sci*. **31**, 1225 (1996).
13. V. Rokhlin and M. A. Stalzer, Scalability of the fast multipole method for the Helmholtz equation, in *Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, MN* (SIAM, Philadelphia, 1997).
14. J. M. Song and W. C. Chew, Multilevel fast-multipole algorithm for solving combined field equations of electromagnetic scattering, *Microwave Opt. Technol. Lett*. **10**, 14 (1995).
15. S. D. Gedney, J. J. Ottusch, P. Petre, J. Visher, and S. Wandzura, Efficient high-order discretization schemes for integral equation methods, in *IEEE Symposium on Antennas and Propagation, Montreal, Canada, July 1997*.
16. J. N. Lyness and D. Jespersen, Moderate degree symmetric quadrature rules for the triangle, *J. Inst. Math. Appl*. **15**, 19 (1975).
17. N. Morita, N. Kumagai, and J. R. Mautz, *Integral Equation Methods for Electromagnetics* (Artech, Boston, 1990).
18. D. Colton and R. Kress, *Integral Equation Methods in Scattering Theory* (Wiley, New York, 1983).
19. J. J. Bowman, T. B. A. Senior, and P. L. E. Uslenghi (Eds.), *Electromagnetic and Acoustic Scattering by Simple Shapes* (Hemisphere, New York, 1987).
20. L. Canino, L. Hamilton, J. J. Ottusch, R. Ross, J. Visher, and S. Wandzura, FastScat performance on EMCC benchmark cases, in *Presentations of Electromagnetic Code Consortium Annual Meeting, Rome, NY, May 1996*.
21. *User's Manual for FISC (Fast Illinois Solver Code)* (Center for Computational Electromagnetics at the University of Illinois and DEMACO, Inc., 1997).
22. V. Rokhlin, personal communication, 1997.
23. A. W. Maue, Toward formulation of a general diffraction problem via an integral equation, *Z. Phys*. **126**, 601 (1949).
24. J.-H. Ma, V. Rokhlin, and S. Wandzura, Generalized Gaussian quadrature rules for systems of arbitrary functions, *SIAM J. Numer. Anal*. **33**, 971 (1996).
25. M. G. Duffy, Quadrature over a pyramid or cube of integrands with a singularity at a vertex, *J. Numer. Anal*. **19**, 1260 (1982).