



Accurate Computation of the Radiosity Gradient with Constant and Linear Emitters

Nicolas Holzschuch, François X. Sillion

► To cite this version:

Nicolas Holzschuch, François X. Sillion. Accurate Computation of the Radiosity Gradient with Constant and Linear Emitters. *Rendering Techniques 1995 (Eurographics Symposium on Rendering)*, Jun 1995, Dublin, Ireland. pp.186-195. inria-00379406v1

HAL Id: inria-00379406

<https://inria.hal.science/inria-00379406v1>

Submitted on 28 Apr 2009 (v1), last revised 13 Apr 2012 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Accurate Computation of the Radiosity Gradient for Constant and Linear Emitters

Nicolas Holzschuch, François Sillion

iMAGIS/IMAG*

Abstract: Controlling the error incurred in a radiosity calculation is one of the most challenging issues remaining in global illumination research. In this paper we propose a new method to compute the value and the gradient of the radiosity function at any point of a receiver, with arbitrary precision. The knowledge of the gradient provides fundamental informations on the radiosity function and its behaviour. It can specially be used to control the consistency of the discretisation assumptions.

1 Introduction

Computing the effect of a given patch on the radiosity of another patch is easily done assuming the radiosity on both patches are constant. In that case, we can express the influence of the emitter on the receiver with a single number, the form-factor. However, assuming the radiosity on both patches is constant is a strong assumption, and it introduces a specific source of error in the resolution algorithm.

In 1994, Arvo et al. [2] recorded all possible sources of error in global illumination algorithms, and introduced a framework for the analysis of error. Errors can occur at several levels in the resolution process:

- During modeling: our geometry is not exactly that of the scene we want to compute, and the BRDF are not exact either.
- During discretisation: our set of basis functions is not able to represent the real solution, but only an approximated one.
- During computation: we do not compute transfer elements exactly, but only within finite precision.

Lischinski et al. [9] presented an error driven refinement strategy for hierarchical radiosity. They were able to maintain upper and lower bounds on computed radiosity, and to concentrate their work in places where the difference was too large.

However, practical tools are still lacking to measure discretisation error. The problem is to efficiently reconstruct the radiosity function, with only a small number of samples. The best position for sampling points can only be found with total knowledge of the radiosity function.

In practice, at each step, we have to intuit the behaviour of the function from our current set of samples, in order to guess if we should – or not – introduce new sampling points, and where.

* iMAGIS is a joint research project of CNRS/INRIA/INPG/UJF. Postal address: B.P. 53, F-38041 Grenoble Cedex 9, France. E-mail: Nicolas.Holzschuch@imag.fr.

Knowing the radiosity derivatives allows better sampling, and thus reduction of discretisation error. Heckbert [6] and Lischinski et al. [7] predicted an efficient surface mesh using derivatives discontinuities. Drettakis and Fiume [4, 5] used information on the structure of the function to accurately reconstruct the illumination. Vedel and Puech [11] presented a refinement criterion based on gradient values at the nodes.

However, these authors usually resorted to approximated values of the partial derivatives, using several computations of radiosity and finite differences. Computing accurate values for the gradient allows arbitrary precision on our refinement criterion.

Arvo [1] presented a method to compute the irradiance Jacobian in case of partially occluded light sources. His method is presented with constant emitters. This paper introduces a new formulation of the radiosity gradient, valid for arbitrary radiosity functions on the emitter. The derivation is presented in the case of total visibility, i.e. without occluders. However, we shall see that extending the algorithm to the case of partial visibility is easy using Arvo's technique, since the two algorithms are largely independant.

2 Reformulating the Radiosity Equation

We will consider only diffuse surfaces, characterised by their radiosity function $B(x)$, without any assumption about B .

We want to know the value of B at a point x on a given patch A_1 , due to the emission of light from another polygon A_2 . We will assume a reflectivity of ρ at point x .

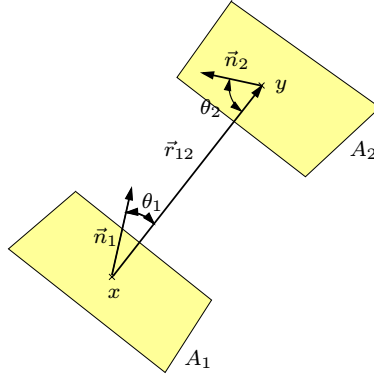


Fig. 1. Geometry of the problem

Our knowledge of radiosity at the receiving point derives from the integral equation:

$$B(x) = \frac{\rho}{\pi} \int_{A_2} \frac{B(y) \cos \theta_1 \cos \theta_2}{\|\vec{r}_{12}\|^2} dA_2 \quad (1)$$

where $\vec{r}_{12}$ is the vector joining point x on the receiver and point y on the emitter. θ_1 is the angle between $\vec{r}_{12}$ and the normal on the receiver, θ_2 the angle between $\vec{r}_{12}$ and the normal on the emitter, and dA_2 the area element on the emitter around point y (see Fig. 1).

Should any occluders be present between point x and emitter A_2 , the integral would only be over the part of A_2 visible from x .

We can reformulate Equation 1 as the expression of the flux of a vector field through surface A_2 :

$$B(x) = \int_{A_2} \vec{F} \cdot d\vec{A}_2 \quad (2)$$

where $\vec{F}$ is:

$$\vec{F} = -\frac{\rho B(y)(\vec{r}_{12} \cdot \vec{n}_1)\vec{r}_{12}}{\pi \|\vec{r}_{12}\|^4}$$

A classic way to deal with flux integrals as Equation 2 is to transform them into a linear integral using Stoke's theorem²:

$$\int_A (\nabla \times \vec{V}) \cdot d\vec{A} = \oint_{\partial A} \vec{V} \cdot d\vec{x} \quad (3)$$

These linear integrals can be easier to compute, and are also easier to estimate if there are no closed forms. However, to use Stoke's theorem (3), we need to express the vector field $\vec{F}$ as the curl of another vector field, $\vec{V}$.

A classic property is that this is equivalent to $\vec{F}$ having a null divergence ($\nabla \cdot \vec{F} = 0$). Basically, the divergence of a vector flux is a quantity that express at each point how much does the flux “radiates away” from this point, while the curl of a vector field “turns around” it at each point. The divergence of a curl is always null ($\nabla \cdot (\nabla \times \vec{V}) = 0$), and if a field has a null divergence, it can be expressed as a curl.

An easy computation shows that the divergence of $\vec{F}$ with respect to point y on surface A_2 is³:

$$\nabla \cdot \vec{F} = -\frac{\rho}{\pi} \frac{\vec{r}_{12} \cdot \vec{n}_1}{\|\vec{r}_{12}\|^2} (\nabla(B) \cdot \vec{r}_{12}) \quad (4)$$

and hence is null if the gradient of B on the emitting surface is null. That is to say, if the radiosity of the emitter is constant.

We can always separate $\vec{F}$ in two parts:

$$\vec{F} = \nabla \times (\vec{V}) + \vec{G}$$

Namely:

$$\begin{aligned} \vec{V} &= \rho B(y) \frac{\vec{r}_{12} \times \vec{n}_1}{2\pi \|\vec{r}_{12}\|^2} \\ \vec{G} &= -\rho \nabla(B) \times \left(\frac{(\vec{r}_{12} \times \vec{n}_1)}{2\pi \|\vec{r}_{12}\|^2} \right) \end{aligned}$$

and thus cut Equation 2 in two integrals:

$$B(x) = \oint_{\partial A_2} \vec{V} \cdot d\vec{x}_2 + \int_{A_2} \vec{G} \cdot d\vec{A}_2 \quad (5)$$

Using the properties of cross-products and dot-products, we can rewrite Equation 5 as:

$$\frac{2\pi}{\rho} B(x) = -\vec{n}_1 \cdot \oint_{\partial A_2} B(y) \frac{\vec{r}_{12} \times d\vec{x}_2}{\|\vec{r}_{12}\|^2} + \int_{A_2} \frac{\vec{r}_{12}}{\|\vec{r}_{12}\|^2} \cdot (\vec{n}_1 \times (\nabla(B) \times \vec{n}_2)) dA_2 \quad (6)$$

² ∂A stands for the contour of A , and $\oint$ expresses that this contour is closed.

³ In this section, all derivative signs (∇ , $\nabla \cdot$, $\nabla \times$) are relative to point y on surface A_2 .

Note that this rewriting process does not make any assumption whatsoever on $B(y)$. Hence it can be used in any case. An interesting case is when $B(y)$ is constant: then $\vec{G} = \vec{0}$, and the second term is null. Another interesting case is $B(y)$ being linear: then its gradient is constant and can be carried out of the second integral, leaving only a pure geometric factor to compute. Appendix A presents a detailed study of these two cases.

This rewriting process separates the radiosity in two terms, a contour integral that we can generally compute, provided that we know the radiosity on the emitter, and a surface integral, generally harder to compute as an exact term. But, as shown later, having an integral form of this term, we can compute its value with an arbitrary precision.

3 The Radiosity Gradient

An interesting quantity to describe scalar fields, such as $B(x)$ is their gradient. Gradient is the extension of derivation for function of several variables. Basically, $\nabla(B)(x) \cdot \vec{v}$ gives the derivative of function B at point x in the direction of $\vec{v}$.

3.1 Computing the Gradient

The radiosity gradient can be computed from an equation such as Equation 1 or 6:

$$\nabla(B)(x) = \nabla \left(\int_{A_2} \vec{F} \cdot d\vec{A}_2 \right) \quad (7)$$

In case the emitter A_2 does not depend on the position of the point x – that is to say, in case there are no occluder between point x and the emitting surface A_2 – this equation is equivalent to:

$$\nabla(B)(x) = \int_{A_2} \nabla \left(\vec{F} \cdot d\vec{A}_2 \right)$$

Or, if we use Equation 5:

$$\nabla(B)(x) = \oint_{\partial A_2} \nabla \left(\vec{V} \cdot d\vec{x}_2 \right) + \int_{A_2} \nabla(\vec{G} \cdot d\vec{A}_2) \quad (8)$$

If the emitter depends on the position of point x – that is, if there are occluders – the expression of $\nabla(B)(x)$ is the sum of two terms; the first one takes into account the variation of $\vec{F}$, and is exactly the term we are discussing, and the second one takes into account the variation of the emitter. Thus, it is easy to merge a method to compute the gradient with occluders and a constant emitter, as in Arvo [1], and our method to compute the gradient with an arbitrary emitter, but without occluders.

Note that in this section, we are taking a derivative with respect to point x on the receiving surface, not with respect to point y on the emitting surface. So for our derivating operator, the radiosity on the emitting point $B(y)$ can be regarded as constant, as well as its gradient, $\nabla(B)(y)$.

Using the properties of the gradient of a scalar product, starting from Equation 8, we can express the gradient of radiosity at the receiving point:

$$\begin{aligned} \frac{2\pi}{\rho} \nabla(B)(x) &= \vec{n}_1 \times \oint_{\partial A_2} B(y) \frac{d\vec{x}_2}{\|\vec{r}_{12}\|^2} + 2 \oint_{\partial A_2} B(y) \frac{\vec{n}_1 \cdot \vec{r}_{12}}{\|\vec{r}_{12}\|^4} (\vec{r}_{12} \times d\vec{x}_2) \\ &\quad + \int_{A_2} (\vec{n}_1 \times (\nabla(B)(y) \times \vec{n}_2)) \frac{dA_2}{\|\vec{r}_{12}\|^2} \\ &\quad - 2 \int_{A_2} \frac{(\vec{n}_1 \times (\nabla(B)(y) \times \vec{n}_2)) \cdot \vec{r}_{12}}{\|\vec{r}_{12}\|^4} \vec{r}_{12} dA_2 \end{aligned} \quad (9)$$

This equation, like the radiosity equation (6) is divided in two parts: a contour integral which usually has a closed form, and a surface integral that we can estimate to any arbitrary precision.

As before, two interesting cases occur: if the gradient on the emitter is null, that is if we assume a constant radiosity on the emitter, all surface integrals vanish. And if the gradient on the emitter is constant, that is if we assume a linear radiosity on the emitter, it can be carried out of the surface integrals, leaving us with purely geometrical factors or vectors to compute. Please refer to Appendix A for a detailed study of these cases.

3.2 Using the gradient

Knowing the gradient at a point gives very valuable information on the function we are studying. As previous authors pointed out, the gradient may be used either to reconstruct the illumination function before display, or to check the consistency of our discretisation hypothesis.

Reconstructing the illumination function If we know the radiosity values and the gradient at our sample points, we can then reconstruct the radiosity function as, e.g. a bicubic spline.

Salesin et al. [10] and Bastos et al. [3] proposed such methods for reconstruction of radiosity using estimates of gradient. Ward and Heckbert [12] computed irradiance gradients to interpolate irradiance on receiving surfaces.

Refinement criterion Many radiosity algorithms assume a constant radiosity over patches. It may seem strange to compute the gradient of radiosity in that case, but in fact the information given by the gradient can also be used there.

Using the derivatives allows precisely to check whether our discretisation hypothesis were correct or not, and if they were not, it also gives a hint on where it would be best to refine in order to minimize the discretisation error.

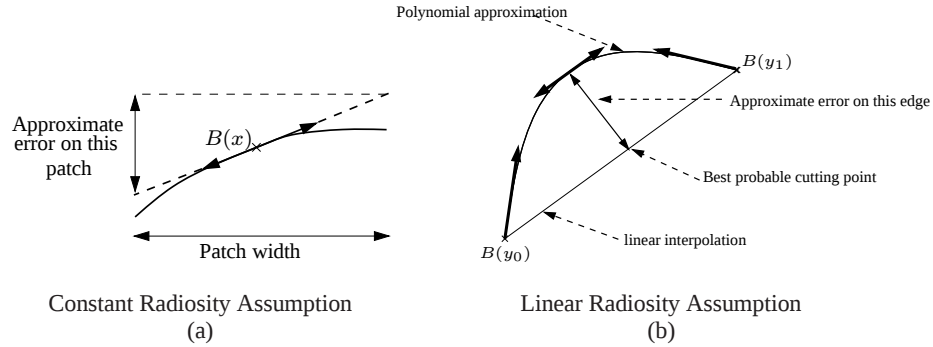


Fig. 2. Using the gradient to measure discretisation error

If we assume constant radiosity on our patches, the gradient gives a first estimate of how much does the function vary over the patch: $\nabla(B)(x) \cdot \vec{v}$ is approximately the difference between radiosity at point $x + \vec{v}$ and radiosity at point x . The norm of the gradient times the width of the patch gives an approximation of how much does radiosity varies over the patch (see Fig. 2a for an example in 2D). The direction of the gradient gives the best probable direction of refinement.

If we assume linear radiosity on our patches, we can compute a cubic interpolant over the patch using the radiosity and gradient values at each vertices, and then test how much

this cubic interpolant differs from our linear assumption (see Fig. 2b for an example in 2D). We can even compute the difference between linear and cubic interpolant without explicitly computing the interpolants. This criterion also gives the best next sampling point, the position of the maximum difference between the two interpolants.

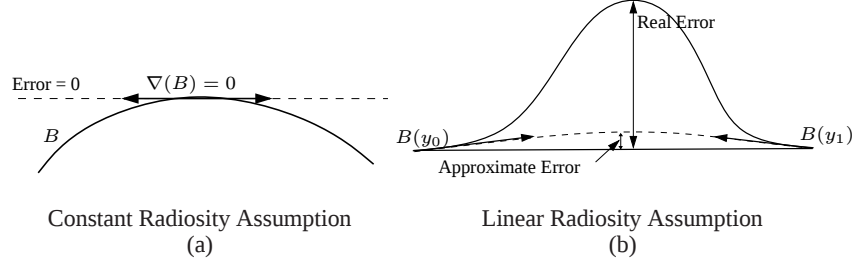


Fig. 3. Sample cases where the proposed refinement criterion fail

Although none of these refinement criterion are foolproof (see Fig. 3 for an example where these two criterion fail to detect an important discretisation error), they provide a way to measure and quantify the discretisation error.

Also, the points where these refinement criterions are more likely to be fooled are basically the extrema of the radiosity functions. We know that a single convex emitter induces only one maximum on the receiver (see, for example, Drettakis [4]).

So, to study the interaction between two patches so as to minimize discretisation error, we would, first, find the theoretical position of the maximum of radiosity, then sample it, then refine the receiving patch using our gradient-based criterion.

4 Implementation and First Results

We have implemented the gradient and radiosity formulas described in appendix A, for both constant and linear emitters⁴. Using a C++ class for vectors, with definitions of cross- and dot-products makes the implementation very straightforward, being a mere recopy of the formulas. The only special attention it needs is avoiding to recompute quantities already computed at previous steps. Most of the quantities needed to express the gradient were also used for the radiosity.

In the color plates, Fig. A shows the radiosity values on a plane, due to a triangular emitter parallel to that plane (see Fig. E for the geometry of the scene). Fig. B shows the norm of the gradient of this radiosity.

Fig. C and D show the same quantities if we assume a linear emitter.

5 Conclusion and Future Work

We have provided a way to compute the gradient of radiosity at the receiving point with any distribution of illumination on the emitter. The gradient can be used in several ways, and specially to compute the discretisation error. It can also be used to find the best next refining point.

Future work will include a complete gradient computation, using the method described by Arvo in [1] to take the possible occluders into account.

⁴ Source code and documentation for this implementation is available at <ftp://safran.imag.fr/pub/holzschu/gradient.tar.gz>.

The ability to compute radiosity gradients for linear emitters is especially interesting when using linear basis functions or linear wavelets. In that case, the discretisation error can be precisely isolated.

Our next step will be a complete implementation of the refinement criterion described in section 3.2, to effectively reduce the discretisation error, within a hierarchical radiosity framework with linear radiosity.

We will then have the possible background for a complete radiosity algorithm with all possible sources of error (visibility, discretisation, computational) recorded and monitored, thus allowing to focus the computing resources at the points where this error is large.

6 Acknowledgements

Color pictures were computed by Myriam Houssay-Holzschuch using the GMT package, developed by Wessel and Smith [13].

The authors would like to thank the anonymous reviewers for useful insights and positive criticism.

References

1. Arvo, J.: The Irradiance Jacobian for Partially Occluded Polyhedral Sources. *SIGGRAPH* (1994) 343–350
2. Arvo, J., Torrance, K., Smits, B.: A Framework for the Analysis of Error in Global Illumination Algorithms. *SIGGRAPH* (1994) 75–84
3. Bastos, R. M., de Sousa, A. A., Ferreira, F. N., Reconstruction of Illumination Functions using Bicubic Hermite Interpolation. *Fourth Eurographics Workshop on Rendering* (June 1993) 317–326
4. Drettakis, G., Fiume, E.: Concrete Computation of Global Illumination Accurate and Consistent Reconstruction of Illumination Functions Using Structured Sampling. *Computer Graphics Forum (Eurographics 1993 Conf. Issue)* 273–284
5. Drettakis, G., Fiume, E.: Concrete Computation of Global Illumination Using Structured Sampling. *Third Eurographics Workshop on Rendering* (May 1992) 189–201
6. Heckbert, P. S.: *Simulating Global Illumination Using Adaptive Meshing*. PhD Thesis, University of California, Berkeley, June 1991.
7. Lischinski, D., Tampieri, F., Greenberg, D. P.: Discontinuity Meshing for Accurate Radiosity. *IEEE Computer Graphics and Applications* 12,6 (November 1992) 25–39
8. Lischinski, D., Tampieri, F., Greenberg, D. P.: Combining Hierarchical Radiosity and Discontinuity Meshing. *SIGGRAPH* (1993)
9. Lischinski, D., Smits, B., Greenberg, D. P.: Bounds and Error Estimates for Radiosity. *SIGGRAPH* (1994) 67–74
10. Salesin, D., Lischinski, D., DeRose, T.: Reconstructing Illumination Functions with Selected Discontinuities. *Third Eurographics Workshop on Rendering* (May 1992) 99–112
11. Vedel, C., Puech, C.: Improved Storage and Reconstruction of Light Intensities on Surfaces. *Third Eurographics Workshop on Rendering* (May 1992) 113–121
12. Ward, G. J., Heckbert, P. S.: Irradiance Gradients. *Third Eurographics Workshop on Rendering* (May 1992) 85–98
13. Wessel, P. and Smith, W. H. F.: Free Software helps Map and Display Data. *EOS Trans. Amer. Geophys. U.*, vol. 72, 441–446, 1991

A Application to Constant and Linear Emitters

A.1 Case of a constant emitter

In the case of a constant emitter the Equations 6 and 9 reduce to:

$$\frac{2\pi}{\rho}B(x) = -\vec{n}_1 \cdot \oint_{\partial A_2} B(y) \frac{\vec{r}_{12} \times d\vec{x}_2}{\|\vec{r}_{12}\|^2} \quad (10)$$

$$-\frac{2\pi}{\rho}\nabla(B)(x) = \vec{n}_1 \times \oint_{\partial A_2} B(y) \frac{d\vec{x}_2}{\|\vec{r}_{12}\|^2} + 2 \oint_{\partial A_2} B(y) \frac{\vec{n}_1 \cdot \vec{r}_{12}}{\|\vec{r}_{12}\|^4} (\vec{r}_{12} \times d\vec{x}_2) \quad (11)$$

If A_2 is a polygon, these integrals have a closed form, and yield:

$$\begin{aligned} \frac{2\pi}{\rho}B(x) &= -B_2 \vec{n}_1 \cdot \sum_i I_1(i) (\vec{r}_i \times \vec{e}_i) \\ -\frac{2\pi}{\rho}\nabla(B)(x) &= B_2 \sum_i I_1(i) (\vec{n}_1 \times \vec{e}_i) \\ &\quad + 2B_2 \sum_i (\vec{r}_i \times \vec{e}_i) \cdot \vec{n}_1 (I_2(i)\vec{r}_i + J_2(i)\vec{e}_i) \end{aligned}$$

where the sum extends on all the edges of the polygon, and B_2 is the radiosity of the emitter. $\vec{r}_i$, $\vec{e}_i$, $I_1(i)$, $I_2(i)$ and $J_2(i)$ stand for (see also Fig. 4):

$$\begin{aligned} \vec{r}_i &= \overrightarrow{x E_i} \\ \vec{e}_i &= \overrightarrow{E_i E_{i+1}} \\ I_1(i) &= \frac{\gamma_i}{\|\vec{r}_i \times \vec{e}_i\|} \\ I_2(i) &= \frac{1}{2\|\vec{r}_i \times \vec{e}_i\|^2} \left(\frac{\vec{r}_{i+1} \cdot \vec{e}_i}{\|\vec{r}_{i+1}\|^2} - \frac{\vec{r}_i \cdot \vec{e}_i}{\|\vec{r}_i\|^2} + \|\vec{e}_i\|^2 I_1(i) \right) \\ J_2(i) &= \frac{1}{2\|\vec{e}_i\|^2} \left(\frac{1}{\|\vec{r}_i\|^2} - \frac{1}{\|\vec{r}_{i+1}\|^2} - 2I_2(i)\vec{r}_i \cdot \vec{e}_i \right) \end{aligned}$$

and γ_i is the angle sustained by edge $\vec{e}_i$ from point x .

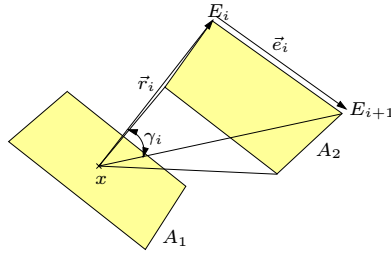


Fig. 4. Geometric Notations Used

Computing B at point x requires roughly 63 multiplications, 6 divisions, 54 additions or subtractions, 6 square roots and 3 arc cosines. This equals approximately 300 additions on an SGI Indy computer, with no optimisations and the standard compiler.

Computing $\nabla(B)(x)$ requires roughly 87 multiplications more, 57 additions more and 3 divisions more. Which, with the same material, equals approximately 150 additions.

Although this computational cost may depend on implementation details as well as on the computer used (some compilers have very fast implementations of arc cos and square root), computing the gradient along with the radiosity does not over-increase computation time.

A.2 Case of a linear emitter

If the emitter is not constant, the gradient of radiosity on the emitter is not null, and must be used in our computations. However, if we assume the radiosity of the emitter is linear, then its gradient is constant and can be carried out of the integrals. Moreover, this gradient is orthogonal to $\vec{n}_2$, and can be expressed as:

$$\nabla(B)(y) = \vec{n}_2 \times \vec{k}$$

with $\vec{k}$ orthogonal to $\vec{n}_2$. $\vec{k} = \frac{1}{2A_2}((B_2 - B_0)\vec{e}_0 + (B_1 - B_0)\vec{e}_2)$

Using the properties of $\vec{k}$, we can express Equation 6 as:

$$\frac{2\pi}{\rho}B(x) = -\vec{n}_1 \cdot \oint_{\partial A_2} B(y) \frac{\vec{r}_{12} \times d\vec{x}_2}{\|\vec{r}_{12}\|^2} + (\vec{n}_1 \cdot \vec{n}_2)(\vec{k} \cdot (\vec{m} \times \vec{n}_2)) + (\vec{m} \cdot \vec{n}_2)(\vec{n}_2 \cdot (\vec{n}_1 \times \vec{k}))$$

with:

$$\vec{m} = \int_{A_2} \frac{\vec{r}_{12}}{\|\vec{r}_{12}\|^2} dA_2 = \int_{A_2} \nabla(\ln(r_{12})) dA_2$$

Computing the contour integrals does not induce any particular difficulties. However, computing $\vec{m}$ is harder. We can make use of Ostrogradsky's theorem, similar to Stoke's:

$$\int_A \nabla(V) \times d\vec{A} = - \oint_{\partial A} V d\vec{x}$$

to express $\vec{m} \times \vec{n}_2$.

$\vec{m} \cdot \vec{n}_2$ is null if point x is on polygon A_2 . If point x is not on polygon A_2 , it can be estimated with arbitrary precision.

The formula for $B(x)$ is then:

$$\begin{aligned} \frac{2\pi}{\rho}B(x) = & -\vec{n}_1 \cdot \oint_{\partial A_2} B(y) \frac{\vec{r}_{12} \times d\vec{x}_2}{\|\vec{r}_{12}\|^2} - (\vec{n}_1 \cdot \vec{n}_2)\vec{k} \cdot \oint_{\partial A_2} \ln(r_{12})d\vec{x}_2 \\ & + (\vec{m} \cdot \vec{n}_2)(\vec{n}_2 \cdot (\vec{n}_1 \times \vec{k})) \end{aligned}$$

If we derive this formula rather than use Equation 9, we find:

$$\begin{aligned} -\frac{2\pi}{\rho}\nabla(B)(x) = & \vec{n}_1 \times \oint_{\partial A_2} B(y) \frac{d\vec{x}_2}{\|\vec{r}_{12}\|^2} + 2 \oint_{\partial A_2} B(y) \frac{\vec{n}_1 \cdot \vec{r}_{12}}{\|\vec{r}_{12}\|^4} (\vec{r}_{12} \times d\vec{x}_2) \\ & - (\vec{n}_1 \cdot \vec{n}_2) \oint_{\partial A_2} \frac{\vec{r}_{12}}{\|\vec{r}_{12}\|^2} (\vec{k} \cdot d\vec{x}_2) \\ & + (\vec{n}_2 \cdot (\vec{n}_1 \times \vec{k})) (\vec{n}_2 X_1 - 2(\vec{n}_2 \cdot \vec{r}_0)\vec{p}) \end{aligned}$$

with:

$$\vec{p} = \int_{A_2} \frac{\vec{r}_{12}}{\|\vec{r}_{12}\|^4} dA_2$$

$$X_1 = \int_{A_2} \frac{dA_2}{\|\vec{r}_{12}\|^2}$$

Computing $\vec{p}$ is exactly like computing $\vec{m}$: we can compute $\vec{p} \times \vec{n}_2$, and we can estimate $\vec{p} \cdot \vec{n}_2$ with arbitrary precision. Then we use:

$$\vec{p} = \vec{n}_2 \times (\vec{p} \times \vec{n}_2) + (\vec{p} \cdot \vec{n}_2) \vec{n}_2$$

Hence:

$$\begin{aligned} \frac{2\pi}{\rho} B(x) &= -\vec{n}_1 \cdot \sum_i (B_i I_1(i) + \delta B_i J_1(i)) (\vec{r}_i \times \vec{e}_i) \\ &\quad - (\vec{n}_1 \cdot \vec{n}_2) \sum_i (\vec{k} \cdot \vec{e}_i) K_1(i) \\ &\quad + (\vec{r}_0 \cdot \vec{n}_2) (\vec{n}_2 \cdot (\vec{n}_1 \times \vec{k})) X_1 \\ -\frac{2\pi}{\rho} \nabla(B)(x) &= \sum_i (B_i I_1(i) + \delta B_i J_1(i)) (\vec{n}_1 \times \vec{e}_i) \\ &\quad + 2 \sum_i \vec{n}_1 \cdot (\vec{r}_i \times \vec{e}_i) B_i (I_2(i) \vec{r}_i + J_2(i) \vec{e}_i) \\ &\quad + 2 \sum_i \vec{n}_1 \cdot (\vec{r}_i \times \vec{e}_i) \delta B_i (J_2(i) \vec{r}_i + K_2(i) \vec{e}_i) \\ &\quad - (\vec{n}_1 \cdot \vec{n}_2) \sum_i (\vec{k} \cdot \vec{e}_i) (I_1(i) \vec{r}_1 + J_1(i) \vec{e}_1) \\ &\quad - (\vec{n}_2 \cdot (\vec{n}_1 \times \vec{k})) (\vec{n}_2 \cdot \vec{r}_0) \vec{n}_2 \times \sum_i I_2(i) \vec{e}_i \\ &\quad + (\vec{n}_2 \cdot (\vec{n}_1 \times \vec{k})) (X_1 - 2(\vec{r}_0 \cdot \vec{n}_2)^2 X_2) \vec{n}_2 \end{aligned}$$

With:

$$\begin{aligned} \delta B_i &= B_{i+1} - B_i \\ J_1(i) &= \frac{1}{\|\vec{e}_i\|^2} \left[\ln \left(\frac{\|\vec{e}_{i+1}\|}{\|\vec{e}_i\|} \right) + (\vec{r}_i \cdot \vec{e}_i) I_1(i) \right] \\ K_1(i) &= \frac{1}{2\|\vec{e}_i\|^2} (\vec{r}_{i+1} \cdot \vec{e}_i \ln(\|\vec{e}_{i+1}\|^2) - \vec{r}_i \cdot \vec{e}_i \ln(\|\vec{e}_i\|^2) + 2\|\vec{r}_i \times \vec{e}_i\| \gamma_i) \\ K_2(i) &= \frac{1}{\|\vec{e}_i\|^2} (I_1(i) - \|\vec{r}_i\|^2 I_2(i) - 2(\vec{r}_i \cdot \vec{e}_i) J_2(i)) \\ X_2 &= \int_{A_2} \frac{dA_2}{\|\vec{r}_{12}\|^4} \end{aligned}$$

If the the distance between point x and the emitter surface is null, $\vec{m} \cdot \vec{n}_2$ and $\vec{p} \cdot \vec{n}_2$ are both null. If it is not, we prefer to estimate X_1 and X_2 . As we know bounds on the values of the function and its derivatives, we make use of a Gaussian quadrature.