MODELLING AND INVERSION – PROGRESS, PROBLEMS, AND CHALLENGES

ART RAICHE

CSIRO Division of Exploration and Mining, P.O. Box 136, North Ryde, NSW, Australia, 2113

(Accepted 30 August 1993)

Abstract. Researchers in the field of electromagnetic modelling and inversion have taken advantage of the impressive improvements of new computer hardware to explore exciting new initiatives and solid extensions of older ideas. Finite-difference time-stepping methods have been successfully applied to full-domain 3D models. Another new method combines time-stepping with spatial frequency solutions. The 2D model 3D source (2.5D) problem is also receiving fresh attention both for continental and sea floor applications.

The 3D inversion problem is being attacked by several researchers using distorted Born approximation methods. Q-domain inversions using transformation to pseudo-wave field and travel time tomography have also been successfully tested for low contrast problems. Subspace methods have been successful in dramatically reducing the computational burden of the under-determined style of inversion. Static magnetic field interpretation methods are proving useful for delineating the position of closely-spaced multiple targets.

Novel ("appeals to nature") methods are also being investigated. Neural net algorithms have been tested for determining the depth and offset of buried pipes from EM ellipticity data. Genetic algorithms and simulated annealing have been tested for extremal model construction.

The failure of researchers to take adequate account of the properties of the mathematical transformation from algorithms to the number domain represented by the computing process remains a major stumbling block. Structured programming, functional languages, and other software tools and methods are presented as an essential part of the serial process leading from EM theory to geological interpretation.

1. Introduction

Although the title of this paper may seem self-explanatory, let me confuse the issue by telling you exactly what I had in mind. As each new discovery in modelling and inversion methods is presented, it is easy to convey the impression that at last, all the significant problems capable of solution have been solved. In fact, we know that the principle of conservation of difficulty exists and that each new method generates its own set of frustrations. One of the aims of this paper therefore is to convey the impression that no matter how brilliant some of the methods described at this meeting may seem, there is still very considerable scope for new research in modelling and inversion. I also wish to present a few new challenges to those who may wish to look at the process of modelling and inversion (M & I) in different ways.

The progress will consist of two parts, one of which is an explicit sampling of several new modelling and inversion algorithms. This year has been particularly exciting because several 3-D modelling algorithms have appeared which allow for

Surveys in Geophysics **15**: 159–207, 1994.

a general variation of subsurface conductivity rather than being limited to a blob in an otherwise uniform half-space. Also, 3-D inversion is becoming at last, a practical reality rather than just a dream (numerical nightmare?). However, these algorithms are not very useful unless we can encapsulate them into reliable computer software. Thus the other part of this review will cover something which we tend to ignore, the tremendous advancement made in software productivity tools, including higher level languages, symbolic manipulation software, and structured programming.

Before defining what I mean by "problems", let's examine some of the reasons why we might choose a career in EM modelling and inversion. On the basic level, it is certainly a much more comfortable existence than working on a factory assembly line or other repetitive manual labour jobs. Next, there is ego gratification, the thrill of being the first to find the solution to a difficult logical problem. But lastly in my list, there is the desire to understand how to use electromagnetic methods to find resources and/or to study the structure of the earth's crust and possibly the crust-mantle interface. It is this latter reason which gives rise to what I mean by "problem". How do we know that the numbers at the end of the computational process are "correct"?

Most of us regard the work of modelling and inversion as devising mathematical and numerical schemes to compute responses. Inevitably, these require machine computations. The work of writing programs to produce the numbers is usually regarded as a tedious mechanical task, much akin to typing a paper and drawing diagrams. It should be, but in fact, it is not. The act of computation is much more profound than this. It is a mathematical transformation of the ideas of the modelling algorithm onto the field of numbers of finite precision. These transformations are not unique. Each has its own set of rules and conditions, many of which are not obvious. As with all transformations, if rules and conditions are violated, the results may appear reasonable but still contain gross errors. If the computational structure is not reliable, then the numbers and the algorithmic development are useless. The problem is how to design computational structures so that our results are a correct mirror of how well the algorithm itself simulates the interaction of the electromagnetic field with the earth. One way is use old tools and methods as best we can. A better way is to take advantage of the power and reliability of new languages and more structured procedures.

Figure 1 is a process diagram of the M & I (modelling and inversion) process. It is a series process with feedback loops, only one of which is shown. Series process means that if any one part of the chain is in error, the results are unreliable. It is possible to achieve apparently reasonable results by fortuitous error cancellation.

Since few of us tend to dispute the validity of Maxwell's equations, I have assumed that these are the root point of any M & I procedure. We can make the procedure of obtaining a solution to these equations as complicated as we wish

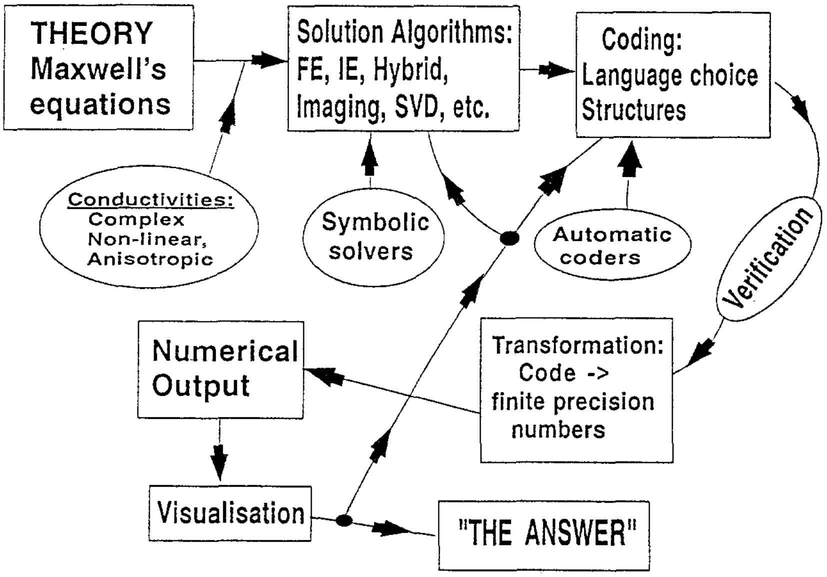

Fig. 1. Modelling and inversion process diagram.

by including anisotropy as well as letting the conductivities vary with respect to field strength and frequency. The next stage is the selection of model-source geometry. Current geometries of interest are 3-D, 2.5-D, and thin sheets. The first two can be broken down into full domain heterogeneous or localised heterogeneous.

The next block, solution algorithms, is the major topic of this paper. Typically, these include finite elements/differences, integral equations, direct inversion, tomographic reconstruction, image solutions etc. Also included in this block is choice of domain (frequency-time and spatial frequency-spatial co-ordinate).

It is after this point in the chain that many of us suspend rational thought and enter a dream world of childlike belief that by scribbling out a few hundred or a few thousand lines of Fortran, a magical device called a computer will interpret our thoughts and give us numbers which are an accurate simulation of how electromagnetic energy has interacted with earth structure. The James Joyce "stream of consciousness" style of Fortran programming may express the poet within us but then we must also be prepared to regard the output as fantasy.

If the numbers do not behave according to our physical intuition, we assume that the algorithm is in error, or that we may have made a few silly typing errors. The adventurous may use unexpected numerical results as the basis for announcing "new discoveries". It is only when all else fails that we may possibly stop to consider the importance of the computational structure itself on the numerical outcome.

The really serious problem occurs when a computationally invalid structure yields plausible results due either to the fortuitous by-passing of error producing parts or fortuitous cancellation of errors. This is similar to putting your trust in a seemingly honest used car salesman who makes a good first impression.

Before reviewing some of the advances which have been made in modelling and inversion methods, I would like to discuss some aspects of the computational process; and offer suggestions on how to eliminate problems which we may unnecessarily inject into the modelling process. I would also like to suggest exciting new ways of computing which will make the person-machine interface more transparent.

2. Computational Process and Structure

Suppose for the moment that we were going to be so rational about computing that we would actually specify a list of criteria for producing software. Such a list might include the following:

1. The program should represent a "true computation" of the algorithm.
2. The programming procedure should correspond as closely as possible to how we think about the problem; i.e., model specification and the mathematics of the algorithm.

3a SPEED The time required to go from the algorithm to stable computational process yielding correct answers should be minimised. (Why is it that there is never enough time to do something right the first time but there is always enough time to do it over?)

3b SPEED The program should run as quickly as possible and should not exceed available memory. Disk access requests should be minimised.

4. It should be easy to program correctly changes to the algorithm.
5. The code should be easy to debug.
6. The code should be portable. That is it should be able to be compiled the first time on any computer with sufficient memory using a standard compiler for the program language. The results (aside from minor precision differences) should not be machine dependent.
7. The code elements should be re-usable. That is one should be able to pull out various computational procedures for use in other programs.
8. The code should be easy to understand (at least by the author) a year after the program was last changed. (A very severe requirement indeed!)

Although these goals may seem obvious, there is not a great deal of software around in our field which meets them. Structured programming (SP) is a methodology which has grown up over the last fifteen years specifically designed to help the programmer meet the goals listed above.

A. Structured programming

In most of what follows, I will use Fortran as the basis for discussion since it is the most widely used language for serious numerical computation. However, the concepts apply to the other procedural languages such as C[1], Pascal, Modula-II etc.

It has been estimated that failure to use SP methods will, on average, result in major errors every 200 lines of executable code. (Of course this doesn't apply to programs which you or I would write, only to people less gifted than ourselves!) Using strict SP methods will reduce the error rate to about 1 error every 4,000 lines of executable code. Since there are a number of excellent treatises on SP (Dahl *et al.*, 1972; Dijkstra, 1976; Linger *et al.*, 1979) I will present only a limited description of the discipline.

Modularity, and eliminating side effects are the two main aspects of SP methodology. I include a third, documentation. Modularity means that the program should be broken down into as many self-contained modules and sub-modules as is logically feasible with a minimum of inter-module connections. To start at the top, the main program should perform no work function other than to call subroutines to read in data, present the results, plus those which will organise the major computational functions. In turn, each subroutine and function should be kept quite short by breaking down the work function into sub-modules which are themselves divided into smaller modules.

Now we come to a crucial point, the elimination of side effects. In the case of Fortran, side effects refer to globally defined variables which reside in common statements. The output of subroutines and functions should be STRICTLY determined by its calling arguments and executable statements. There should be no variables in Common statements. This means that a change in one subroutine will not be able to influence what happens in another subroutine except through explicit calling arguments. Failure to observe this convention can wreak enormous havoc and seemingly inexplicable behaviour.

Modularity and the elimination of side effects yield great benefits. Each subroutine can be tested for correctness independently of all of the others. Code can be reused since independent subroutines can be inserted into other programs. Programs composed of independent modules can be more easily improved or changed. But there is another demon to be eliminated, the GOTO statement.

It is important that each subroutine has exactly one entry point and one exit point and that control flows strictly from entry to exit (hopefully top to bottom). This is achieved by using DO loops and IF-THEN-ELSE constructions and ban-

[1] C is a very dangerous language because its commands are so powerful that it encourages programmers to write cryptic compact code with hidden complexities. In fact, some people refer to C as "Write Only" code.

ning the use of the GOTO statement. The use of GOTO is one of main sources of errors and confusion in non-structured programs.

B. Documentation

Documentation should be the first rather than the last activity in producing a program. It is well known that if left to last, it invariably is done poorly, if at all. However, there is a more profound reason for doing it first. It is similar to using an outline as the first step in writing a paper. By writing statements in the spoken language of ones choice into the subroutine as to what is intended and how it is to be done, writing the executable program consists of simply filling in the blanks with Fortran statements. Debugging is made easier because one can first ask whether the intended logic is correct and then whether the subsequent Fortran statements actually do what the spoken language statement says it does. Lastly, a properly documented program can actually be understood by the writer as well as others, particularly after some time has passed since its genesis. The use of indentation and blank lines; i.e., making the layout look pretty, can be a great aid to comprehension, the same as for written text.

C. Verification

As mathematical geophysicists, we have a tendency to think of computers as useful for computing arithmetical functions. As a result, we do a number of things manually that a computer could do much faster. If a program gives unexpected answers, we begin a process of laboriously printing out intermediate computational stages hoping to find a reason for the strange behaviour. Since at any stage of the computation we have a good idea as to the allowable range of variables, why not write functions which will trap values outside these ranges and either print messages or take some other form of appropriate action? These trapping functions should be written with switches which allow them to be de-activated when we are confident that the program is working properly. Another powerful trap is to use the bounds checking options on compilers when running test data sets.

It is important to include these traps in the first few editions of a program instead of waiting until the program definitely fails. Very often there are dormant errors in programs which may be only weakly activated by test data sets. This can create false confidence in subsequent program output which may contain more serious errors. Trapping underflows can assist the programmer to place more restricted limits on DO loops which can be useful in reducing execution times.

Lastly, there exist software checking tools which will report on unused or undefined variables, non-standard code, and a number of other things which may lead to pre-execution identification of program errors. FORCHECK is an example of one such product.

D. Precision

Analytic solutions often pose hidden precision problems. We tend to believe that the computer can compute closed form analytic expressions more accurately than it could perform the equivalent numerical integration. The opposite is often true, especially when there are no singularities. This belief is a legacy of applied mathematics courses taught from a 19th century perspective and fails to take account of the implications of a machine which performs arithmetic with finite digit numbers.

Usually, analytic solutions for EM propagation in the earth involve two or more expressions of differing sign, which can cause severe precision problems in some part of the relevant domain. Numerical integration procedures; i.e., the addition of weighted function values, will in many cases be less prone to precision problems if we choose the correct methods. This is not to deny the value of analytic solutions. Rather it is to point out the importance of exploring the domain of these solutions for precision problems. In many cases, it may be possible to use series expansions or alternatively phrased solutions to eliminate the precision problem.

Consider two examples, one trivial, one less so. The roots of the quadratic equation $ax^2 + bx + c = 0$ are usually expressed as:

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \qquad x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}.$$

If $b > 4000ac$, and the computer carries 6 significant digits then the solution for x_2 will be meaningless. Careful programming would encourage expressing x_2 in the form

$$x_2 = \frac{2c}{ax_1}.$$

The second example comes from an analytic expression for the time-domain electric field E on the surface of a half-space of conductivity σ at a distance ρ from a horizontal dipole transmitter.

$$E = \frac{1}{2\pi\sigma\rho^3}\left[\mathrm{erf}(\sqrt{X}) - 2\sqrt{\frac{X}{\pi}}e^{-X}\right]$$

where

$$X = \frac{\sigma\mu\rho^2}{4t}$$

for small values of X; i.e., late times or very resistive half-spaces, the above form is unstable. Expanding the error function, erf, results in an alternate form which should be used for small X.

$$E = \frac{1}{2\pi\sigma\rho^3}\sqrt{\frac{X}{\pi}}\,e^{-X}\sum_{n-1}^{\infty}\frac{(2X)^n}{(2n+1)!!}$$

The promiscuous use of double precision is a dangerous panacea. Although double precision would work quite well for the above problem, its use can mask deeper precision problems for which double precision is not sufficient. An example of this occurs when using Summary Representation methods to limit the mesh extent of finite-difference or finite element computations. In Tarlowski and Raiche (1984), it appeared that precision problems existed with the numerical finite difference part of the solution. It improved as the grid size was decreased but continued to remain unacceptable. The use of double precision gave only a moderate improvement. A series expansion of the Summary Representation particular solution, which resulted in the elimination of the three leading terms of the series, solved the problem.

In general, double precision should be used primarily for those cases where the precision problem is well understood such as the Gaver-Stehfest method for inverting Laplace transforms (Knight and Raiche, 1982).

The inversion community may feel quite smug about the precision issue because of the close attention paid to regularising the sensitivity matrices. However, what about the underlying forward solutions? An SVD analysis of the matrices used for frequency-domain solutions or implicit time-stepping solutions might be quite revealing. This is one of the challenges.

E. Speed of execution

In spite of the astonishing growth in hardware performance, execution speed is still a vital issue. On the more academic side, random search inversion methods such as simulated annealing and genetic algorithms applied to 3-D models will bring any machine to its knees. Those of us who build software for industrial use find that the best modelling program in the world will be unused if it cannot be run in a relatively short time on a desktop. There are a number of ways we can dramatically decrease CPU times for our programs.

One method to reduce CPU times is through applying approximation methods with adjustable precision. A common example of this strategy is the use of Hankel transform algorithms (including Fourier transforms) based upon the digital filter coefficients developed by researchers such as Anderson (1979) and Johansen and Sorensen (1979). This represented a very substantial breakthrough for computational EM geophysics. Another breakthrough (which appears to be much less often used) consists of representing computationally intensive functions, their integrals and their first derivatives, by cubic spline interpolants. Hohmann (1975) used two-dimensional splines to replace explicit Green's tensor element evaluation. It was discovered (Gupta *et al.*, 1987) that it was much more efficient to use layers of one-dimensional splines than the two-dimensional splines used by Hohmann.

Finally, it was shown that it was possible to improve computational speeds by more than an order of magnitude (with no sacrifice of accuracy) by using nested layers of interpolating functions (Raiche, 1987).

Procedures which require the computation of a function over an extended domain can often be made more efficient by using simple functional forms to extrapolate (predict) the function when its behaviour is known to be simple. For example, when computing the frequency spectrum of a function for transformation into the time-domain, it is usually the case that the low frequency end approaches the DC limit in a simple fashion. Empirical analysis of the functional form will often indicate a suitable extrapolation such as substituting simple linear or exponential forms for complex function evaluations before the DC limit is reached.

Poor structure is an obvious source of excessive computer overheads. Programs which were not originally properly structured, which subsequently were then "corrected" or given added functionally over time, tend to contain a number of unnecessary loops which can greatly extend execution times. Using simple restructuring procedures, people in my group are often able to achieve order of magnitude reduction in CPU runtimes for software kindly given to us by other research groups. If SP methodology is invoked from the start and rigidly adhered to, it is very difficult to program redundant loops without being aware of it.

However, let me be suggest a firm rule at this point. Never sacrifice clarity and structure for cleverness. Failure to observe this rule is very costly in the long term. If you must be clever, make it very obvious in the program just what it is that has been done.

Matrix operations form the backbone of much of our M & I computing tasks. The way in which we phrase these tasks can make literally orders of magnitude difference in runtimes. This is because programs which can access pieces of data contiguously will run faster (in some cases very much faster) than those which have to hop back and forth looking for addresses of randomly accessed data. Failure to access data contiguously, when part of the data is disk resident, can increase execution times spectacularly.

Efficient computation of matrix operations is a large field of research in itself. Rather than attempting to make this paper into a lecture on the subject, let me suggest a very readable book which has become one of the bibles on the subject, *Matrix Computations* (Golub and Van Loan, 1989).

However, I will use a simple example from this book to lead into another topic. It is fairly common knowledge that Fortran stores a two-dimensional matrix by columns. Thus if one has the matrix A(I,J) in a double loop, it is much more efficient to have J in the outer loop and I in the inner loop since in this way we will be accessing data contiguously. However, it is easy to fall into a trap when doing something simple such as taking the product of two matrices and storing the result. Consider two different codings to construct a matrix which is the product of two other matrices. C(I,K) = A(I,J)*B(J,K)

```
for I = 1, N1                  for K = 1, N2
  for K = 1, N2                  for J = 1, N3
    for J = 1, N3                  for I = 1, N1
      C(I,K) = A(I,J)*B(J,K)         C(I,K) = A(I,J)*B(J,K)
    end                            end
  end                            end
end                            end
```

The code on the right will access data more efficiently and thus may run much faster than the code on the left if N1, N2, and N3 are large. However, from the standpoint of developing correct code quickly, the algorithm on the left offers the advantage of clarity because it is closer to the style in which the mathematics is expressed. Should we write for clarity or speed?

In this case, the best choice is none of the above because we should not be writing code at this level any more than we should have to program at machine level. The machine software should be performing these index ordering tasks. To a small degree, this is done using the vector commands of Fortran 90. However, I believe that we number crunchers should follow the lead of those in the artificial intelligence field. We should be using higher order custom built languages.

F. Higher level languages

The concept of a higher level language for us would mean that we would specify our model, variables, and domains in something close to natural language (English, Finnish, Russian, Kiwiese etc.). We would then express the mathematical formulation in a form that was as close as possible to the actual mathematics of the problem. Some procedure specification would be necessary but a lot would be done by the language constructs and function libraries. The compiler would produce Fortran or C code which could then be compiled on any machine with standard compilers. The need for the user to use structured programming disappears because the structure is in the language. Programming for the user is reduced to filling in templates. Consider the current alternative.

The finite-element method is a powerful, well proven numerical method for many problems in engineering and science. Although the theory is fairly simple, writing FE programs is a formidable task. The way in which nodes are numbered can affect bandwidth significantly, and hence execution times. Correlating node numbers with element numbers is a complicated bookkeeping task, especially for higher order elements. This becomes even more complicated if one uses a frontal solution. Application of different types of boundary conditions, changes to element functions, and changes to geometry open the opportunity for programmer-induced error yet further.

At least 90 percent of finite-element programming consists of tedious bookkeeping tasks that the computer can perform more reliably and efficiently than we can.

Computer languages constitute a whole sub-discipline in computer science. We should be working with computing science people to develop specialised languages for finite elements, inversion, finite differences etc.

The development of DPML (data parallel meta-language) is an example of such a collaboration between the CSIRO Divisions of Information Technology and Atmospheric Research (Abramson, pers. com. and Francis, pers. com.) The goal was to build an explicit time-stepping finite difference program for climate modelling which not only made provision for wind dynamics, solar radiation, clouds, rain, snow and ice; but also for such things as evaporation through vegetation and coupling with oceans. The results for the prototype language, DWARF, were published in the CTAC-91 conference proceedings (Abramson, *et al.* 1991)

Programming in DPML, is very simple and quite far removed from what we might consider a program to be. In a typical program, the user specifies the values of constant data, number of physical dimensions (2-D or 3-D say) and the domains; i.e., the grids. (The climate model operates on 4 staggered grids.) The user then defines the fundamental and derived variables and assigns them to the appropriate grid. The transforms which relate the various grids are specified next. Fortran is used for the input and output functions so module interfaces must be specified.

The user then specifies the boundary conditions, the spatial differencing and the time-stepping algorithm. Lastly, the physics of the particular model; i.e., the mathematical relationships, are expressed in terminology very close to the way in which equations are written. A typical program can be written in about 100 lines.

The DPML compiler produces Fortran code but not in a form which could be understood very easily. It targets this intermediate code to specific machines so as to achieve maximum efficiency. Fine-grained massively parallel SIMD machines such as the Connection Machine or the Maspar organise their data in different ways. The DPML compiler will produce intermediate Fortran code to optimise the way the grid variables are assigned to the different processors in each case. There are separate compiler options for Crays, unix workstations, and machines using VMS.

DPML represents a significant step in developing the concept of special purpose languages for problems in computational physics. Special languages for tasks other than explicit time-stepping finite differences may be slow to be developed. In the meantime, functional languages offer an interesting alternative to what we do now.

G. Functional languages

Fortran, C, Pascal, Modula-II etc. belong to a category known as procedural or imperative languages so named because programs written in these languages define procedures and the order in which these are to be executed. On the other hand, a program written in a functional language is itself a function defined in terms of a hierarchy of functions until at the bottom level, the functions are language primitives. These functions are very much like ordinary mathematical functions. Starting a program in an imperative language begins a time sequenced series of processes defined by the user. Running a program in a functional language is more like logging in to an environment and asking the computer to evaluate a function.

A functional program is a collection of mathematical expressions comprised of user defined and intrinsic functions which are well-defined and determinate. They define unique mappings between domains and ranges. There are no assignment statements so variables, once given a value, never change. Thus, when passed the same set of values, a function will always yield the same answer regardless of the environment. This establishes what is called referential transparency which means that one can freely replace variables by their value and vice versa.

Thus there are no side effects. Consequently, the order of execution is irrelevant since the lack of side effects means that a function can be evaluated at any time. One implication of this is that the programs are implicitly parallel.

In an excellent introductory article on functional programming, Hughes (1989) discusses another powerful aspect of FL (functional languages), the "glue". Having provided the facility for the programmer to write correct modules quickly, FL also supply powerful tools to glue simple modules together. For example, on the micro level, FL programs commonly operate on list structures. The elements of these lists can be constants but they can also be functions. FL have the facility to perform higher order operations on lists as a whole or on designated elements. This enables the programmer to build structures of arbitrary complexity from simple elements using simple commands.

On the macro level, another form of FL "glue", called lazy evaluation, makes it easy to run two or more programs together with no extra programming. An FL program is a function, which can call any other function including another program. Lazy evaluation refers to the fact that a function is memory resident only when it is being called so no extra storage is required. By contrast, if we were to use the output of one imperative program to feed another imperative program, we would have to anticipate the entire range of data and provide a large array to store it, or alternatively provide memory for the other program(s).

The lack of side effects, the elimination of the need to specify flow control, and the close relationship with mathematical formulation means that functional languages make it easier to write, prove, analyse, maintain and extend programs. They allow a level of abstraction over conventional imperative languages which separates the programmer from the machine and emphasises problem definition and algorithm mathematics over the mechanics of computer solution. Although programs written in FL are not necessarily faster than those written in Fortran in terms of execution speed, the development time from algorithm conception to acceptable numerical results can be an order of magnitude shorter for FL compared to that for procedural languages.

Functional languages are not a new development. LISP and Prolog have been an integral part of artificial intelligence work for many years. In fact, LISP is as old as Fortran. What is relatively new is the interest in applying FL to large scale numerical problems. Until recently, there had been a problem with execution speeds, the so-called copying problem. Since strict FL do not allow variable reassignment, a change to an N by N matrix required the whole matrix to be recopied each time an element was updated. Thus if N = 1,000, there are one million matrix copy operations. Recently, this problem has been solved with the development of in place operations (termed copy elimination) which do not violate FL strictures.

At the present time, SISAL, developed at LLNL (Lawrence Livermore National Laboratory), would appear to be the most mature of the general purpose numerically inclined functional languages. Its compiler produces intermediate C or Fortran and leaves final machine code generation up to resident compilers. Cann (1992) gives examples where large numerical simulation programs from different disciplines were run on various Cray configurations in both SISAL and Fortran. Sometimes SISAL was faster, sometimes the Fortran was, depending upon the program, and in some cases, the specific iteration. SISAL is a public domain software package.

An example of applying FL programming to 2-D finite-element stress computations is described in Liu *et al.* (1992).

H. Other tools – artificial intelligence applications

I would like to conclude this section with a brief discussion of two types of software tools, one of which exists, the other of which doesn't. We tend to think of applied mathematics as a cognitive activity of high order.[2] However, activities such as analytic integration, differentiation, solving algebraic equations, series expansions etc., consist largely of pattern matching and the application of well-defined rules. Computers are quite good at pattern matching and applying well-defined rules as well as the creation of new rules from input data.

There are several symbolic manipulation packages now extant (Reduce, Macsyma, and Mathematica to name three) which can perform algebraic and calculus tasks. These could play an invaluab!e role for verifying (and indeed, even for deriving) complex mathematical equations. Mathematica (Wolfram, 1991), in particular is very interesting because the language ties the symbolic mathematics facility to numerical evaluation and 2-D and 3-D screen graphics options. It also has another powerful aspect. Mathematica makes it possible to perform floating point operations with numbers of any digit length, say 72 for example. This feature is invaluable for examining the extent of precision problems for small pieces of code. Mathematica is useful mostly for analysis rather than practical execution.

[2] It is much easier to build a machine which can derive theorems and manipulate mathematical equations than it is to build one to play tennis.

Using it to perform entire large numerical simulations would give the user an intimate feeling for geologic time scales.

A tool which would make AI (artificial intelligence) directed numerical computation easy to program would be invaluable. A large part of our numerical computations consist of approximations which represent a trade-off between the required accuracy and speed of execution. Mesh design, and setting the knots for spline computation are but two examples of this. These are set before runtime as a sort of compromise. Thus some parts are over-specified and other parts are inadequate. If we had much faster brains which could take in the information of how selected numerical functions developed as they were being computed, and had the ability to change the mesh density and extent and knot distribution as the computation progressed (in a metaphoric sense, wander around inside the computer memory), our programs could be made to have the desired degree of accuracy and efficiency. In principle, we could program this in Fortran but it would be very difficult. The point is that it would be very useful to have an AI facility to let the computer examine itself and vary program control during the numerical computation process. The facility should then report on changes. This tool does not yet exist but I offer you the challenge of developing it.

3. Forward Modelling

The elusive (controversial) goals of inversion depend upon many things including the development of fast, reliable, and general forward modelling algorithms. Curiously, we tend to lose sight of the fact that numerical forward modelling is really an inversion process. The unknowns are the local functions used to represent fields and possibly the mesh point locations. The data is a modeller defined functional of the primary source field. The moving finite-element (MFE) work of Travis and Chave (1989) is one of the few acknowledgements of this fact. Indeed, they had to use regularisation methods to preserve element shapes. Their work exhibited two other advanced features which are not yet in wide use: the use of an automatic mesh generator, and the use of incomplete factorisation methods instead of full Gaussian elimination. At this time, there is not much new to report on MFE modelling.

In the last three or four years, forward modelling activities have in fact taken a great leap forward. Previously, practical EM forward modelling programs (integral equations, compact finite elements) had all been based upon modelling the response of a blob in an otherwise uniform half-space. Druskin and Knizhnerman (1988) published a more general, 3-D time-domain finite-difference solution which was free from this restriction. (I will refer to this as full-domain modelling.) They used a spectral Lanczos decomposition method (SLDM) rather than a time-stepping approach to obtain time-domain solutions. This required an implicit finite-difference solution but in Hördt *et al.* (1992), it is claimed that the solution time for the SLDM increases only as $\sqrt{t}$.

In what follows, I want to cover some interesting new full-domain work which is in the process of being published. The first is Wang and Hohmann's 3-D time-stepping finite-difference work (Wang and Hohmann, 1993). This is followed by Lee's new approach of time-stepping in the spatial frequency domain (Lee, 1991). There has also been a resurgence of interest in the full-domain 2.5-D problem. Sugeng and Raiche (1992) have developed a program for the active source minerals exploration whilst Unsworth, *et al.* (1993) as well as Everett and Edwards (1992) have been oriented towards sea floor active source applications. This section will conclude with some general observations on the current state of modelling with suggestions on new directions.

A. Time-stepping with staggered grids and first order equations

Finite-difference and finite-element solutions for EM induction have been in the literature for more than two decades. Until recently, it was standard practice to solve for a second order equation on a single grid.

Yee (1966) published a solution for coupled first order equations defined on a staggered grid, which formed the basis for Wang and Hohmann's (1993) 3-D finite-difference time-stepping method. In the frequency domain, Smith (pers. com.) and (Madden and Mackie, 1989) have also used the staggered grid approach. The approach is best understood by looking at Figure 2. Two of Maxwell's equations expressed in terms of the usual symbols are;

$$\nabla \times \mathbf{e}(\mathbf{r}, t) + \mu\partial_t\mathbf{h}(\mathbf{r}, t) = 0 \tag{1}$$

$$\nabla \times \mathbf{h}(\mathbf{r}, t) = \sigma\mathbf{e}(\mathbf{r}, t) \tag{2}$$

Two grids are defined such that the corner of a cell of one is in the centre of the cell of the other. At an initial time, t_0, an analytic solution is used to compute the half-space electric field on the E grid. At time step t_1, the electric field values are frozen and Equation (1) is used to compute the magnetic field on the B grid. At time step t_2, the magnetic field values are frozen and Equation (2) is used to compute the electric field on the E grid. This process is continued iteratively until the maximum desired delay time is reached. Stokes' theorem can be invoked to show that the staggered grid implies that the curl is computed without additional discretisation error; i.e., it has the same accuracy as the nodal values on the other grid. In theory, this time-stepping method is a very simple and accurate process. However, getting it to perform adequately in practice has required a lot of skill and work.

In order to avoid a matrix inversion at every step, Wang and Hohmann used a modified Dufort-Frankel scheme which could be made unconditionally stable.

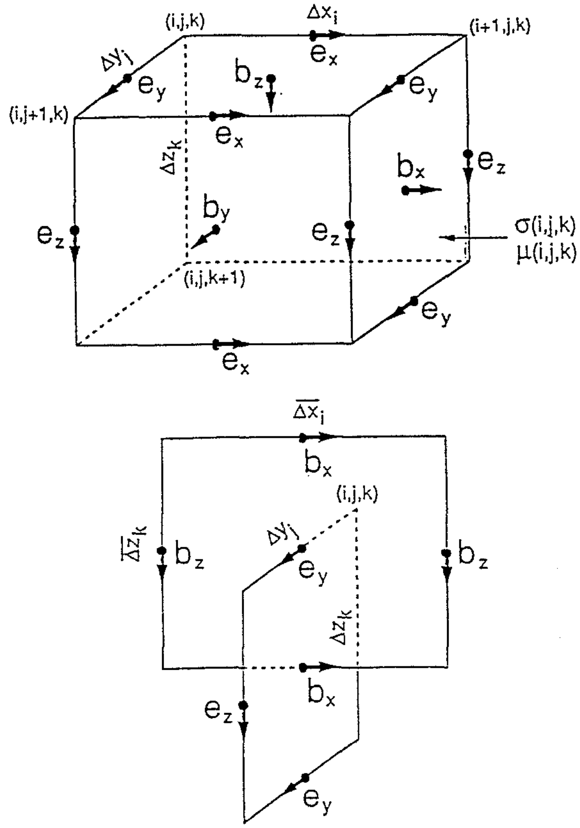

Fig. 2. Staggered grid scheme and orthogonal field pattern.

When applied to the first order Maxwell's equations, this required that a pseudowave term be added to Equation (2) which then becomes

$$\nabla \times \mathbf{h}(\mathbf{r}, t) = \sigma\mathbf{e}(\mathbf{r}\ t) + \gamma\partial_t\mathbf{e}(\mathbf{r}, t) \qquad (3)$$

The added term on the right appears to be a restoration of the displacement current but in fact its magnitude is considerably larger than an actual displacement. Its raison d'etre is to give stability to the solution. γ is chosen as a function of the square of the ratio of the time step to the minimum grid spacing. If the time step is chosen to be too large, the fictitious displacement current can dominate the solution so the time step must be increased only gradually. The process of scheduling time-step increases is discussed in Oristaglio and Hohmann (1984).

Ideally, one might use absorbing boundary conditions (Engquist and Majda, 1977) to limit the extent of the grid. Since diffusion processes imply a spread of phase velocities, it is quite difficult to define appropriate absorbing boundary conditions for this problem. Thus the boundaries at the side and at the bottom were taken out far enough for homogeneous Dirichlet boundary conditions to

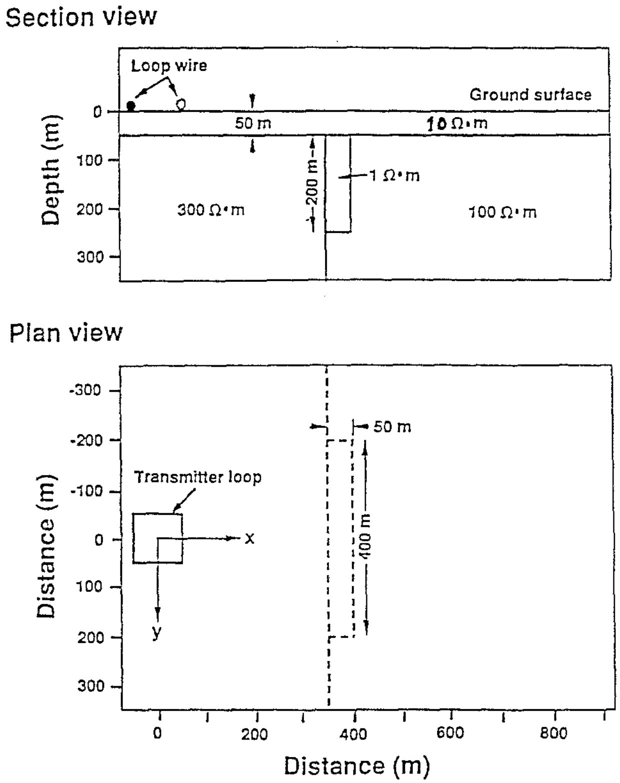

Fig. 3. Model with target in a non-uniform basement.

apply. Since Laplace's equation can be used to compute the magnetic field in air, an analytic continuation process was used to compute the fields in air. Thus the top of the grid could be terminated one layer of nodes above the earth's surface.

The choice of the initial time step is crucial because if it is too small, the solution will take too long and additional errors could develop before the desired end time was reached. The maximum t_0 is determined by how long it takes the source field to reach the first heterogeneity because the initial values are set assuming a uniform half-space. During the earliest time steps, a fourth-order difference scheme is used so that the very high frequency components are adequately modelled. As these die out, the second order scheme is used.

Figure 3 illustrates an important type of model which cannot be computed using either Sugeng and Raiche's compact finite-element method or Newman and Hohmann's integral equation method because the host resistivity is no longer restricted to vary only with depth. There is good agreement on the TDEM response of this model between Wang and Hohmann's method and the Spectral Lanczos method of Druskin and Knizhnerman.

At the present time, the numerical procedure of Wang and Hohmann is used to compute the total rather than the secondary field. This necessitates a relatively fine mesh in comparison to that which would be required for computing the secondary field. However, the alternative required computing the primary field analytically at every mesh point at every time step which proved to be more computationally expensive than using the fine mesh.

B. The $t - k$ method – time-stepping in the spatial frequency domain

The use of numerical derivatives necessitated by finite difference/element modelling methods can be an unwanted source of error. To some degree, this can be alleviated by the use of fitting splines in the pre-derivative stage, or apparently better yet, through the use of staggered grids. Seunghee Lee (1991) sidesteps this problem completely by Fourier-transforming Maxwell's equations from the spatial domain (x, y, z) to the spatial frequency (or wavenumber) domain $(k_x,\ k_y,\ k_z)$. Spatial derivatives in the wavenumber amount to simple multiplication; e.g., differentiating with respect to x consists of multiplying the quantity by ik_x where $i = \sqrt{-1}$.

Lee defines two scalar potentials f and g in terms of the vertical magnetic field $\hat{H}_z$ and the vertical induced current $\hat{J}_z$ where the hat denotes that the variables are defined in the wavenumber domain.

$$f(k_x, k_y, k_z) = i\frac{(k_x^2 + k_y^2 + k_z^2)}{(k_x^2 + k_y^2)}\hat{H}_z \tag{4}$$

$$g(k_x, k_y, k_z) = \frac{k_x}{(k_x^2 + k_y^2)}\hat{J}_z \tag{5}$$

By writing Maxwell's equations in the wavenumber domain and performing some simple algebraic manipulations, the currents can be expressed in terms of these potentials as

$$\hat{J}_x = k_y f - k_x g \tag{6}$$

$$\hat{J}_y = -\ k_x f - k_y g \tag{7}$$

$$\hat{J}_z = \frac{k_x^2 + k_y^2}{k_z} g \tag{8}$$

Further manipulation then yields the equations of propagation as

$$\mu_0 \partial f = \frac{k_x^2 + k_y^2 + k_z^2}{k_x^2 + k_y^2}(k_x\hat{E}_y - k_y\hat{E}_x) \tag{9}$$

$$\mu_0 \partial_t g = \frac{k_z^2}{k_x^2 + k_y^2}(k_x\hat{E}_x - k_y\hat{E}_y) - k_z\hat{E}_z \tag{10}$$

The procedure for solving these equations consists of:

1. At an initial time t_0, H_z, and J_z can be computed analytically in the space domain and transformed into the wavenumber domain. t_0 must be less than the time required for the diffusion front to reach the first heterogeneity. The potentials f and g can thus be computed from Equations (4) and (5).
2. The wavenumber current densities $\hat{J}_x$, $\hat{J}_y$, and $\hat{J}_z$ are computed from these potentials using Equations (6–8).
3. The wavenumber domain electric field $\hat{\mathbf{E}}$ can be obtained either by transforming the wavenumber domain current densities into the space domain and multiplying by the resistivity $\mathbf{E}(x, y, z) = \rho(x, y, z)\mathbf{J}(x, y, z)$ and transforming the result back into the wavenumber domain – or – performing a convolution $\hat{\rho} \otimes \hat{\mathbf{J}}$ in the wavenumber domain. The first of these procedures turns out to be computationally more efficient due to the use of the FFT.
4. The resulting electric fields in the wavenumber domain are inserted into the propagation Equations (9) and (10) and the values of the potentials f and g at the next time step are obtained through the use of an appropriate time differencing formula.
5. These potentials are fed back into step 2 to compute new current densities.

As with Wang and Hohmann (1993) and Oristaglio and Hohmann (1984), it is easy enough to use Euler's method to define a stable time step. However if the computation is to take less time than a geological process, methods must be found to increase the step size as the computation progresses. Lee (1991) makes an analogy with the exponential time decay of a distribution of magnetic dipoles in wavenumber domain, and uses this to derive his time-differencing procedure.

Once again, since absorbing boundary conditions are difficult to apply to diffusion problems the mesh was extended sufficiently far to allow homogeneous Dirichlet conditions to be employed. Also, analytic continuation was used at the earth's surface so that the mesh did not have to be continued into the air region.

Lee (1991) was able to use an FFT method for transformations between the wavenumber and spatial domains for a non-uniform, monotonically expanding grid through the relationship

$$F_{l,m,n}(\alpha\delta x, \beta\delta y, \gamma\delta z) = FET^{-1}\left\{f\left(\frac{k_x}{\alpha}, \frac{k_y}{\beta}, \frac{k_z}{\gamma}\right)\right\}$$

Lee achieved reasonable agreement when comparing his solutions with analytic half-spaces, integral equation solution for a block in a half-space, and semi-analytical solutions for a sphere in half-space. The problem is that sharp boundaries require very high Fourier bandwidths. As the resistivity contrast increases, the errors associated with limiting this band increase rapidly. Lee states that the maximum practical resistivity contrast for the $t - k$ method is about 10:1.

The $t - k$ method may not appear very useful for modelling structures with sharp conductivity boundaries – but it could be very effective for modelling conductivity structures which can be represented by continuous functions. The importance of this is discussed in the Inversion Commentary section of this paper.

C. 2.5-D (2-D GEOLOGY – 3-D SOURCE)

In addition to the usual justification that the earth's geology can, in many cases be represented by 2-D structure, this model is important for letting us study the EM response of complex structures at a much reduced computing cost compared to that required by equivalent 3-D models. For controlled source modelling, it is necessary to incorporate the 3-D nature of the source field because 2-D approximations don't work very well except perhaps for very long wavelength induction sources.

Flosadóttir (1990) used a local spectral representation to compute the 2.5-D EM response from a distant dipole source using 1-D eigenfunctions. In what follows, we will concentrate on methods which can be used for all sources at all ranges.

Specifying the model as constant along the y-direction, the problem is usually formulated in the spatial frequency (k_y) domain; e.g., the magnetic field $\mathbf{H}(x, y, z)$ is represented as

$$H(x, y, z) = \frac{1}{\pi}\int_0^\infty \mathbf{H}(x, k_y, z)e^{-ik_y y}\,\mathrm{d}k_y$$

Computations for all required fields are then performed in the k_y domain. In earlier work; e.g., (Lee and Morrison, 1985) the problem was formulated in terms of three components of the electric field but this has the conflict that the finite-element solution assumes the continuity of all components whereas, in fact, the fields normal to a conductivity boundary are discontinuous. An alternative (Sugeng and Raiche, 1992) and (Unsworth *et al*, 1991), is to formulate the problem in terms of the along-strike components of the electric and magnetic field E_y and H_y. In addition to reducing the dimensionality of the computation from 3 to 2, one solves for fields which are everywhere continuous.

Defining the primary field as that due to a uniform half-space, the best approach is to solve for the primary field analytically and formulate a finite difference/element solution for the secondary fields E^s_y and H^s_y. Then after manipulating Maxwell's equations in the frequency-domain (Hohmann, 1987) the other two components of the magnetic field can be found in the k_y domain from

$$\hat{H}^s_x = \frac{1}{k_e^2}(-ik_y\partial_x\hat{H}^s_y + \sigma\partial_z\hat{E}^s_y + ik_y\sigma_a\hat{E}^p_z)$$

and

$$\hat{H}_z^s = \frac{1}{k_e^2}(-ik_y\partial_z\hat{H}_y^s + \sigma\partial_x\hat{E}_y^s + ik_y\sigma_a\hat{E}_x^p)$$

where the hat indicates transformed quantities, $\hat{E}_i^p$, is the *i*th component of the electric field in the k_y, domain,

$$k_a^2 = -i\omega\mu_0\sigma_a$$

and

$$k_e^2 = k_y^2 - k^2 = k_y^2 + i\omega\mu_0\sigma .$$

σ is the actual conductivity and σ_a is the anomalous conductivity; i.e., the difference between σ and the half-space conductivity.

In a collaboration initiated by Hohmann (pers. com.) and Rijo, Sugeng and Raiche (1992) solve for $\hat{H}_y^s$ and $\hat{E}_y^s$ using the frontal finite-element method (Irons, 1970) with isoparametric elements. This is similar to Gaussian elimination except that the global matrix is never assembled. Instead, the elimination procedure is performed during the assembly of columns of elements (the front). Then only the non-zero elements of the matrix are actually stored. The back substitution process is performed in the usual way. Thus, the number of arithmetic operations is greatly reduced.

Sugeng and Raiche were interested in constructing software to model the response of 2.5-D models for a wide range of time-domain EM systems, necessitating frequency-domain solutions in the range from 1 Hz to 100 kHz. The ratio of E_y^s to H_y^s decreases linearly with frequency so that at low frequencies, this ratio can be several orders of magnitude. In principle, the solution could be ill-conditioned. However, in the k_y domain, this ratio is much smaller, almost always below 100 for the models studied. For this as well as computational efficiency reasons, the derivatives were performed in the k_y domain rather than in the spatial domain.

Sugeng and Raiche compute 25 k_y domain solutions for k_y values ranging from 10^{-5} to 1, at a density of five per decade, and integrate the subsequent spline representation. The resulting frequency-domain solutions are computed at a density of six per decade, splined, and transformed into the time-domain using the fast Hankel transform coefficients derived from Johansen and Sorensen (1979). The response of different time-domain systems is constructed by convolving the step response with the transmitter waveform and receiver windows in the time domain.

Both Unsworth *et al.* (1991) and Everett and Edwards (1992) were interested in computing the EM response of a mid-oceanic ridge for transmitters and receivers on the ocean floor. Both used linear finite elements. This problem is easier than the air-ground problem because the sea-water acts as an absorber. The absence of an air layer apparently permitted the use of the Gaver-Stehfest inverse Laplace transform algorithm in the time-domain program of Everett and Edwards. This requires considerably fewer "frequency-domain" solutions per time point than is the case with the inverse sine transform as well as allowing the computation to avoid complex arithmetic. In practice, however, the CPU saving is negligible since one must still construct a frequency-domain representation sufficiently broad to serve as a basis for the time-domain range of interest.

Unsworth also used the absence of an air layer to advantage by applying a mixed boundary condition (in essence an infinite exponential element) instead of requiring that the mesh extend sufficiently far from the target to allow the application of a homogeneous Dirichlet boundary condition. His finite element program had another interesting innovation.

Instead of solving coupled equations for $\hat{E}_y^s$ and $\hat{H}_y^s$, he took advantage of the fact that $\hat{E}_y^s$ would be dominant. Thus, he initially set $\hat{H}_y^s$ to zero, solved for $\hat{E}_y^s$, and then used this to solve for $\hat{H}_y^s$. He then used $\hat{H}_y^s$ to update for a new E_y^s and continued this iterative procedure until convergence. This approach greatly reduced storage and CPU times. In principle, this is a subset of a more general class of iterative matrix solutions which could be tried.

In some preliminary comparisons, the frontal finite-element solution of Sugeng and Raiche and Unsworth's iterative finite-element solution with absorbing boundary conditions required similar computation times.

D. Commentary

Mesh design is still a major question. How far do meshes need to extend? How fine do they have to be as a function of conductivity structure? To what degree will increasing the order of the local functions allow the use of coarser meshes? How does this affect time-stepping procedures or the required spectral density for the Lanczos spectral decomposition (Druskin and Knizhnerman, 1988)?

The moving finite-element work of Travis and Chave (1989) is a very interesting first step in this direction. Although it ignores the above questions, it answers a related one. Given N mesh points, what is the most suitable arrangement for them? It would be interesting to see the MFE work extended to the 3-D time domain. In principle, the mesh would simply move with the induced currents as they developed with time. There is a commonality with problems in computational fluid dynamics which we should be exploring.

Sugeng and Raiche (1989) showed that by defining the conductivity at the boundary as the geometric average of the volume-weighted conductivities of surrounding elements (*n*th root of n products), the allowable conductivity contrast increased from 300:1 to over 10,000:1). In essence, this corresponds to using a smoothly varying conductivity contrast and then contracting it back to a line boundary. There is still a tendency to use arithmetic averages in many other modelling programs. Consider that if the conductivity of one element is 1 and that of its neighbour is 10^{-4}, does a conductivity of 0.5 or 10^{-2} represent the better average?

What is the best way to terminate a finite-element or finite-difference mesh? This is crucial in 3-D but it is even important in 2-D. In a paper to be given at this meeting, Sugeng and Raiche simply use a very large mesh and use homogeneous Dirichlet conditions for the secondary field. Unsworth *et al.* (1991) is somewhat more sophisticated by using a mixed boundary condition equivalent to an infinite element at the boundary. A third possibility would be the use of Summary Representation as a hybrid method. Raiche and Tarlowski (1984) showed how to model the response of a heterogeneous region in a host which could consist of uniform blocks of different conductivities. Analytic solutions apply to the uniform blocks. Finite element/difference methods need be used only in the heterogeneous region. In principle, this could be applied to 3-D modelling with the aid of symbolic manipulation software such as Mathematica.

Let me offer the challenge of a very useful, but time consuming research project which would consist of phrasing various forward modelling algorithms in terms of an explicit inversion problem for node positions as well as function coefficients. In particular, it would be invaluable to perform SVD decompositions so that we could understand the interaction of node position with function coefficients through the eigenparameters. The eigenparameters of such an inversion may contain certain combinations of nodes which are non-local, especially for complicated anomalous structures. This may explain why it is so difficult to determine the best node spacings simply on the basis of skin depth.

The last issue concerns the best way to model in the time domain. Time-stepping is possibly the most direct way of time-domain modelling and also offers the opportunity to extend the work of Zhdanov and Frenkel (1983) on inversion as reverse time migration. Wang (pers. com.) has suggested treating the data residuals at each receiver source as artificial sources which radiate backwards in time. There are some disadvantages however. The fact that the explicit time-stepping scheme may be stable does not mean that the correct answer will result. Round-off errors can grow with each step since each computation depends upon all of the preceding ones. There is a lot of art in getting the 3-D time-stepping solution to work.

In this regard, the spectral Lanczos decomposition method (SLDM) of Druskin and Knizhnerman would appear to have an advantage. Because it is a spectral method, the errors do not propagate forward with time. Also, it is believed to be faster since computation time grows only as $\sqrt{t}$. However, it is not clear how one could perform Wang's reverse time migration scheme using SLDM.

Lee's $t - k$ method is quite interesting but the fact that large conductivity contrasts will blow out the computing times because of the need for a larger k-band is disconcerting. However, it may be regarded as an important advance for computing the response of structures with smoothly varying conductivity.

Computing time-domain solutions by transforming frequency-domain solutions has several advantages. Firstly, it is easy to add new transmitters and new waveforms at minimal computational costs. Secondly it is a spectral method so each frequency solution can be computed independently. Thus representing the frequency spectrum as a spline function constrains the error propagation in one part of the spectrum from propagating as a function of time. Most models can be computed using 30 frequency-domain solutions. Both Newman and Hohmann (1986), and Sugeng and Raiche (1989) have demonstrated that even for very high contrast models; i.e. greater than 10,000 to 1, six frequency-domain solutions per decade provide adequate sampling. Indeed, as spectrum prediction methods improve, even fewer FD solutions will be needed. Certainly we should be experimenting with the coupled first-order equation, staggered grid method of Smith (pers. com.) in the frequency domain.

Direct time-domain methods have the disadvantage of an additional mesh variable, the time step. How ıs this affected by conductivity contrast? To date, no high conductivity contrast models have been run and no late time solutions have been published. However, frequency-domain solutions have had a much longer maturation time than have direct time-domain solutions so making a definitive judgement at this stage is premature. The important point is that new ground has been broken and there is considerable room for developing these new methods further.

4. Inversion

Having previously discussed the inversion aspects of forward modelling, this section will concentrate on the inversion problem of estimating conductivity structures from EM data. I want to concentrate on new styles of inversion which have not yet been published rather than review the body of existing published work. Having said this, I would be remiss in not mentioning that details of two outstanding advances in inversion methodology, which were discussed at the 1990 Mexico meeting, have since been published: the approximate inverse method (Oldenburg and Ellis, 1991) and the Smith and Booker (1991) rapid inversion for magnetotelluric data. Both of these are undergoing maturation; i.e., being tested on a number of problems.

In what follows, I discuss Pellerin's version of distorted Born inversion followed by Ki-Ha Lee's innovative work in the Q-domain. The profile analysis approach of Silic is presented as a counterpoint to the heavily mathematical paradigms. Methodologies based on metaphors of Nature; i.e., genetic algorithms, simulated annealing, and neural nets are the subject of section 5.

A. Distorted born approximation inversion of time-domain soundings

The motivation behind this work (Pellerin, 1992) is to use 3-D inversion methods to improve 1-D resistivity estimates from time-domain sounding data. Time-domain

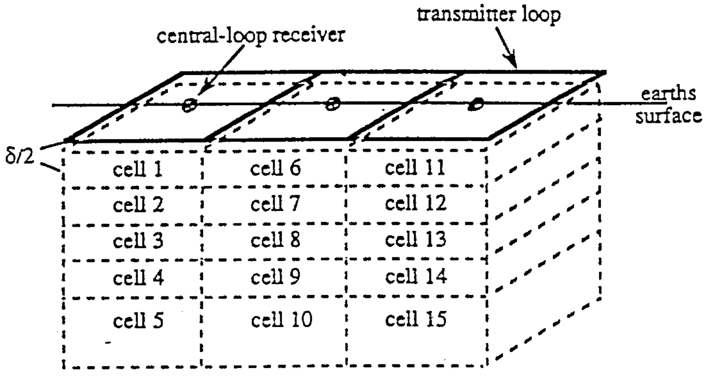

Fig. 4. Block model schematic for Born inversion.

soundings are made over each column of blocks comprising the model (Figure 4) and the results converted to frequency domain. The model consists of block structure anomalies in an otherwise uniformly layered half-space. The dimensions and discretisation of the heterogeneous region are fixed throughout a given inversion run. The initial guess for the 3-D model is obtained using either the image solution of Eaton and Hohmann (1989) or common 1-D layered earth inversion methods. The resistivity and thickness of the layers are also obtained using these initial 1-D methods.

The basic concept of this inversion is to realise that the Frechet derivative is inherently contained in the integral equation formulation. The perturbed vertical magnetic field, H_z^+, arising from a change in the conductivity of the anomalous blocks, $\delta\sigma$, can be computed from the previous magnetic field H_z using the integral equation:

$$H_z^+(\mathbf{r}_0) = H_z(\mathbf{r}_0) + \int [\overleftrightarrow{\mathbf{G}}^H(\mathbf{r}_o, \mathbf{r}') \cdot \mathbf{E}(\mathbf{r}')]_z \delta\sigma(\mathbf{r}') \, \mathrm{d}V' \tag{11}$$

where $\overleftrightarrow{\mathbf{G}}$ is the 3-D (not the layered-earth) magnetic Green's tensor, **E** is the previously computed total electric field in the target, $\mathbf{r}_o$ and $\mathbf{r}'$ are the "observation" and "source" points respectively. **E** is assumed constant within each cell.

The Born approximation consists of assuming an initial value for the electric field derived from the 1-D estimates and neglecting the second order term $\delta\mathbf{E}\delta\sigma_a$ and updating the field iteratively. The distorted Born approximation updates the Green's tensor at each iteration as well as the electric fields. Thus it can recover a much wider range of conductivity contrasts.

From Equation (11), it can be seen that $D_{i,j}$, the Frechet kernel relating the change in response of the ith data point, δH_i, to the change in conductivity in cell j, $\delta\sigma_j$, is simply

$$D_{i,j} = \mathrm{vol}_j[\overleftrightarrow{\mathbf{G}}^H(\mathbf{r}_i, \mathbf{r}_j) \cdot \mathbf{E}_j(\mathbf{r}_j)]_z \tag{12}$$

Initially, the inversion begins with the 1-D layered earth magnetic Green's tensor. However, since a new electric field is computed at each inversion iteration, the 3-D magnetic Green's tensor elements are computed by invoking reciprocity. In particular, the z component of the magnetic field at $\mathbf{r}_0$, $G_{zi}(\mathbf{r}_0, \mathbf{r}')$ due to the ith component of a unit electric dipole at $\mathbf{r}'$ is related to the electric field E_{iz} at $\mathbf{r}'$ due to a unit magnetic dipole at $\mathbf{r}_0$ as

$$G_{zi}(\mathbf{r}_0, \mathbf{r}') = -\frac{1}{i\omega\mu} E_{iz}(\mathbf{r}', \mathbf{r}_0)$$

Thus as new electric fields are computed, the Green's tensor is updated. In discretised form, Equation (11) becomes

$$\delta H_i = H_i^+ - H_i = \sum_{j=1}^{M} D_{i,j}\delta\sigma_j\,. \tag{13}$$

In what follows, H_i^d is defined as the ith data point, H_i^k as the data value after k iterations, and e_i^k as the residual error.

$$H_i^{k+1} = H_i^k + \sum_{j=1}^{M} D_{i,j}^k \delta\sigma_i^k \tag{14}$$

$$e_i^k = H_i^d - H_i^k\,.$$

Thus the expression for the error residual to be minimised is:

$$e_i^k - e_i^{k+1} = \sum_{j=1}^{M} D_{i,j}^k \delta\sigma_i^k \tag{15}$$

Three regularisation schemes were tried for the least squares solution of Equation (15). The first (Fullagar and Oldenburg, 1984) creates an under-determined problem of dimension $M + N$ by using the future residual e_i^{k+1} to augment the Frechet kernel such that

$$\mathbf{e}_i^k = \sum_{j=1}^{M+N} \Gamma_{i,j}^k \delta\mathbf{m}_j^k$$

where for $j = 1, M$,

$$\Gamma_{i,j}^k = \mathbf{D}_{i,j}^k, \qquad \delta\mathbf{m}_j^k = \delta\sigma_j^k$$

and for $l = 1, N$

$$\Gamma_{i,l+M}^k = \delta_{i,l} \qquad \delta\mathbf{m}_{i,l+M}^k = \mathbf{e}_j^{k+1}$$

where $\delta_{i,l}$ is the Kronecker delta. This leads to the minimum length solution

$$\delta\mathbf{m}^k = \overleftrightarrow{\Gamma}^T(\overleftrightarrow{\Gamma}\overleftrightarrow{\Gamma}^T)^{-1}\delta\mathbf{e}^k$$

Philosophically, this approach was interesting because it explicitly attempted to minimise the future error rather than setting it to zero. Unfortunately, in spite of attempts to scale the matrix, it produced steepest descent step sizes which were far too small for practical inversions.

This approach was abandoned in favour of two, more traditional approaches. The first, referred to as the smallest step solution, was the familiar Marquardt method where one minimises the functional

$$E = \mathbf{e}^T\mathbf{e} + \lambda\delta\mathbf{m}^T\delta\mathbf{m}$$

The second approach, called the flattest step solution, is basically a second order version of the recent craze for smoothness. Instead of minimising the roughness between adjacent parameters, Pellerin minimised the roughness in the change of adjacent model parameters by minimising the functional

$$E = \mathbf{e}^T\mathbf{e} + \lambda(\Delta\delta\mathbf{m})^T(\Delta\delta\mathbf{m})$$

The inversion method consisted of first obtaining an initial parameter estimate using either the 1-D image inversion of Eaton and Hohmann or layered-earth inversion. At each iteration, an integral equation was used to compute the electric and magnetic fields, the Frechet kernel and the updated residual. Then either the traditional Marquardt method, or the flattest step method was applied to update the parameters.

In comparing the smallest step and the flattest step methods on artificial data, Pellerin found that when the Lagrange multiplier was sufficiently reduced, both methods converged near the same solution. When $\lambda > 1$, the smallest step method creeps towards a solution. The flattest step method is the more robust of the two if the starting model is far from the true model but the final solution exhibits horizontal banding as could be expected. Since the aim of her work was to develop a scheme to refine 1-D inversions, where the starting model could be expected to be near the final model, Pellerin used the smallest step method.

Pellerin applied her methodology to a number of low contrast problems, one of which is shown in Figure 5, representing a conductive contaminant plume. The inversions which used initial estimates from half space inversions resolved the location of the plume bottom but tended to shift the top of the plume upwards and overestimated the plume resistivity. The half-space resistivity was, of course, well resolved. The starting model based on image inversion does the reverse. The top of the plume and its resistivity are well resolved but the base of the plume and the host resistivity are not as well resolved.

There are a number of other interesting issues in her work such as stopping criteria and data partitioning which are still under investigation.

Newman (1992) used Born inversion also (Born again) to invert single frequency data for cross hole transmitters and receivers. Although much of what he did was similar to the work of Pellerin, he used different regularisation procedures. Be-

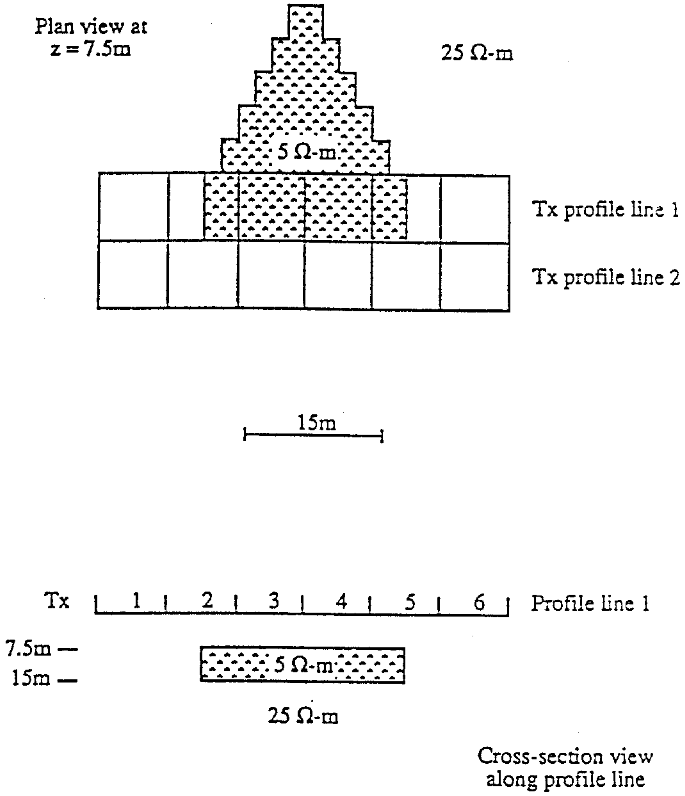

Fig. 5. Pollution plume model.

cause of the similarity and the fact that he will be presenting his work in this session (Newman, 1992), I will not include details here.

B. Q domain – a pseudo wave equation approach

For distances beyond a few metres, the conductivity of the earth limits EM propagation to being a diffusion process. In addition to the implied lack of resolution, this poses a concomitant computational difficulty. Fields that satisfy the diffusion equation are global in the sense that their value at any one point of space-time, is a consequence of the entire history of the field in that domain.

The Q-domain inversion method of Lee and Xie (1993) starts from the point that diffusion fields can be related uniquely to a fictitious wave field, defined in what is termed the Q-domain, through an integral transform. Thus it should be possible to analyse EM data using techniques developed for seismic data analysis. The basic idea of Q-domain inversion is to transform magnetic field data into pseudo-wave field data, compute travel times, and then use tomographic methods to construct a conductivity map.

In the frequency-domain, one cannot help but notice that the diffusion equation can be transformed into the wave equation by replacing the real angular frequency ω with a pseudo-frequency ν defined as $\nu = \sqrt{i\omega}$. This has prompted several papers such as Isaev and Filatov (1981), and Filatov (1984) to investigate the link between scalar diffusion fields and the corresponding wave equations in the time domain. Lee and Morrison (1989) generalised the relationship to include vector EM fields.

The starting point is the time-domain equation for the magnetic field, $\mathbf{H}(\mathbf{r}, t)$ in a homogeneous region (ignoring the displacement current)

$$\nabla^2\mathbf{H}(\mathbf{r}, t) - \mu\sigma(\mathbf{r})\frac{\partial}{\partial t}\mathbf{H}(\mathbf{r}, t) = -\nabla \times \mathbf{S}(\mathbf{r}, t)$$

where $\mathbf{H}(\mathbf{r}, 0) = 0$. An equivalent pseudo-wave equation can be written in the Q-domain as

$$\nabla^2\mathbf{U}(\mathbf{r}, q) - \mu\sigma(\mathbf{r})\frac{\partial^2}{\partial q^2}\mathbf{U}(\mathbf{r}, t) = \mathbf{F}(\mathbf{r}, q)$$

with

$$\mathbf{u}(\mathbf{r}, 0) = 0; \qquad \frac{\partial}{\partial q}\mathbf{U}(\mathbf{r}, t)\big|_{q=0}$$

If the same boundary conditions hold in each domain, then Lee and Xie (1993) show that the magnetic field $\mathbf{H}$ and the wave field $\mathbf{U}$ are related uniquely as

$$\mathbf{H}(\mathbf{r}, t) = \frac{1}{2\sqrt{\pi t^3}}\int_0^\infty q e^{-q^2/4t}\mathbf{U}(\mathbf{r}, q)\, \mathrm{d}q \tag{16}$$

This wave field propagates with a velocity which is inversely proportional to the square root of conductivity. Thus if one can use tomographic techniques to map the slowness (square root of the conductivity permeability product), the conductivity map follows directly.

There were two essential steps to making this procedure work. The first was to find a method to invert Equation (16) in order to be able to transform time-domain magnetic field data into Q-domain waveform data. The second was to develop a travel-time tomography algorithm capable of handling high conductivity contrasts. (In seismic applications, slowness contrasts seldom exceed a factor of two.)

The inversion of Equation (16) is an ill-posed problem due to the strongly damped kernel. Lee and Xie solved this problem using a singular value decomposition technique. First, Equation (16) is discretised in q and t using the trapezoidal rule and collocation to produce the matrix equation,

$$H = AU\,. \tag{17}$$

An important point is the discretisation of t into n values t_i where

$$t_i = \frac{(i-1)\Delta T}{(n-1)}$$

and ΔT is the time-window occupied by the time-domain window. In previous work, Lee and Morrison (1989) developed an inversion for Equation (16) which required four decades of time-domain or frequency-domain data with a maximum allowable noise level of 3 percent. Since no EM system yet exists with this wide-band capability, a better transform method was needed. By expressing A in the singular value decomposed form $A = WDV^T$, the regularised solution of Equation (17) can be formally expressed as

$$U = V(D^2 + \alpha I)^{-1}\, DW^T H$$

Lee and Xie used the "quasi-optimality' method proposed by Leonov (1978) to compute α. In this method α is chosen as the smallest positive root of the equation

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\left\{\left(\frac{\eta}{\alpha}\right)^2 + \alpha^2 \sum_{k=1}^{p} \frac{\bar{u}_k}{(\alpha + \lambda_k^2)^4}\right\} = 0 \tag{18}$$

where η is the norm of the residual of the null space of A, λ_k are the p non-zero eigenvalues of A, and $\bar{u}_k$ are the corresponding non-zero members of $DW^T H$. The parameter η depends upon H. When the data are noisy, η will increase resulting in a smoother wave field solution with decreased resolution.

In order to test the above transform from time domain to Q-domain, Lee and Xie (1993) computed time-domain magnetic fields from a magnetic dipole in a whole space, computed the wave field in Q-domain, and compared the travel times with those computed directly from the time-domain data. This was done for a number of values of α, and different time windows. The agreement was good provided that α was picked within the minimum and maximum bounds of the solutions to Equation (18). In particular for time windows of only 1.5 decades, spatial resolutions of better than 1 percent of transmitter-receiver separation were obtained. Added noise did not significantly affect travel time estimates as long as the noise level was kept under a 3–5 percent limit.

Non-linear methods are necessary for constructing a tomographic algorithm for EM inversion because linear ray paths cannot be used. Since the slowness model depends on the ray paths and the ray paths depend on the slowness model, this becomes an iterative process. Assuming a 2-D model such that x is the horizontal propagation direction and z the depth, a discretised slowness model is constructed which is piecewise linear in x and cubic spline continuous in z, in each element. Thus the ray paths are smooth and can bend continuously. Using Fermat's principle that the travel time between any two points must be a minimum, a two point ray tracing algorithm is used to construct the paths. A consequence of the method is

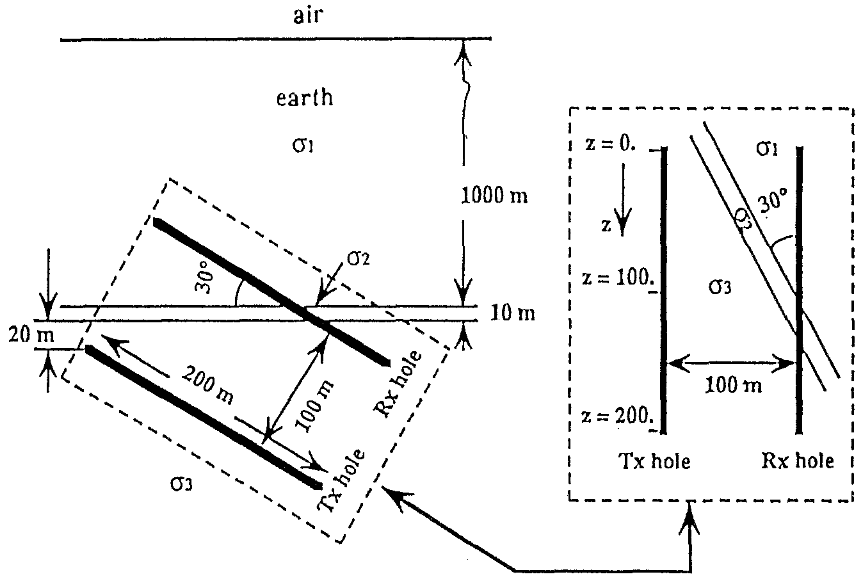

Fig. 6. Cross-hole inversion target for Q-domain inversion.

that the rays can move only in the positive x-direction so that backscattering is not allowed.

Having constructed the ray paths, the travel time data is used to construct an updated slowness model. This once again is a nonlinear process requiring regularisation methods. Interested readers are referred to the derivation in Lee and Xie (1993).

In order to test the method, Lee and Xie used the three layer model shown in Figure 6 which was rotated 60 degrees so as to simulate a dipping dyke between two boreholes. The conductivity values on the diagram represent a conductivity contrast of 10 to 1, the highest contrast studied by Lee and Xie.

Lee and Xie constructed Q-domain data sets from magnetic field data computed in the time range 0.01 to 0.5 ms. They were able to recover both the location of the dyke as well as the three conductivities quite well in 120 iterations requiring about 5 hours CPU time on a SPARC2.

The results of Lee and Xie are quite impressive; and they leave some interesting questions to research. Can this method be extended to work for conductivity contrasts in excess of 10:1? Will it work for more complex models? Would it be better to abandon Q-domain and use the new tomographic procedures directly in time domain?

C. Time-domain pattern recognition studies

Over the past decade, exploration geophysicists have derived empirical rules for analysing time-domain EM data sets. These are based largely upon bumps and

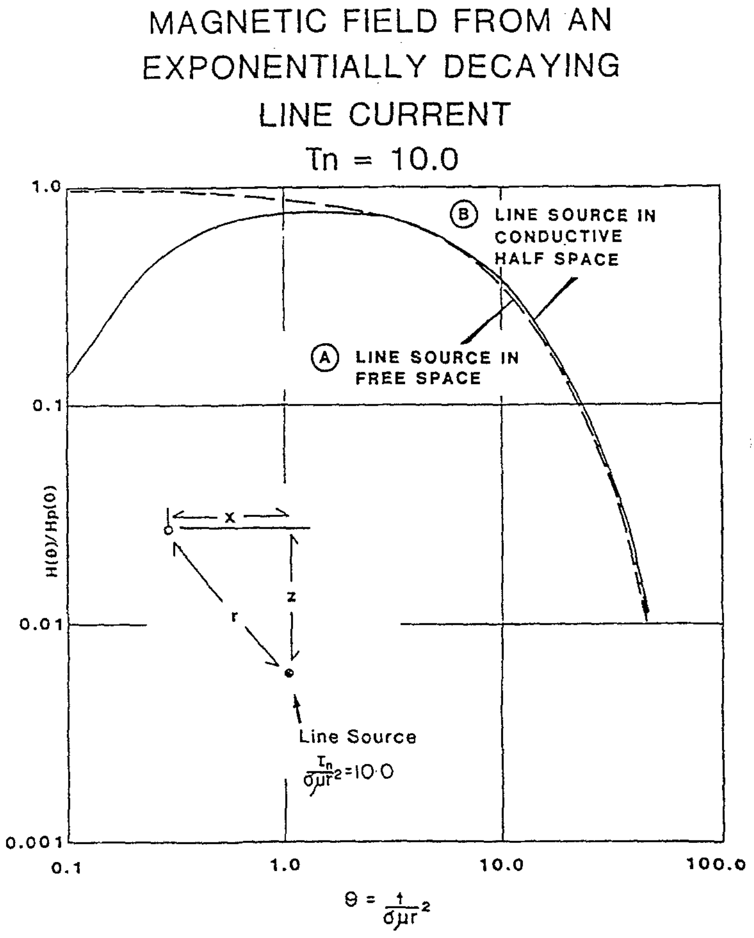

Fig. 7. Free space vs half-space decay of line current source.

cross-overs (sign changes) in the profiles (at specific delay times) of vertical and horizontal components of the magnetic field and its time derivative. However, using raw profile data alone, it is difficult to distinguish between responses due to a broad step in the overburden from those due to deeper conducting bodies. Silic (pers. com.) has developed a method based upon combinations of spatial derivatives, which has proven effective in separating the responses of different conductors both in analog model studies and field data. The background to this work can be found in Silic (1989)

Silic compared the response of a line current source in a conductive half-space with that of the same source in free-space. As can be seen from Figure 7, the two responses are very close for delay times, $t_n > \sigma\mu r^2$ where σ is the half-space conductivity and r is the distance between the receiver and the buried line source.

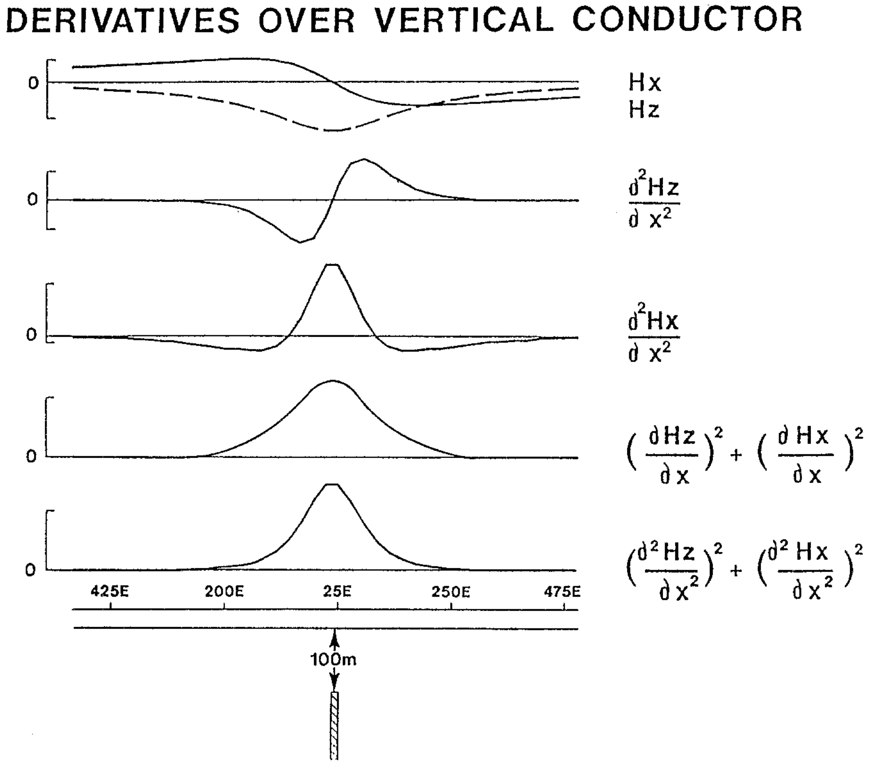

Fig. 8. Derivatives over vertical conductor.

The basis for this is the fact that if the receiver is in air, then at any specific delay time, the EM diffusion equation reduces to Poisson's equation. Thus, the time-domain response can be accurately modelled by DC current sources, the location of which move with time. When discrete conductors are present, these equivalent DC current sources will become stationary at the location of the conductor during a delay time window, the width and position of which depend upon relative conductivities and geometry. This is the basis of Barnett's DC current filament inversion method (Barnett, 1984) which has proven to be quite successful for locating thin sheet targets.

Thus, $t_n > \sigma\mu r^2$ corresponds to a regime where equivalent DC currents are stationary within discrete conductors. Moreover, potential field theory can be invoked to show that the horizontal derivatives of the response can be computed as the weighted sum of edge currents where the weights correspond to the appropriate *free-space* Green's function.

Silic used analog modelling to obtain the time-domain response of a number of simple models. Figure 8 shows the response of different horizontal derivative

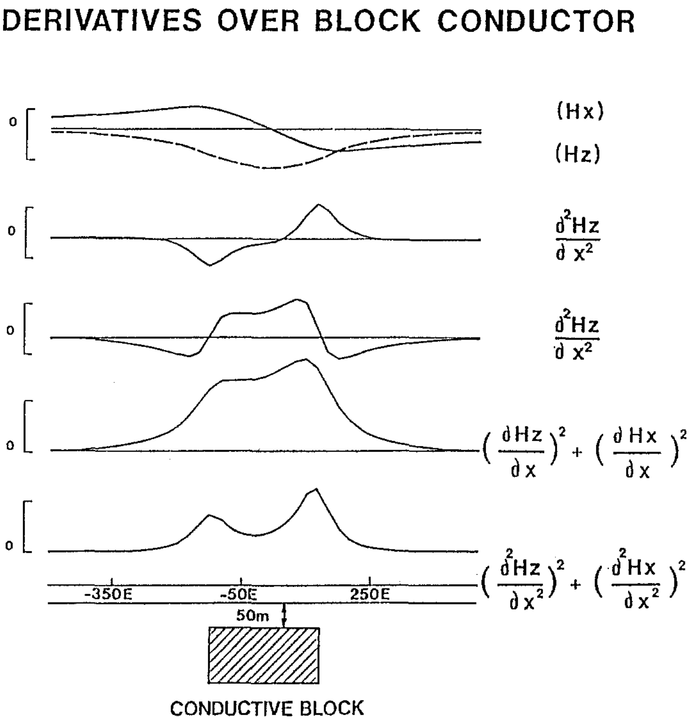

Fig. 9. Derivatives over block conductor

combinations of the time-domain response of a vertical dyke. The half-width of the second derivative responses is proportional to the depth of burial. Figure 9 shows the responses over a finite block model. The peaks appear over the edges. The responses of the two vertical edges add linearly consistent with the DC regime assumption. For these simple targets, all of the derivative combinations give useful information.

From analog model data for the more complicated structure shown in Figure 10, only the indicator $(\partial^2 H_z/\partial x^2)^2 + (\partial^2 H_x/\partial x^2)^2$ shows a pattern which is diagnostic of the whole model. Peaks appear over the edges of the overburden steps. There is a long wavelength feature corresponding to the deeper vertical dyke.

Analog model data is one thing, field data another. The presence of temporal and geologic noise makes it difficult to perform meaningful spatial derivatives on field data. However Silic smoothed his data using cubic spline techniques which minimised the power in the higher derivatives. When applied to field data from exploration surveys, Silic's interpretation technique has successfully identified drilling targets in a complex, deformed terrain.

This methodology is significant for a number of reasons. Firstly it has shown the value of potential field edge-finding techniques for qualitatively interpreting the time-domain EM response of complicated conductivity structures where more

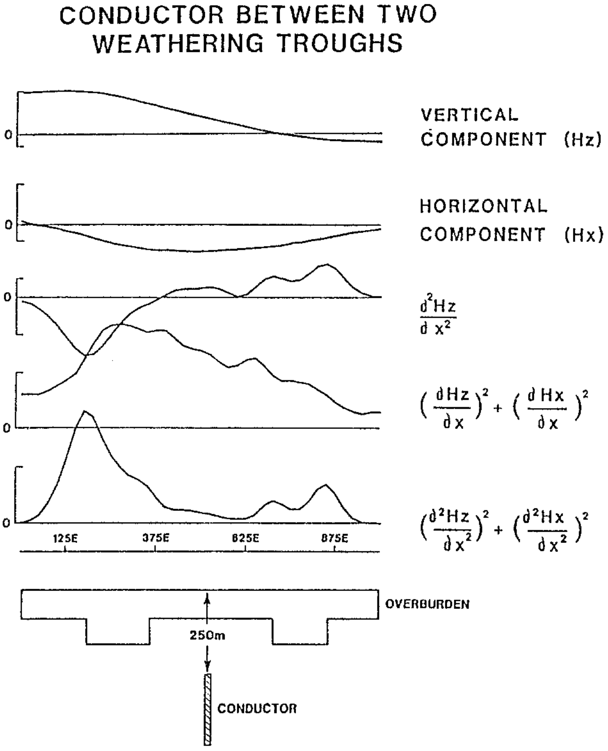

Fig. 10 Derivatives over conductor between two weathering troughs.

than one conductor is present. Secondly, it can provide starting models for more quantitative mathematical inversion methods. Silic is currently working on DC inversion techniques to make the method yield more quantitative estimates of size and depth.

D. Commentary

Consider an inversion problem with ND data points and MP unknown parameters. Suppose, for the moment, that we represent the linearised part of an inversion by the linear equation

$$A\mathbf{u} = \mathbf{g} \tag{19}$$

where the matrix A contains the physics of the problem, $\mathbf{u}$ is an MP dimensional vector containing the unknown model parameters, and $\mathbf{g}$ is an ND dimensional vector containing the data or some appropriate functional thereof. Let us also assume for the moment that our model consists of a 2-D or 3-D half-space divided into MP cells and that our inversion problem is to find the conductivity of these cells from ND data points. In general, this will be an under-determined problem, even if MP < ND. To paraphrase Lanczos (1961), this is because the matrix A is not activated in all the dimensions of the spaces spanned by $\mathbf{u}$ and $\mathbf{g}$. One way to solve this problem is to increase the activation of A by including model prejudice.

This strategy was employed in a classic paper on stochastic inversion by Franklin (1970). He augmented Equation (19) by including an unknown noise vector $\mathbf{n}$.

$$A\mathbf{u} + \mathbf{n} = \mathbf{g} \tag{20}$$

The unknowns, $\mathbf{u}$ and $\mathbf{n}$ are regarded as samples drawn from two random processes. Equation (20) now expresses the stochastic relationship between signal, noise, and data processes. In this context, inversion consists of asking which random variable obtained from the data process is the best approximation to the random variable acted on by the signal process? What does best mean?

Working from a signal processing point of view, Franklin (1970) chose "best" as finding a solution using an autocorrelation function, R_1, which has been constructed from some a-priori conviction concerning the size and smoothness of an admissible solution.

Assuming that the signal and noise processes are independent, Franklin (1970) thus computes the best estimate for the model $\mathbf{u}$ in the form:

$$\mathbf{u} = R_1 A^T (A R_1 A^T + R_2)^{-1} \mathbf{g} \tag{21}$$

where R_2 is the noise autocorrelation. Assuming white noise of small amplitude, this noise correlation matrix has the form $R_2 = \lambda^2 I$ where I is the identity matrix. Thus Equation (21) can be expressed

$$\mathbf{u} = R_1 A^T (A R_1 A^T + \lambda^2 I)^{-1} \mathbf{g} \tag{22}$$

which looks very much like regularised inversion solutions. Choosing various forms for R_1 gives the inversion great power to explore different model constraints. Indeed, we can see that several of the various inversion paradigms which are used in EM are simply special cases of Equation (22). The autocorrelation function R_1 can be used to force model coherency or to enforce smoothness constraints.

We can phrase the ideas of Equation (22) in a more familiar way as follows.

$$\phi(\mathbf{m}, \mathbf{m}_0 \mathbf{d}, \mathbf{d}_0) = f(\mathbf{m}, \|\mathbf{d} - \mathbf{d}_0\|) + \lambda g(\|\mathbf{m} - \mathbf{m}_0\|) \tag{23}$$

The first term on the right, f represents choosing a set of model parameters, $\mathbf{m}$ which minimise the misfit between model data $\mathbf{d}$ and observed data $\mathbf{d}_0$, subject to appropriate regularisation techniques. The second term, g seeks to minimise the deviation of the inverted model parameters $\mathbf{m}$ from a model $\mathbf{m}_0$ with pre-specified properties. λ is a Lagrange multiplier. Often, the model misfit component tries to enforce smoothness constraints on adjacent cell conductivity values, producing the smoothest model consistent with a level of data misfit.

Using a non-zero λ corresponds to adding geological prejudice to somehow compensate for the lack of uniqueness caused by data noise and incomplete data. It is equivalent to solving an equation by specifying a particular solution and adding it to the general solution of a homogeneous equation. To quote from Lanczos (1961): "While this procedure is formally correct, it has the disadvantage that the 'particular solution' from which we start, can be chosen with a high degree of arbitrariness. This hides the fact that our operator gives a very definite answer to our linear problem in all those dimensions in which the operator is activated, and, on the other hand, fails completely in all those dimensions in which the operator is not activated. But then it seems more adequate to the nature of the operator to give the unique solution in all those dimensions in which this solution exists and ignore those dimensions which are outside the realm of the operator. If we give some particular solution, this condition is not fulfilled because in all probability our solution will have some projection in the field V_0 which is not included by the operator."

Thus, performing a grossly under-determined inversion, constrained by requiring the smoothest model, is by no means the least presumptuous solution. It is, in fact, a very strong assumption which will yield solutions outside the space spanned by the operator which links the physics of the process to the data.

If we solve for the natural inverse of Equation (19) (equivalent to setting $\lambda = 0$ in Equation (23)) using the Jupp SVD regularisation algorithm (Jupp and Vozoff, 1975), we can gain a great deal of insight into the inversion process regardless of whether MP < ND or vice versa. This regularisation method is based upon the NSR (noise to signal ratio) and the natural eigenparameters of the model rather than preconceived conductivity distributions; i.e., it does not act outside the space spanned by A in Equation (19). In essence, it damps the effect of eigenparameters whose relative singular eigenvalue (normalised with respect to the largest one) is below a specified threshold. Initially, this threshold is set quite high (usually above 10 percent) and is allowed to decrease down to the NSR value as a function of data misfit. Thus, in the early stages, only the most important eigenparameters determine the solution path. The less important eigenparameters come into play, only as the data misfit decreases. The only remnant smoothing occurs as a result of excluding the effect of eigenparameters whose relative singular eigenvalue is below the estimated NSR. The point is that this type of smoothing (including the selection of the Marquardt parameter) is a data driven consequence of the model rather than a consequence of the ideology of the person doing the inversion.

Moreover, we can use this method to adjust the grid to reduce the level of non-uniqueness. The eigenparameters corresponding to large eigenvalues contain information as to which combinations of physical parameters are important in fitting the data. Similarly, those corresponding to very small eigenvalues indicate which physical parameters are unresolvable. Thus for a grid cell model, this analysis indicates which cells can be subdivided and which can be combined. Thus the model becomes rougher in some parts and smoother in others, once again, entirely driven by the data rather than preconception.

This inversion paradigm yields two additional pieces of information. One is the noise to signal ratio which can be used to govern the regularisation process in a subsequent inversion. The second is the average predicted residual error (APRE); i.e., the error which would arise in predicting a data value if each data point were removed, one at a time; the process averaged over all data points. When the number of parameters for a given model is consistent with what is actually activated by the EM field, APRE will have a value consistent with the standard error. If the model is appreciably over or under parameterised, APRE will be significantly higher than the standard error (Hohmann and Raiche, 1987).

One point, which doesn't seem to be well appreciated, is that prior geological information can be introduced into the inversion via the starting model without the need for constraints. If the inversion is well conditioned, and the results do not support the form of the initial model, then it can be truly said that this information is inconsistent with the data. Moreover, the use of constraints distorts the formation of the true eigenparameters of the model.

Regardless of whether or not one chooses to use explicit constraints, realistic 2-D and 3-D inversion can impose such a heavy computational burden that one is driven to look for ways to reduce the dimensionality of the problem. Oldenburg *et al.* (1992) have developed subspace methods which limit the number of search vectors and thus dramatically reduce the computational burden. They still use explicit model constraints in the solution. In (Oldenburg, 1992) this method was applied to a 3-D DC problem. They are currently applying this paradigm to a number of different inversion tasks such as 2-D MT inversion.

At this point we can mention two different inversion philosophies. One seeks to invert for the whole subsurface conductivity map in some detail. In this case, the above subspace approach represents a very interesting advance. An alternative philosophy (which would also permit a substantial reduction of the computational problem) consists of an initial localisation of distinct anomalous regions followed by a more detailed inversion to better resolve the anomalous structure. Thus, it would use a relatively fine discretisation over the anomalous regions and a very coarse discretisation to cover the rest of the illuminated volume. Eigenparameter analysis could be used to adaptively adjust the discretisation as the inversion proceeded.

The initial localisation can be accomplished in a number of ways. The most obvious is to use stitched 1-D inversions to locate regions of interest and restrict 2-D or 3-D grid cell inversions to these regions. (Pellerin, 1992) is an example of this procedure. Alternatively, one could use data processing techniques such as the derivative techniques of (Silic, 1989) to localise the structures of interest. One of the implicit strategies of this method is to locate equivalent edge currents. This is quite interesting because; in fact, the magnetic and electric fields which one measures are caused by induced current distributions which are related to conductivity structures. Thus the equivalent current filament method (Barnett, 1984) and the equivalent dipole sources method (Macnae *et al.*, 1991) can be regarded as variations on the same theme – finding the currents which denote anomalous conductivity zones.

However, an inversion procedure based upon localisation of anomalous regions followed by detailed parametric estimation will be of limited success when the underlying conductivity structure is smoothly varying. On the other hand, it seems ridiculous to invert for say 10,000 unknowns when the subsurface conductivity might be described by a function with perhaps 20 degrees of freedom. The fact that the possible choices for this conductivity might be limited by smoothness constraints still makes the problem far more complicated then it need be. In this context, the subspace inversion method of Oldenburg *et al.* (1992), where the inversion is limited to a few search vectors, represents an implicit recognition of the fact that a full inversion with 10,000 degrees of freedom is vastly over-complicating the task of subsurface interpretation.

Alternatively, the inversion problem could be reduced greatly by representing the earth's conductivity in terms of continuous functions (with relatively few parameters) rather than as constant conductivity cells. One candidate for two and three dimensional conductivity modelling would be products of 1-D splines. One would then invert for knot positions and function amplitudes at the knots. The inversion would start with a small set of functions because even a grossly underparameterised model would yield useful information about the location of scatterers and the degree of smoothness of the model. Subsequent inversion runs would increase the functional complexity until the averaged predicted residual error reached a minimum. In this way, one starts with a smooth model which builds complexity without preconception because additional knots would be added as a function of the magnitude of spatial derivatives of the conductivity functions. The forward modelling part of this paradigm could be based upon spectral methods such as Lee's $t - k$ domain (1991). Indeed, this becomes very appealing because the bandwidth necessary to describe smoothly varying structures is much less than that required for structures described by sharp boundaries.

Nabighian (pers. com.) has suggested that functional (rather than block conductivity) models could be based on EM versions of alpha centres and wavelets. The problem with the latter is that most of the basic references on wavelet methods hide the useful information in a blancmange of formal mathematics.

5. Appeals to nature

Many commonly observed natural phenomena can be thought of as the results of "Nature's optimisation" so it is not surprising when weary warriors in the trenches of mathematical inversion join others in seeking to emulate these "natural inversion" solution methods. Two of these, genetic algorithms, and simulated annealing, are aimed at trying to find a global minimum through directed Monte Carlo methods. A third, neural nets, is an adaptive learning pattern matching paradigm. It is perhaps only a matter of time until inversion methods emulating the olfactory search optimisation methods of ants, fleas, flies or bees become published.

The problem with emulating nature is that even apparently simple natural phenomenon often contain such deep complexities that our best endeavours produce highly simplistic approximations which fail to capture the real essence of the actual natural process. This is particularly true in the field of neural networks. The second problem is that Nature does things on a far more massively parallel scale than we can hope to emulate in the next few years. Unlike our current fine-grained parallel computer systems, Nature does not get bogged down with shared memory hot-spots or excessive communication overheads. Nature also does not have to pay for computer time

Despite the above, natural emulations are an important part of research because not only might they generate improved solutions; they can also lead us into lines of enquiry which might otherwise never have been pursued.

A. Genetic algorithms

"Natural selection" is presumably an optimisation process based on genetics although sometimes observation would indicate that optimisation may not be the most appropriate word. Basically, the idea of GA's is to start with an initial, randomly chosen population of say N possible models, whose parameters are coded in binary form into "chromosomes". The genetic processes of selection, cross-over, and mutation are applied to update the population. Cross-over means exchanging bits between pairs of chromosomes (models) whilst mutation changes one bit at random. After these processes are applied, the new models are compared to the old chromosomes and acceptance depends upon an update probability. The best N are "selected" and the mutation and cross-over process begins anew. (Goldberg, 1989) is a good basic text for this paradigm.

There appears to be considerable interest in applying GA techniques for seismic problems. One example is the moderate success achieved by Sen and Stoffa (1992) using genetic inversion of AVO data. In a paper presented at this meeting, Schultz (pers. com.) used a GA to explore the solution space for a 1-D MT problem.

Initially, linearised inversion was used to pick thicknesses for a 24 layered earth model using data from 27 frequencies converted to Shmucker C values. The inversion parameters are log conductivities. After generating a random population of starting models, mutation, cross-over and selection procedures are performed based upon χ^2 misfit. The GA algorithm can perform about 0.5 million searches per hour of Sparc 2 CPU time. The final population of models shows strong consistency in regions where the EM resolution is expected to be good and considerable divergence where the opposite is true.

B. Simulated annealing

Annealing crystalline materials consists of melting them and then cooling them quite slowly, resulting in increased strength because slow cooling results in larger crystals and concomitant increase in long range order. At high temperatures, thermal agitation allows transitions to higher energy states as well as those of lower energy but as the temperature decreases, the probability for transition to higher energy states decreases to near zero. The slow cooling prolongs the thermal agitation process, thus increasing the probability that the annealed material will reach a global minimum energy state and thus achieve its greatest strength.

A famous algorithm (bearing the name of the lead author only) which simulated this behaviour was developed by Metropolis *et al.* (1953). It was subsequently formulated as an optimisation problem by Kirkpatrick *et al.* (1983) where (now) obvious parallels were drawn between achieving a global minimum and achieving the lowest energy state. The method consists of randomly perturbing the model parameters and computing the change to the objective function. If the objective function decreases, the new model replaces the old one. If it increases, a number is generated from a probability distribution, the value of which may allow the new model to be accepted. The probability of acceptance depends upon the value of the objective function increase and the number of previous iterations. As the process proceeds, the probability of acceptance of new models which increase the objective function is lowered (the annealing schedule) until near the end, only those model changes which lower the objective function will be accepted (freezing).

Dosso and Oldenburg (1991) used simulated annealing as a method of constructing extremal models for fitting 1-D models to MT data. The goal was to construct maximum and minimum values of conductivity over specified depth intervals subject to achieving an acceptable fit to the data. A conservative approach was taken which accepted 90 percent of the perturbations at the start and very slowly lowered the "temperature" (probability of acceptance). For the four layer model discussed in their paper, Dosso and Oldenburg (1991) report that the process took two days on a Sun 4/310 workstation. This contrasts with 3–5 minutes taken by a linear appraisal method which they had previously developed (Dosso and Oldenburg, 1989) The two methods showed good agreement near the "region of maximization" but gave different deep structure values. This difference is probably not important since the resolution at depth is expected to be poor.

C. Neural nets

Crudely speaking, neural nets consist of an array of interconnected, simple processing elements (PE's), the output of each of which is an amplitude-limited monotonic function of the weighted sum of those outputs of other elements connected to its input channels. The weights are set adaptively by correlating input patterns with a desired output pattern. Neural net paradigms are characterised by the connection

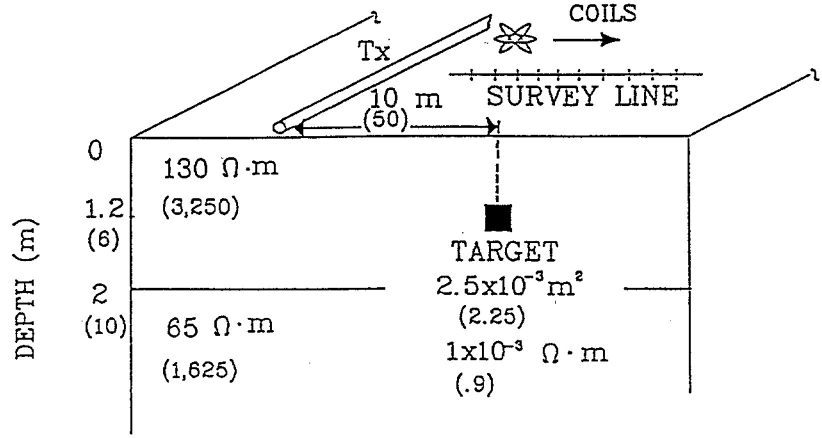

Fig. 11 Model geometry for neural net experiments.

topology, the PE output function, and "learning method" (the way in which the magnitudes of the weights are determined). A more detailed description will not be presented here because NN (neural net) methods have been discussed widely in both the popular and technical literature. (Raiche, 1991) contains an extensive review of many neural net paradigms in the context of using pattern recognition methods to solve geophysical inverse problems.

Generally speaking, neural nets are designed either as classifiers or as interpolators. What makes them interesting is that instead of specifying the statistical classification or mathematical interpolation paradigm directly, the NN designer specifies the net topology, PE output function and the learning rule. General inversion combines these tasks. Is there a target and of what type (classification)? Where is it and what are its properties (interpolation)?

In two papers, Poulton *et al.* (1992, 1992a) compared the ability of several different NN methods to use ellipticity data to estimate the position and conductivity area product of a horizontal galvanised pipe. The 60 m long pipe (with an outside diameter of 6 cm) was buried 1.27 m deep parallel to a grounded transmitter wire with a horizontal offset of 10 m as illustrated in Figure 11. The radial and vertical components of the magnetic field, H_r and H_z, generated from a grounded wire transmitter, were measured at several offset distances for 11 frequencies ranging from 150 to 6814 Hz. The magnetic fields were converted to ellipticity, e, defined below in terms of the tilt angle, α.

$$e = \left| \frac{H_z \cos \alpha - H_r \sin \alpha}{H_r \cos \alpha + H_z \sin \alpha} \right| \tag{24}$$

$$\tan(2\alpha) = \frac{2\left|\frac{H_z}{H_r}\right| \cos(\phi_z - \phi_r)}{1 - \left(\frac{H_z}{H_r}\right)}$$

The project consisted of generating a synthetic data set using a 2-D finite-element program to compute the ellipticity for several different offset distances and depths of the pipe and using the resulting image pixel data to train various neural net configurations. The experiment was simply to determine how well the different neural net methods could recover these parameters from both the artificial data and some field data as well. Initially, the input data set for each pipe position consisted of the percent ellipticity values from 660 pixels.

Although using the entire input data set produced results within 10 percent accuracy, it required more than a day to train the network on a PC. Decreasing the data set can greatly speed up training times so different representations were tried. These included 2-D spatial FFT, sampling peak heights and widths, and using different sub-samples of pixel data. The FFT data set produced results almost as good as those using the whole image. For data sets of less than 100 pixels, directed random search and function link worked best. These apparently did not scale upwards very well because for large data sets, the combination of the self-organising map with backpropagation worked best.

There are some things which presumably could have improved this work. Expanding the input data set for function net to include common pattern indicators such as peak height and half-width of the magnetic field profiles could possibly have improved parameter recovery. The use of non-localised node updating methods would also improve efficiency. Nonetheless, from an inversion point of view, this was a very easy problem and Poulton *et al.* had to work very hard to get fairly mature NN methods to yield a reasonable answer. What would have been the case for a problem with 10 parameters?

D. Commentary

1. *Neural Nets*

In a program based upon numerical or logical manipulation, the programmer controls the computation and can ascertain the state of this computation at any stage by having selected variables report their values. This control is not possible in NN computations. The best that can be done is watching how selected weights change value. The real (but often hidden question) is: Is there any magic? By giving up control of the process, can something new be discovered? Can NN methods discover hidden patterns and perform classification and interpolation tasks more efficiently than ordinary statistical and AI methods? All the evidence points to one answer: No.

I am unaware of any significant pattern recognition task performed by neural nets which has not been done (and probably been done better) by either AI or statistical methods. For a start, most neural net methods in practical use are analogue realisations of pre-existing statistical or AI methods. For example Kohonen's vector quantisation method for self-organising nets is the analogue equivalent of a well known statistical clustering method, the *k*-means algorithm (Duda and Hart, 1973). Backpropagation is a simple form of linear regression without regularisation. Using neural net training rather than direct programming just makes these things more difficult to implement and monitor.

Neural nets will perform much more efficiently if problem specific information (rules and mathematical relationships) can be used to shape the form of the input data and the training method. This is the basis for the success of Pao's function net. Why not make things easier and go the whole way with logic and numeric programming? Two reasons were given in (Raiche, 1991). The first is that by experimentation with different types of feedback terms, one might discover better methods of classification not covered by existing AI or statistical methods. To date, this does not seem to have happened, and neural network research is now a big field.

The second reason was that once trained, a neural net could perform classifications much faster than say a statistical algorithm. In fact, a neural net can be represented by a set of filter coefficients much the way as Hankel transforms. Why not compute these coefficients and use them directly without using neural nets to obfuscate the issue?

Raiche (1991) was oriented towards developing an NN approach to geophysical inversion. Paradoxically, the application of geophysical inversion techniques could significantly improve the training of feed-forward networks. The question of how many nodes to use and how they should be connected is usually answered by trial and error methods. Connection weights are updated individually. However, it is easy to formulate the training problem as an SVD inversion. Regularisation methods would undoubtedly improve the stability of the networks. Moreover, by examining the eigen-weights rather than simply updating the actual network weights, it is easier to determine which are the most important weight combinations and which are irrelevant.

However, I would like to augment the above judgement with a mention of the development of a new paradigm in behavioural neurodynamics. The 1st Appalachian Conference on Behavioural Neurodynamics (Radfield University, Radfield, Virginia, Sept., 1992) will feature papers discussing the application of quantum field theory and non-linear dynamics to the processing of neuroelectric signals. Heisenberg matrix and symmetry group methods are apparently now being applied to imaging and object perception. Whether any of this can be used eventually to improve current inversion processes is a matter of conjecture and future research.

In summary, there is a clear trend towards recognising neural net methods as a collection of mathematical pattern recognition techniques rather than an emulation of "biological computing".

2. *Fast Forward Model*

However, the body of neural net research has been useful in that it suggests that we re-examine the computational possibilities of different multi-dimensional interpolation methods. One possibility is the development of the fast forward model, similar in concept to Pelton *et al.* (1978). The idea is to compute a model suite of the EM responses of a low parameter model such as a 3-D prism in a two-layer half-space over a wide range of model parameter combinations. A clustering algorithm would be used to study the variation of the responses as a function of different combinations of parameters. In principle, one could use multi-dimensional interpolating functions to represent responses in each cluster. A rule-based program would direct the computation into the appropriate cluster. Having devised such a program, one could then compute model responses in seconds on a PC rather than hours on a supercomputer. The question is: how long would it take to compute a suitably representative model suite to feed the clustering and interpolation algorithms. This depends strongly on the efficacy of the interpolator.

Multi-dimensional interpolation is a topic of considerable recent interest, some of which has been sparked off by the neural net boom. One method is the use of radial basis functions (Poggio and Girosi (1990), Powell (1985), Powell (1987)). Another is the multiquadratic-biharmonic method (Hardy (1990), Kansa (1990)). (Broomhead and Lowe, 1988) discuss the relationship between multi-variable interpolations and adaptive networks. The point of this is, why not use direct mathematical methods instead of hoping for magic by training neural nets to represent model responses.

At the present time, we have no indication as to how efficiently radial basis functions or multiquadrics will work for EM models. Poulton (pers. com.) is now investigating the use of radial basis functions.

3. *Directed Monte Carlo Methods*

Both Schultz and Dosso and Oldenburg (1991) tackled aspects of a 1 -D MT problem, the former with GA's, the latter with simulated annealing. Both directed Monte Carlo methods gave tight responses in regions where the resolution was anticipated to be good, and scattered results where the physical resolution was expected to be poor. The problem of course is that the time to evaluate a 1-D MT response is negligible compared to that required to compute the response of a 2-D or 3-D model response of an active EM source. It is possible that we may be able to design directed Monte-Carlo methods which are better directed than simulated annealing or GA's. Alternatively, another method of 3-D inversion is to wait for geological weathering and tectonic processes to either expose the target at depth or bring it to the surface.

The following quote from Adam Schultz (pers. com.) describes the simulated annealing problem as well as it does GA's. "Eigenparameters as a paradigm have got to be better than the blind groping GA is doing. GA is just like rolling a marble on a bumpy rubber surface. You let the membrane vibrate just a little and you have a better chance of the marble falling into the deepest depression in the surface. Don't vibrate it at all, and it falls into whatever's handy. Vibrate it too much and the thing jumps all over the place. So how much do you vibrate it? You vibrate it JUST RIGHT. This sounds like an eigenvalue problem if only we could figure out how to describe the misfit manifold." I leave this as a challenge for the reader.

6. Conclusions

The early 2-D numerical modelling papers in the 1970's used to begin with phrases such as "with the advent of powerful new computers". Over the past two decades, the development of even newer computers has allowed us to make moderate progress in modelling and inversion methodology but some of us may have felt that this progress was more a case of refinement rather than real innovation or breakthroughs. However, "with the advent of powerful new computers"....

New innovations both in hardware and software are allowing researchers to test ideas which had previously been judged to be computationally unfeasible. New frequency and time-stepping modelling methods using staggered grids defined on the full spatial domain will free us from the restriction of localised heterogeneities in homogeneous hosts and allow us at last to model the EM response of true geologic complexity. A new generation of relatively inexpensive, massively parallel, fine-grained computers will allow us to compute these models in minutes. Developments in problem based data languages, symbolic processing and visualisation technology will enable new ideas to be prototyped and tested in a fraction of the time now required.

Inversion methodology should virtually leap ahead based upon new abilities to compute complex models quickly. New weapons technology should make the debate between competing inversion theologies much more interesting.

In short, two decades of good scientific research by our colleagues, coupled with advances in information technology, have now brought us to the threshold of a new era in modelling and inversion. However, in all of this excitement, we should perhaps ask one question. What can be done about the low resolution of our EM systems caused by that low pass filter called the earth?

References

Abramson, D., Dix, M., Francis, R., Mathieson, I. and Whiting, P.: 1991, 'A Retargettable Programming Environment for Studying Climate Models', *Computational Techniques and Applications (CTAC-91)*, Adelaide.

Anderson, W.: 1979, 'Numerical Integration of Related Hankel Transforms of Orders 0 and 1 by Adaptive Digital Filtering', *Geophysics* **44**, 1287–1305.

Barnett, C. J.: 1984, 'Simple Inversion of Time-Domain Electromagnetic Data', *Geophysics* **49**, 925–933.

Broomhead, D. S. and Lowe, D.: 1988, 'Multivariable Functional Interpolations and Adaptive Networks', *Complex Systems* **2**, 321–355.

Cann, D.: 1992, *Retire Fortran? A Debate Rekindled.*

Dahl, O. E., Dijkstra, E., and Hoare, C.: 1972, *Structured Programming*, Academic Press.

Dijkstra, E.: 1972, 'Structured Programming', in: J. Buxton *et al.* (eds.), *Software Engineering, Concepts and Techniques*, Van Nostrand Reinhold.

Dosso, S. E. and Oldenburg, D. W.: 1991, 'Magnetotelluric Appraisal Using Simulated Annealing', *Geophys. J. Int.* **106**, 379–385.

Dosso, S. E. and Oldenburg, D. W.: 1989, 'Linear and Non-linear Appraisal Using Extremal Models of Bounded Variation', *Geophys. J. Int.* **99**, 483–495.

Druskin, V. L. and Knizhnerman, L. A.: 1988, 'A Spectral Semi-discrete Method for the Numerical Solution of 3D Nonstationary Problems in Electrical Prospecting', *Physics of the Solid Earth* **24**, 63–74.

Duda, R. O. and Hart, P. E.: 1973, *Pattern Classification and Scene Analysis*, John Wiley and Sons.

Eaton, P. A. and Hohmann, G. W.: 1989, 'A Rapid Inversion Technique for Transient Electromagnetic Soundings', *Physics of the Earth and Planetary Interiors* **53**, 394–404.

Engquist, B. and Majda, A.: 1977, 'Absorbing Boundary Conditions for the Numerical Simulation of Waves', *Math. Comp.* **31**, 629–651.

Everett, M. E. and Edwards, R. N.: 1992, 'Transient Marine Electromagnetics: The 2.D Forward Source Problem', To appear in *Geophys. J. Int.*

Filatov, V. V.: 1984, 'Construction of Focusing Transformations of Transient Electromagnetic Fields', *Geol. i Geofiz (Soviet Geology and Geophysics)* **25**, 89–95.

Isaev, G. A. and Filatov, V.V.: 1981, 'Physicomathematical Principles of Visualisation of Transient Electromagnetic Fields', *Geol. i Geofiz (Soviet Geology and Geophysics)* **22**, 89–95.

Flosadóttir, A. H.: 1990, *The Response of the Oceanic Lithosphere to Electromagnetic Controled Source Transmitter Using Local Spectral Representation*, PhD Thesis Univ. of California, Scripps Institution of Oceanography, SIO Ref No. 90-23.

Franklin, J.: 'Well-Posed Stochastic Extensions of Ill-Posed Linear Problems', *J. Math. Anal. Appl.* **31**, 682–716.

Fullagar, P. K. and Oldenburg, D. W.: 1984, Inversion of Horizontal Loop Frequency Soundings', *Geophysics* **49**, 150–164.

Goldberg, D. E.: 1989, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley

Golub, G. H. and Van Loan, C. F.: 1989, *Matrix Computations*, second edition, The Johns Hopkins University Press.

Gupta, P. K., Bennett, L. A., and Raiche, A. P.: 1987, 'Hybrid Calculations of the Three-Dimensional Electromagnetic Response of Buried Conductors', *Geophysics* **52**, 301–306.

Hardy, R. L.: 1990, 'Theory and Applications of the Multiquadratic-Biharmonic Method', *Computers Math. Applic.* **19**, 163–208.

Hohmann, G. W.: 'Three-Dimensional Induced Polarization and Electromagnetic Modeling', *Geophysics* **40**, 309–324.

Hohmann, G. W.: 1987, 'Numerical Modelling for Electromagnetic Methods of Geophysics', in M. N. Nabighian (ed.), *Electromagnetic Methods in Applied Geophysics* 3, vol. 1, *Soc. Expl. Geophys.* 313–364.

Hohmann, G. W. and Raiche, A. P.: 1987, 'Inversion of Controlled Source Electromagnetic Data', in: M. N. Nabighian (ed.), *Electromagnetic Methods in Applied Geophysics 3*, vol. 1, *Soc. Expl. Geophys.* 469–505.

Hordt, A., Dhuskin, V. L., Knizhnerman, L. A., and Strack, K. M.: 1992, 'Interpretation of 3-D Effects in Long-offset Transient Electromagnetic (LOTEM) Soundings in the Münsterland Area/Germany', submitted to Géophysics.

Hughes, J.: 1989, 'Why Functional Programming Matters', *The Computer Journal* **32**, 98–107.

Irons, B. M.: 1970, 'A Frontal Solution Program for Finite Element Analysis', *International Journal for Numerical Analysis in Engineering* **2**, 5–32.

Johansen, H. K. and Sorensen, K.: 1979, 'Fast Hankel Transforms', *Geophysical Prospecting* **27**, 876–901.

Jupp, D. L. B. and Vozoff, K.: 1975, 'Stable Iterative Methods for the Inversion of Geophysical Data', *Geophys. J. R. Astr. Soc.* **42**, 957–976.

Kansa, E. J.: 1990, 'Multiquadrics – A Scattered Data Approximation Scheme with Applications to Computational Fluid Dynamics', *Computers Math. Applic.* **19**, 127–145.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. P.: 1983, 'Optimisation by Simulated Annealing: Science', **220**, 671–680

Knight J. H. and Raiche, A. P.: 1982, 'Transient Electromagnetic Calculations Using the Gaver-Stehfest inverse Laolace Transform Method', *Geophysics* **47**, 47–50.

Lanczos, C.: 1961, *Linear Differential Operators*, Chapter 3, D. Van Nostrand Co.

Lee, K. H. and Morrison, H. F.: 1985, 'A Numerical Solution for the Electromagnetic Scattering by a Two-Dimensional Inhomogeneity', *Geophysics* **50**, 466–472.

Lee, K. H. and Morrison, H. F.: 1989, 'A New Approach to Modeling the Electromagnetic Response of Conductive Media', *Geophysics* **54**, 1180–1192.

Lee, K. H. and Xie, G. Q.: 1992, 'A New Approach to Imaging with Low Frequency Electromagnetic Fields', *Geophysics* **58**, 780–796.

Lee, S.: 1991, 'Modelling of 3-D Electromagnetic Responses Using the Time-Wavenumber Method', PhD Thesis, University of California, Berkeley.

Linger, Mills and Witt, 1979, Structured Programming – Theory and Practice, Addison-Wesley.

Liu, J., Kelly, P. H. J., Cox, S. M., and Taylor, F. S.: 1992, 'Functional Programming for Scientific Computation', submitted to *Engineering Computation*.

Macnae, J. C., Smith, R., Polzer, B. D., Lamontagne, Y., and Klinkert, P.: 1991, 'Conductivity-Depth Imaging of Airborne Electromagnetic Step-Response Data', *Geophysics* **56**, 102–114.

Madden T. R. and Mackie, R. L.: 1989, 'Three-Dimensional Modeling and Inversion', *Proc. IEEE* **77**, 318–333.

Mteropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E.: 1953, 'Equation of State Calculations by Fast Computing Machines', *J. Chemical Physics* **21**, 1087–1092.

Newman, G. A. and Hohmann, G. W.: 1986, 'Transient Electromagnetic Response of High-Contrast Prisms in a Layered Earth', *Geophysics* **53**, 691–706.

Oldenburg, D. W. and Ellis, R. G.: 1991, 'Inversion of Geophysical Data Using an Approximate Inverse Mapping', *Geophys. J. Int.* **105**, 325–353.

Oldenburg, D. W., McGillivray, P. R., and Ellis, R. G.: 1992, 'Generalised Subspace Methods for Large Scale Inversion', submitted to *Geophys. J. Int.*

Oristaglio, M. L. and Hohmann, G.W.: 1984, 'Diffusion of Electromagnetic Fields into a Two-Dimensional Earth; A Finite Difference Approach', *Geophysics* **46**, 870–894.

Pellerin, L.: 1992, *Application of* 3-*D Modeling in EM Exploration*, PhD thesis, University Of Utah, Salt Lake City.

Pelton, W. H., Rijo, L., and Swift Jr., C. M.: 1978, 'Inversion of Two-Dimensional Resistivity and Induced Polarization Data'. *Geophysics* **43**, 788–803.

Poggio, T. and Girosi, F.: 1990, 'Regularisation Algorithms for Learning that are Equivalent to Multilayer Networks'. *Science* **247**, 978–982.

Poulton, M. W., Sternberg, B. K., and Glass, C. E.: 1992, 'Location of Subsurface Targets in Geophysical Data Using Neural Networks', *Geophysics* **57**, 1534–1544.

Poulton, M. W., Sternberg, B. K., and Glass, C. E.: 1992a, 'Neural Network Pattern Recognition of Subsurface Images', *Journal of Applied Geophysics* **29**, 21–36.

Powell, M. J. D.: 1987, Radial Basis Function Approximations to Polynomials, DAMPT preprint (presented at the 1987 Dundee Numerical Analysis Conference).

Powell, M. J. D.: 1985, Radial Basis Functions for Multivariable Interpolation: a Review, IMA Conference on "Algorithms for the Approximation of Functions and Data", RMCS Shrivenham.

Raiche, A. P.: 1987, 'Transient Electromagnetic Field Computations for Polygonal Loops on Layered Earths', *Geophysics* **52**, 785–793.

Raiche, A. P.: 1991, 'A Pattern Recognition Approach to Geophysical Inversion Using Neural Nets', *Geophys J. Int.* **105**, 629–648.

Raiche, A. P. and Tarlowski, Z. K.: 1984, 'A Hybrid Method for Solving the Boundary Value Problems for Helmholtz's Equation in Two-Dimensional Domains', *Computational Techniques & Applications (CTAC-83),* J. Noye & C. Fletcher, Editors, Elsevier (North Holland).

Sen, M. K. and Stoffa, P. L.: 1992, 'Genetic Inversion of AVO', *The Leading Edge*, 27–29.

Silic, J.: 1989, 'Interpretation of TDEM Data Using First and Second Spatial Derivatives and Time Decay Analysis', *Exp. Geophys.* **20**, 57–64.

Smith, J. T. and Booker, J. R.: 1991, 'Rapid Inversion of Two- and Three-Dimensional Magnetotelluric Data', *J. Geophys. Res.* **96**, 3905–3922.

Sugeng, F. and Raiche, A. P.: 1989, 'Predicting the Transient EM Response of Complex Structures Using the Compact Finite-Element Method', *Exp. Geophys.* **20**, 51–56.

Sugeng, F. and Raiche, A. P.: 1992, 'Comparing 3-D Controlled Source Time-Domain Response of 2-D and Elongated 3-D Conductors in Layered Conducting Hosts', IAGA 11th Workshop on EM Induction in the Earth, Wellington, NZ.

Tarlowski, C. Z. and Raiche, A. P.: 1984, 'The Use of Summary Representation for Electromagnetic Modeling', *Geophysics* **49**, 1506–1516.

Travis, B. J. and Chave, A. D.: 1989, 'A Moving Finite Element Method for Magnetotelluric Modelling', *PEPI* **53**, 432–443.

Unsworth, M. J., Travis, B. J., and Chave, A. D.: 1993, 'Electromagnetic Induction by a Finite Dipole over a 2-D Earth', *Geophysics* **58**, 198–214.

Wang, T. and Hohmann, G. W.: 1993, 'A Finite-Difference Time-Domain Solution for Three-Dimensional Electromagnetic Modelling', *Geophysics* **58**, 780–796.

Wolfram, S.: 1991, *Mathematica, A System for Doing Mathematics by Computer*, Addison-Wesley.

Yee, K. S.: 1966, 'Numerical Solution of Initial Boundary Problems Involving Maxwell's Equations in Isotropic Media', *IEEE Trans. Ant. Prop.* **AP-14**, 302–309.

Zhdanov, M. S. and Frenkel, M. A.: 1983, 'The Solution of the Inverse Problems on the Basis of the Analytical Continuation of the Transient Electromagnetic Field in Reverse Time', *J. Geomag. Geoelectr.* **35**, 747–765.