EVALUATION OF THE ANISOTROPIC GREEN'S FUNCTION AND ITS DERIVATIVES

Mark A. Sales* and L. J. Gray†

November 25, 1996

Abstract

An algorithm for computing the anisotropic elastic Green's function and its derivatives is presented. The method is based upon the calculus of residues, the main requirement being the computation of the roots of a sixth degree polynomial. It is demonstrated herein that this procedure is faster and more accurate than the standard Wilson-Cruse interpolation scheme, and moreover the need for extensive precalculated tables is eliminated.

1 Introduction

Boundary integral equation analysis depends upon the ability to evaluate a Green's function $\mathbf{G}(\mathbf{P},\mathbf{Q})$ (fundamental solution) and its derivatives. These functions take on a relatively simple analytical form for isotropic elasticity [2, 4], but the situation is far more complicated for a general anisotropic material. The anisotropic three-dimensional Green's function can be expressed as a line integral [1, 12]

$$\mathbf{U}(\mathbf{x},\mathbf{y}) = \frac{1}{8\,\pi^2\,r}\oint_{S^1}\mathbf{K}^{-1}(\boldsymbol{\xi})\,ds(\boldsymbol{\xi}) \equiv \frac{1}{8\,\pi^2\,r}\tilde{\mathbf{U}}\ , \tag{1}$$

*Department of Mathematics, Vanderbilt University, Nashville TN 37235. Current address: Department of Physics, University of California, Berkeley CA 94720

†Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831-6367

where the Christoffel matrix $\mathbf{K}(\boldsymbol{\xi})$, $\boldsymbol{\xi} \in \mathcal{R}^3$, is defined in terms of the elastic constants $\mathbf{C}_{iljm}$ of the material,

$$K_{ij} = \sum_{l,m} C_{iljm} \xi_l \xi_m \ , \tag{2}$$

and $r = \|\mathbf{x} - \mathbf{y}\|$. The integration path is the unit circle in the plane having normal $\mathbf{x} - \mathbf{y}$,

$$S^1 = S^1(\mathbf{x}, \mathbf{y}) = \left\{ \boldsymbol{\xi} \in \mathcal{R}^3 \ \ | \ \ \|\boldsymbol{\xi}\| = 1, \ \boldsymbol{\xi} \cdot (\mathbf{x} - \mathbf{y}) = 0 \right\} \ . \tag{3}$$

A boundary integral calculation requires numerous Green's function evaluations, and thus direct numerical quadrature to obtain $\tilde{\mathbf{U}}$ (and its derivatives) is simply too time consuming to be practical [17, 20].

The best available algorithm has been the method proposed by Wilson and Cruse in 1978 [20] (the recent monograph by Schclar [15] provides a good overview of anisotropic boundary element methods). In their approach, computation time is brought back to a reasonable level (roughly twice that of an isotropic analysis) by employing cubic interpolation from tabulated precalculated values. While successful, the storage required for these tables can be significant. There are two independent variables, the spherical angles (θ, ψ) defining the orientation of $\mathbf{R} = \mathbf{Q} - \mathbf{P}$, and the Green's tensor has six independent components. Calculating first derivatives of $\mathbf{G}$ therefore involves twelve more two-dimensional tables, while second derivatives add another eighteen. Second derivatives arise when the 'derivative' (hypersingular) boundary integral equation is employed in the analysis. These equations are essential for fracture mechanics [4, 9] and many other applications [10, 13].

As noted in [15], another possible difficulty with the Wilson/Cruse approach is accuracy. The variation in the function values for a highly anisotropic material is likely to be significantly larger than for a mildly anisotropic solid, and thus the number of precalculated points (θ, ψ) needed to maintain accuracy is a function of the anisotropy of the material. Identifying appropriate values for the parameters $\Delta\theta$ and $\Delta\psi$ defining the tables may require effort. A judicious choice of grid points will likewise be needed with a recently proposed variation of the Wilson/Cruse algorithm [5]. This method interpolates the Green's function from pre-constructed bivariate cubic spline approximations.

Two alternatives to the Wilson-Cruse algorithm, both based upon the relative 'simplicity' of the isotropic case, have recently been proposed. The approach in [8] is based upon a perturbation series in which the zeroth order term is an appropriately chosen isotropic Green's function. Although this series yields fast algorithm, without stored tables, it is only practical for materials in which the anisotropy is 'not too strong'. The second technique [15] is specific to the boundary integral method, in that the Green's function is not directly evaluated. The fundamental solution is split into isotropic and anisotropic parts, and the anisotropic remainder is handled via the Dual Reciprocity Method [14]. The choice of

the appropriate number and location of internal points required by the Dual Reciprocity approximation is problem dependent and not an easy task, and an adaptive strategy has been proposed [15]. Thus, although no computation times are given, this method is not likely to be competitive with Wilson/Cruse. Moreover, this approach has not yet been employed in conjunction with the derivative boundary integral equation.

This paper presents an algorithm based upon a simple expression for $\tilde{\mathbf{U}}$ obtained by Dederichs and Leibfried [6] using residue calculus. It has generally been assumed that the calculation of the Green's function from this formula is not a viable algorithm [6]. However, it is shown that this is not the case, and that the extension to evaluating the first and second derivatives of $\tilde{\mathbf{U}}$ is equally successful. This approach appears to run faster than the Wilson/Cruse interpolation algorithm, and it eliminates the need for large stored tables. Moreover, this direct evaluation method is essentially exact and is not sensitive to the anisotropy of the material.

Recently, the residue approach has also been exploited by Wang [18, 19] to obtain forms for the Green's function and derivatives. However, these papers have been primarily theoretical, and have not focused on the computations.

2 Residue Evaluation

The contour integral in Eq. (1) can be written as

$$\tilde{\mathbf{U}}(\theta,\psi) = \int_0^{2\pi} \mathbf{K}^{-1}(\boldsymbol{\xi}(t))\, dt, \tag{4}$$

where $\boldsymbol{\xi}$ is written parametrically as

$$\boldsymbol{\xi}(t) = \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \\ \xi_3(t) \end{bmatrix} = \begin{bmatrix} \sin(\theta)\cos(t) + \cos(\theta)\cos(\psi)\sin(t) \\ -\cos(\theta)\cos(t) + \sin(\theta)\cos(\psi)\sin(t) \\ -\sin(\psi)\sin(t) \end{bmatrix}. \tag{5}$$

The integral in Eq. (4) can be evaluated using residue calculus by transforming the integrand into a rational function and expanding the range of the integral to include all real numbers. As Dederichs and Leibfried demonstrated [6], one way to achieve this transformation is with the substitution

$$Z = \tan(t)\,. \tag{6}$$

From Eq. (5) and Eq. (6), $\boldsymbol{\xi}$ can be written as

$$\boldsymbol{\xi} = \cos(t)\mathbf{s}(Z)\,, \tag{7}$$

where

$$\mathbf{s}(Z) = \begin{bmatrix} \sin(\theta) + Z\cos(\theta)\cos(\psi) \\ -\cos(\theta) + Z\sin(\theta)\cos(\psi) \\ -Z\sin(\psi) \end{bmatrix} . \tag{8}$$

Since the elements of $\boldsymbol{\xi}$ are linear functions of Z, and $\mathbf{K}$ is a quadratic function of $\boldsymbol{\xi}$ (Eq. (2)), the elements of $\mathbf{K}$ are second order polynomials in Z. Thus, by Cramer's rule, the elements of $\mathbf{K}^{-1}$ are rational functions of Z,

$$K_{jk}^{-1} = \frac{1}{cos^2(t)} \frac{P_{jk}(Z)}{Q(Z)} , \tag{9}$$

where the cofactor $P_{jk}(Z)$ and the determinant $Q(Z)$ are polynomials of Z,

$$P_{jk}(Z) = \sum_{l=0}^{4} a_{jkl} Z^l, \tag{10}$$

$$Q(Z) = \sum_{l=0}^{6} b_l Z^l . \tag{11}$$

From Eq. (9) and Eq. (6), and the observation that $1/\cos^2(t) = dZ/dt$, the integral in Eq. (4) can be written in the form

$$\tilde{U}_{jk} = 2\int_{-\infty}^{\infty} \frac{P_{jk}(Z)}{Q(Z)} dZ \tag{12}$$

$$= 4\pi i \sum_{n=1}^{3} Residue_{Z=\lambda_n} \frac{P_{jk}(Z)}{Q(Z)} \tag{13}$$

$$= 4\pi i \sum_{n=1}^{3} \frac{P_{jk}(Z)\,(Z-\lambda_n)}{Q(Z)} |_{Z=\lambda_n} \tag{14}$$

$$= 4\pi i \sum_{n=1}^{3} \frac{P_{jk}(\lambda_n)}{Q_n(\lambda_n)} \tag{15}$$

where λ_1, λ_2, and λ_3 represent the three roots of $Q(Z)$ in the upper half of the complex plane ($\Im(Z) > 0$). The factor of 2 in Eq. (12) appears because the change of variables $Z = \tan(t)$ covers the range $(-\infty, \infty)$ twice in the interval $[0, 2\pi]$. Note that $Q_n(Z)$ is equal to $Q(Z)$ with a factor of $(Z - \lambda_n)$ cancelled out, and thus Eq. (13) clearly assumes that the three pairs of roots are distinct. The question of multiple roots will be taken up in Section 4.2.

The derivatives of $\mathbf{U}(\mathbf{x}, \mathbf{y})$ with respect to the coordinates of $\mathbf{x}$ or $\mathbf{y}$ can be expressed in terms of derivatives of $\tilde{\mathbf{U}}$ with respect to θ and ψ, and a similar method can be applied

for their evaluation. Let α represent either angle, and denote partial differentiation by a subscript following a comma. Differentiating through the integral sign in Eq. (4) and once again employing residues yields

$$\begin{aligned}
\tilde{U}_{jk,\alpha} &= 2\int_{-\infty}^{\infty} K^{-1}_{jk,\alpha}(Z)\,dZ && (16)\\
&= 2\int_{-\infty}^{\infty} \frac{P_{jk,\alpha}(Z)\,Q(Z) - P_{jk}(Z)\,Q_{,\alpha}(Z)}{[Q(Z)]^2}\,dZ && (17)\\
&= 4\pi i \sum_{n=1}^{3} \frac{d}{dZ}\left\{\frac{\{P_{jk,\alpha}(Z)\,Q(Z) - P_{jk}(Z)\,Q_{,\alpha}(Z)\}}{[Q_n(Z)]^2}\right\}\Bigg|_{Z=\lambda_n}. && (18)
\end{aligned}$$

Note that the required roots λ_n are the same as in Eq. (15), and that $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{U}}_\alpha$ are fairly simple functions of these roots. Thus, fast calculation of *both* $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{U}}_\alpha$ can be accomplished once the roots are determined.

After some manipulation, Eq. (18) can be written as

$$\tilde{U}_{jk,\alpha} = 4\pi i \sum_{n=1}^{3} \left\{M_{1n}\,P_{jk}(\lambda_n) + M_{2n}\,P_{jk,Z}(\lambda_n) + M_{3n}\,P_{jk,\alpha}(\lambda_n)\right\}, \qquad (19)$$

where

$$\begin{aligned}
M_{1n} &= \frac{Q_{n,Z}(\lambda_n)\,Q_{,\alpha}(\lambda_n) - 2\,Q_n(\lambda_n)\,Q_{,Z\alpha}(\lambda_n)}{[Q_n(\lambda_n)]^3}, && (20)\\
M_{2n} &= \frac{-Q_{,\alpha}(\lambda_n)}{[Q_n(\lambda_n)]^2}, && (21)\\
M_{3n} &= \frac{Q_{,Z}(\lambda_n)}{[Q_n(\lambda_n)]^2}. && (22)
\end{aligned}$$

This form has several computational advantages. Note that $Q(Z)$ is the same for each tensor element, and thus all expressions involving Q, Q_n, and derivatives are likewise independent of $\{jk\}$. Thus, M_{in} can be computed once for the entire tensor, and can be treated as constants. Second, the expression for $\tilde{U}_{jk,\alpha}$ is now a simple function of $P_{jk},\ P_{jk,\alpha}, and\ P_{jk,Z}$, which are all polynomials of Z. The advantages of this are covered in Section 3.

As discussed in the introduction, the second derivatives of $\tilde{\mathbf{U}}$ with respect to θ or ψ are also needed for boundary integral analyses. The second derivatives can be obtained by differentiating twice through the integral sign in Eq. (4), and then reducing the expression to a form similar to Eq. (19). However, the manipulations become rather tedious, so it is

easiest to let a symbolic manipulation program such as Maple handle the differentiation and manipulation. The second derivatives can eventually be put into the form

$$\tilde{U}_{jk,\alpha\beta} = 4\pi i \sum_{n=1}^{3} [L_{1n}\, P_{jk} + L_{2n}\, P_{jk,\alpha} + L_{3n}\, P_{jk,\beta} + L_{4n}\, P_{jk,Z} + L_{5n}\, P_{jk,\alpha\beta} + L_{6n}\, P_{jk,Z\alpha} + L_{7n}\, P_{jk,Z\beta} + L_{8n}\, P_{jk,ZZ}] \tag{23}$$

where β, also denotes either θ or ψ. As with M_{in}, the coefficients L_{in} are independent of the tensor indices $\{jk\}$. They depend on Q and Q_n and their derivatives evaluated at $Z = \lambda_n$.

The expressions for Eq. (19) and Eq. (23) can be made computationally more amenable by observing that $Q_{,Z}(\lambda_n) = Q_n(\lambda_n)$. This can be expressed more generally as

$$Q^{(m)}(\lambda_n) = m\, Q_n^{(m-1)}(\lambda_n), \tag{24}$$

where the superscripts in parenthesis represent the number of derivatives with respect to Z. Aside from reducing the number of quantities to be calculated, this substitution increases the accuracy of $Q^{(m)}(Z)$ near multiple roots. This can be clearly seen in the case when $m = 1$, as

$$Q_{,Z}(Z) = Q_n(Z) + Q_{n,Z}(Z)\,(Z - \lambda_n). \tag{25}$$

Since Newton's method only determines Z within a fixed accuracy of the root λ_n, there is an error term which remains relatively constant near a double root. However, the expression $Q_n(Z)$ is approaching zero near a double root, and hence calculation of $Q_{,Z}$ is becoming increasingly inaccurate near a double root unless this error term is eliminated.

3 Numerical Implementation

The three roots λ_n can be calculated using a number of methods [3]. Although the fastest available general algorithm appears to be one based upon the QR eigenvalue method [7], we have found that Newton's method works better for this specific problem (a comparison of timing results can be found in Section 4.1). This algorithm can be used in conjunction with Horner's method [3] for efficient polynomial evaluation.

The polynomial coefficients are real, and it is known that none of the roots can be real (by the physical constraint that $\mathbf{K}$ must be invertible). Thus, the roots occur in complex conjugate pairs. After locating the first pair, these roots can be factored out of the polynomial to simplify the subsequent Newton iterations. Although the algorithm for

analytically solving a quartic equation could now be used, it is not likely to be faster than a second use of Newton's method. After factoring out the second pair of roots, the quadratic formula can be used to quickly identify the remaining solution.

We have found that $1+i$ and $-1+i$ are reasonably good starting points for each of the two uses of Newton's method. These implementations of Newton's method typically converge, error less than of 10^{-7}, in less than 10 iterations. However, in boundary integral calculations multiple Green's function evaluations are being carried out, usually with small changes in θ and ψ from one evaluation to the next. In this case, the locations of the previously found roots are clearly better initial values for Newton's method and can be used to achieve some additional efficiency. The use of these initial values roughly halves the computational time of finding the roots λ_n. Since the calculation of the roots typically takes around one third of the total time of calculating $\tilde{\mathbf{U}}$ and its derivatives, there is a noticeable increase in speed when this method is used. However, for consistency, all results given in this paper employ $1+i$ and $-1+i$ for the initial values.

Once the roots have been found, Eq. (15) and Eq. (19) can be used to calculate the tensor elements of $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{U}}_\alpha$. If the second derivatives are required, Eq. (23) can be used. In order to evaluate Eq. (19), expressions must be available for $P_{jk}(Z), P_{jk,\alpha}(Z), Q_{,\alpha}(Z)$ and their derivatives with respect to Z. (The derivatives of $Q(Z)$ with respect to Z are directly determined from the roots λ_n using Eq. (24).) All of these expressions (plus those needed for Eq. (23)) are low order polynomials of Z, and thus these functions can be conveniently stored in terms of the coefficients. This method has two principal advantages. First, the calculation of the coefficients can be accomplished using only real numbers, maximizing the amount of work done before introducing the complex number $Z=\lambda_n$. Second, with the coefficients available, Horner's method can be used to evaluate the derivatives without much additional work.

The generation of expressions for the coefficients of P_{jk}, $P_{jk,\alpha}$, $Q_{,\alpha}$, and optionally for $P_{jk,\alpha\beta}$ and $Q_{,\alpha\beta}$, as well as expressions for the coefficients of $Q(Z)$ (needed to calculate the roots λ_n) can be found using a symbolic manipulation program such as Maple. For optimal efficiency, these expressions can be generated for each material.

4 Calculations

In this section, the results of test calculations employing the materials silicon and tin are presented. Silicon is a cubic crystal, and is therefore specified by three elastic constants, whereas tin has tetragonal symmetry and six constants are required. The values of these constants are listed in Table 1. The values for silicon were taken from [11] and those for

tin from [16, p. 280].

	c_{11}	c_{12}	c_{44}	c_{33}	c_{13}	c_{66}
silicon	1.667	0.639	0.796	-	-	-
tin	0.735	0.234	0.22	0.87	0.28	0.2265

Table 1: Elastic constants for silicon and tin, in units of $10^{12} dynes/cm^2$.

To examine the accuracy of the residue method, the results were compared with essentially exact results obtained from a direct numerical quadrature. For $\tilde{U}$ the appropriate line integral is given by Eq. (4), while for first and second derivatives these expressions are [20]

$$\begin{aligned}
\tilde{\mathbf{U}}_\alpha &= -\oint_{S^1} \mathbf{K}^{-1}(\boldsymbol{\xi})\,\mathbf{K}_\alpha(\boldsymbol{\xi})\,\mathbf{K}^{-1}(\boldsymbol{\xi})\,ds(\boldsymbol{\xi}) \\
\tilde{\mathbf{U}}_{\alpha\beta} &= \oint_{S^1} \Big[\mathbf{K}^{-1}\,\mathbf{K}_\alpha\,\mathbf{K}^{-1}\,\mathbf{K}_\beta\,\mathbf{K}^{-1} + \\
&\qquad \mathbf{K}^{-1}\,\mathbf{K}_\beta\,\mathbf{K}^{-1}\,\mathbf{K}_\alpha\,\mathbf{K}^{-1} - \mathbf{K}^{-1}\,\mathbf{K}_{\alpha\beta}\,\mathbf{K}^{-1}\Big]\,ds(\boldsymbol{\xi}) \qquad (26)
\end{aligned}$$

The integrations were carried out using Simpson's rule, dividing the interval $[0, 2\pi]$ into 150 subintervals.

Figures 1-3 display the error for $\tilde{U}_{11}(\theta, \pi/4)$, $\tilde{U}_{11,\theta}(\theta, \pi/4)$, and $\tilde{U}_{11,\theta\theta}(\theta, \pi/4)$ for silicon. The accuracy of these calculations are clearly more than sufficient, as the values C_{jlkm} are typically only known to 3 significant digits. The small errors are to be expected, as Newton's method yielded the roots λ_n with a relative error less than 10^{-7}.

Figures 4-6 display corresponding results for tin, this time however as a function of ψ for a fixed value of θ. As tin is more anisotropic than silicon, it is encouraging that the approximations for $\tilde{U}$ and its derivatives remain highly accurate.

4.1 Computation Time

To be a practical algorithm, the residue approach must be at least competitive with Wilson/Cruse in regards to computational speed. The test calculations indicate that the new algorithm is in fact significantly faster than cubic interpolation from tables. This (possibly surprising) result is due to the fact that once the roots have been determined *all* function evaluations reduce to simple polynomial evaluation and proceed very quickly.

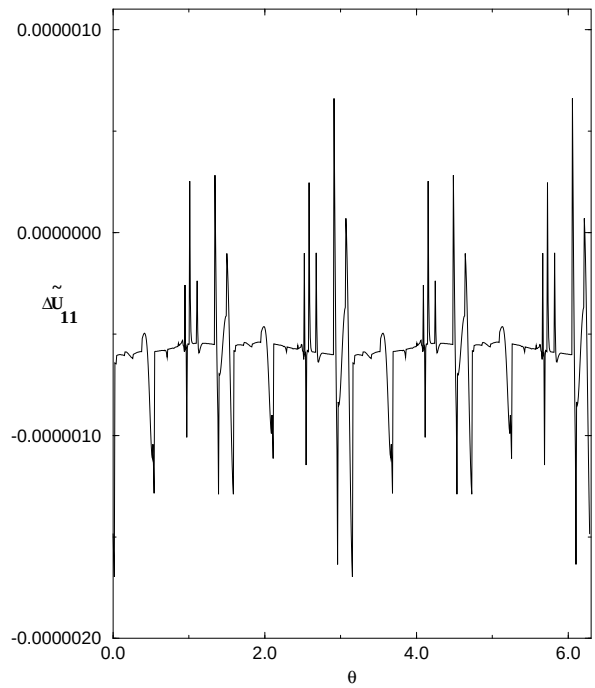

Figure 1: Error of $\tilde{U}_{11}(\theta, \pi/4)$ for silicon.

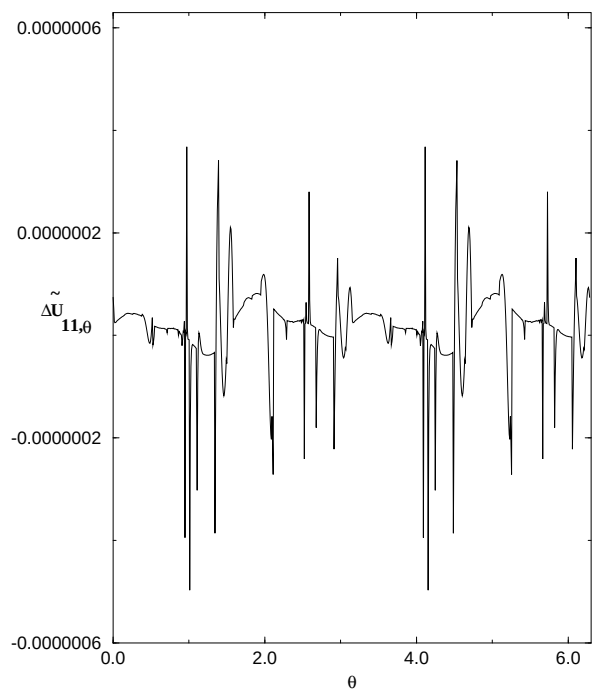

Figure 2: Error of $\tilde{U}_{11,\theta}(\theta, \pi/4)$ for silicon.

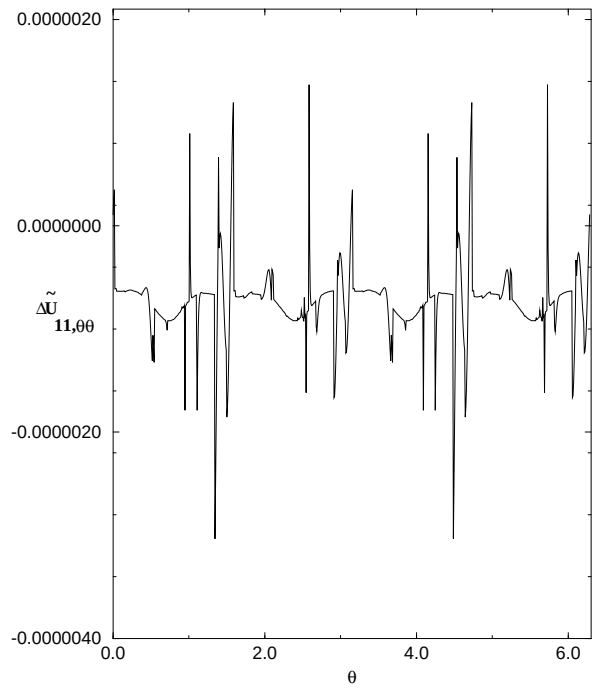

Figure 3: Error of $\tilde{U}_{11,\theta\theta}(\theta, \pi/4)$ for silicon.

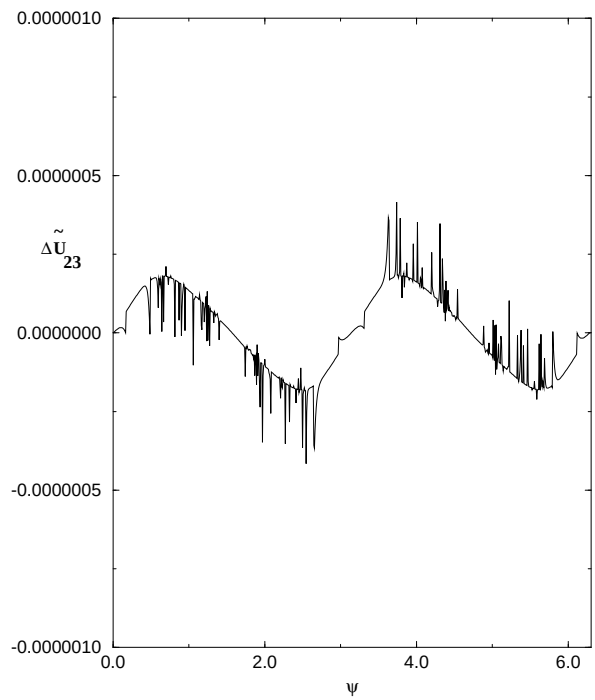

Figure 4: Error of $\tilde{U}_{23}(\pi/4, \psi)$ for tin.

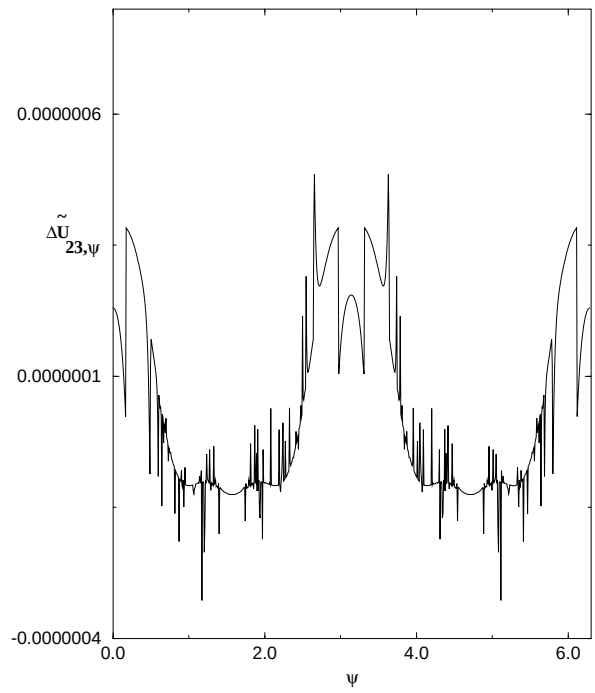

Figure 5: Error of $\tilde{U}_{23,\psi}(\pi/4, \psi)$ for tin.

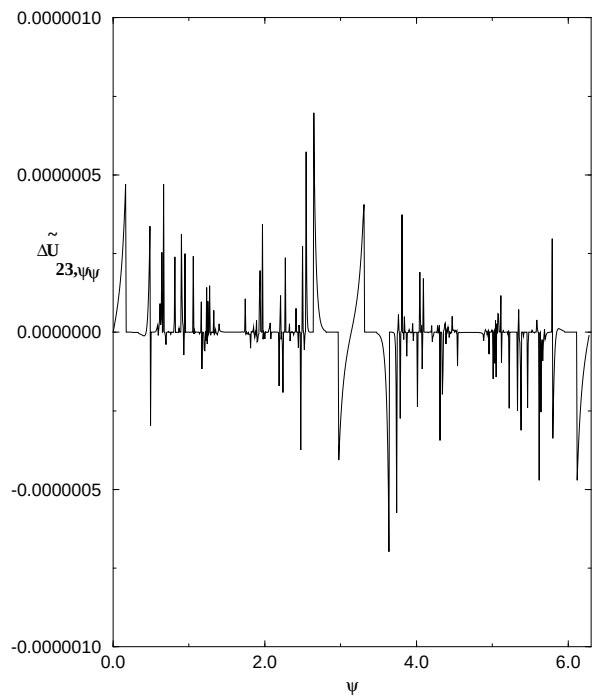

Figure 6: Error of $\tilde{U}_{23,\psi\psi}(\pi/4, \psi)$ for tin.

The timing results are shown in Table 2. The first set of numbers indicates the time needed to evaluate all 6 components of $\tilde{\mathbf{U}}$ plus the 12 values comprising of $\tilde{\mathbf{U}}_\alpha$. The second set is the time required if the 18 values of the second derivative $\tilde{\mathbf{U}}_{\alpha\beta}$ are also computed.

All calculations were performed on an IBM RS/6000-520 workstation, and the reported timings were obtained by reading the computer clock at the start and end of the calculation. As thus other activities by the operating system could interfere, the reported numbers are the minimum value obtained by running the calculation ten times. For the most part, the timing numbers fluctuated only slightly over the ten trials, on the order of 1%, indicating that the vagaries of the timing mechanism are not significant factors. However, with the residue calculation of tin the fluctuations were somewhat larger, probably due to variations in the rate of convergence of Newton's method. The two times for tin in Table 2 frequently increased to around 1×10^{-3} seconds and 1.9×10^{-3} seconds. The slightly longer time needed for tin, as compared to silicon, likely is a consequence of having fewer zeros in the C_{jlkm} tensor of elastic constants. Other materials may vary from these results, but probably not by a significant degree.

	$\tilde{\mathbf{U}}_\alpha(\theta, \psi)$	$\tilde{\mathbf{U}}_{\alpha\beta}(\theta, \psi)$
Residue (Silicon)	0.801×10^{-3}	1.746×10^{-3}
Residue (Tin)	0.857×10^{-3}	1.800×10^{-3}
Interpolation	3.156×10^{-3}	6.312×10^{-3}

Table 2: Time (in seconds) for evaluating all components of $\tilde{\mathbf{U}}(\theta, \psi)$ and $\tilde{\mathbf{U}}_\alpha(\theta, \psi)$. The last column is the time if all components of the second derivatives are also computed.

These timing results are for evaluating the functions at a single (θ, ψ) point. As discussed earlier, the time required by the residue algorithm, as part of a boundary element code, should be a little smaller than these values. This is due to having better initial approximations to the roots (from the previous evaluation), and thus fewer Newton iterations.

In general, the root finding algorithm of Goedecker [7] will be faster and more reliable than Newton's method. However, computational tests indicate that for the 6th degree polynomials which arise in this particular application, the QR scheme is roughly an order of magnitude slower than Newton's method. Table 3 lists some typical computation times required by the two approaches. The speed using Newton is due in part to the fact that only 5 to 8 iterations were generally required to reach convergence. In addition, convergence of the QR scheme may be slowed a little because all roots are complex.

Newton	QR
.0003067255	.0036886930
.0003061295	.0027679205
.0003062487	.0025386810
.0003072023	.0025392771
.0003248453	.0025418997
.0003420115	.0025440454
.0006994009	.0029073954
.0003420115	.0025415421
.0003415346	.0025422573
.0003414154	.0025383234
.0005424023	.0025502443

Table 3: Typical computation times (in seconds) for evaluating the roots of $Q(Z)$ with Newton's method and with Goedecker's QR algorithm.

4.2 Multiple Roots

The only potential concern of this method is the existence of multiple roots in the polynomial $Q(Z)$. The formulas used to calculate $\tilde{\mathbf{U}}$ and its derivatives are only valid under the assumption that the roots are simple. Moreover, Newton's method converges much slower at multiple roots, and thus the timing results presented above may not be applicable. In the materials examined, and probably in other materials as well, multiple roots only occur at a few isolated points (θ, ψ). As discussed below, this occurrence does not appear to be a significant problem, as the algorithm functions very well at a very small distance away from the multiple root. Nevertheless, methods for identifying and handling multiple roots warrants some further study.

The test calculations indicate that the residue algorithm produces sufficiently accurate results very close to a multiple root. For instance, there is a multiple root in $Q(Z)$ for tin at approximately $\theta = \pi/4$, $\psi = 1.9296$, but there is no significant error in the graphs (Figures 4-6) near this point. Further calculations show that the error does not become significant until the function is evaluated within less than 10^{-5} radians of the multiple root. Therefore, the values of the function within these small domains surrounding multiple roots can be accurately approximated by the value a small distance away. This will still produce Green's function values with more than acceptable accuracy. Though Newton's method will converge slower close to multiple roots, resulting in an increase in computation time, the results are accurate as long as the function is not evaluated too close to a multiple

root. One possible way for an algorithm to ensure this is to monitor the quantity

$$|\lambda_1 - \lambda_2||\lambda_1 - \lambda_3||\lambda_2 - \lambda_3|. \tag{27}$$

Whenever this quantity becomes less than around 0.0003, a nearby point $(\theta + \delta_1, \psi + \delta_2)$ can be used to evaluate the Green's function, or if more accuracy is needed, two points on opposite sides of the root can be evaluated, and the average can be used.

5 Conclusions

An algorithm based upon residue calculations has been shown to be a fast and accurate technique for evaluating the anisotropic Green's function and its derivatives. The computational tests indicate that this algorithm runs three to four times as fast as the standard Wilson/Cruse cubic interpolation method. Moreover, this approach avoids the need for massive tables, and the accuracy and speed of the algorithm is not dependent on the anisotropy of the material. The only potential difficulty arises from the question of multiple roots, wherein the residue is not a simple pole. Although the test calculations indicate that multiple roots do not pose a significant numerical problem, it is likely that our scheme for this situation can be improved.

Acknowledgments

This research was supported by the Laboratory Directed Research and Development Program of the Oak Ridge National Laboratory, and by the Applied Mathematical Sciences Research Program of the Office of Mathematical, Information, and Computational Sciences, U.S. Department of Energy, under contract DE-AC05-96OR22464 with Lockheed Martin Energy Research Corp. The authors are grateful to Professors L. Reichel and T. A. Cruse for pointing out Goedecker's method and reference [15], respectively.

References

[1] D. M. Barnett, *The precise evaluation of derivatives of the anisotropic elastic Green's functions*, Phys. Stat. Sol. (b), 49 (1972), pp. 741–748.

[2] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel, *Boundary Element Techniques*, Springer-Verlag, Berlin and New York, 1984.

[3] R. L. Burden and J. D. Faires, *Numerical analysis*, PWS-Kent Publishing, Boston, 5 ed., 1989.

[4] T. A. Cruse, *Boundary Element Analysis in Computational Fracture Mechanics*, Kluwer Academic Publishers, Boston, 1988.

[5] A. Deb, J. D. P. Henry, and R. B. Wilson, *Alternate BEM formulations for 2- and 3-d thermoelasticity*, Int. J. Solids Structures, 27 (1991), pp. 1721–1738.

[6] P. H. Dederichs and G. Liebfried, *Elastic Green's function for anisotropic cubic crystals*, Phys. Rev., 188 (1969), pp. 1175–1183.

[7] S. Goedecker, *Remark on algorithms to find roots of polynomials*, SIAM J. Sci. Comput., 15 (1994), pp. 1059–1063.

[8] L. J. Gray, D. Ghosh, and T. Kaplan, *Evaluation of the anisotropic Green's Function in three dimensional elasticity*, Computational Mechanics, 17 (1996), pp. 255–261.

[9] L. J. Gray, L. F. Martha, and A. R. Ingraffea, *Hypersingular integrals in boundary element fracture analysis*, Int. J. Numer. Meth. Engrg., 29 (1990), pp. 1135–1158.

[10] F. Hartmann, C. Katz, and B. Protopsaltis, *Boundary elements and symmetry*, Ingenieur-Archiv, 55 (1985), pp. 440–449.

[11] H. B. Huntington, *The elastic constants of crystals*, in Solid State Physics, Advances in Research and Applications, Vol. 7, F. Seitz and D. Turnbull, eds., Academic Press, 1958, pp. 213–351.

[12] I. M. Lifschitz and L. N. Rozentsweig, Zh. Eksperim. Teor. Fiz., 17 (1947), p. 783.

[13] G. Maier, G. Novati, and S. Sirtori, *On symmetrization in boundary element elastic and elastoplastic analysis*, in Discretization methods in structural mechanics, G. Kuhn and H. Mang, eds., Springer-Verlag, Berlin and New York, 1990, pp. 191–200.

[14] P. W. Partridge, C. A. Brebbia, and L. C. Wrobel, *The Dual Reciprocity Boundary Element Method*, Computational Mechanics Publications, Southampton and Boston, 1992.

[15] N. A. Schclar, *Anisotropic Analysis using Boundary Elements*, Computational Mechanics Publications, Southampton and Boston, 1994.

[16] F. Seitz and D. Turnbull, eds., *Solid State Physics*, vol. 7, Academic Press, New York, 1958.

[17] S. M. Vogel and F. J. Rizzo, *An integral equation formulation of three-dimensional anisotropic elastostatic boundary value problems*, J. Elast., 3 (1973), pp. 203–216.

[18] C.-Y. Wang, *Two-dimensional elastic Green's functions for general anisotropic solids and generalization of Stroh's formalism*, Int. J. Solids Structures, 31 (1994), pp. 2591–2597.

[19] ——, *Green's tensors for solids of general anisotropy*, in Boundary Elements XVII, C. A. Brebbia, ed., Southampton, 1995, Computational Mechanics.

[20] R. B. Wilson and T. A. Cruse, *Efficient implementation of anisotropic three dimensional boundary-integral equation stress analysis*, Int. J. Numer. Meth. Engrg., 12 (1978), pp. 1383–1397.