GEOPHYSICS, VOL. 67, NO. 2 (MARCH-APRIL 2002); P. 525–535, 8 FIGS.
10.1190/1.1468613

Multiple-source Euler deconvolution

R. O. Hansen* and Laura Suciu‡

ABSTRACT

Rapid three-dimensional (3-D) source location methods can be extremely useful in framing a subsurface structural model from gravity or magnetic data. However, existing implementations of Euler deconvolution are limited to a single source in each window. This can be a significant limitation in areas of complex structure. We have generalized the method to the multiple-source case, and implemented the 3-D algorithm. Results from synthetic data and from the Gold Acres mining district in Nevada suggest that the new algorithm can be a useful interpretive tool.

INTRODUCTION

Since its introduction by Reid et al. (1990), three-dimensional (3-D) Euler deconvolution has gained considerable popularity as a source location tool. It is one of the few fully 3-D methods that yields rapid and useful results in a wide variety of geological settings. This is primarily because the method parameterizes the inverse problem so that it becomes linear, a characteristic which it shares with Werner deconvolution (Hartman et al., 1971; Hansen and Simmonds, 1993) and CompuDepth (O'Brien, 1971). However, these latter algorithms are two-dimensional.

One of the limitations of the Euler method is that it is a single-source technique. A 3-D multiple-source algorithm has been developed by Wang and Hansen (1990) and applied successfully by Arndt et al. (1994), but this method is intrinsically more complex than the Euler method. A multiple-source generalization of the Euler algorithm, therefore, would be useful. One approach to this has been given by Stavrev (1997). We show a new, robust approach to making the Euler algorithm more useful.

THE EULER METHOD

A field f measured at the point (x, y, z) is said to satisfy Euler's equation with structural index n if

$$(x - x_0)\frac{\partial f}{\partial x} + (y - y_0)\frac{\partial f}{\partial y} + (z - z_0)\frac{\partial f}{\partial z} = -nf \qquad (1)$$

for some choice of constants (x_0, y_0, z_0) and n, which is often but not always restricted to be an integer. In particular, if

$$f = \frac{A}{R^n} \qquad (2)$$

where A is a constant and $R = \sqrt{(x-x_0)^2+(y-y_0)^2+(z-z_0)^2}$, then f satisfies Euler's equation with structural index n. For $n = 3$, f can be the magnetic field of a point dipole; for $n = 2$, the field of a magnetic rod; for $n = 1$, the field of a thin sheet source; and Reid et al. (1990) showed that dipping contacts satisfy a variant of equation (1), discussed in Appendix B, with $n = 0$.

Because equation (1) is linear in the variables (x_0, y_0, z_0) to be determined, it can be used as a tool for rapid source location. Rearranging equation (1) so that the terms containing only known or specified quantities are moved to the right-hand side yields

$$x_0\frac{\partial f}{\partial x} + y_0\frac{\partial f}{\partial y} + z_0\frac{\partial f}{\partial z} = nf + x\frac{\partial f}{\partial x} + y\frac{\partial f}{\partial y} + z\frac{\partial f}{\partial z}. \qquad (3)$$

This now can be seen to be a linear equation in the variables (x_0, y_0, z_0). By evaluating the right-hand side and the coefficients on the left-hand side for at least three points, a system of linear equations can be obtained which can be solved for (x_0, y_0, z_0). Thus, Euler's equation leads to a source location algorithm (Hood, 1965; Thompson, 1982; Barongo, 1984; Reid et al., 1990). Notice that this algorithm does not depend on f satisfying Laplace's equation, although it is almost always used for potential fields. The Euler algorithm is comparable to Werner deconvolution and CompuDepth, but more general in that it can handle a variety of models with different indices.

The question addressed in this paper is whether the Euler method can also be extended to multiple bodies. Since there is essentially no difference between the 2-D and 3-D cases in the one-body algorithm, we set out from the beginning to develop a 3-D multiple-source algorithm. The 2-D case has been partially investigated; we give the results in Appendix A. The complete theory for the 3-D case is derived in the following section.

Manuscript received by the Editor March 23, 2000; revised manuscript received May 11, 2001.
*Pearson, de Ridder, and Johnson, Inc., 12640 West Cedar Drive, Suite 100, Lakewood, Colorado 80228. E-mail: rohansen@prj.com.
‡Formerly Bell Geospace, Inc., Lamont-Doherty Earth Observatory, Columbia University, Palisades, New York 10964; presently Target Health Inc., 305 Madison Avenue, 25th Floor, New York, New York 10165. E-mail: lsuciu@targethealth.com.

THEORY

Several alternative approaches were considered for the theoretical development. The main difficulty in extending the 2-D algorithm sketched in Appendix A to three dimensions is generalizing the complex polynomial equation (A-11) which determines the source depths and locations once the generalized Euler equation (A-10) has been solved. This step uses complex variables (and thus two-dimensionality) in an essential way. One possibility is to use a series of 2-D slices, as in Wang and Hansen (1990). A solution along those lines appears possible, but this slicing was one of the least satisfactory aspects of the earlier algorithm. Quaternionic formulations were also investigated, but no workable algorithm using quaternions was found.

Ultimately, standard tensor methods were adopted. The development which follows assumes some familiarity with Cartesian tensor notation, such as that in Hadsell (1995). However, we will develop the theory for the two-source case step-by-step first. The general case, for which a detailed development would be far too lengthy, will then be given in abbreviated form.

We write the coordinates (x, y, z) in the form

$$R_1 = x, \tag{4}$$

$$R_2 = y, \tag{5}$$

$$R_3 = z, \tag{6}$$

and refer to the ith coordinate as R_i. Partial derivatives are written in the form

$$\nabla_1 = \frac{\partial}{\partial x}, \tag{7}$$

$$\nabla_2 = \frac{\partial}{\partial y}, \tag{8}$$

$$\nabla_3 = \frac{\partial}{\partial z}. \tag{9}$$

The summation convention is used throughout, so that if A_i and B_i are the components of vectors,

$$A_i B_i = A_1 B_1 + A_2 B_2 + A_3 B_3 \tag{10}$$

is the inner product of the two vectors. Then, the single-source Euler equation can be written

$$(R_i - R_{0i})\nabla_i f + nf = 0, \tag{11}$$

where R_{0i} is the ith component of the source location.

For two sources which each satisfy an Euler equation with structural index n, we have

$$(R_i - R_{1i})\nabla_i f_1 + nf_1 = 0, \tag{12}$$

$$(R_i - R_{2i})\nabla_i f_2 + nf_2 = 0, \tag{13}$$

but of course, only the superposition $f = f_1 + f_2$ can be measured. The objective is to derive an equation involving only f from which R_{1i} and R_{2i} can be extracted. Note that all sources must have the same structural index; this is apparently a fundamental limitation of the algorithm.

Differentiating equations (12) and (13), contracting the first resulting equation with $(R_i - R_{2i})$ and the second with $(R_i - R_{1i})$, adding, substituting $f - f_2$ for f_1 and $f - f_1$ for f_2, and using the equations again to eliminate terms not involving $f_1 + f_2$ in combination yields

$$(R_i - R_{1i})(R_j - R_{2j})\nabla_i\nabla_j f + (n+1) \times (R_i - R_{1i} + R_i - R_{2i})\nabla_i f + n(n+1)f = 0. \tag{14}$$

Collecting terms of order two, one, and zero in R_{1i} and R_{2i} gives

$$R_{1i}R_{2j}\nabla_i\nabla_j f - (R_{1i} + R_{2i})[R_j\nabla_i\nabla_j f + (n+1)\nabla_i f] + R_i R_j \nabla_i\nabla_j f + 2(n+1)R_i\nabla_i f + n(n+1)f = 0. \tag{15}$$

We observe that because f satisfies Laplace's equation

$$\nabla_i\nabla_i f = 0, \tag{16}$$

and because partial derivatives commute, so that $\nabla_i\nabla_j f = \nabla_j\nabla_i f$, only the symmetric, trace-free part of $R_{1i}R_{2j}$ can be determined from equation (15):

$$R_{1i}R_{2j}\nabla_i\nabla_j f = \left(R_{1(i}R_{2j)} - \frac{1}{3}\delta_{ij}R_{1k}R_{2k}\right)\nabla_i\nabla_j f, \tag{17}$$

where the parentheses on indices indicate symmetrization:

$$R_{1(i}R_{2j)} = \frac{1}{2}(R_{1i}R_{2j} + R_{1j}R_{2i}) \tag{18}$$

and δ_{ij} is the unit or identity tensor, defined by

$$\begin{aligned} \delta_{ij} &= 1, \quad i = j, \\ &= 0, \quad i \neq j, \end{aligned} \tag{19}$$

so the trace over i and j vanishes:

$$R_{1i}R_{2i} - \frac{1}{3}\delta_{ii}R_{1k}R_{2k} = R_{1i}R_{2i} - \frac{1}{3}\cdot 3R_{1k}R_{2k} = 0. \tag{20}$$

Then, the Euler equation for two sources is

$$\left(R_{1i}R_{2j} - \frac{1}{3}\delta_{ij}R_{1k}R_{2k}\right)\nabla_i\nabla_j f - (R_{1i} + R_{2i}) \times [R_j\nabla_i\nabla_j f + (n+1)\nabla_i f] + R_i R_j\nabla_i\nabla_j f + 2(n+1)R_i\nabla_i f + n(n+1)f = 0. \tag{21}$$

In principle, it would be possible to solve equation (21) directly for R_{1i} and R_{2i} using some type of nonlinear optimization scheme. However, drawing on experience from CompuDepth, multiple-source Werner deconvolution, and the frequency-domain inversion of Wang and Hansen (1990), we are led to seek instead a two-stage inverse scheme in which a set of parameters are fit to the data by a linear least-squares problem, followed by a nonlinear step in which the coordinates are determined from the parameters determined from the linear fit. This strategy has a tremendous advantage if the nonlinear problem to be solved is comparatively "tame," as general nonlinear optimization problems are notoriously difficult to solve. Thus, the success of the two-stage procedure relies heavily on finding an appropriate way of extracting the coordinate locations from the linear fit parameters.

Define

$$A_{1i} = -(R_{1i} + R_{2i}) \tag{22}$$

$$A_{2ij} = R_{1(i}R_{2j)} - \frac{1}{3}\delta_{ij}R_{1k}R_{2k}. \tag{23}$$

Then, the two-source Euler equation takes the form

$$A_{2ij}\nabla_i\nabla_j f + A_{1i}[R_j\nabla_i\nabla_j f + (n+1)\nabla_i f] + R_iR_j\nabla_i\nabla_j f + 2(n+1)R_i\nabla_i f + n(n+1)f = 0. \tag{24}$$

This is a linear system in the unknown tensor coefficients A_{1i} and A_{2ij}, which can be solved by standard least squares methods to be discussed in the "Implementation" section.

Equations (22) and (23) must now be solved for R_{1i} and R_{2i} using the solutions A_{1i} and A_{2ij} just determined. Considered as a system of equations for R_{1i} and R_{2i}, equations (22) and (23) form an overdetermined system—there are 8 independent equations in the 6 unknown coordinates. Furthermore, the system is nonlinear. The problem, then, is to find a way of reducing the system to a tractable one.

Again, guided by experience from other multiple-source algorithms, we seek to combine equations (22) and (23) into a single system. It can be verified by direct substitution that for each i, j, R_{1i} and R_{2i} are the roots of the following equation:

$$a_{ij} = R_iR_j - \frac{1}{3}\delta_{ij}R_kR_k + A_{1(i}R_{j)} - \frac{1}{3}\delta_{ij}A_{1k}R_k + A_{2ij} = 0. \tag{25}$$

This equation is a tensor generalization of the polynomial equation (A-9) of Appendix A. The equation is overdetermined—there are 5 independent equations in the 3 unknown coordinates. Since A_{1i} and A_{2ij} are not exactly known, there will generally be no solutions. However, it is possible to minimize $a_{ij}a_{ij}$. The question is whether this yields the solutions we want.

To determine the minima of $a_{ij}a_{ij}$, note that if z_{ij} is any invertible matrix, then

$$a_{ij}a_{ij} = z^{-1}_{ik}a'_{kl}z_{lm}z^{-1}_{mj}a'_{nj}z_{ni}. \tag{26}$$

Since a_{ij} is symmetric, we can choose z_{ij} to diagonalize a'_{ij} (by similarity transformations). Then,

$$a_{ij}a_{ij} = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \tag{27}$$

where $\{\lambda_1, \lambda_2, \lambda_3\}$ are the eigenvalues of a_{ij}. Thus $a_{ij}a_{ij}$ vanishes only if $a_{ij} = 0$, so the minima of $a_{ij}a_{ij}$ are roots of $a_{ij} = 0$ and therefore correspond to the source locations and depths as required.

We now turn to the general case of m sources. It turns out to be convenient to work with the quantities $a^{(m)}_{ki_1\cdots i_k}$ defined recursively by

$$a^{(0)}_0 = 1, \tag{28}$$

$$a^{(1)}_0 = 1, \tag{29}$$

$$a^{(1)}_{1i_1} = R_{i_1} - R_{1i_1}, \tag{30}$$

$$a^{(m)}_{ki_1\cdots i_k} = a^{(m-1)}_{ki_1\cdots i_k} + \mathcal{C}[(R_{i_1} - R_{mi_1})a^{(m-1)}_{k-1i_2\cdots i_k}] \tag{31}$$

where $\mathcal{C}[\cdots]$ stands for the totally symmetric, trace-free part of the object in brackets (Perng, 1999). To obtain this quantity, we proceed in two steps: first, we calculate the totally symmetric part of the tensor, and then we remove all of the trace terms from the symmetrized tensor. If $\rho_{i_1\cdots i_m}$ is an arbitrary tensor of rank m, its totally symmetric part is

$$\sigma_{i_1\cdots i_m} = \rho_{(i_1\cdots i_m)} = \frac{1}{m!}\sum_{k=1}^{m!}\rho_{\mathcal{P}_k\{i_1\cdots i_m\}}, \tag{32}$$

where $\mathcal{P}_k\{i_1\cdots i_m\}$ is the kth permutation of the indices $i_1\cdots i_m$; and its totally symmetric, trace-free part is

$$\mathcal{C}[\rho_{i_1\cdots i_m}] = \sum_{k=0}^{[m/2]}(-1)^k\frac{(d+2(m-2-k))!!}{(d+2(m-2))!!} \times\frac{m!}{2^k k!(m-2k)!}\delta_{(i_1i_2}\cdots\delta_{i_{2k-1}i_{2k}}\sigma_{i_{2k+1}\cdots i_m)j_1j_1\cdots j_kj_k}, \tag{33}$$

where $[m/2]$ is the largest integer less than or equal to $m/2$, d is the dimension (here $d = 3$), and $q!! = q(q-2)\cdots 1$.

Then,

$$A^{(m)}_{ki_1\cdots i_k} = a^{(m)}_{ki_1\cdots i_k}\big|_{R_j=0}, \tag{34}$$

and the multiple-source Euler equation for m sources takes the form

$$\sum_{k=0}^{m}\frac{(n+m-1)!}{(n+k-1)!}a^{(m)}_{ki_1\cdots i_k}\nabla_{i_1}\cdots\nabla_{i_k}f = 0 \tag{35}$$

where, as usual, n is the structural index. The inductive proof of equation (35), which is straightforward but rather tedious, is in Appendix B.

As it stands, equation (35) is not directly useful; it needs to be rewritten in terms of the constant coefficients. This results in

$$a^{(m)}_{ki_1\cdots i_k} = \sum_{j=0}^{k}\frac{(m-j)!}{(k-j)!(m-k)!}\mathcal{C}[A^{(m)}_{ji_1\cdots i_j}R_{i_{j+1}}\cdots R_{i_k}]. \tag{36}$$

Thus the kth term of equation (35) can be written

$$\sum_{j=0}^{k}\frac{(n+m-1)!}{(n+k-1)!}\frac{(m-j)!}{(k-j)!(m-k)!}\mathcal{C}[A^{(m)}_{ji_1\cdots i_j}R_{i_{j+1}}\cdots R_{i_k}]. \tag{37}$$

Writing the required equation now involves only collecting the terms in $A^{(m)}_{ji_1\cdots i_j}$.

We now have the overdetermined but linear equations for the coefficients $A^{(m)}_{ji_1\cdots i_j}$; it only remains to write the equations which determine R_{ki} from the coefficients. Fortunately this is a straightforward generalization of equation (25):

$$a^{(m)}_{mi_1\cdots i_m} = \sum_{k=0}^{m}\mathcal{C}[A^{(m)}_{ki_1\cdots i_k}R_{i_{k+1}}\cdots R_{i_m}] = 0. \tag{38}$$

Equation (38) is again overdetermined, but we can form $a^{(m)}_{mi_1\cdots i_m}a^{(m)}_{mi_1\cdots i_m}$ and minimize the resulting scalar expression; as before, the minima occur at the values corresponding to the R_{ki}. Finding all of the minima of a highly nonlinear function is not, in general, a tractable problem. Fortunately, in this case a procedure analogous to finding the roots of a complex polynomial is possible. This procedure is outlined in Appendix C.

IMPLEMENTATION

The multiple-source Euler algorithm is implemented in two programs, analogous to the implementation of the

frequency-domain inversion by Wang and Hansen (1990). In the first program, the required derivatives of the field are computed in a batch process using frequency-domain techniques. Although customized to the multiple-source Euler algorithm, the methods are well-known (see, for example, Hildenbrand, 1983).

The second program performs the bulk of the calculation. First, the user selects a data window to be analyzed, the number of sources for that window, and an Euler index n for the sources in the window. Using the precomputed field derivatives, the necessary coefficients for equation (35) are computed. Because the $A_{i_1 \ldots i_k}$ are symmetric and trace free, only $2k+1$ of them are independent, so some care is required in collecting terms to form the coefficients. Once this is done, however, the linear system can be solved readily using standard techniques such as QR decomposition (Dongarra et al., 1979). Error estimates are obtained at the same time from the data misfit and covariance.

The minimization of $a_{mi_1 \ldots i_m} a_{mi_1 \ldots i_m}$ is easily seen to be a standard nonlinear least-squares problem in which the vector to be minimized is the vector of independent components of $a_{i_1 \ldots i_m}$. However, written in this form, the inner product is in a nonorthogonal basis, and so cannot be solved by standard minimization software. Modified Gram-Schmidt orthogonalization (Lawson and Hanson, 1974) is therefore used to write the solution vector in an orthogonal basis.

At first, we attempted to use the nonlinear minimization package NL2SOL (Dennis et al., 1981a, b) to locate the minima. Although this package performed well in finding individual solutions, a problem frequently arose in determining all roots. Successive solutions were obtained by a process analogous to deflation of a polynomial. Usually this worked well, but in certain classes of geometries one or a few solutions were strongly dominant, so that no matter what initial values were used in NL2SOL these solutions would almost always emerge as the final answer. An example is a tetrahedron (Figure 1) with one shallow vertex and three deep ones. In this case, it is not surprising that the shallow vertex is a dominant solution. An alternative approach, tailored to the exact problem at hand, was therefore developed.

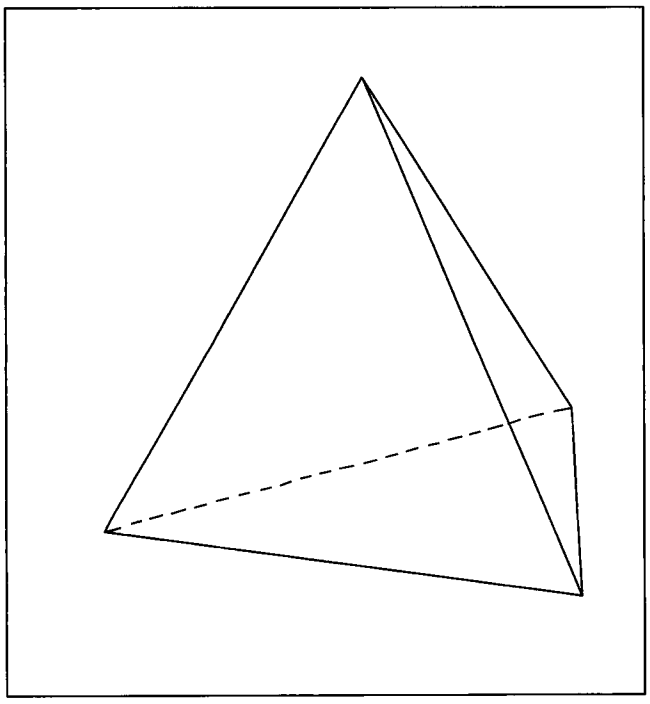

FIG. 1. Tetrahedral model. Depth to top is 5 units; depth to bottom is 15 units.

Our new approach is based on the observation that equation (36) is an overdetermined analog of the problem of finding the roots of a polynomial. Positive experience with the Durand-Kerner method (Aberth, 1973) in multiple-source Werner deconvolution (Hansen and Simmonds, 1993) suggested developing a least-squares generalization of this technique. Because the Durand-Kerner method finds all solutions of the system of equations simultaneously, the "root collapse" problem which arose with NL2SOL cannot occur. In fact, the technique forces the solutions of the system of equations to separate, which is precisely what is desired in this situation.

Although straightforward, the generalized Durand-Kerner approach is, to our knowledge, original and therefore requires a detailed explanation, which is found in Appendix C.

By calculating the pseudoinverse of the least-squares matrix at the final solution points, the error covariance obtained from the least-squares fit can be propagated through the nonlinear solution to obtain a final error covariance for each vertex coordinate. This step is conceptually straightforward. Note that the misfit variance in the nonlinear step is ignored. Generally, this omission is justified as the part of the variance arising from the nonlinear minimization is expected to be much smaller than the data misfit variance.

The output for each window, then, consists of a set of 3-D source locations together with error estimates for each of these locations. The output can be viewed in a variety of ways, including 3-D perspective plots, traditional posted plots, or cross-sections.

Our implementation allows for a maximum window size of 100×100, which is large enough for any practical application, and a maximum number of sources per window of six. In practice, the maximum usable number of sources per window is five because the computational overhead rises very rapidly as a function of the number of sources. Typically, the numerical stability is also poor in the six-source case, except for noiseless synthetic data.

PROGRAM USAGE

It is our experience that learning to use a source location algorithm optimally is a process which requires extensive experimentation. Thus, any recommendations we make at this stage of the development are necessarily provisional. Nevertheless, we will describe the way in which we have been using our implementation to date.

Because the algorithm is a multiple-source scheme, we chose large window sizes. Our usual practice has been to plot the analytic signal of the data to determine a first approximation to the source boundaries, and to lay out rectangular regions whose boundaries roughly coincide with the minima of the analytic signal. These are variable in size, but almost always much larger than is typical for a single-source Euler analysis. Roughly speaking, the windows are large enough to enclose a source body.

We almost always set the structural index to zero because the software is normally used to define the shape of three-dimensional bodies. The exception occurs when the sources in the window are believed to be thin sheets, in which case we use a structural index of one. No larger structural indices have been used except for testing purposes. Presumably it would be appropriate to use a structural index of two for locating kimberlite pipes, and three for unexploded ordnance, but we have not yet had the opportunity to try these on real data sets.

Our program usage is radically different from accepted practice for the single source algorithm (cf. Reid et al., 1990). This makes it complicated to compare our algorithm with the single-source algorithm as conventionally used. There is no reason that our algorithm could not be used in a manner similar to the usual practice for the single-source algorithm. That we have not done so reflects our own philosophy of program usage, which may evolve as we gain further experience.

RESULTS

Synthetic data

A large number of tests were performed on synthetic data to verify the correctness of the algorithm and its implementation, and to gain insight into the way the software should be used to obtain optimal results on real data. This last issue is particularly important: the way the program is used can influence the results very strongly. In this respect, the multiple-source Euler algorithm is similar to most other semi-automatic source location methods we have encountered.

We will restrict ourselves to a single example which, though not closely related to a typical geological problem, clearly illustrates the strengths of the algorithm. Figure 1 shows the source, a tetrahedron whose top is located at a depth of 5 units and whose base is at a depth of 15 units. Since we are considering the magnetic anomaly of this body, the units are irrelevant. Figure 2 shows the corresponding magnetic anomaly at 90° inclination, calculated on a 100×100 one-unit grid.

Because the lateral dimensions of this body are comparable to its depth extent, it would be virtually impossible for a single-source depth estimation algorithm to locate the bottom corners of the tetrahedron correctly. However, the generalized Durand-Kerner scheme forces the roots to be separated, and the version using this technique has little difficulty with the model.

The results from the multiple-source Euler algorithm, applied to a 50×50 unit window extracted from the center of the grid, are shown in Figure 3. These results are from a single four-source calculation with structural index zero. The horizontal locations of all vertices are essentially exact, and so is the depth to the top vertex. The estimated depths to the bottom vertices are 2–3% too shallow, which is a typical error under optimum conditions of the type shown here.

The depth estimates for the depth to the bottom vertices degrade rapidly with the addition of noise; in the presence of 5% noise, the bottom vertices cannot be distinguished. This result is to be expected for any source location algorithm. The moral is that high-quality data is always required to obtain depth-to-bottom estimates with any reliability.

Gold Acres

Figure 4 shows the location of the Gold Acres district, Lander County, north-central Nevada. Figure 5 is a total magnetic intensity map of the district. The area is underlain by a quartz monzonite stock, which is the source of the large magnetic anomaly in the center of the map. Various volcanic flows at depths ranging from the surface to several hundred feet, and of various ages, are represented by magnetic anomalies scattered

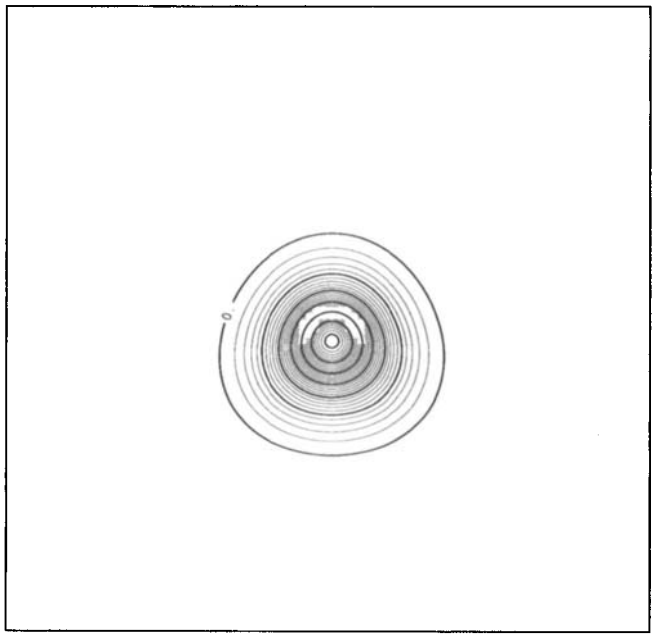

FIG. 2. Map of magnetic anomaly at 90° inclination for the tetrahedral model, calculated on a 100×100 one-unit grid. Contour interval is 1 nT.

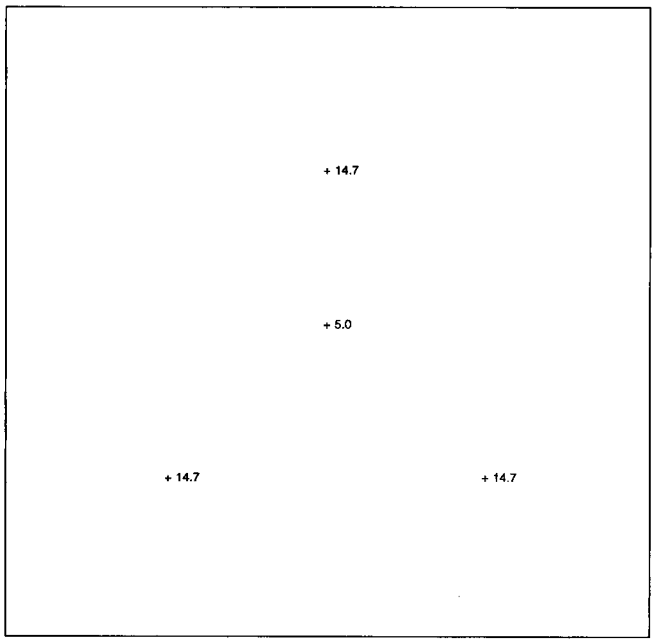

FIG. 3. Depth estimates obtained using the multiple-source Euler program on the tetrahedral model of Figure 1.

over the map. For example, it is apparent that the flows just to the southwest of the stock are reversely magnetized. Thus, the interpretation represents sufficient complexity to be challenging. Because there are active gold mines in the district, identifying the magnetic features accompanying the mineralization is a matter of considerable economic interest.

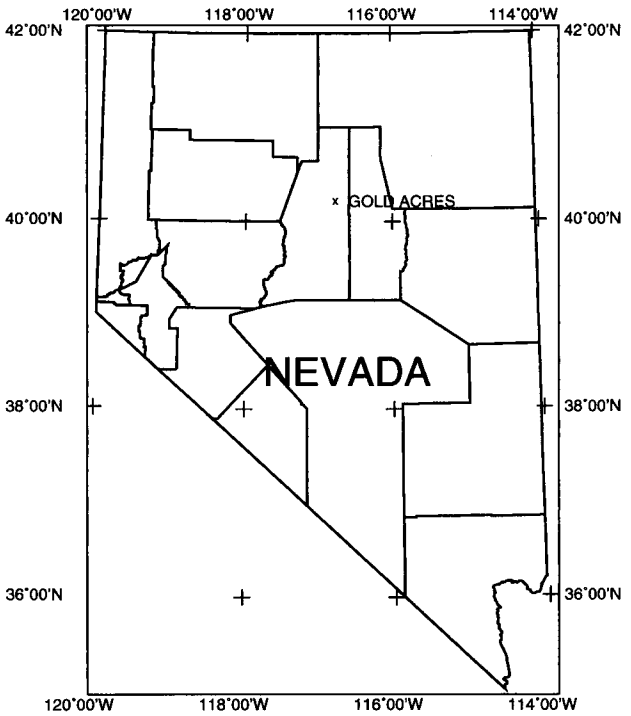

FIG. 4. Index map showing the general location of the Gold Acres district.

We have not attempted a comprehensive interpretation of the data. The results we present are only intended to show that the algorithm would probably be useful as part of such an interpretation.

The analysis was divided into two parts: mapping of the volcanic flows and their associated structures, and finding the depth and configuration of the stock. Because of the spatial extent and complexity of the volcanic flows, no source location scheme could be expected to isolate the stock without some enhancement of the longer wavelengths. Therefore, the data were Wiener filtered to suppress the anomalies mostly due to the shallow volcanics. Figure 6 shows the filtered data used in this part of the analysis.

For both sets of anomalies, windows for analysis were selected using the minima associated with the 3-D analytic signal, as discussed in the "Program Usage" section. Thus, the window sizes varied across the map. A structural index of one was used in analyzing the shallow volcanics, which are effectively thin sheets; a structural index of zero was used for the long-wavelength analysis. The number of sources in each window was selected by examining the goodness of fit, variance, and scatter in the solutions. These selection criteria varied from window to window.

The results are shown in Figures 7 and 8. Here the solutions have merely been gridded and contoured; artifacts of this treatment are obvious, and we are working on improved displays. However, several features are clear. First, there is significant depth structure to the shallow volcanic flows. For example, the areas southwest and northeast of the stock are correctly mapped as extremely shallow. Interestingly, the shallow area mapped at the southeast end of the stock coincides almost exactly with the Gold Acres mining district.

The depths obtained from the Wiener-filtered data show an elongated feature dipping smoothly to the east-southeast. The general configuration and depths agree well with the known

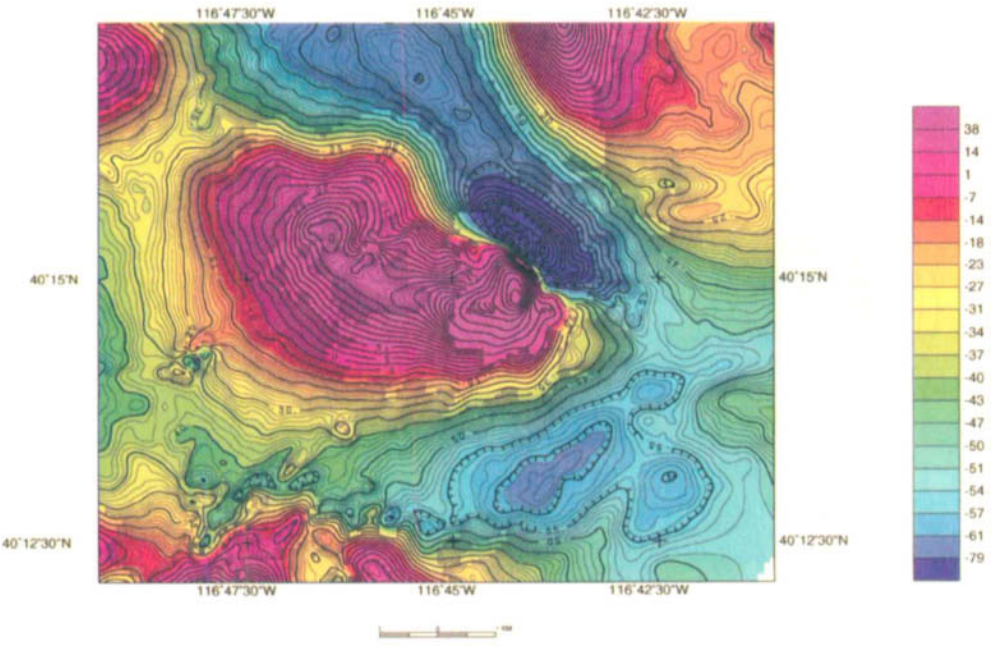

FIG. 5. Total magnetic intensity of the Gold Acres area. Contour interval is 1 nT.

characteristics of the stock. Thus, the multiple-source Euler method appears to have correctly determined most of the significant features of the area.

DISCUSSION

Despite the apparent complication in the algorithm due to the presence of tensor indices, this generalization of the Euler method to any number of sources is more straightforward than the generalization of CompuDepth to three dimensions by Wang and Hansen (1990). For example, there are no transformations to the frequency domain, the overdetermination in the nonlinear system giving the body coordinates from the coefficients of the linear system can be handled in a natural way, and so on. In these respects, the algorithm is much like the multiple-source Werner scheme (Hansen and Simmonds, 1993), but generalizes to three dimensions. Furthermore, the additional term introduced by Reid et al. (1990) to handle the

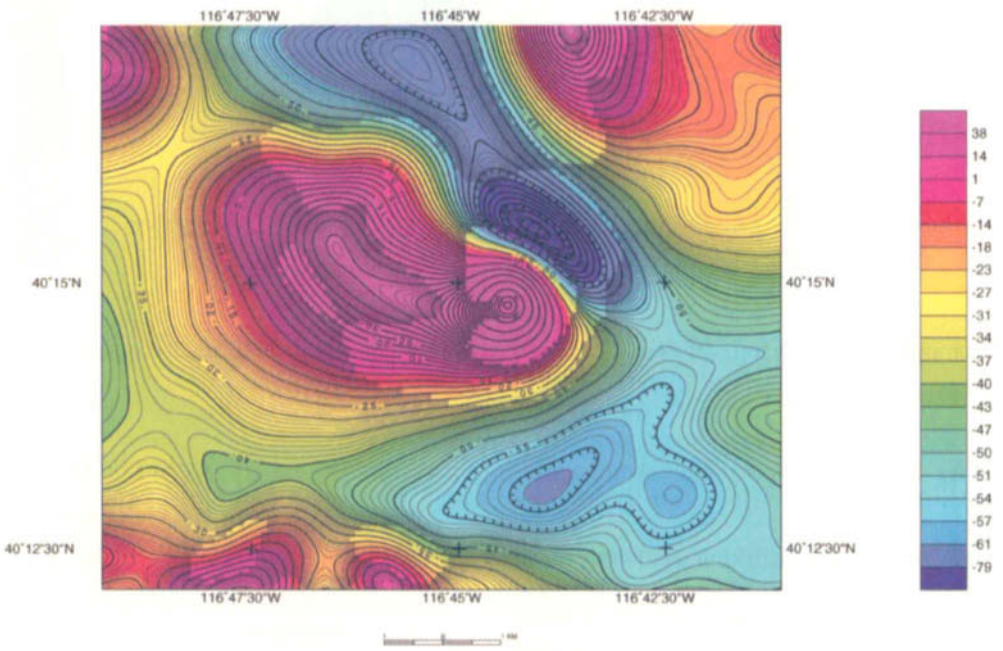

FIG. 6. Wiener-filtered total magnetic intensity for the Gold Acres area. Contour interval is 1 nT.

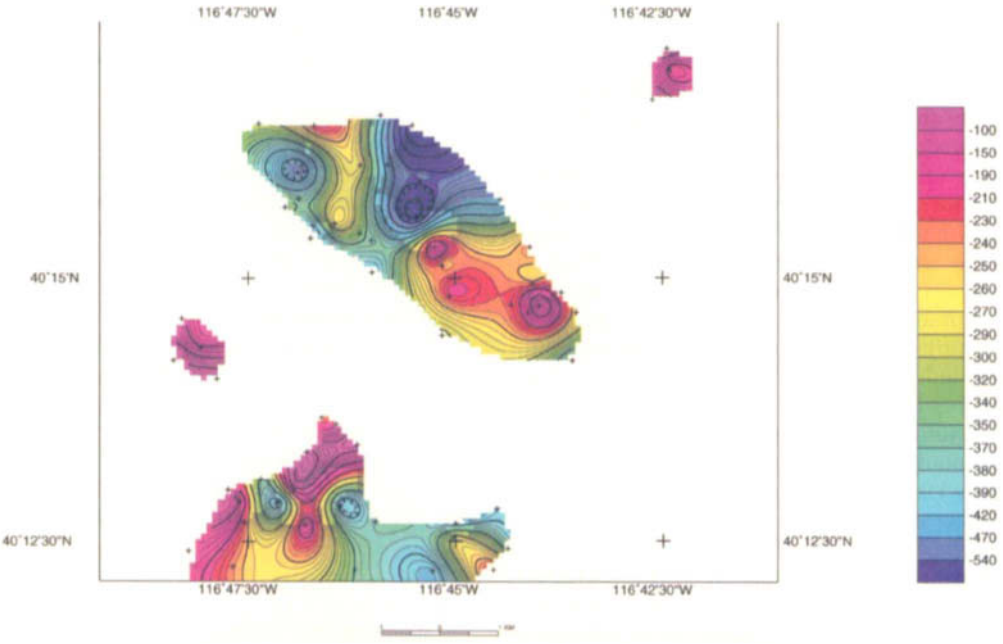

FIG. 7. Mapped shallow volcanic flows for the Gold Acres area. Contour interval is 20 ft. Small crosses indicate locations of the solutions.

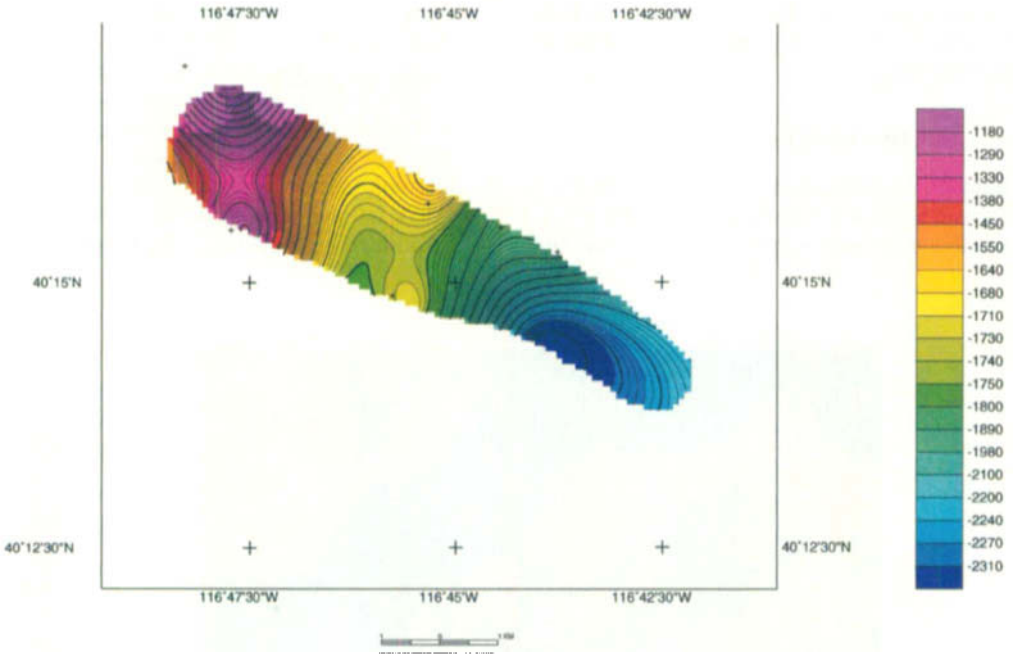

FIG. 8. Mapped depth to the top of the Gold Acres stock. Contour interval is 20 ft. Small crosses indicate locations of the solutions.

important zero-index case can be incorporated with no difficulty (see Appendix B, where it is included for an arbitrary structural index).

There are two drawbacks to the algorithm, neither of which is serious. First, all sources in a given window must have the same structural index. It is possible to imagine situations where it would be desirable to be able to relax this requirement, but it does not appear that any straightforward generalization of the Euler algorithm will handle the mixed-index situation. This state of affairs, however, is no worse than for the single-source case.

The other drawback is a computational one. For both the 3-D inverse of Wang and Hansen (1990) and for the multiple-source Werner algorithm, it is possible to solve the system of linear equations in a window for the coefficients of a high-order system, then to test different numbers of sources by simply truncating the fit. This strategy does not work for the multiple-source Euler algorithm because the coefficients of the lower-order terms change as additional sources are added. Thus, to test different numbers of sources for a given data window, it is necessary to form and solve a new least-squares problem at each step. This turns out not to make the algorithm computationally intractable, but it adds significantly to the effort required for the program.

We have also developed a multicomponent generalization of the algorithm, suitable for such measurements as full tensor gravity gradiometry. Using this generalization, errors due to instrument noise and cross-track undersampling can be reduced and an even more accurate subsurface image can be developed. Finally, using the result of Nabighian and Hansen (2001) that the 3-D Hilbert transforms of a potential field satisfying an Euler equation also satisfy an Euler equation, we anticipate further stabilization of the solutions even in the single-source case.

CONCLUSIONS

We have developed a multiple-source generalization of Euler deconvolution which is capable of handling complex systems that the single-source algorithm can only deal with approximately. Our algorithm has been tested successfully on a variety of synthetic data, and on such areas as the Gold Acres district, Nevada, with satisfactory results.

ACKNOWLEDGMENTS

We thank our employers, Pearson, deRidder and Johnson, Inc., and Bell Geospace, Inc., for support and encouragement during this project and for permission to publish the results. We thank Tien Grauch, Misac Nabighian, Jeff Phillips, Mike Webring, and Bob van Niewenhuise for helpful discussions and encouragement; and the reviewers, Alan Reid, Kwan Cho Hing, and Jeff Phillips, and the Associate Editor, David Chapin, for their very thoughtful comments. Richard Hansen also wishes to acknowledge the intellectual legacy of Doug O'Brien, for without CompuDepth the present work would be unthinkable; the encouragement of Sheldon Breiner, who urged him to implement the Euler algorithm many years ago; the patience of Cathy Skokan and our research groups at the Colorado School of Mines, who listened to presentations of very early work on the algorithm; and the skills learned from Bob Geroch, who taught him to manipulate symmetric, trace-free tensors as a graduate student. Laura Suciu acknowledges Robin Bell for encouragement and financial support.

REFERENCES

Aberth, O., 1973, Iteration methods for finding all zeros of a polynomial simultaneously: Mathematics of Computation, **27**, 339–344.

Arndt, R., Romer, A., and Heinz, H., 1994, 3D non-iterative inversion of aeromagnetic and gravity anomalies in Austria: 56th Mtg., Eur. Assn. Expl. Geophys., Extended Abstracts, session I033.

Barongo, J. O., 1984, Euler's differential equation and the identification of the magnetic point-pole and point-dipole sources: Geophysics, **49**, 1549–1553.
Dennis, J. E., Gay, D. M., and Welsch, R. E., 1981a, An adaptive nonlinear least-squares algorithm: ACM Transactions on Mathematical Software, **7**, 348–368.
———1981b, NL2SOL—An adaptive nonlinear least-squares algorithm: ACM Transactions on Mathematical Software, **7**, 369–383.
Dongarra, J. J., Moler, C. B., Bunch, J. R., and Stewart, G. W., 1979, LINPACK users' guide: Soc. Ind. Appl. Math.
Grau, A. A., 1971, The simultaneous Newton improvement of a complete set of approximate factors of a polynomial: SIAM J. Numer. Anal. **8**, 425–438.
Hadsell, F., 1995, Tensors of geophysics for mavericks and mongrels: Soc. Expl. Geophys.
Hansen, R. O., and Simmonds, M., 1993, Multiple-source Werner deconvolution: Geophysics, **58**, 1792–1800.
Hartman, R. R., Teskey, D. J., and Friedberg, J. L., 1971, A system for rapid aeromagnetic interpretation: Geophysics, **36**, 891–918.
Hildenbrand, T. G., 1983, FFTFIL: A filtering program based on two dimensional Fourier analysis: U.S. Geol. Surv. Open-File Report 83-237.
Hood, P., 1965, Gradient measurements in magnetic prospecting: Geophysics, **30**, 891–902.
Lawson, C. L., and Hanson, R. J., 1974, Solving least squares problems: Prentice-Hall, Inc.
Nabighian, M. N., and Hansen, R. O., 2001, Unification of Euler and Werner deconvolution in three dimensions via the generalized Hilbert transform: Geophysics, **66**, 1805–1810.
O'Brien, D. P., 1971, CompuDepth—A new method for depth-to-basement calculation: Presented at the 42nd Ann. Mtg., Soc. Expl. Geophys.
Perng, S.-M., 1999, On conserved quantities at spatial infinity: J. Math. Phys., **40**, 1923–1950.
Reid, A. B., Allsop, J. M., Granser, H., Millett, A. J., and Somerton, I. W., 1990, Magnetic interpretation in three dimensions using Euler deconvolution: Geophysics, **55**, 80–91.
Stavrev, P., 1997, Euler deconvolution using differential similarity transformations of gravity or magnetic anomalies: Geophys. Prosp., **45**, 207–246.
Thompson, D. T., 1982, EULDPH: A new technique for making computer-assisted depth estimates from magnetic data: Geophysics, **47**, 31–37.
Wang, X., and Hansen, R. O., 1990, Inversion for magnetic anomalies of arbitrary three-dimensional bodies: Geophysics, **55**, 1321–1326.

APPENDIX A

TWO-DIMENSIONAL CASE

We begin with the two-source case. Suppose that $f(x) = f_1(x) + f_2(x)$, where f_1 and f_2 each satisfy an Euler equation with structural index n:

$$(x - x_1)\frac{\partial f_1}{\partial x} + (z - z_1)\frac{\partial f_1}{\partial z} + nf_1 = 0, \quad \text{(A-1)}$$

$$(x - x_2)\frac{\partial f_2}{\partial x} + (z - z_2)\frac{\partial f_2}{\partial z} + nf_2 = 0. \quad \text{(A-2)}$$

From this form of the Euler equations, it is not at all clear how to proceed. Motivated by the use of the analytic signal in multiple-source Werner deconvolution (Hansen and Simmonds, 1993), we write $w = x + iz$, $\varphi = f - i\mathcal{H}f$, where $\mathcal{H}$ is the Hilbert transform operator. If f satisfies Laplace's equation and Euler's equations with structural index n, then $\mathcal{H}f$ does also (Nabighian and Hansen, 2001), so equations (A-1) and (A-2) take the form

$$(w - w_1)\frac{d\varphi_1}{dw} + n\varphi_1 = 0, \quad \text{(A-3)}$$

$$(w - w_2)\frac{d\varphi_2}{dw} + n\varphi_2 = 0. \quad \text{(A-4)}$$

From these equations, it follows that

$$(w - w_1)(w - w_2)\frac{d^2\varphi}{dw^2} + (n + 1)(w - w_1 + w - w_2)\frac{d\varphi}{dw} + n(n + 1)\varphi = 0, \quad \text{(A-5)}$$

where $\varphi = \varphi_1 + \varphi_2$ is the observable analytic signal. Collecting terms yields

$$w_1 w_2 \frac{d^2\varphi}{dw^2} - (w_1 + w_2)\left[w\frac{d^2\varphi}{dw^2} + (n + 1)\frac{d\varphi}{dw}\right] + w^2\frac{d^2\varphi}{dw^2} + 2(n + 1)w\frac{d\varphi}{dw} + n(n + 1)\varphi = 0. \quad \text{(A-6)}$$

The objective is to solve this last equation for w_1 and w_2, which is equivalent to solving for (x_1, z_1) and (x_2, z_2). This objective cannot be achieved in a straightforward way. However, the equation can be solved (by least-squares methods, for example) for the coefficients of $[w\frac{d^2\varphi}{dw^2} + (n + 1)\frac{d\varphi}{dw}]$ and $\frac{d^2\varphi}{dw^2}$:

$$C_1 = -(w_1 + w_2) \quad \text{(A-7)}$$

$$C_2 = w_1 w_2. \quad \text{(A-8)}$$

Then w_1 and w_2 can be obtained as the roots of

$$w^2 + C_1 w + C_2 = 0. \quad \text{(A-9)}$$

It is fairly clear how this form of the Euler equation generalizes to any number of sources. We give only the result; the proof follows exactly the pattern of Appendix B. The Euler equation for m sources is

$$\sum_{j=0}^{m} C_j \sum_{k=j}^{m} \frac{(n + m - 1)!}{(n + k - 1)!}\frac{(m - j)!}{(k - j)!(m - k)!} w^{k-j}\frac{d^k\varphi}{dw^k} = 0, \quad \text{(A-10)}$$

where $C_0 = 1$, and the source locations are the roots of the polynomial

$$\sum_{j=0}^{m} C_j w^{m-j} = 0. \quad \text{(A-11)}$$

Comparison with equation (18) of Hansen and Simmonds (1993) suggests a close relationship between this algorithm and multiple-source Werner deconvolution; this relationship has not yet been investigated in detail.

APPENDIX B

DERIVATION OF THE MULTIPLE-SOURCE EULER EQUATION

We proceed by induction on the number of sources. For a single source, the result is by definition given by

$$(R_i - R_{1i})\nabla_i f^{(1)} + nf^{(1)} = \alpha_1, \tag{B-1}$$

where the constant α_1 has been included to accommodate the inhomogeneous term which appears in the treatment of the case of structural index $n=0$ by Reid et al. (1990). Now assume that the mth-order Euler equation is given by

$$\sum_{k=0}^{m} \frac{(n+m-1)!}{(n+k-1)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)} = \frac{(n+m-1)!}{n!} \sum_{k=1}^{m} \alpha_k. \tag{B-2}$$

Then the left-hand side of the $(m+1)$th-order equation is

$$\begin{aligned}
&\sum_{k=0}^{m+1} \frac{(n+m)!}{(n+k-1)!} a^{(m+1)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f^{(m+1)} \\
&= \sum_{k=0}^{m+1} \frac{(n+m)!}{(n+k-1)!} \big[a^{(m)}_{ki_1\cdots i_k} \\
&+ \mathcal{C}\big[(R_{i_1} - R_{m+1i_1}) a^{(m)}_{k-1i_2\cdots i_k}\big] \nabla_{i_1} \cdots \nabla_{i_k} \big(f^{(m)} + f_{m+1}\big) \\
&= (n+m) \sum_{k=0}^{m} \frac{(n+m-1)!}{(n+k-1)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)} \\
&+ \sum_{k=1}^{m+1} \frac{(n+m)!}{(n+k-1)!} \mathcal{C}\big[\Delta R_{m+1i_1} a^{(m)}_{k-1i_2\cdots i_k}\big] \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)} \\
&+ \sum_{k=0}^{m+1} \frac{(n+m)!}{(n+k-1)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f_{m+1} \\
&+ \sum_{k=1}^{m+1} \frac{(n+m)!}{(n+k-1)!} \mathcal{C}\big[a^{(m)}_{k-1i_1\cdots i_{k-1}} \Delta R_{m+1i_k}\big] \nabla_{i_1} \cdots \nabla_{i_k} f_{m+1} \\
&= \frac{(n+m)!}{n!} \sum_{k=1}^{m} \alpha_k \\
&+ \sum_{k=0}^{m} \frac{(n+m)!}{(n+k)!} \mathcal{C}\big[\Delta R_{m+1i_1} a^{(m)}_{ki_2\cdots i_{k+1}}\big] \nabla_{i_1} \nabla_{i_2} \cdots \nabla_{i_{k+1}} f^{(m)} \\
&+ \sum_{k=0}^{m} \frac{(n+m)!}{(n+k-1)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f_{m+1} \\
&+ \sum_{k=0}^{m} \frac{(n+m)!}{(n+k)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} \big(\Delta R_{m+1i_{k+1}} \nabla_{i_{k+1}} f_{m+1}\big) \\
&- \sum_{k=0}^{m} k \frac{(n+m)!}{(n+k)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f_{m+1} \\
&= \frac{(n+m)!}{n!} \sum_{k=1}^{m} \alpha_k + \sum_{k=0}^{m} \frac{(n+m)!}{(n+k)!} \mathcal{C}\big[\Delta R_{m+1i_1} a^{(m)}_{ki_2\cdots i_{k+1}}\big] \\
&\times \nabla_{i_2} \cdots \nabla_{i_{k+1}} f^{(m)} \\
&+ \sum_{k=0}^{m} \frac{(n+m)!}{(n+k-1)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f_{m+1} \\
&+ \frac{(n+m)!}{n!} \alpha_{m+1} - n \sum_{k=0}^{m} \frac{(n+m)!}{(n+k)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f_{m+1} \\
&- \sum_{k=0}^{m} k \frac{(n+m)!}{(n+k)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f_{m+1} \\
&= \frac{(n+m)!}{n!} \sum_{k=1}^{m} \alpha_k + \sum_{k=0}^{m} \frac{(n+m)!}{(n+k)!} \\
&\times \mathcal{C}\big[\Delta R_{m+1i_1} \nabla_{|i_1|} a^{(m)}_{ki_2\cdots i_{k+1}}\big] \nabla_{i_2} \cdots \nabla_{i_k} f^{(m)},
\end{aligned} \tag{B-3}$$

where the symbols $|\cdots|$ around an index or group of indices indicate exclusion from the symmetrization operation.

Equation (B-3) is the desired result if the second sum on the last right-hand side vanishes. To see that this is the case, first note that by an easy induction

$$\nabla_j a^{(m)}_{ki_1\cdots i_k} = (m-k+1)\mathcal{C}\big[\delta_{ji_1} a^{(m)}_{k-1i_2\cdots i_k}\big]. \tag{B-4}$$

Thus,

$$\begin{aligned}
&\nabla_j \sum_{k=0}^{m} \frac{(n+m-1)!}{(n+k-1)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)} \\
&= \sum_{k=0}^{m} \frac{(n+m-1)!}{(n+k-1)!} (m-k+1) \delta_{j(i_1} a^{(m)}_{k-1i_2\cdots i_k)} \\
&\times \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)} + \sum_{k=0}^{m} \frac{(n+m-1)!}{(n+k-1)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)} \\
&= \sum_{k=1}^{m} \frac{(n+m-1)!}{(n+k-1)!} (m-k+1) a^{(m)}_{k-1i_1\cdots i_{k-1}} \\
&\times \nabla_j \nabla_{i_1} \cdots \nabla_{i_{k-1}} f^{(m)} \\
&+ \sum_{k=0}^{m} \frac{(n+m-1)!}{(n+k-1)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)} \\
&= \sum_{k=0}^{m-1} \frac{(n+m-1)!}{(n+k)!} (m-k) a^{(m)}_{ki_1\cdots i_k} \nabla_j \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)} \\
&+ \sum_{k=0}^{m} \frac{(n+m-1)!}{(n+k-1)!} a^{(m)}_{ki_1\cdots i_k} \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)} \\
&= \sum_{k=0}^{m-1} \frac{(n+m-1)!}{(n+k)!} (m-k+n+k) a^{(m)}_{ki_1\cdots i_k} \\
&\times \nabla_j \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)} + a^{(m)}_{mi_1\cdots i_m} \nabla_j \nabla_{i_1} \cdots \nabla_{i_m} f^{(m)} \\
&= \sum_{k=0}^{m} \frac{(n+m)!}{(n+k)!} a^{(m)}_{ki_1\cdots i_k} \nabla_j \nabla_{i_1} \cdots \nabla_{i_k} f^{(m)}.
\end{aligned} \tag{B-5}$$

However, the left-hand side of equation (B-5) vanishes by the induction hypothesis. Thus, the second sum on the right-hand side of equation (B-3) vanishes, and the induction is complete.

APPENDIX C

THE GENERALIZED DURAND-KERNER METHOD

The Durand-Kerner method is explained in Aberth (1973). To set the stage, we recapitulate Aberth's derivation.

Suppose P is a complex polynomial of order m:

$$P(z) = \prod_{j=1}^{m}(z - z_j) = (z - z_1)(z - z_2)\cdots(z - z_m) \quad \text{(C-1)}$$

We are given the polynomial in the coefficient form

$$P(z) = a_0 + a_1 z + \cdots + a_{m-1}z^{m-1} + z^m, \quad \text{(C-2)}$$

where $a_0, a_1, \ldots, a_{m-1}$ are known, and we wish to find the roots $z_1, z_2, \ldots, z_m$.

The process begins with some initial guesses $z_1^{(0)}, z_2^{(0)}, \ldots, z_m^{(0)}$ for the roots. Suppose that after q iterations we have approximations $z_1^{(q)}, z_2^{(q)}, \ldots, z_m^{(q)}$. Then the errors are $\delta z_j^{(q)} = z_j^{(q)} - z_j$.

Rearranging, $z_j = z_j^{(q)} - \delta z_j^{(q)}$, so

$$\prod_{j=1}^{m}\left(z - z_j^{(q)} + \delta z_j^{(q)}\right) = P(z). \quad \text{(C-3)}$$

Expanding to first order in $\delta z_j^{(q)}$ yields

$$\prod_{j=1}^{m}\left(z_j - z_j^{(q)}\right) + \sum_{j=1}^{m}\prod_{\substack{k=1\\k\neq j}}^{m}\left(z - z_k^{(q)}\right)\delta z_j^{(q)} \approx P(z). \quad \text{(C-4)}$$

Substituting $z = z_i^{(q)}$ gives

$$\prod_{\substack{k=1\\k\neq i}}^{m}\left(z_i^{(q)} - z_k^{(q)}\right)\delta z_i^{(q)} \approx P\left(z_i^{(q)}\right). \quad \text{(C-5)}$$

Finally, dividing by the product on the left gives an approximate expression for $\delta z_i^{(q)}$:

$$\delta z_i(q) \approx \frac{P\left(z_i^{(q)}\right)}{\prod_{\substack{k=1\\k\neq i}}\left(z_i^{(q)} - z_k^{(q)}\right)}. \quad \text{(C-6)}$$

This approximation can be used to update $z_i^{(q)}$:

$$z_i^{(q+1)} = z_i^{(q)} - \delta z_i^{(q)}. \quad \text{(C-7)}$$

Repeating for each i yields an update for all roots, and the process can be repeated until convergence is achieved. This scheme can be shown to be equivalent to a Newton-Raphson method, quadratically convergent near a root. It has proved to work well in the multiple-source Werner implementation (Hansen and Simmonds, 1993).

Note that in this form the Durand-Kerner method is unsuitable for equations with multiple roots, as equation (C-6) diverges as the multiple roots approach their true values. This is not a concern for the multiple-source Werner application, as multiple roots correspond to coincident vertices. In fact, the structure of the update equation tends to force the roots apart, a desirable feature. However, it is worth mentioning that the method has been generalized to equations having multiple roots by Grau (1971).

The analog of the polynomial in the multiple-source Euler system is

$$\mathcal{C}\left[\prod_{j=1}^{m}\left(R_{i_j} - R_{ji_j}\right)\right] = a_{i_1\cdots i_m}(\mathbf{R}). \quad \text{(C-8)}$$

There are two essential differences between this system and the polynomial: the equations are real rather than complex, so that we are not guaranteed roots; and the system is overdetermined for $m > 1$. Thus, what is required is a least-squares version of the Durand-Kerner process.

Fortunately, the derivation follows formally exactly the same path as before. Assume that at the qth iteration we have approximations $R_{ji_j}^{(q)}$ for the vertex locations and errors $\delta R_{ji_j}^{(q)}$. Then, expanding equation (C-8) to first order in $\delta R_{ji_j}^{(q)}$ gives

$$\mathcal{C}\left[\prod_{j=1}^{m}\left(R_{i_j} - R_{ji_j}^{(q)}\right)\right] + \mathcal{C}\left[\sum_{k=1}^{m}\prod_{\substack{j=1\\j\neq k}}^{m}\left(R_{i_j} - R_{ji_j}^{(q)}\right)\delta R_{ki_k}^{(q)}\right]$$
$$\approx a_{i_1\cdots i_m}(\mathbf{R}). \quad \text{(C-9)}$$

Now substitute $\mathbf{R} = \mathbf{R}_l^{(q)}$. The first term on the left side of the preceding equation disappears, and the second collapses to the term $k = l$, so

$$\mathcal{C}\left[\prod_{\substack{j=1\\j\neq l}}^{m}\left(R_{li_j}^{(q)} - R_{ji_j}^{(q)}\right)\delta R_{li_l}^{(q)}\right] \approx a_{i_1\cdots i_m}\left(\mathbf{R}_l^{(q)}\right). \quad \text{(C-10)}$$

Unlike the update for the complex-variable case, it is not possible to divide by the factor on the left-hand side of equation (C-10). In fact, the system of equations is overdetermined for $m > 1$. However, it is possible to solve the system in the least-squares sense, and this is precisely how the update has been implemented. LINPACK routines (Dongarra et al., 1979) are used to factor the system by QR decomposition and then to solve for the correction $\delta R_{li_l}^{(q)}$. Convergence properties of this generalized Durand-Kerner scheme have not been formally investigated, but empirically excellent results are obtained for the systems which occur in the multiple-source Euler problem.