Advances in Computational Mathematics 12 (2000) 151–174

A fast matrix–vector multiplication method for solving the radiosity equation

Kendall Atkinson [a,*] and David Da-Kwun Chien [b]

[a] *Department of Computer Science and Department of Mathematics, University of Iowa, Iowa City, IA 52242, USA*

[b] *Department of Mathematics, California State University – San Marcos, San Marcos, CA 92096, USA*

Received September 1998; revised September 1999
Communicated by I. Sloan

A "fast matrix–vector multiplication method" is proposed for iteratively solving discretizations of the *radiosity equation* $(I - \mathcal{K})u = E$. The method is illustrated by applying it to a discretization based on the centroid collocation method. A convergence analysis is given for this discretization, yielding a discretized linear system $(I - K_n)\mathbf{u}_n = \mathbf{E}_n$. The main contribution of the paper is the presentation of a fast method for evaluating multiplications $K_n\mathbf{v}$, avoiding the need to evaluate K_n explicitly and using fewer than O(n^2) operations. A detailed numerical example concludes the paper, and it illustrates that there is a large speedup when compared to a direct approach to discretization and solution of the radiosity equation. The paper is restricted to the surface S being unoccluded, a restriction to be removed in a later paper.

Keywords: radiosity equation, integral equation, numerical analysis, collocation methods, fast matrix–vector multiplication

AMS subject classification: 65R20

1. Introduction

The *radiosity equation* is a mathematical model for the brightness of a collection of one or more surfaces when their reflectivity and emissivity are given. The equation is

$$u(P) - \frac{\rho(P)}{\pi} \int_S u(Q)G(P,Q)V(P,Q)\,\mathrm{d}S_Q = E(P), \quad P \in S, \tag{1}$$

with $u(P)$ the "brightness" or *radiosity* at P and $E(P)$ the *emissivity* at $P \in S$. The function $\rho(P)$ gives the *reflectivity* at $P \in S$, with $0 \leqslant \rho(P) < 1$. In deriving this equation, the reflectivity is assumed to be uniform in all directions; that is, the surface is a *Lambertian diffuse reflector*.

* This researcher was supported in part by NSF grant DMS-9403589.

The function G is given by

$$G(P,Q) = \frac{\cos\theta_P \cos\theta_Q}{|P-Q|^2} = \frac{[(Q-P)\cdot\mathbf{n}_P][(P-Q)\cdot\mathbf{n}_Q]}{|P-Q|^4}. \tag{2}$$

In this, $\mathbf{n}_P$ is the inner unit normal to S at P, θ_P is the angle between $\mathbf{n}_P$ and $Q-P$, and $\mathbf{n}_Q$ and θ_Q are defined analogously. The function $V(P,Q)$ is a "line of sight" function. More precisely, if the points P and Q can "see each other" along a straight line segment which does not intersect S at any other point, then $V(P,Q)=1$; and otherwise, $V(P,Q)=0$. An *unoccluded* surface is one for which $V \equiv 1$ on S, and the numerical solution of this case by some collocation methods was studied previously in [5]. Note that S need not be connected, and it is usually only piecewise smooth. General introductions to the derivation, numerical solution, and application of the radiosity equation (1) can be found in the books of Cohen and Wallace [8] and Sillion and Puech [17]. In this paper, we consider only the unoccluded case, with $V \equiv 1$. In a later paper, we will consider the more practical occluded case; but the major aspects of the fast matrix–vector multiplication will remain the same.

We often write (1) in the simpler form

$$u(P) - \int_S K(P,Q)u(Q)\,\mathrm{d}S_Q = E(P), \quad P \in S, \tag{3}$$

or in operator form as

$$(I-\mathcal{K})u = E. \tag{4}$$

In section 2, we review the properties of G and $\mathcal{K}$, along with the solvability of (1). When the boundary S is smooth, the operator $\mathcal{K}$ is a compact operator on various function spaces associated with (3). However, for the more practical case that S possesses edges and corners, the mapping properties of $\mathcal{K}$ are more subtle and are as yet not as well-understood. A major contribution to the study of the properties of $\mathcal{K}$ when S contains edges has recently been given by Rathsfeld [15], and related results are given in [4] for the planar radiosity equation. We give some additional details in later sections.

In this paper we propose a "fast method" for solving discretizations of (1). The predominant form of discretization of (1) has been the Galerkin method with piecewise constant functions as the approximations. We believe collocation methods are more efficient, in part because they involve simpler integrations. To illustrate our fast solution method, we apply it to the very simple *centroid method*, the simplest of collocation methods. It has the same rate of convergence as the Galerkin method that is based on piecewise constant approximations. The stability and convergence of the centroid method is discussed in section 3; and we also discuss an iteration method for the solution of the linear system $A\mathbf{u} = \mathbf{E}$, $A = I - K$, associated with the centroid method. The iteration method used is again very simple, being equivalent to the

standard Neumann series solution of $(I - K)\mathbf{u} = \mathbf{E}$. Improvements of this iteration method are needed, and we are examining such methods.

The most important characteristic of this linear system $A\mathbf{u} = \mathbf{E}$ is that typically it has a very large order n, usually in the thousands. Consequently, the cost of calculating A is quite large, the storage requirement for A is large, and the cost of the iterative solution of $A\mathbf{u} = \mathbf{E}$ is large. Herein we extend an idea developed for boundary integral equations to the "fast solution" of this linear system. The method does not calculate K explicitly, avoiding the calculation of most of the O(n^2) integrals making up K. For each given vector $\mathbf{v}$, it produces the matrix–vector product $K\mathbf{v}$ to within acceptable accuracy in O($n \log^5 n$) operations, which is an improvement on the O(n^2) operations of the standard methods for larger values of n. When combined with the iteration method, we have a numerical method for solving (1) in $O(n \log^5 n)$ operations, again an improvement over standard procedures. This method is described in section 4; and numerical examples are presented in section 5. At the conclusion of section 5, we give a cost comparison with the direct setup and solution of $A\mathbf{u} = \mathbf{E}$; and there is a very large speedup with our fast method.

Other "fast solution methods" have been studied for solving $A\mathbf{u} = \mathbf{E}$. These include methods based on the *fast multipole method*, used previously for solving boundary integral equations, and *wavelet compression methods*. As examples, see [8, chapter 7; 10,12].

The centroid method has a very low order of convergence, but it is sufficient for illustrating the fast matrix–vector multiplication method discussed in the paper. Moreover, many of the methods discussed in the literature have the same order of convergence as that presented here. We have also avoided considering higher order methods, as they need much more attention to be given to the meshing of S. Again, we note that this paper is restricted to S being unoccluded.

2. Properties of the radiosity equation

We quote some results from other sources, especially [5, section 2], and we introduce some additional remarks on the properties of $\mathcal{K}$ in the case S is only piecewise smooth. The properties of the integral operator $\mathcal{K}$ are not yet fully understood when S is not a smooth surface, but they are similar to properties of the double layer boundary integral operator on piecewise smooth surfaces, from the subject of potential theory.

For S to be *piecewise smooth*, we mean the following. The surface S can be decomposed into a finite union,

$$S = S_1 \cup \cdots \cup S_J \tag{5}$$

with each S_j a smooth surface, i.e., there is a function

$$F_j : R_j \xrightarrow[\text{onto}]{1-1} S_j \tag{6}$$

with R_j a closed simply-connected polygon in $\mathbb{R}^2$ and F_j a twice continuously differentiable function on R_j. The closed subsurfaces S_j are allowed to intersect only along common edges and corners. Note that it is quite possible that S is a disconnected set. As notation, the expression

$$f \in \widetilde{C}^m(S)$$

will have the meaning

$$f \circ F_j \in C^m(R_j), \quad j = 1, \ldots, J,$$

with no restrictions of continuity across edges of S. In particular, $f \in \widetilde{C}(S)$ will mean only that f is continuous over each subsurface S_j.

The radiosity equation will be considered within the framework of the function space $L^\infty(S)$ with the uniform norm $\|\cdot\|_\infty$. For work in later sections, it is important to note that $L^\infty(S)$ is a space of cosets of functions. When we say $x \in L^\infty(S)$, we often treat x as a function, whereas, in fact, $x = [x_0]$ with x_0 a measurable function on S and

$$[x_0] = \big\{y \mid y \text{ measurable on } S,\ y(t) = x_0(t) \text{ almost everywhere on } S\big\}.$$

When considering the values of $\mathcal{K}u(t)$, we can dispense with all t in some set of measure zero; and for our surfaces S, the collection of edges and corners form such a set of measure zero.

At all points $P, Q \in S$ at which S is a smooth surface, the kernel function $G(P, Q)$ is continuous for $P \neq Q$; and for $P \in S$ a point at which S is smooth, it can be shown that

$$\big|G(P, Q)\big| \leqslant c$$

for all points $Q \in S$ that are sufficiently close to P. It is not true, however, that this bound is uniform as P, Q vary over S, with $P \neq Q$ (e.g., see the below example (15)). In the kernel

$$K(P, Q) = \frac{1}{\pi}\,\rho(P)G(P, Q)V(P, Q)$$

we assume the reflectivity $\rho \in \widetilde{C}(S)$ and

$$\|\rho\|_\infty < 1. \tag{7}$$

With these assumptions on S and ρ, the function $K(P, Q)$ is defined except on a set of measure zero and it is a measurable function on $S \times S$. In addition, it is relatively straightforward to show that $\mathcal{K}$ maps $L^\infty(S)$ to $L^\infty(S)$.

From the surface being unoccluded, we obtain easily that

$$K(P, Q) \geqslant 0, \quad P, Q \in S,$$

at all points at which $K(P, Q)$ is defined. The following results are basic to the solvability theory for (3).

Lemma 1. Assume S is the boundary of a convex bounded open set Ω, and assume S is a surface to which the Divergence Theorem can be applied. Let $P \in S$, and let S be smooth in an open neighborhood of P. Then $G(P,Q) \geqslant 0$ for $Q \in S$ and not on an edge of S, and

$$\int_S G(P,Q)\,\mathrm{d}S_Q = \pi. \tag{8}$$

Proof. See [5, section 2]. □

Lemma 2. Assume S is a piecewise smooth surface in $\mathbb{R}^3$ satisfying (5)–(6); and further assume $S \subset \widehat{S}$, with $\widehat{S}$ a surface to which the Divergence Theorem applies. Then for almost all $P \in S$,

$$\int_S K(P,Q)\,\mathrm{d}S_Q \leqslant \|\rho\|_\infty < 1. \tag{9}$$

Consequently, $\mathcal{K}: L^\infty(S) \to L^\infty(S)$ is bounded and

$$\|\mathcal{K}\| \leqslant \|\rho\|_\infty < 1. \tag{10}$$

Proof. This is a consequence of lemma 1. □

Theorem 3. Let S satisfy the assumptions of lemma 2. Then $I - \mathcal{K}: L^\infty(S) \xrightarrow[\text{onto}]{1-1} L^\infty(S)$ has a bounded inverse, and

$$\left\|(I-\mathcal{K})^{-1}\right\| \leqslant \frac{1}{1-\|\mathcal{K}\|} \leqslant \frac{1}{1-\|\rho\|_\infty}. \tag{11}$$

Moreover, for each $E \in L^\infty(S)$, the equation $(I-\mathcal{K})u = E$ has a unique solution $u \in L^\infty(S)$, and it satisfies

$$\|u\|_\infty \leqslant \frac{\|E\|_\infty}{1-\|\rho\|_\infty}.$$

Proof. This is immediate from the preceding results and the Geometric Series Theorem. In fact, the Geometric Series Theorem allows us to write

$$u = E + \mathcal{K}E + \mathcal{K}^2E + \cdots \tag{12}$$

and to know this is convergent in $L^\infty(S)$. □

2.1. Regularity results

When solving (1) numerically, it is important to know the differentiability properties of the solution u, as that affects the rate of convergence of the numerical approximants to u. When S is smooth, those properties are discussed in [5, section 2].

In brief, if S is the boundary of a convex bounded region and is $(m+2)$-times differentiable, and if E is m-times differentiable on S, then u is also m-times differentiable on S. The more practical situation is that S is only piecewise smooth, and we give a brief example of that case, to give some intuition of what to expect. For polyhedral regions with edges and with a piecewise constant reflectivity function ρ, Rathsfeld [15] gives the behavior of the radiosity solution in the vicinity of the edges. He shows that

$$u(P) \approx c_0 + c_1 r^{\gamma}, \quad c_0, c_1 \text{ constants}, \tag{13}$$

with P near to an edge of S and r equal to the distance of P from that edge. The exponent $\gamma \in (0,1)$, and it depends on both ρ and the angle at which the two subsurfaces meet to form the edge. This is also examined in some detail in [4] for the planar radiosity equation.

Let S be a polyhedral surface and consider the values of $\mathcal{K}u(P)$ as P approaches a point P_0 belonging to an edge of S. For definiteness, we assume the edges meet at a right angle; and we comment on the use of a general angle later, preceding (17). It suffices to consider the simple surface

$$S = S_{xz} \cup S_{xy}, \tag{14}$$

$$S_{xz} = \big\{(x,0,z) \mid 0 \leqslant x, z \leqslant 1\big\}, \qquad S_{xy} = \big\{(x,y,0) \mid 0 \leqslant x, y \leqslant 1\big\},$$

which are unit squares in the xz and xy-planes in $\mathbb{R}^3$, respectively. Let $P = (x,y,z)$ and $Q = (\xi,\eta,\zeta)$ belong to S. Then

$$G(P,Q) = \begin{cases} \dfrac{y\zeta}{[(x-\xi)^2 + y^2 + \zeta^2]^2}, & P \in S_{xy},\ Q \in S_{xz}, \\ \dfrac{z\eta}{[(x-\xi)^2 + \eta^2 + z^2]^2}, & P \in S_{xz},\ Q \in S_{xy}, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Note that $G(P,Q)$ has an unbounded discontinuity as P, Q approach a common edge point P_0 from different sections of S.

Lemma 4. Assume $u \in \widetilde{C}(S)$. For the surface S of (14), let $P_0 = (x_0, 0, 0)$ with $0 < x_0 < 1$, and assume the limits

$$\lim_{\substack{Q \to P_0 \\ Q \in S_{xy}}} u(Q) \equiv u_{xy}(P_0), \qquad \lim_{\substack{Q \to P_0 \\ Q \in S_{xz}}} \rho(Q) \equiv \rho_{xz}(P_0)$$

exist. Then

$$\lim_{\substack{P \to P_0 \\ P \in S_{xz}}} \mathcal{K}u(P) = \frac{1}{2} u_{xy}(P_0)\rho_{xz}(P_0). \tag{16}$$

This shows $\mathcal{K}$ acts like convolution with a Dirac delta function at edge points of S. Moreover, $\mathcal{K}u(P)$ is continuous on S at $P = P_0$ if and only if

$$u_{xy}(P_0)\rho_{xz}(P_0) = u_{xz}(P_0)\rho_{xy}(P_0)$$

with the obvious meaning for $u_{xz}(P_0)$ and $\rho_{xy}(P_0)$.

Proof. Using (15), write

$$\mathcal{K}u(P) = \frac{\rho(P)}{\pi}\int_0^1\int_0^1 \frac{z\eta u(\xi,\eta,0)\,\mathrm{d}\xi\,\mathrm{d}\eta}{[(x-\xi)^2+\eta^2+z^2]^2}, \quad 0<\xi,\eta\leqslant 1, \quad P=(x,0,z)\in S_{xz}.$$

For simplicity, we ignore the term $\rho(P)$, since it is easily dealt with. Let $\delta > 0$ be a parameter satisfying $\delta < \min\{x_0, 1-x_0\}$; and then decompose the above integral into two integrals, one over a δ-neighborhood of P_0 and one over the remaining portion of S_{xy}. Let

$$S_\delta = \left\{(\xi,\eta,0) \mid 0\leqslant\eta\leqslant\delta,\ x_0-\tfrac{1}{2}\delta\leqslant\xi\leqslant\ x_0+\tfrac{1}{2}\delta\right\}$$

and write

$$I_1 = \frac{1}{\pi}\iint_{S_\delta}\frac{z\eta u(\xi,\eta,0)\,\mathrm{d}\xi\,\mathrm{d}\eta}{[(x-\xi)^2+\eta^2+z^2]^2}$$

and I_2 as the integral over $S_{xy}\backslash S_\delta$. For each fixed $\delta > 0$, it is trivial that

$$\lim_{\substack{P\to P_0\\ P\in S_{xz}}} I_2 = 0.$$

Write $I_1 = I_{1,1} + u(x_0,0,0)I_{1,2}$ with

$$I_{1,1} = \frac{1}{\pi}\iint_{S_\delta}\frac{z\eta[u(\xi,\eta,0)-u(x_0,0,0)]\,\mathrm{d}\xi\,\mathrm{d}\eta}{[(x-\xi)^2+\eta^2+z^2]^2},$$
$$I_{1,2} = \frac{1}{\pi}\iint_{S_\delta}\frac{z\eta\,\mathrm{d}\xi\,\mathrm{d}\eta}{[(x-\xi)^2+\eta^2+z^2]^2}.$$

It is straightforward that

$$|I_{1,1}| \leqslant c\max_{Q\in S_\delta}|u(Q)-u(P_0)|, \quad c = \frac{1}{2}\left(2+\frac{1}{\sqrt{2}}\right).$$

Pick δ such that

$$|I_{1,1}| \leqslant \varepsilon$$

for a given $\varepsilon > 0$, and then fix δ. Directly,

$$I_{1,2} = \frac{1}{2\pi}\left\{\pi - \arctan\left(\frac{z}{x-x_0+\varepsilon/2}\right) - \arctan\left(\frac{z}{x_0+\varepsilon/2-x}\right)\right\}$$
$$- \frac{z}{2\pi\sqrt{\varepsilon^2+z^2}}\left\{\arctan\left(\frac{x-x_0+\varepsilon/2}{\sqrt{\varepsilon^2+z^2}}\right) + \arctan\left(\frac{x_0+\varepsilon/2-x}{\sqrt{\varepsilon^2+z^2}}\right)\right\}$$

and

$$\lim_{\substack{P\to P_0\\ P\in S_{xz}}} I_{1,2} = \frac{1}{2}.$$

Combining these results,

$$\limsup_{\substack{P\to P_0\\ P\in S_{xz}}} \left| \frac{1}{\pi} \int_0^1 \int_0^1 \frac{z\eta u(\xi,\eta,0)\,\mathrm{d}\xi\,\mathrm{d}\eta}{[(x-\xi)^2+\eta^2+z^2]^2} - \frac{1}{2}\,u_{xy}(P_0) \right| \leqslant \varepsilon.$$

Since ε was arbitrary, this proves (16). The continuity of $\mathcal{K}u(P)$ at $P = P_0$ follows by combining (16) with the analogous result for $P_0 \in S_{xy}$. □

The analogous result for planes meeting at other than a right angle can be proven similarly, although the intermediate formulas are a bit more complicated. Let

$$S = S_\varphi \cup S_{xy}$$

with S_φ a unit square meeting S_{xy} at an angle of φ, $0 < \varphi < \pi$. Then

$$\lim_{\substack{P\to P_0\\ P\in S_\varphi}} \mathcal{K}u(P) = (1+\cos\varphi)\,\frac{1}{2}\,u_{xy}(P_0)\rho_\varphi(P_0). \tag{17}$$

3. The centroid method

To describe and illustrate the fast matrix–vector multiplication method, we use the simplest of collocation methods, the *centroid method*. In it, we subdivide the surface S into closed elements $\{\Delta_j\}$ and we approximate u by an unknown constant over each element. We then determine the constants by forcing the approximating piecewise constant function to satisfy the integral equation (1) exactly at the centroids of the elements. In this section, we review the convergence theory for the method, and we discuss and illustrate the iterative solution of the associated linear system.

We use the framework for collocation methods that is described in [1, chapters 5 and 9], and only the most pertinent points are summarized here. An implementation of the numerical methods of this paper makes use of the boundary element package described in [3], to which the reader is referred for more detail. It includes examples of the centroid method applied to the radiosity equation.

We assume there is a sequence of triangulations of S, $\mathcal{T}_n = \{\Delta_{n,k} \mid 1 \leqslant k \leqslant n\}$, with some increasing sequence of integer values n converging to infinity. In our codes, the values of n increase by a factor of 4. For example, if S is an ellipsoid, then we often subdivide S into a sequence of triangulations $\{\mathcal{T}_n \mid n = 8, 32, 128, \ldots\}$. There are standard assumptions made on the triangulations. We describe the triangulation process briefly, and the details are left to [1, chapter 5]. More is said about the refinement process in section 4 below.

Associated with most surfaces are parameterizations of the surface, as in (5)–(6). Consider only one such parameterization function, say

$$F_j : R_j \xrightarrow[\text{onto}]{1-1} S_j$$

with R_j a polygonal region in the plane and some $1 \leqslant j \leqslant J$. Triangulate R_j, say as

$$\{\widehat{\Delta}_{n,k}^j \mid k = 1, \ldots, n_j\}. \tag{18}$$

This need not be a "conforming" triangulation, in contrast to the situation with finite element methods for solving partial differential equations. Next, triangulate the corresponding subsurface S_j using

$$\Delta_{n,k}^j = F_j(\widehat{\Delta}_{n,k}^j), \quad k = 1, \ldots, n_j. \tag{19}$$

For S as a whole, define

$$\mathcal{T}_n = \bigcup_{j=1}^{J} \{\Delta_{n,k}^j \mid k = 1, \ldots, n_j\}.$$

Often we will dispense with the subscript n, although it is to be understood implicitly. The *mesh size* of this triangulation is defined by

$$h \equiv h_n = \max_{1 \leqslant j \leqslant J} \max_{1 \leqslant k \leqslant n_j} \text{diameter}(\widehat{\Delta}_{n,k}^j).$$

As noted earlier, the elements of $\mathcal{T}_n$ are denoted collectively by $\mathcal{T}_n = \{\Delta_{n,k} \mid 1 \leqslant k \leqslant n\}$.

For purposes of numerical integration and interpolation over the triangular elements in $\mathcal{T}_n$, we need a parameterization of each such triangular element with respect to a standard reference triangle in the plane. Let $\Delta_{n,k} \in \mathcal{T}_n$, and assume $\Delta_{n,k}$ is associated with a planar triangle $\widehat{\Delta}_{n,k}$ and a mapping function F_j, so that $\Delta_{n,k} = F_j(\widehat{\Delta}_{n,k})$. Our reference triangle is the unit simplex,

$$\sigma = \{(s,t) \mid 0 \leqslant s, t, s+t \leqslant 1\}.$$

Let the vertices of $\widehat{\Delta}_{n,k}$ be denoted by $\{v_1, v_2, v_3\}$, and define a parameterization function $m_k : \sigma \xrightarrow[\text{onto}]{1-1} \Delta_{n,k}$ by

$$m_k(s,t) = F_j(uv_3 + tv_2 + sv_1), \quad (s,t) \in \sigma, \tag{20}$$

with $u = 1 - s - t$. Using this, we can write

$$\int_{\Delta_k} f(Q)\, \mathrm{d}S_Q = \int_{\sigma} f\big(m_k(s,t)\big) \big|(D_s m_k \times D_t m_k)(s,t)\big| \mathrm{d}\sigma \tag{21}$$

and this can be used to numerically evaluate the left-hand integral by using numerical integration formulas developed for the region σ. The *centroid* of $\Delta_{n,k}$ is defined as

$$P_k = m_k\big(\tfrac{1}{3}, \tfrac{1}{3}\big) = F_j\big(\tfrac{1}{3}(v_3 + v_2 + v_1)\big). \tag{22}$$

Define the operator $\mathcal{P}_n$ by

$$(\mathcal{P}_n f)(P) = f(P_k), \quad P \in \Delta_k, \quad k = 1, \ldots, n, \tag{23}$$

for $f \in \widetilde{C}(S)$. We are not concerned with the values of $\mathcal{P}_n f$ on the boundaries of the elements Δ_k, since $\mathcal{P}_n f$ is to be regarded as an element of $L^\infty(S)$ and need only be defined almost everywhere. We want to extend the above definition to all elements of $L^\infty(S)$, and to do so we call on the mathematical construction of point functionals as defined and analyzed in [7].

Let $\mathcal{C}(S)$ denote the closed subspace of $L^\infty(S)$ consisting of all cosets based on continuous functions:

$$\mathcal{C}(S) = \big\{[f] \mid f \in C(S)\big\}.$$

For a point $P \in S$, define a linear functional on $\mathcal{C}(S)$ by

$$\ell_P\big([f]\big) = f(P), \quad f \in C(S).$$

It is bounded with $\|\ell_P\| = 1$. Then using the Hahn–Banach theorem, the functional ℓ_P can be extended (in more than one way) to a linear functional on all of $L^\infty(S)$, with preservation of norm. We continue with the same notation for the extension; and properties of the extension are studied in [7], to which we refer the reader. The following is an important property, however, of all such extensions. Let $[f] \in L^\infty(S)$ and suppose that f is continuous at the point P. Then

$$\lim_{Q \to P} \ell_Q\big([f]\big) = \ell_P\big([f]\big) = f(P).$$

Thus the value of $\ell_P([f])$ possesses the expected value, without requiring that $f \in C(S)$.

The operator $\mathcal{P}_n$ can now be extended to $L^\infty(S)$:

$$\big(\mathcal{P}_n[f]\big)(P) = \ell_{P_k}\big([f]\big), \quad P \in \Delta_k, \quad k = 1, \ldots, n, \quad [f] \in L^\infty(S). \tag{24}$$

The operator $\mathcal{P}_n$ is a projection on $L^\infty(S)$, with

$$\|\mathcal{P}_n\| = 1, \quad n \geqslant 1. \tag{25}$$

The range of $\mathcal{P}_n$ is the set of all cosets of functions that are piecewise constant over the triangulation $\mathcal{T}_n$. Rather than continuing with the somewhat convoluted notation of (24), we will use the less precise notation of (23) as a shorthand for the more precise version of (24).

3.1. Stability and convergence

The collocation method for solving $(I - \mathcal{K})u = E$ is given in its abstract formulation by

$$(I - \mathcal{P}_n \mathcal{K})u_n = \mathcal{P}_n E \tag{26}$$

with $\mathcal{P}_n$ an interpolatory projection operator, generally on $L^\infty(S)$ or $\widetilde{C}(S)$. The centroid method uses the particular choice (24).

For stability of the centroid method, combine (10) and (25) to obtain

$$\|\mathcal{P}_n\mathcal{K}\| \leqslant \|\mathcal{K}\| < 1,$$
$$\left\|(I-\mathcal{P}_n\mathcal{K})^{-1}\right\| \leqslant \frac{1}{1-\|\mathcal{K}\|}, \quad n \geqslant 1.$$

For convergence, combine the above with the identity

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

to obtain

$$\|u-u_n\|_\infty \leqslant \frac{\|u-\mathcal{P}_n u\|_\infty}{1-\|\mathcal{K}\|}, \quad n \geqslant 1. \tag{27}$$

Theorem 5. Assume the equation (1) has a solution $u \in \widetilde{C}^1(S)$. Further, assume $F_j \in C^2(R_j)$, $1 \leqslant j \leqslant J$, for the parameterization functions of (6). Then

$$\|u-u_n\|_\infty = \mathrm{O}(h). \tag{28}$$

If instead the function u has a power singularity of the form given in (13), then

$$\|u-u_n\|_\infty = \mathrm{O}\big(h^\gamma\big). \tag{29}$$

Proof. The proof of (28) follows from the straightforward interpolation error bound

$$\|u-\mathcal{P}_n u\|_\infty = \mathrm{O}(h)$$

for a function $u \in \widetilde{C}^1(S)$. The result (29) follows similarly. □

The result (29) can be improved to $\mathrm{O}(h)$ by using a mesh that is suitably graded towards the edges of S, and that idea is explored in [4]. Neither of the results (28) or (29) is a rapid rate of convergence, but methods with such a rate of convergence are widely used. In part this follows from the nature of the additional approximations which are made in connection with setting up and solving the linear system associated with (26). It is also due to the difficulties of properly using higher order approximations. This includes the need to define the mesh so as to respect discontinuities in u and its derivatives. Less well-known, it also requires using a mesh that is properly graded near edges of S and near lines of discontinuity of derivatives of u. This has been explored previously in the literature on the numerical solution of boundary integral equations, and we refer the reader to Elschner [9] and Rathsfeld [14], for Galerkin and collocation methods, respectively. For a general discussion of graded meshes in one variable, see [1, p. 125].

Superconvergence results can also be proven in some cases. These are obtained by applying the bound

$$\max_{1\leqslant k\leqslant n}\big|u(P_k)-u_n(P_k)\big|\leqslant c\max_{1\leqslant k\leqslant n}\big|\big(\mathcal{K}(I-\mathcal{P}_n)u\big)(P_k)\big| \tag{30}$$

(see [1, pp. 449–450] for the general idea used in proving this bound). Using this bound, it is shown in [5] that if $u\in C^2(S)$ and if S is a C^3 surface, then

$$\max_{1\leqslant k\leqslant n}\big|u(P_k)-u_n(P_k)\big|\leqslant \mathrm{O}\big(h^2\big).$$

In the case of one-variable problems, it is known that this rate extends to piecewise smooth boundaries when the mesh is suitably graded near corner points of the integration region (e.g., see [4]). We expect that this result will extend to solving problems over piecewise smooth surfaces.

Example. Let S be the surface of the unit cube

$$\Omega=[0,1]\times[0,1]\times[0,1]. \tag{31}$$

The reflectivity is allowed to vary over S, being equal to 0.5 in the two sides parallel to the xz-plane, 0.4 in the two sides parallel to the xy-plane, and 0.3 in the two sides parallel to the yz-plane. We generate our example with the known true solution

$$u(x,y,z)=x^2+y^2+z^2.$$

The emissivity $E(x,y,z)$ is generated by means of a highly accurate numerical integration of $E=u-\mathcal{K}u$, and then the equation (1) is solved by the centroid method. Some numerical results are given in table 1. In the table,

$$\|\mathbf{u}-\mathbf{u}_n\|_\infty\equiv\max_{1\leqslant k\leqslant n}\big|u(P_k)-u_n(P_k)\big|.$$

Note that the rate of convergence is fairly consistent with a rate of convergence of O(h), which would be predicted from the C^1 behavior of the solution u over each smooth section of the surface.

Table 1
Errors in centroid method example.

n	$\|\mathbf{u}-\mathbf{u}_n\|_\infty$	Ratio
12	7.61E−2	
48	2.58E−2	2.95
192	1.37E−2	1.88
768	7.28E−3	1.88
3072	3.77E−3	1.93
12288	1.92E−3	1.96

In general, one would expect a slower rate of convergence, probably $O(h^\gamma)$ with $\gamma \doteq 0.763$, based on the results given in [15] and [4, table 1]. The value of γ varies with the reflectivity around the edge and it varies with the angle formed at the edge by the two adjoining subsurfaces. It is not yet known how the behavior at the corners will affect the rate of convergence, and the latter estimate of γ assumes the edge effects are more important than the corner effects.

3.2. The linear system

Equation (26) has the equivalent formulation

$$\mathcal{P}_n(I - \mathcal{K})u_n = \mathcal{P}_n E.$$

From this, we obtain the equivalent linear system

$$(I - K_n)\mathbf{u}_n = \mathbf{E}_n, \tag{32}$$

$$(\mathbf{u}_n)_i = u_n(P_i), \quad (\mathbf{E}_n)_i = E(P_i), \quad i = 1, \ldots, n,$$

$$(K_n)_{i,j} = \int_{\Delta_j} K(P_i, Q)\, \mathrm{d}S_Q, \quad i, j = 1, \ldots, n. \tag{33}$$

To set up the linear system requires the evaluation of these integrals. These must usually be evaluated by numerical integration, taking advantage of the formulation (21). For polyhedral surfaces S, these integrations can be made somewhat simpler, resulting in a large speedup in our codes. Note also that $(K_n)_{i,i} = 0$ since $G(P_i, Q) \equiv 0$ over Δ_i; and thus the system (32) contains no singular integrals, although some are likely to be nearly singular. For codes evaluating these integrals numerically, see [3].

If the surface is curved, then we generally approximate S by a simpler boundary. Because of the low rate of convergence in (28), it suffices to use a piecewise planar approximation of the boundary, based on using the vertices of the triangulation $\mathcal{T}_n$. This leads to a further error of size $O(h)$, but that is again adequate when considering (28). For simplicity, all of our examples will be for polyhedral surfaces S; and this seems to be the main case of interest in applications.

3.3. The iterative solution of the linear system

The iterative solution of (32) is based on fixed point iteration. Define

$$\mathbf{u}_n^{(k+1)} = \mathbf{E}_n + K_n \mathbf{u}_n^{(k)}, \quad k = 0, 1, \ldots, \tag{34}$$

for some given initial guess $\mathbf{u}_n^{(0)}$. Easily,

$$\mathbf{u}_n - \mathbf{u}_n^{(k+1)} = K_n\left[\mathbf{u}_n - \mathbf{u}_n^{(k)}\right] \tag{35}$$

and the method converges if $\|K_n\| < 1$. In this case, the matrix norm being used is the row norm, compatible with the uniform norm on $L^\infty(S)$. Examining the elements of K_n, the elements are all nonnegative; and for the sum of the ith row, we have

$$\sum_{j=1}^{n}(K_n)_{i,j} = \int_S K(P_i, Q)\,\mathrm{d}S_Q.$$

Thus

$$\|K_n\| \leqslant \|\mathcal{K}\|,$$

which, by (10), is bounded away from 1. This proves the convergence of (34), with a rate that is independent of the size of n.

Example 2. Solve the linear systems of example 1 by using the iteration of (34). Then after a few iterations, the ratios

$$\frac{\|\mathbf{u}_n - \mathbf{u}_n^{(k+1)}\|_\infty}{\|\mathbf{u}_n - \mathbf{u}_n^{(k)}\|_\infty}, \quad k = 0, 1, 2, \ldots, \tag{36}$$

approach a constant value of approximately 0.40, independent of n.

4. A fast matrix–vector multiplication method

There are two main practical problems in dealing with the linear system

$$(I - K_n)\mathbf{u}_n = \mathbf{E}_n \tag{37}$$

when n is large. First the setup time for the matrix K_n is proportional to n^2, with a large constant of proportionality. Second, the cost of the iteration method is also proportional to n^2, even though only a few iterations are needed with the centroid collocation method, because the error in the latter is not all that small. The setup time seems the greater expense in practice, as some of our timings, given later, will suggest. We propose a method to reduce the cost of both the matrix setup and the iteration procedure, to something of theoretical order $n \log^5 n$. The method is based on the work of Hackbusch and Nowak [11], who developed such a method for solving boundary integral equations. In the following we describe their work and adapt it to the radiosity equation. We consider the collocation method with only piecewise constant functions, but their ideas are suitable for piecewise polynomial approximations of any fixed degree. *To simplify our presentation and the construction of the programs, we assume the surface S is polyhedral throughout this and the following section. We further assume, as stated earlier, that the surface S is unoccluded.* These assumptions will be relaxed in a future paper and our results extended, but the most important ideas are presented more easily and intuitively with these assumptions.

We do not compute K_n explicitly. Rather, for a given vector $\mathbf{u}$, we compute an approximation to $K_n\mathbf{u}$. The cost will be much less than the setup cost for computing

K_n, followed by the cost of the matrix–vector multiplication $K_n\mathbf{u}$. This approximation process is under the control of two parameters M and η to control the approximation error, and the user supplies these parameters. The parameter M is a positive integer, and the parameter $\eta \in (0, 1)$; they are introduced below.

Recall

$$(K_n\mathbf{u})_i = \sum_{j=1}^{n} u_j \int_{\Delta_j} K(P_i, Q)\,\mathrm{d}S_Q, \quad i = 1, \ldots, n. \tag{38}$$

For each P_i, we separate out those elements Δ_j of the triangulation $\mathcal{T}_n$ that are near to P_i, and collectively these are called the "near field". The remaining elements are said to make up the "far field". This separation is associated with the parameter η. For elements Δ_j in the near field, the corresponding integrals in (38) are computed by standard methods; and we try to minimize the number of such integrations. For Q belonging to an element in the far field, we approximate $K(P_i, Q)$ based on a Taylor polynomial of degree M, and then we carry out all integrations exactly.

We have a sequence of triangulations $\mathcal{T}_n$, $n = n_0, n_1, \ldots, n_c$, with $\mathcal{T}_{n_c}$ the most current subdivision of S. For simplicity, we assume $n_\ell = 4^\ell n_0$, $\ell \geqslant 0$. Let

$$\mathcal{T}_{n_\ell} = \left\{\Delta_k^\ell\colon\ k = 1, \ldots, n_\ell\right\}, \quad \ell = 0, 1, \ldots, c. \tag{39}$$

We refer to this triangulation as being at "level ℓ". For every triangular element $\Delta \in \mathcal{T}_n$ for some n, let the center of a circumscribing circle be denoted by $A(\Delta)$ and let the radius of this circle be denoted by $R(\Delta)$. The collection

$$\mathcal{U} = \mathcal{T}_{n_0} \cup \cdots \cup \mathcal{T}_{n_c} \tag{40}$$

is a tree structure of clusters of the elements from the finest subdivision $\mathcal{T}_{n_c}$ of S, from the various levels of the refinement process.

Given a collocation point P_i, we say a cluster $\tau \in \mathcal{U}$ is *admissible* if

$$R(\tau) \leqslant \eta\left|P_i - A(\tau)\right|. \tag{41}$$

(The present meaning of η is that of $\overline{\eta}$ in (3.6) of [11].) If an element $\Delta \in \mathcal{T}_{n_c}$ is not admissible, we say it is in the *near field* of P_i; and the *far field* is in the closure of the complement in S of the near field. As η decreases, the size of the near field increases, and this generally increases the cost of computing $K_n\mathbf{u}$.

An *admissible covering* of S with respect to P_i is a collection

$$\mathcal{C} = \{\tau_1, \ldots, \tau_m\} \subset \mathcal{U}, \quad S = \bigcup_1^m \tau_i,$$

with each $\tau \in \mathcal{C}$ satisfying

$$\tau \in \mathcal{T}_{n_c} \quad \text{or} \quad \tau \text{ admissible}.$$

It is shown in [11, proposition 3.9] that for each P_i, there is a unique minimal admissible covering, unique in the sense of containing a minimum number of clusters τ.

To evaluate $(K_n\mathbf{u})_i$, begin by writing the unique minimal admissible covering of S with respect to P_i by

$$S = \left[\Delta_{i_1}^c \cup \cdots \cup \Delta_{i_q}^c\right] \cup [\tau_1 \cup \cdots \cup \tau_p] \tag{42}$$

with each τ_i an admissible cluster and each $\Delta_{i_j}^c$ a non-admissible element of the current triangulation $\mathcal{T}_{n_c}$. Then

$$(K_n\mathbf{u})_i = \sum_{j=1}^{q} u_{i_j} \int_{\Delta_{i_j}^c} K(P_i, Q)\,\mathrm{d}S_Q + \sum_{j=1}^{p} \int_{\tau_j} u_n(Q)K(P_i, Q)\,\mathrm{d}S_Q. \tag{43}$$

The function $u_n(Q)$ is the piecewise constant function over $\mathcal{T}_{n_c}$ with values determined from $\mathbf{u}$. The first integrals are evaluated by traditional means, by numerical integration, which is discussed later in this section. The remaining integrals are evaluated as follows. For the integral over τ_j, we use Taylor polynomial approximations involving $K(P_i, Q)$ and then perform the remaining integration exactly. How to do this efficiently is explained below.

4.1. The far field integration

Let $\tau \in \mathcal{U}$. Using a Taylor approximation in Q of degree M about $A(\tau)$, write

$$\frac{[(Q-P)\cdot\mathbf{n}_P]}{|P-Q|^4} \approx \sum_{\alpha\in I_M} a_\alpha(P)\big(Q - A(\tau)\big)^\alpha, \quad Q \in \tau, \tag{44}$$

with $A(\tau)$ the center of the circumscribing circle for τ, as introduced earlier. The set I_M consists of all multi-integers $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ with

$$\alpha_1, \alpha_2, \alpha_3 \geqslant 0, \qquad \alpha_1 + \alpha_2 + \alpha_3 \leqslant M.$$

As customary, if $Q = (\xi, \eta, \zeta)$, then $Q^\alpha = \xi^{\alpha_1}\eta^{\alpha_2}\zeta^{\alpha_3}$. From (44), we can obtain an analogous approximation of $K(P, Q)$:

$$\frac{[(Q-P)\cdot\mathbf{n}_P][(P-Q)\cdot\mathbf{n}_Q]}{|P-Q|^4} \approx \sum_{\alpha\in I_M} b_\alpha(P)\big(Q - A(\tau)\big)^\alpha. \tag{45}$$

Note that $\mathbf{n}_Q$ is constant over any $\tau \in \mathcal{U}$. More important in obtaining (45) from (44), the quantity $(P-Q)\cdot\mathbf{n}_Q$ is constant over an element τ. To see this, decompose $P-Q$ into components perpendicular and parallel to τ. All variation in $P-Q$ takes place parallel to τ, and the part perpendicular to τ remains constant.

Next, expand and rearrange the terms in (45) into the form

$$K(P, Q) \approx \sum_{\alpha\in I_M} \kappa_\alpha\big(P, A(\tau)\big)Q^\alpha, \quad Q \in \tau. \tag{46}$$

Return to (43), to the approximation of

$$\int_{\tau_j} u_n(Q)K(P_i, Q)\,\mathrm{d}S_Q.$$

Then use (46) to write

$$\int_{\tau_j} u_n(Q)K(P_i,Q)\,\mathrm{d}S_Q \approx \sum_{\alpha\in I_M} \kappa_\alpha\big(P,A(\tau)\big)\int_{\tau_j} u_n(Q)Q^\alpha\,\mathrm{d}S_Q \tag{47}$$

The integrals

$$\int_{\tau} u_n(Q)Q^\alpha\,\mathrm{d}S_Q, \quad \tau\in\mathcal{U}, \quad \alpha\in I_M,$$

can be evaluated explicitly, and they can be obtained in a preprocessing step before beginning the iterative solution of (37). As they are independent of the field point P and the center of expansion $A(\tau)$, they can be obtained in O(n) steps.

In greater detail, let $\tau\in\mathcal{U}$. Then we can write

$$\tau = \Delta_{i_1}^c \cup\cdots\cup \Delta_{i_k}^c$$

for some elements $\Delta_{i_j}^c \in \mathcal{T}_{n_c}$. Then

$$\int_{\tau} u_n(Q)Q^\alpha\,\mathrm{d}S_Q = \sum_{j=1}^{k} u_n(Q)\int_{\Delta_{i_j}^c} Q^\alpha\,\mathrm{d}S_Q \tag{48}$$

since $u_n(Q)$ is constant over each element in $\mathcal{T}_{n_c}$. Produce the integrals over the elements of $\mathcal{T}_{n_c}$, and then extend those to the remaining clusters $\tau\in\mathcal{U}$ using (48). Since we are working with S a polyhedral surface, the integrals on the right hand side of (48) can be evaluated explicitly.

In Hackbusch and Nowak [11], it is shown that with appropriate choices of $\eta=\eta(n)$ and $M=M(n)$, the quantity $K_n\mathbf{u}_n$ can be approximated, say by $\widetilde{K_n\mathbf{u}_n}$, at a cost of O($n\log^5 n$) operations with

$$\big\|K_n\mathbf{u}_n - \widetilde{K_n\mathbf{u}_n}\big\|_\infty = \mathrm{O}(h)\|\mathbf{u}_n\|_\infty. \tag{49}$$

This assumes the near field integrals in (43) are evaluated exactly, or that they are evaluated with an error consistent with (49). This is an excellent theoretical result; and in practice, our results seem even better.

4.2. The near field integration

Consider the evaluation of the near field integrals

$$\int_\Delta K(P_i,Q)\,\mathrm{d}S_Q$$

of (43). Let σ denote the unit simplex in the st-plane, and let

$$m:\sigma \xrightarrow[\text{onto}]{1-1} \Delta$$

be an affine mapping that parameterizes Δ. Then

$$\int_{\Delta} K(P_i, Q)\,\mathrm{d}S_Q = 2\mathrm{Area}(\Delta)\int_{\sigma} K\big(P_i, m(s,t)\big)\,\mathrm{d}\sigma. \tag{50}$$

We divide the evaluation into two cases.

Case (a). Let $P_i \in \Delta$. Then the integrand in (50) is zero because $\mathbf{n}_Q$ is orthogonal to $P - Q$ in this case, thus making $K(P, Q) = 0$ as Q varies over Δ.

Case (b). Let $P_i \notin \Delta$. Then the integrals (50) are nonsingular. However, they vary from being almost singular when P_i is very near Δ, to having very well-behaved integrands when P_i is distant from Δ.

An efficient quadrature method with an exponential rate of convergence is defined and analyzed in Schwab [16]. We have used the following somewhat simpler schema of [3], and it is reasonably efficient in most cases. Introduce a parameter μ to indicate the number of levels of subdivision of Δ for a composite quadrature scheme. For the basic quadrature scheme, we use the 7-point rule T2:5-1, of degree of precision 5, from Stroud [18].

1. If $0 < \mathrm{dist}(P_i, \Delta) \leqslant h$, use μ levels of subdivision of Δ and apply the basic quadrature scheme to each of the resulting 4^{μ} subelements.
2. If $h < \mathrm{dist}(P_i, \Delta) \leqslant 2h$, use $\mu - 1$ levels of subdivision of Δ and apply the basic quadrature scheme to each of the resulting $4^{\mu-1}$ subelements.
3. Continue this until no subdivision of Δ takes place, so that the basic quadrature scheme is applied directly to Δ.

As n increases to $4n$ to $16n$, it seems best empirically to increase μ to $\mu + 1$, although this is not always necessary. Typically, we have used $\mu = 0, 1, 2, 3$, with $\mu = 3$ reserved for only the largest values of n, say $n = 4^5 n_0$. With $k \geqslant 0$ levels of subdivision, the number of integrand evaluations in the composite quadrature scheme over Δ is $7 \cdot 4^k$. Thus we want to keep μ as small as possible, whereas it needs to be large enough to ensure accurate evaluation of the integrals in (50). For additional discussion in the case of solving boundary integral equations, see [2].

At present, we are also exploring methods for the exact evaluation of the near field integrals, and we hope to present those results in a future paper.

4.3. Practicalities

The approximation of $K_n\mathbf{u}_n$ requires a knowledge of μ, η, and M. For the most part, knowing how to choose these parameters is mostly a matter of developing intuition through experimentation. Of particular use in developing this intuition is the identity

$$\int_S K(P, Q)\,\mathrm{d}S_Q = 1$$

for surfaces S that are the boundary of a bounded simply connected convex region in $\mathbb{R}^3$ (i.e., the surface is unoccluded). The identity is for all points $P \in S$ other than edge and corner points. For additional test cases, we chose radiosity functions $u(Q)$ and then defined the emissivity function $E(P)$ accordingly from (1), evaluating it by high accuracy numerical integration. Some examples of the results are given in the following section.

There is an additional problem with storing the large number of quantities produced in the preprocessing stage of the computation. To do this, we created a large buffer in the main memory, dumping it as needed to disk. Then for each iteration, it was read into the core buffer and used in generating the approximation to $K_n\mathbf{u}_n$. Examples of the needed size of disk storage are also given in the following section.

5. Numerical examples

We illustrate the use of our fast matrix–vector multiplication process by solving $(I - K_n)\mathbf{u}_n = \mathbf{E}_n$ by the iterative method of (34). All of the results are for the example given following (31). We vary the parameters μ of section 4.2, η of (41), and M of (44). We give the resulting error

$$\|\mathbf{u} - \mathbf{u}_n\|_\infty \equiv \max_{1 \leqslant k \leqslant n} \big|u(P_k) - u_n(P_k)\big|, \tag{51}$$

some associated times, and disk storage costs. The errors should be compared with those of table 1. These results were computed on an IBM RS-6000/360 workstation. The machine has a 50 MHz processor and a main memory of 64 megabytes.

In table 2, we fix the degree $M = 1$ and vary both μ and η. For each cell in the table, the top figure is the error $\|\mathbf{u} - \mathbf{u}_n\|_\infty$ resulting when solving the linear system using iteration and the fast matrix–vector multiplication method. This is the error in the answer obtained after 20 iterations, although far fewer than 20 were needed. The

Table 2
Fast matrix–vector multiplication: degree $M = 1$.

n	(η, μ)					
	(0.2, 1)	(0.2, 2)	(0.2, 3)	(0.1, 1)	(0.1, 2)	(0.1, 3)
768	7.48E−3	7.06E−3	7.05E−3	8.56E−3	6.65E−3	6.67E−3
	4.72E+1	1.49E+2	5.52E+2	2.53E+2	9.09E+2	3.53E+3
	7.49	7.49	7.49	1.55E+1	1.55E+1	1.55E+1
3072	7.45E−3	7.49E−3	7.49E−3	5.24E−3	3.26E−3	3.29E−3
	1.74E+2	3.96E+2	1.28E+3	7.61E+2	2.40E+3	8.95E+3
	3.91E+1	3.91E+1	3.91E+1	1.46E+2	1.46E+2	1.46E+2
12288	7.74E−3	7.76E−3	7.76E−3	3.54E−3	1.87E−3	1.87E−3
	1.22E+3	1.68E+3	3.54E+3	2.80E+3	6.40E+3	2.08E+4
	1.93E+2	1.93E+2	1.93E+2	6.46E+2	6.46E+2	6.46E+2
	79	79	79	290	290	290

Table 3
Fast matrix–vector multiplication: degree $M = 2$.

n	(η, μ)					
	$(0.2, 1)$	$(0.2, 2)$	$(0.2, 3)$	$(0.1, 1)$	$(0.1, 2)$	$(0.1, 3)$
768	9.20E−3	7.29E−3	7.31E−3	9.17E−3	7.26E−3	7.28E−3
	4.72E+1	1.48E+2	5.50E+2	2.57E+2	9.06E+2	3.52E+3
	9.51	9.51	9.51	1.81E+1	1.81E+1	1.81E+1
3072	5.76E−3	3.78E−3	3.80E−3	5.72E−3	3.75E−3	3.77E−3
	1.73E+2	3.95E+2	1.28E+3	7.59E+2	2.39E+3	8.91E+3
	5.03E+1	5.03E+1	5.03E+1	1.79E+2	1.79E+2	1.79E+2
12288	3.95E−3	1.93E−3	1.96E−3	3.91E−3	1.90E−3	1.92E−3
	1.21E+3	1.68E+3	3.53E+3	2.80E+3	6.38E+3	2.07E+4
	2.45E+2	2.45E+2	2.45E+2	8.36E+2	8.36E+2	8.36E+2
	79	79	79	290	290	290

Table 4
Fast matrix–vector multiplication: degree $M = 3$.

n	(η, μ)					
	$(0.2, 1)$	$(0.2, 2)$	$(0.2, 3)$	$(0.1, 1)$	$(0.1, 2)$	$(0.1, 3)$
768	9.22E−3	7.30E−3	7.32E−3	9.17E−3	7.26E−3	7.28E−3
	4.73E+1	1.48E+2	5.52E+3	2.53E+2	9.09E+2	3.53E+3
	1.43E+1	1.43E+1	1.43E+1	2.43E+1	2.43E+1	2.42E+1
3072	5.77E−3	3.79E−3	3.82E−3	5.73E−5	3.75E−3	3.77E−3
	1.74E+2	3.91E+2	1.28E+3	7.62E+2	2.40E+3	8.93E+3
	7.73E+1	7.93E+1	7.74E+1	2.53E+2	2.60E+2	2.53E+2
12288	3.96E−3	1.95E−3	1.97E−3	3.91E−3	1.90E−3	1.92E−3
	1.22E+3	1.72E+3	3.54E+3	2.80E+3	6.39E+3	2.08E+4
	3.80E+2	3.80E+2	3.81E+2	1.25E+3	1.25E+3	1.26E+3
	79	79	79	290	290	290

second figure is the time (in seconds) for the setup, the third figure is the average cost of the 20 iterations (including disk access). The final figure, at the bottom of each column, is the fixed disk storage in megabytes (given only for the most expensive case of $n = 12288$). Tables 3 and 4 contain the corresponding results for degrees $M = 2, 3$.

A more detailed examination of the timings for different parts of the program yields additional information. In a typical iteration for larger values of n (say $n \geqslant 768$), the cost of computing the near field integrals is at most 2–3% of the total time; and for $n = 12288$, it was at most 1%. For the setup costs, the near field integrals are usually the major cost, depending on the value of μ, η, and n; see table 5. Thus efficient near-field integration is still important, just as for the case in which the matrix K_n is calculated explicitly.

From these various tables, it appears that the choice of parameters $(\eta, \mu, M) = (0.2, 2, 2)$ is sufficient for the values of n considered here. Of course, the needed

Table 5
Setup costs: proportion due to near field integrals.

n	(η, μ)					
	(0.2, 1)	(0.2, 2)	(0.2, 3)	(0.1, 1)	(0.1, 2)	(0.1, 3)
768	0.76	0.92	0.98	0.93	0.98	0.99
3072	0.46	0.76	0.93	0.76	0.93	0.98
12288	0.14	0.38	0.70	0.46	0.75	0.92

parameter values will be problem dependent, but this gives some indication of the values needed in practice. One need not go to large values of μ and M or to small values of η in order to obtain satisfactory accuracy. Of course, as the accuracy of the underlying solution $\mathbf{u}_n$ of the original collocation method increases, we also need to change the parameters (η, μ, M) so as to maintain that accuracy in the further approximations implicit in the fast matrix–vector multiplication method. Again, it will be necessary to do some experimentation to obtain some sense of the needed size for these parameters.

5.1. Comparison with direct setup and solution

To give a comparison with the direct setup and solution of the linear system, we give comparative timings for $(\eta, \mu, M) = (0.2, 2, 2)$ and $n = 3072$. These timings were done after those given above, on a Hewlett-Packard C200 workstation with a 200 MHz PA-8200 CPU, 768 MB RAM, and a 4 GB hard disk for local swapping. This is much faster than the machine used for the earlier calculations, and therefore the timings given below are much smaller than those given earlier for the comparable case. For the comparison, we did not do the case $n = 12288$ since our machine does not have sufficient main memory (RAM) for setting up the complete system; it would require 1.21 GB RAM to store the coefficients of this linear system.

At the request of a referee, we have changed the integration rule used in the computation of the near field integrals in section 4.2. We have replaced the 7-point rule T2:5-1 of Stroud [18] with a lower order rule, namely the 3-point rule

$$\int_\sigma f(x,y)\,\mathrm{d}\sigma \approx \tfrac{1}{6}\left[f\left(0,\tfrac{1}{2}\right) + f\left(\tfrac{1}{2},0\right) + f\left(\tfrac{1}{2},\tfrac{1}{2}\right)\right],$$

which has degree of precision 2. Since the centroid collocation method has only degree of precision 0, the above 3-point integration rule should not change the order of convergence of the overall method when used properly. We also use this same rule, together with the scheme described in case (b) of section 4.2, to evaluate the integrals in the direct setup and solution of the linear system. In fact, the value of $\mu = 2$ is not sufficient for reproducing the accuracy of the collocation method, and $\mu = 3$ is now necessary. Nevertheless, we did the comparisons for $\mu = 2$.

- *Complete setup and direct solution of linear system*. This uses the setup of the linear system as described earlier, using the numerical integration scheme described

in section 4.2 with $\mu = 2$. All the collocation integrals of (33) are calculated numerically. Our program took special note that the surface was polyhedral, thereby simplifying the calculation of surface differentials and normals. The program used is given in the package [3], with the substitution of the 3-point integration rule, as noted above.

– The matrix setup cost was approximately 214 seconds. For $n = 12288$, the setup costs would have been roughly

$$4^2 \cdot 214 \approx 3400 \text{ seconds}. \tag{52}$$

– The linear system was solved directly using the LINPACK routine DGESL, and the runtime was approximately 1180 seconds. Note that if the system for $n = 12288$ was set up and solved by this same method, the cost of solving the linear system would have been approximately

$$4^3 \cdot 1180 \approx 75000 \text{ seconds}. \tag{53}$$

Compare this with the figures given below for our "fast" solution of this problem.

– When iteration was used to solve the linear system, as in formula (34) and with the matrix K_n already computed, the cost per iteration was approximately 6.33 seconds. Starting from an initial guess of $\mathbf{u}_n^{(0)} = \mathbf{0}$, approximately $k = 13$ iterations were needed to reach an accuracy such that $\mathbf{u} - \mathbf{u}_n^{(k)} \doteq 3.74 \times 10^{-3}$, which is the best possible for the given values of $\mu = 2$ and $n = 3072$. The geometric rate of convergence, as defined in (36), converged very quickly to 0.397. It is clearly better, when $n = 3072$, to use iteration rather than a direct method of solving the linear system; and this is even more true for $n = 12288$ when we expect the cost of each iteration to be approximately

$$4^2 \cdot 6.33 \approx 101 \text{ seconds}.$$

- *The fast matrix–vector multiplication method.* The near field integrals were calculated with the scheme described in section 4.2, and the results were stored on the local hard disk for later retrieval in the iteration method. Then the fast matrix–vector multiplication was used in performing the iteration scheme of (34). The timings include the costs of the disk usage, both in the setup of the near field integrals and the matrix–vector multiplications for each iteration.

  – The cost of refining the triangulations and of calculating the needed clustering of the triangulations into near and far fields for $n = 3072$ (which includes the earlier triangulations beginning with $n = 12$) was approximately 23 seconds. There was an additional 187 seconds to go to the case of $n = 12288$.

  – The cost of calculating the near field integrals $n = 3072$ was approximately 14 seconds, and it was approximately 29 seconds for $n = 12288$. This is done only once, stored in RAM or on disk, and then retrieved as needed during the iteration process. We also produce information regarding clustering of the triangulation

and basic integrals over these clusters, storing this information on disk for later retrieval.

– For $n = 3072$, each iteration cost approximately 15 seconds, with again 13 iterations needed to reach the best possible accuracy with respect to **u**. For the case of $n = 12288$, the cost per iteration of the fast matrix–vector multiplication was approximately 68 seconds.
– The total solution cost for $n = 3072$, including 13 iterations, was approximately 230 seconds. Compare this to the more than 1400 seconds for the direct setup and solution of the collocation linear system; or compare to the approximately 300 seconds for the direct setup and the iterative solution. For $n = 12288$, including 15 iterations, it was approximately 1260 seconds with our fast method. For comparison, the direct setup and solution costs would be vastly greater, as indicated in (52)–(53); and the direct setup and iterative solution is estimated to be around 4900 seconds.

Note that our fast matrix–vector multiplication algorithm has larger costs per iteration for $n = 3072$. Our implementation of the direct method did not make use of disk storage, whereas our fast matrix–vector method does make use of disk storage for intermediate quantities (with several megabyte-sized buffers being used). Had the direct method used such disk storage, its timing costs per iteration would have been increased somewhat; and such disk storage will be needed with larger values of n such as 12288. Nonetheless, the increased iteration cost of our fast method is more than compensated for by the significant decrease in the total solution cost. With the direct method, the setup costs are $O(n^2)$ and the cost of solving the linear system is $O(n^3)$. These increase much more rapidly with n than does our fast method. The cost of solving directly the linear system becomes excessive as n becomes large, and iteration must be used, even if sufficient memory is available for a direct solution. The setup cost of the complete matrix and the subsequent cost of an interative solution is more manageable; but our method is still superior for such larger values of n.

Clearly, these comparisons will vary with the speed of convergence of the iteration method and with the nature of the solution being sought. But our example demonstrates that significant decreases in computing costs are possible with our method.

References

[1] K. Atkinson, *The Numerical Solution of Integral Equations of the Second Kind* (Cambridge University Press, 1997).
[2] K. Atkinson, The numerical solution of boundary integral equations, in: *The State of the Art in Numerical Analysis*, eds. I. Duff and G. Watson (Clarendon Press, Oxford, 1997) pp. 223–259.
[3] K. Atkinson, User's guide to a boundary element package for solving integral equations on piecewise smooth surfaces (Release #2), Reports on Computational Mathematics, No. 103, Department of Mathematics, University of Iowa, Iowa City (1998). The programs and guide are available via the URL http://www.math.uiowa.edu/~atkinson/bie.html.

[4] K. Atkinson, The planar radiosity equation and its numerical solution, IMA J. Numer. Anal., to appear. Also available as Reports on Computational Mathematics, No. 109, Department of Mathematics, University of Iowa (1998).
[5] K. Atkinson and G. Chandler, The collocation method for solving the radiosity equation for unoccluded surfaces, J. Integral Equations Appl. 10 (1998) 253–290.
[6] K. Atkinson and D. Chien, Piecewise polynomial collocation for boundary integral equations, SIAM J. Sci. Comput. 16 (1995) 651–681.
[7] K. Atkinson, I. Graham and I. Sloan, Piecewise continuous collocation for integral equations, SIAM J. Number. Anal. 20 (1983) 172–186.
[8] M. Cohen and J. Wallace, *Radiosity and Realistic Image Synthesis* (Academic Press, New York, 1993).
[9] J. Elschner, The double layer potential operator over polyhedral domains II: Spline Galerkin methods, Math. Methods Appl. Sci. 15 (1992) 23–37.
[10] S. Gortler, P. Schröder, M. Cohen and P. Hanrahan, Wavelet radiosity, in: *Computer Graphics Proceedings*, Annual Conference Series (1993) pp. 221–230.
[11] W. Hackbusch and Z. Nowak, On the fast matrix multiplication in the boundary element method by panel clustering, Numer. Math. 54 (1989) 463–491.
[12] P. Hanrahan, D. Salzman and L. Aupperlie, A rapid hierarchical radiosity algorithm, Computer Graphics 25(4) (1991) 197–206.
[13] R. Lewis, Solving the classic radiosity equation using multigrid techniques, Preprint, University of British Columbia (1992).
[14] A. Rathsfeld, The invertibility of the double layer potential operator in the space of continuous functions defined on a polyhedron: The panel method, Appl. Anal. 45 (1992) 135–177.
[15] A. Rathsfeld, Edge asymptotics for the radiosity equation over polyhedral boundaries, Preprint 369, Weierstraß-Institut für Angewandte Analysis und Stochastik, Berlin (1997).
[16] C. Schwab, Variable order composite quadrature of singular and nearly singular integrals, Comput. 53 (1994) 173–194.
[17] F. Sillion and C. Puech, *Radiosity and Global Illumination* (Morgan Kaufmann, San Francisco, CA, 1994).
[18] A. Stroud, *Approximate Calculation of Multiple Integrals* (Prentice-Hall, Englewood Cliffs, NJ, 1971).