

Parallel computation of optimized arrays for 2-D electrical imaging surveys

M. H. Loke¹, P. B. Wilkinson² & J. E. Chambers²

¹ Geotomo Software, 115 Cangkat Minden Jalan 5, Minden Heights, 11700 Gelugor, Penang, Malaysia.

² British Geological Survey, Natural Environment Research Council, Kingsley Dunham Centre, Keyworth, Nottingham, NG12 5GG, UK

Accepted 2010 August 30. Received 2010 August 9; in original form 2010 January 25

Abbreviated title : Parallel computation of optimized arrays

Corresponding author : M.H.Loke, 115 Cangkat Minden Jalan 5, Minden Heights, 11700 Gelugor, Penang, Malaysia. Email : drmhloke@yahoo.com, Tel : +60 4 6574525, Fax : +60 4 6588437.

Published in *Geophysical Journal International* (Royal Astronomical Society / Blackwell Publishing). The definitive version is available at www.blackwell-synergy.com

Geophysical Journal International (2010) **183**, 1302-1315

<http://onlinelibrary.wiley.com/doi/10.1111/j.1365-246X.2010.04796.x/pdf>

Summary

Modern automatic multi-electrode survey instruments have made it possible to use non-traditional arrays to maximize the subsurface resolution from electrical imaging surveys. Previous studies have shown that one of the best methods for generating optimized arrays is to select the set of array configurations that maximizes the model resolution for a homogeneous earth model. The Sherman-Morrison Rank-1 update is used to calculate the change in the model resolution when a new array is added to a selected set of array configurations. This method had the disadvantage that it required several hours of computer time even for short 2-D survey lines. The algorithm was modified to calculate the change in the model resolution rather than the entire resolution matrix. This reduces the computer time and memory required as well as the computational round-off errors. The matrix-vector multiplications for a single add-on array were replaced with matrix-matrix multiplications for 28 add-on arrays to further reduce the computer time. The temporary variables were stored in the double-precision SIMD registers within the CPU to minimize computer memory access. A further reduction in the computer time is achieved by using the computer graphics card GPU as a highly parallel mathematical coprocessor. This makes it possible to carry out the calculations for 512 add-on arrays in parallel using the GPU. The changes reduce the computer time by more than two orders of magnitude. The algorithm used to generate an optimized data set adds a specified number of new array configurations after each iteration to the existing set. The resolution of the optimized data set can be increased by adding a smaller number of new array configurations after each iteration. Although this increases the computer time required to generate an optimized data set with the same number of data points, the new fast numerical routines has made this practical on commonly available microcomputers.

Keywords : Parallel, computation, optimized, arrays, electrical, imaging

INTRODUCTION

In the past decade there have been many significant developments in the resistivity exploration method such that it is now one of the standard techniques used in engineering, environmental and mining surveys. Two-dimensional resistivity surveys are widely carried out, and even three-dimensional surveys are becoming more common in areas of very complex geology (Dahlin 1996; Auken *et al.* 2006; Chambers *et al.* 2006). The field applications range from agriculture (Samouëlian *et al.* 2003), groundwater exploration (Seaton & Burbey 2000), engineering site investigation (Kuras *et al.* 2007), environmental assessment (Dahlin *et al.* 2002), mineral exploration (Bingham *et al.* 2006) to even hydrocarbon mapping (Bauman 2005).

The development of automatic multi-electrode survey instruments has made such surveys fast and economical. It has also enabled the user to select the optimum array for the survey problem. Most surveys still use conventional arrays such as the Wenner, Schlumberger, pole-pole, pole-dipole, dipole-dipole and gradient (Dahlin & Zhou 2004). Recently there have been significant developments in algorithms to automatically select arrays to maximize the resolution of the subsurface inversion model (Stummer *et al.* 2004). The 'Compare R' method by Wilkinson *et al.* (2006b) that directly calculates the model resolution has proved to be the best of these methods (Loke *et al.* 2010). However the 'Compare R' method had the disadvantage of requiring much more computer time compared to other faster but less accurate methods that use approximations of the model resolution. Thus the main focus of this paper is on numerical and computational techniques to reduce the time required by this method using commonly available personal computer systems as many electrical imaging surveys are carried out by small geophysical companies.

There is continuous demand for high-performance computing in the geophysical industry (Sava 2010) as more sophisticated survey and data interpretation techniques are

developed to provide increasingly realistic models of the subsurface. More recent trends place less emphasis on increasing CPU speed that has reached a plateau of about 3 to 4 GHz for common microprocessors. There is now more emphasis on highly parallel computational models and more efficient memory to CPU data transfers (Camp and Thierry 2010) and the use of non-conventional techniques such as GPU programming (Kadlec & Dorn 2010; Moorkamp *et al.* 2010) to achieve a greater level of performance. There have been several recent papers on high-performance computing techniques for seismic data processing which is the largest geophysical user of computer resources (Michéa & Komatitsch 2009; Clapp & Fu 2010). We discuss the use of similar techniques in the context of resistivity survey design in this paper. While some of the discussion is specific to the computer architectures used, the general principles can be applied to other systems and potential field problems that use similar numerical matrix algorithms.

This paper describes the numerical and computational techniques devised to reduce the execution time of the optimization algorithm. Then a study is made of the optimum balance between computer time and model resolution in calculating the optimized arrays. This is followed by tests of the optimized arrays using a synthetic model. Finally, we compare the use of the simple damped (Levenberg-Marquardt) and smoothness-constrained versions of the least-squares equation for generating the optimized data sets.

THEORY

(a) Model resolution and point spread functions

The smoothness-constrained least-squares optimization method is frequently used for 2-D inversion of resistivity data (Loke *et al.* 2003). The subsurface model usually consists of a large number of rectangular cells. The linearized least-squares equation that gives the relationship between the model parameters and the measured data is given below.

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{C}) \Delta \mathbf{r}_i = \mathbf{J}^T \mathbf{d} - \lambda \mathbf{C} \mathbf{r}_{i-1}, \quad (1)$$

The Jacobian matrix \mathbf{J} contains the sensitivities of the measurements with respect to the model parameters, \mathbf{C} contains the roughness filter constraint, λ is the damping factor and \mathbf{d} is the data misfit vector. \mathbf{r}_{i-1} is the model parameter vector (the logarithm of the model resistivity values) for the previous iteration, while $\Delta \mathbf{r}_i$ is the change in the model parameters.

It can be shown that the model resolution matrix \mathbf{R} (Menke 1989) is given by

$$\mathbf{R} = (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{C})^{-1} \mathbf{J}^T \mathbf{J}. \quad (2)$$

The main diagonal elements of \mathbf{R} that give an estimate of the model cells resolution have values of between 1.0 and 0.0. The sum of the elements in each row of the \mathbf{R} matrix is equal to 1.0 (Jackson 1972). A model cell has perfect resolution if the resolution value is 1.0 and all other row elements of the \mathbf{R} matrix are 0.0. In practice, the resolution values are less than 1.0 and decreases exponentially with depth (Loke *et al.* 2010). The ‘Compare R’ method by Wilkinson *et al.* (2006b) attempts to determine the set of array configurations that will maximize the average resolution value for a homogeneous earth model.

Some authors (Friedel 2003; Miller & Routh 2007; Oldenborger & Routh 2009) have proposed the use of the point spread function as another measure of the resolution capability of the data. The point spread function for a model cell consists of the corresponding column of the resolution matrix. A spread criterion, that is a weighted sum of the elements of the point spread function, is frequently used as it summarizes the information into a single number. The spread criterion (Miller & Routh 2007) value for the i th model cell, $S(i)$, is given by the following equation.

$$S(i) = \left[\frac{\sum_{j=1}^{j=m} W_{ij} [R(i, j) - \Delta_{ij}]^2 \delta_j}{\alpha + \sum_{j=1}^{j=m} R(i, j)^2 \delta_j} \right]^{1/2} \quad (3)$$

where $W_{ij} = 1 + d_{ij}$

$\Delta_{ij} = 1$ for $i=j$, and $\Delta_{ij} = 0$ for $i \neq j$.

d_{ij} is the normalized distance (distance divided by the unit electrode spacing) between the centers of i th and j th model cells, α is a small value (0.0001) and m is the number of model cells. δ_j is the (normalized) area of the j th model cell. We also use the average spread criterion value, S_{CR} , in this paper that is calculated using the following equation.

$$S_{CR} = \frac{1}{m} \sum_{j=1}^{j=m} S(i) \quad (4)$$

In a later section of this paper, plots of the spread criterion values are displayed together with the model resolution values.

If the data errors are known the least-squares equation can be modified by a data weighting matrix (Menke 1989; Farquharson & Oldenburg 1998). Similarly, a weighted form of the roughness filter and Jacobian matrix is sometimes used to impose an L1-norm constraint on the model roughness and data misfit (Farquharson & Oldenburg 1998) using the iteratively reweighted least-squares method. However, for this paper, we choose the simpler form in equations (1) and (2) so that the results can be directly compared with previous work by Stummer *et al.* (2004) and Wilkinson *et al.* (2006b). For the same reason, we also use the sensitivity values for a homogeneous half-space (Loke & Barker 1995) in calculating the resolution and spread values.

(b) Array optimization algorithm

For a system with N electrodes, there are $N(N-1)(N-2)(N-3)/8$ independent four-electrode configurations. To reduce the number of possible arrays, arrays of the Gamma type configuration (Carpenter & Habberjam 1956) with crossed current and potential electrodes as well as those large geometric factors are excluded (Stummer *et al.* 2004). After excluding these less stable configurations, a local optimization procedure is used to select a subset of

the comprehensive set of all the viable configurations that will maximize the model resolution. A small base data set consisting of the dipole-dipole configurations with the ‘ a ’ dipole length of 1 unit electrode spacing and ‘ n ’ dipole separation factor of 1 to 6 is initially selected. The change in the model resolution matrix \mathbf{R} for each new array when added to the base set is then calculated. The configurations that result in the largest increase in the model resolution, and have a suitable degree of orthogonality to the existing configurations, are then added to the base data set (Wilkinson *et al.* 2006b). We also include the modification by Loke *et al.* (2010) whereby for arrays that are not symmetrical about the center of the survey line, the corresponding array configuration on the other half of the survey line is also included in the optimized data set. This ensures that the distribution of data points (and thus the resulting model resolution section) is symmetrical. In each iteration the number of new configurations added is normally set at about 9% of the present number in the base set. The model resolution for the new base data set (after adding the new configurations) is recalculated using equation (2). This is repeated until the desired number of optimized array configurations is selected. Further details on the optimization procedure are given in Wilkinson *et al.* (2006b) and Loke *et al.* (2010). For the following discussion, we rewrite equation (2) into the following form.

$$\mathbf{R} = \mathbf{B} \mathbf{A} , \text{ where } \mathbf{A} = \mathbf{J}^T \mathbf{J} \text{ and } \mathbf{B} = (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{C})^{-1} \quad (5)$$

(c) The original 'Compare R' method

Wilkinson *et al.* (2006b) used the Sherman-Morrison Rank-1 update (Golub & van Loan 1989) to calculate the main diagonal elements of the model resolution matrix of the base set plus the test configuration. The following set of updating formulae is used to calculate the new resolution matrix \mathbf{R}_{b+1} when a new array is added to the base set

$$\mathbf{A}_{b+1} = \mathbf{A}_b + \mathbf{g} \mathbf{g}^T , \quad \mathbf{B}_{b+1} = \mathbf{B}_b - \frac{\mathbf{z} \mathbf{z}^T}{1 + \mu} , \quad \mathbf{R}_{b+1} = \mathbf{B}_{b+1} \mathbf{A}_{b+1} \quad (6)$$

\mathbf{g} is the sensitivity vector for the new array, $\mathbf{z}=\mathbf{B}_b\mathbf{g}$ and $\mu = \mathbf{g}\cdot\mathbf{z}$. While this method proved to produce the arrays with the highest model resolution, it was also the slowest taking about 6 hours on a 3 GHz PC to determine the optimum arrays for a 2-D survey line with 30 electrodes (Wilkinson *et al.* 2006b). In order to use this method for longer 2-D survey lines, it is necessary to greatly improve its computational efficiency. The number of numerical operations required in equation (6) to calculate the diagonal elements of the updated model resolution matrix \mathbf{R}_{b+1} for a single add-on array is proportional to m^2 where m is the number of model cells. For the survey line with 30 electrodes, the number of arrays in the comprehensive data set is 51283. This increases to 166944 (40 electrodes), 411453 (50 electrodes) and 854224 (60 electrodes). It is the huge number of possible add-on arrays in the comprehensive data set that causes the long computer time needed.

This updating algorithm has three main steps; (i) calculate the updated elements for the \mathbf{A} matrix that is stored in a temporary matrix \mathbf{A}_{b+1} , (ii) a similar calculation for the \mathbf{B} matrix that is stored in a temporary matrix \mathbf{B}_{b+1} , (iii) finally multiply the two temporary matrices. In the implementation used by Wilkinson *et al.* (2006b), updating the \mathbf{A} matrix took about 32% of the total time for one iteration. Updating the \mathbf{B} matrix (including calculating \mathbf{z} vector) took about 40%, the matrix multiplication (for the diagonal entries only) took about 27% and the remaining 1% was used for miscellaneous operations such as finding the add-on arrays that gave the largest increase in the model resolution.

The CPU in modern computer systems can operate at a much higher speed compared to the main memory. As an example, in the computer system used in this work, the CPU operates at a frequency of 2.66 GHz while the main memory (RAM) runs at 533 MHz. Thus the time taken to transfer data between the main memory and the CPU can be much longer compared to the time taken for the numerical operations within the CPU. The calculations were carried out on a 2.66 GHz Intel i7 Quad-Core system (with 12 GB RAM) with a Nvidia

GTX 285 graphics card (with 1 GB RAM). In this test, we use the same damped least-squares formulation ($\mathbf{C}=\mathbf{I}$) and damping factor ($\lambda=0.000025$) as that used by Wilkinson *et al.* (2006b).

(d) The matrix-vector method

The first step to improving the program efficiency is to reduce the traffic between the main memory and the CPU. In order to achieve this, Loke *et al.* (2010) expanded equation (6) for \mathbf{R}_{b+1} into the following form.

$$\mathbf{R}_{b+1} = \mathbf{R}_b + \Delta\mathbf{R}_b, \quad \text{where} \quad \Delta\mathbf{R}_b = -\frac{\mathbf{z}\mathbf{z}^T}{1+\mu}\mathbf{A}_b + \mathbf{B}_b\mathbf{g}\mathbf{g}^T - \frac{\mathbf{z}\mathbf{z}^T}{1+\mu}\mathbf{g}\mathbf{g}^T \quad (7)$$

In this paper, we further simplify the above equation by making use of the fact that $\mathbf{z}=\mathbf{B}_b\mathbf{g}$ and $\mu = \mathbf{g}\mathbf{z}$ as follows.

$$\Delta\mathbf{R}_b = -\frac{\mathbf{z}\mathbf{z}^T}{1+\mu}\mathbf{A}_b + \mathbf{z}\mathbf{g}^T - \frac{\mu\mathbf{z}\mathbf{g}^T}{1+\mu} = -\frac{\mathbf{z}\mathbf{z}^T}{1+\mu}\mathbf{A}_b + \mathbf{z}\mathbf{g}^T - \mathbf{z}\mathbf{g}^T + \frac{\mathbf{z}\mathbf{g}^T}{1+\mu} = \frac{\mathbf{z}}{1+\mu}(-\mathbf{z}^T\mathbf{A}_b + \mathbf{g}^T)$$

This equation can be further reduced to

$$\Delta\mathbf{R}_b = \frac{\mathbf{z}}{1+\mu}(\mathbf{g}^T - \mathbf{y}^T) \quad (8)$$

where $\mathbf{y}=\mathbf{A}_b\mathbf{z}$ (note that \mathbf{A} and \mathbf{B} are symmetric matrices). Equation (8) only involves vector-vector multiplication and subtraction. The bulk of the numerical calculations are in the matrix vector products $\mathbf{B}_b\mathbf{g}$ and $\mathbf{A}_b\mathbf{z}$ required to generate the \mathbf{z} and \mathbf{y} vectors. The simplification of equation (7) to (8) further reduces the calculation time by about 5%.

In equation (8) the change in the resolution matrix $\Delta\mathbf{R}_b$ is calculated rather than the entire updated resolution matrix \mathbf{R}_{b+1} . The main diagonal elements of $\Delta\mathbf{R}_b$ are calculated one by one using equation (8), thus avoiding the use of the temporary matrices \mathbf{A}_{b+1} and \mathbf{B}_{b+1} . This reduces the computer time and memory required. The number of times the main memory is accessed in equation (8) is drastically reduced. It is only necessary to access the matrices \mathbf{A}_b and \mathbf{B}_b (with m^2 elements each) once to calculate the vectors \mathbf{z} and \mathbf{y} (each with

only m elements) for a single add-on test configuration. The computer time can be further reduced by making use of the parallel processing capabilities of modern CPUs.

The computer code was optimized so that all the calculations needed to update the model resolution values can be carried out ‘in situ’ within the CPU floating point registers to reduce the traffic between the CPU and main memory to a minimum. The time critical parts of code were written in assembly language (Leiterman 2005) so that the use of the available floating point CPU registers could be directly hand optimized. The SIMD (Single Instruction Multiple Data) registers in the Intel CPU are used for the floating point calculations (Gerber 2002). Each SIMD register can store two double precision values. The SIMD instructions that can carry out two double-precision operations with a single instruction are used. This allows the calculations of the updated resolution matrix values for two new test configurations to be carried out simultaneously. The calculations for several pairs of the test configurations are then also carried out in parallel by using the multiple cores in modern CPUs (Chandra *et al.* 2001; Chapman *et al.* 2008). The Intel i7 (Nehalem) processor has four physical cores, but it has a hyper-threading capability where each physical core can be used as two logical processors (Gerber 2002). The hyper-threading function can reduce the calculation time by up to 30%.

In this study, the Intel i7 CPU was programmed as an eight cores processor. Together with the use of the SIMD registers, this means that the resolution matrices for 16 test configurations are calculated in parallel. Thus it is only necessary to transfer the elements of the \mathbf{A}_b and \mathbf{B}_b matrices once from the computer memory to the CPU for 16 test configurations. On the 2.66 GHz Intel i7 system, the computational time required for a survey line with 30 electrodes was reduced to about 200 seconds (Table 1).

Table 1 here.

(e) **The matrix-matrix method**

The bulk of the numerical calculations involve matrix-vector multiplications of the form $\mathbf{z}=\mathbf{B}_b\mathbf{g}$ and $\mathbf{y}=\mathbf{A}_b\mathbf{z}$. A single matrix-matrix multiplication is more efficient than a series of equivalent matrix-vector multiplications (Dongarra *et al.* 1998). The next step is to calculate the change in the resolution matrix elements for a large number of add-on arrays at the same time using the following equations.

$$\mathbf{Z} = \mathbf{B}_b\mathbf{G} \text{ and } \mathbf{Y} = \mathbf{A}_b\mathbf{Z}, \text{ where } \mathbf{G} = [\mathbf{g}_1\mathbf{g}_2\dots\mathbf{g}_k] \text{ and } \mathbf{Z} = [\mathbf{z}_1\mathbf{z}_2\dots\mathbf{z}_k] \quad (9)$$

The columns of the matrix \mathbf{G} consist of the sensitivity vectors \mathbf{g}_i for k different test array configurations. The optimum value for k was found to be 28 for 64-bit Intel CPUs with 16 SIMD registers (Leiterman 2005). In the matrix-vector method described previously, a single SIMD register (which can store two double precision values) was used to for the calculation of two \mathbf{z} vectors with a single transfer of the elements of the \mathbf{B}_b matrix from the memory to the CPU. The matrix-matrix method essentially carries out this optimization further by using 14 SIMD registers for the calculation of 28 \mathbf{z} vectors. Using the eight (virtual) cores of the i7 processor, the calculations for 224 \mathbf{z} vectors can be carried out for a single transfer of the elements of the \mathbf{B}_b matrix from the memory to the CPU. A similar optimization is made for the calculation of the \mathbf{Y} matrix. It is only necessary to transfer each element of the \mathbf{A} (or \mathbf{B}) matrix from the computer memory to the CPU once where it can be used 224 times for the same calculations involving different add-on array configurations. Once a data value is transferred from the memory to the CPU, it is stored in a high speed internal data cache that can be accessed by the multiple CPU cores. This reduces the time taken by the memory transfer for these matrices by a factor of about 200 times. The matrix-matrix version of the updating formula reduces the calculation time for the 30 electrodes example by more than half to 87 seconds (Table 1). An examination of the times taken by the different program routines show that 76% of time is used in calculating the matrix-matrix multiplications in

equation (9) for the 30 electrodes test (and 92% for the 50 electrodes test). Thus the next step is to reduce the time taken by the matrix-matrix multiplications.

(f) The GPU matrix-matrix method

The Intel CPU used has 4 physical cores but the Graphics Processor Unit (GPU) in the Nvidia GTX 285 graphics card has several hundred parallel computational units (Owens *et al.* 2007; Nvidia 2008). The GPU is limited to simpler numerical operations than the CPU but it is well suited for matrix-matrix calculations. The calculations for 512 configurations can be carried out simultaneously (i.e. k in equation (9) is now 512) using the GPU. This reduces the calculation time for the 30 electrodes example to 50 seconds (Table 1). An examination of the times taken by the different subroutines in the program reveal that the major part is taken in transferring the data between the main computer (CPU) memory and the graphics card (GPU) memory. It takes about 60% of the overall time taken by the program, whereas the numerical calculations within the GPU for equation (9) take only about 0.2% of the overall time. The main bottleneck for the GPU program version is now the transfer rate of the data between the CPU memory and the GPU memory. The PCI-E 2.0 (Peripheral Component Interconnect Express) graphics card bus in the computer system used has a transfer rate of 500 MB/s. When the program was tested on an older computer system with a PCI-E 1.0 bus (that has a transfer rate of 250 MB/s), the time taken for the data transfer was almost doubled. This confirms that the transfer rate of data between the CPU memory and the GPU memory is the main limiting factor. It is only necessary to transfer the elements of the **A** and **B** matrices from the CPU memory to the GPU memory once in each iteration. However, the sensitivity vectors for different sets of the array configurations in the comprehensive data set have to be transferred repeatedly from the CPU memory to the GPU memory. Similarly, the results of the matrix-matrix products **Z** and **Y** are transferred from the GPU memory to the CPU memory for each set of 512 array configurations.

(g) The single-precision GPU matrix-matrix method

The following function F_{CR} is used to rank the improvement in the model resolution due to an add-on array.

$$F_{CR} = \frac{1}{m} \sum_{j=1}^{j=m} \frac{R_{b+1}(j, j)}{R_b(j, j)} = \frac{1}{m} \sum_{j=1}^{j=m} \left[1 + \frac{\Delta R_b(j, j)}{R_b(j, j)} \right] = 1 + \frac{1}{m} \sum_{j=1}^{j=m} \frac{\Delta R_b(j, j)}{R_b(j, j)} \quad (10)$$

The change in the resolution $\Delta R_b(j, j)$ can be several orders of magnitude smaller than resolution value $R_b(j, j)$. Thus equation (8) is less sensitive to round-off errors compared to the direct use of equation (6). The calculations for $\Delta \mathbf{R}_b$ are next carried out in single-precision in the GPU to further reduce the computer time. Table 1 gives the computer time and average relative model resolution ratios achieved by the different versions of the 'Compare R' method.

The average relative model resolution is given by $S_r = \frac{1}{m} \sum_{i=1}^{i=m} R_b(i, i) / R_c(i, i)$ where $R_b(i, i)$ and $R_c(i, i)$ are the base and comprehensive data set model resolutions of the i th cell for a model with m cells. The GPU single-precision version is about twice as fast as the double-precision version while differences in the average relative model resolutions are less than 1%. To put the numerical improvements made in perspective, the double-precision and single-precision GPU versions respectively take 50 and 29 seconds compared to 6 hours (21600 seconds) in the original version by Wilkinson *et al.* (2006b) for the 30 electrodes example. The computer time has been reduced by more than two orders of magnitude.

Figure 1a shows the change in the average relative model resolution with iteration number for survey lines with 30 to 60 electrodes. In general, there is an initial rapid increase in the model resolution followed by a slower increase. Figure 1b shows a plot of the average relative model resolution values against the ratio of the number of arrays in the optimized data set to the total number in the comprehensive data set. The optimized data sets for different survey lines have similar average relative resolution values when the ratio exceeds 1.5%. Table 1 also gives the total number of array configurations generated after 40

iterations. For a survey line with 30 electrodes, there are 4618 optimized arrays configurations. This is much higher than that normally collected in most field surveys where about 200 to 500 data points are usually collected using conventional arrays (Wilkinson *et al.* 2006a; Loke *et al.* 2010). Thus we next examine the use of the array optimization method for generating smaller optimized data sets. In the following tests, the double-precision GPU version of the routines is used.

Figure 1 here.

RESULTS

(a) Optimizing the number of add-on arrays

In this section, we attempt to find the maximum average model resolution that can be achieved for a given number of array configurations by modifying one of the parameters in the local optimization method used to generate the optimized array data sets. The Sherman-Morrison update calculates the change in the model resolution for a single add-on array. As the change in the model resolution depends on the existing base data set (through the **A** and **B** matrices that in turn depend the **J** Jacobian matrix), this method is only guaranteed to correctly identify the add-on array that gives the largest change in the resolution values. Thus, in theory, the optimal method to generate the data set is to add only the array (or symmetrical pair of arrays) that gives the largest increase in the model resolution values to the base data set after each iteration. However, this approach is very expensive in terms of computer time, particularly for the longer survey lines. Stummer *et al.* (2004) and Wilkinson *et al.* (2006a; 2006b) increased the size of the optimized data set by 9% after each iteration.

The tremendous reduction in the computer time achieved by the techniques described in the previous section now makes it possible to use a smaller step size when augmenting the optimized data set after each iteration. The calculations are made for a survey with 30 electrodes. Figure 2a shows the change in the average relative model resolution when the size

of the base data set is increased by 3, 4.5, 6 and 9 % respectively after each iteration for up to about 800 data points (since our interest is in generating small optimized data sets). The curves start to converge when the number of data points exceeds 700. However for less than 600 data points there are significant differences in the model resolution achieved when different step sizes are used. The data set generated with 3% step size has a significantly higher resolution value compared to that generated with the 9% step size. The resolution curves for the 4.5 and 6 % step sizes lie in between the two curves.

Figure 2 here.

We also present results using a new ‘single step’ algorithm where only the array that gives the largest increase in the average relative model resolution is added to the base set after each iteration. In most cases when the array is not symmetrical about the center of the survey line, the corresponding array on the other half of the survey line is also added. In theory, both arrays give the same increase in the model resolution values when the distribution of the data points in the starting base set is symmetrical. Thus for most iterations, two arrays are added to the base set in each iteration in the ‘single step’ method. The ‘single step’ method produce array sets that have the highest average relative model resolution values (Figure 2a). It represents an upper bound for the model resolution (for the same number of data points) that can be achieved by the array optimization algorithm.

Figure 2b shows the change in the average spread criterion value, S_{CR} , with the number of data points. There is an initial rapid decrease in the average spread value followed by a slower decline after about 400 data points. The curves generated with the smaller step sizes have significantly lower average spread values for less than 500 data points after which the curves tend to converge. In general the average spread curves tend to mirror the behaviour of the average resolution curves with higher resolution values corresponding to lower spread values.

Table 2 lists the number of iterations and computer time required needed to generate an optimized data set with 400 data points using the different step sizes. The average relative model resolution and spread criterion values are also given. The left side of Figure 3 shows the relative model resolution sections that give more information about the performance of the algorithm with different step sizes. The largest difference is in the lower part of the sections. The initial base set consisting of the 147 dipole-dipole arrays with a dipole length of 1 meter and the 'n' factor of 1 to 6 has low relative model resolution values of less than 0.5 below the top 3 meters (Figure 3a). The section for the optimized data set generated with a step size of 9% exhibits low resolution values below the depth of 6.4 meters, particularly around the 15 meters mark near the center (Figure 3b). This low resolution patch is significantly reduced when the step size is reduced to 6%. Reducing the step size to 4.5% and to 3% further increases the resolution values in the lower part of the model sections. Almost the entire section has relative resolution values of above 0.7 when the 3% or 'single step' sizes are used. Although the time taken to generate 400 data points with the 3% step about 3 times that required with a 9% step size, the model resolution is significantly higher. The number of iterations (and computer time) required by the 'single step' method is more than 4 times that required by a step size of 3%.

Table 2 here.

Figure 3 here.

The right side of Figure 3 shows the spread criterion sections for the initial base set and the optimized data sets. The initial base set (Figure 3b) has large spread values below the first few meters reaching up to 108.3 at the bottom left and right corners of the section. The optimized data set with a 9% step size has much lower spread values with a maximum value of about 9.2. The spread values near the bottom of the model section are progressively reduced as the step size is reduced. The maximum spread value is reduced to 8.2 with a 3%

step size, and to 8.0 with a 'single step' size. There is a close correspondence between increasing model resolution and decreasing spread values.

Similar tests were carried out for surveys lines with up to 60 electrodes. As an example, Figure 4a shows the change in the average relative model resolution with number of data points using the different step sizes for a survey line with 50 electrodes. There is a significant increase in the resolution when the step size is reduced from 9% to 6% particularly for less than 2000 data points. The gap between the two curves become smaller as the data set size increases from 2000 to 2500 data points. There are smaller, but still significant, increases in the model resolution when the step size is reduced to 4.5% and 3%. The 'single' step method achieves significantly higher model resolution values compared to the 3% step size up to about 1500 data points. The average spread value curves (Figure 4b) show a similar pattern with the 'single step' method having the lowest spread values and the 9% step size having the highest spread values with all the curves converging above 2000 data points.

Figure 5 shows the relative resolution and spread criterion value sections for the initial base set (with 267 data points) and the optimized data sets (with 1000 data points) generated using the different step sizes. The initial base set with 267 dipole-dipole array data points has low relative model resolution values of less than 0.5 below the top 3 meters (Figure 5a). For the optimized data sets with 1000 data points, the relative model resolution sections show that most of the region below a depth of about 6 meters has values of less than 0.6 when a step size of 9% is used (Figure 5b), but has values of above 0.65 with a step size of 3% (Figure 5e). The average relative resolution value is increased to 0.768 when a 'single step' size is used, compared to 0.662 and 0.733 with step sizes of 9% and 4.5%. The spread criterion sections show a gradual reduction in the spread values, particularly in the lower portion of the model sections, as the step size is reduced (right column of Figures 5). Again, the greatest

benefit of using a smaller step size is an improvement in the resolution of the lower part of the model section. The price of the higher resolution is much higher computer times required to generate the optimized data sets with the smaller step sizes (Table 3), particularly for the 'single step' method. The 'single' step size method took 3.8 hours to generate the optimized data set with 1000 data points, compared to 17 and 26 minutes using step sizes of 4.5% and 3%. The use of a step size of between 3% and 4.5% probably represents the best compromise between maximizing the model resolution and reducing the calculation time for such long survey lines.

Table 3 here.

Figure 4 here.

Figure 5 here.

(b) Synthetic model inversion test

Figure 6a shows a test model with 4 rectangular blocks at different depths in a background medium of 10 Ω .m below a 2-D survey line with 35 electrodes 1 meter apart. Three of the blocks have 100 Ω .m resistivity. One block has a gradational boundary rising from 20 to 100 Ω .m to simulate a smooth edge. The horizontal distance between the edges of the third and fourth deepest blocks is less than the depth to the deepest block. This makes it a more challenging test in separating the two deepest blocks compared to earlier test models used by Wilkinson *et al.* (2006a; 2006b) and Loke *et al.* (2010). Figures 6b and 6c show the apparent resistivity pseudosections for the Wenner-Schlumberger (Pazdirek & Blaha 1996) and dipole-dipole arrays. For both arrays, we use all the possible measurements subject to the restriction that the geometric factor does not exceed that that for a dipole-dipole array with '*a*' equal to 1 m and '*n*' equal to 6 (i.e. a geometric factor of 1056 m). This results in data sets with 599 and 530 data points respectively for the Wenner-Schlumberger and dipole-dipole arrays. The apparent resistivity pseudosection for the optimized array data set (using a 4.5%

step size) with 599 data points is shown in Figure 6d for comparison. The contour pattern in the pseudosection has a more jagged appearance compared to the conventional arrays. This is because the data consists of a mixture of arrays of the Alpha and Beta types (Carpenter & Habberjam 1956) that have different responses to the subsurface resistivity. This is illustrated by the differences in the pseudosections of the Wenner-Schlumberger (an Alpha type) and dipole-dipole (a Beta type) arrays.

Figure 6 here.

The smoothness-constrained Gauss-Newton least-squares optimization method used for the inversion of the data sets is described in Loke and Dahlin (2002) and Loke *et al.* (2003). We use the ‘discrepancy principle’ technique (Farquharson and Oldenburg 2004) in selecting the damping factor. A relatively large damping factor value is initially used (usually about 0.10 to 0.30) that is reduced by half after each iteration until the desired data misfit value is obtained. We choose a target data misfit of 0.5% for the data sets without noise that is similar to the accuracy of the finite-difference forward modelling used. We show the inversion model for the first iteration where the data misfit falls below 0.5% for the noise-free test data sets. The final damping factor used at this iteration is usually between 0.005 and 0.010. The smoothness-constrained inversion method using an L_1 -norm inversion was used (Loke *et al.* 2003). The same inversion settings were used for the different data sets.

The two upper blocks in the resulting inversion model for the Wenner-Schlumberger data set after 6 iterations are fairly well resolved while the third deepest block is barely resolved (Figure 6e). For the dipole-dipole data set, the third deepest block is fairly well resolved in the inversion model (Figure 6f). The deepest block shows up as an area of higher resistivity but it is not well resolved. The anomaly corresponding to this block is just barely separated from the third deepest block.

The next test is with the optimized data sets (with 599 data points) using step sizes of 9% and 4.5%. The previous section showed that using a smaller step size can significantly improve the resolution. In the first optimized data set model (with a step size of 9%) the third deepest block is well resolved with a maximum resistivity of about 31 Ω .m, and the deepest block is now visible (Figure 6g) and clearly separated from the third block. The deepest block is even more clearly resolved (with a maximum resistivity of 17 Ω .m) in the inversion model for the optimized data set using a step size of 4.5% (Figure 6h). This agrees with the earlier observation that the data set using a smaller step size has significantly better resolution in the lower part of the model section. The optimized data sets using step sizes of 9% and 4.5% have average model resolution ratios of 0.770 and 0.804 respectively.

Figure 7 shows the results of tests when noise is added to the data sets. Zhou & Dahlin (2003) demonstrated that the error in resistivity field measurements varies inversely with the measured potential value. Gaussian random noise (Press *et al.* 1992) is added to the potential values (for a current of 1 Ampere) for the different array configurations to simulate such potential dependent noise. The potential values are then converted to apparent resistivity values by multiplying with the geometric factor. The amplitude of the potential noise is chosen so that the readings with the lowest potential (and also the largest geometric factor) have a noise level of 10 percent. The average percentage apparent resistivity noise depends on the geometric factors of the array configurations in the data set used. As an example, Figure 7a shows the dipole-dipole array data set with overlapping data levels with the potential dependent noise added. This should be compared to Figure 6c that shows the same pseudosection without noise. The largest differences between the two pseudosections are in the lower sections that correspond to measurements with the larger geometric factors. The inversion model for the Wenner-Schlumberger array (Figure 7b) shows small changes compared to the model without the noise (Figure 6e). This is because the array is relatively

insensitive to noise. The average geometric factor for this data set is 123 m. In comparison, the dipole-dipole array data set that has an average geometric factor of 322 m is more sensitive to noise. This is also reflected in the data misfits of 0.9% for the Wenner-Schlumberger model and 1.9% for the dipole-dipole array model. The lower part of the dipole-dipole array model where the deepest block is located has significant distortions (Figure 7c). The high resistivity zone corresponding to the deepest block is barely visible, unlike the model obtained for the same data set without noise (Figure 6f). The optimized data set with a 4.5% step size has a higher data misfit of 3.1% due to the higher average geometric factor of 569 m. The deepest block is still resolved (Figure 7d) although there is a slight shift in its position to the left.

Figures 7e and 7f show the inversion models for the dipole-dipole array and optimized data sets using the L-curve method (Farquharson and Oldenburg 2004) to automatically select the damping factor. While there are slight differences in the models obtained using the ‘discrepancy principle’ method (Figures 7c and 7d), the models (Figures 7e and 7f) clearly show that the deepest block is much better resolved by the optimized data set although the L-curve method selected a lower damping factor value (0.008) for the dipole-dipole array data set compared to the optimized data set (0.013). The lower damping value selected by the L-curve method is probably due to the lower average noise level in the dipole-dipole data set (due to its lower average geometric factor compared to the optimized data set). The damping factor for the optimized data set selected by the L-curve method is slightly higher (0.013 compared to 0.010) than that used by the model in Figure 7d. This results in slightly lower resistivity values for the deepest block (comparing Figures 7f and 7d) but the model also has fewer artefacts (such as the low resistivity area under the second upper block) due to the noise.

Figure 7 here.

(c) **Array optimization using the smoothness constraint**

In calculating the model resolution values, and consequently the optimized arrays, we have used the simple damped (Levenberg-Marquardt) least-squares method by setting the \mathbf{C} matrix in equation (1) to be equal to the identity matrix \mathbf{I} . This choice was made so that the results can be directly compared with earlier work by Wilkinson *et al.* (2006a; 2006b). However most 2-D resistivity inversion work use a smoothness-constrained least-squares method (deGroot-Hedlin & Constable 1990; Loke *at al.* 2003) where a roughness filter is used to minimize changes in the resistivity between adjacent model cells. The \mathbf{C} matrix is then given by the following equation.

$$\mathbf{C} = \mathbf{d}_z^T \mathbf{d}_z + \mathbf{d}_x^T \mathbf{d}_x \quad (11)$$

The roughness matrices \mathbf{d}_x and \mathbf{d}_z differences the model parameters between adjacent lateral and vertical model cells. The structures of these matrices are described in the paper by deGroot-Hedlin & Constable (1990).

Figure 8a shows the relative model resolution section for the small optimized data set with 599 data points with a step size of 4.5% used in the previous section that was generated using the damped least-squares equation. For comparison, Figure 8c shows a similar relative model resolution section obtained using the smoothness-constrained least-squares equation. Note the high relative model resolution values of over 0.95 are more highly concentrated near the surface for the damped least-squares section compared to the smoothness-constrained least-squares section. The smoothness-constrained section has a more uniform distribution of the high relative resolution values (with significantly higher values in the bottom half) compared to the damped least-squares section. A second test was carried out using a larger optimized data set with 2401 data points that is about four times the size of the small optimized data set. The relative model resolution section for the smoothness-constrained method (Figure 8d) has slightly higher values in the lower part of the model section

compared to that obtained with the damped least-squares method (Figure 8b). The corresponding spread criterion sections are shown on the right side of Figure 8. There is a decrease in the average spread value when the number of data points is increased for both the damped constraint (Figures 8e and 8f) and smoothness constraint (Figures 8g and 8h). The smoothness constraint sections have slightly higher average resolution and lower spread values than the damped constraint sections probably due to differences in the C matrix.

Figure 8 here.

The inversion models for the small optimized data sets generated by both methods are very similar (Figures 9a and 9c) with no significant differences. The deepest block in the model obtained from the inversion of the large optimized data set has a maximum value of 20 $\Omega.m$ (Figure 9b). This is slightly higher than the maximum value of 17 $\Omega.m$ in the inversion model of the small optimized data set (Figure 9a). The inversion model for the large optimized data set generated using the smoothness-constrained least-squares method achieves a higher value of 21 $\Omega.m$ at the location of the deepest block. The base of the block is also slightly better resolved (Figure 9d) compared to the model for large optimized data set with the damped constraint (Figure 9b) This could be due to the slightly higher relative model resolution values in the lower part of the smoothness-constrained model resolution section. Note that the third deepest block is significantly better resolved by both large optimized data sets (where it reaches a maximum value of about 46 $\Omega.m$) compared to the small optimized data sets (maximum value of 36 $\Omega.m$).

Figure 9 here.

CONCLUSION

Optimized arrays generated by maximizing the model resolution have significantly better resolution and greater depth of investigation than conventional arrays for 2-D

resistivity surveys. The computer time required to generate the optimized arrays is greatly reduced by using numerical algorithms that can make the best use of the CPUs and GPUs in modern personal computer systems. The computer time was reduced through three steps. Firstly, the equation used to calculate the change in the model resolution for a single add-on array was simplified so that the final stage only involves vector-vector operations. Secondly, the computer program was optimized at the CPU level by reducing the time taken to transfer data between the main computer memory and the CPU registers through the use of matrix-matrix multiplication algorithms, by storing the temporary variables in the CPU floating point registers, and by using the parallel processing capabilities of modern CPUs. A final reduction in the computer time is achieved by using the graphics card GPU as a highly parallel mathematical coprocessor.

The resolution for small data sets can be significantly improved by using a smaller step size for adding new configurations in the iterative algorithm used to generate the optimized data sets. For small optimized data sets the algorithm is largely insensitive to the type of model constraint used in the optimisation; the simple damped and smoothness-constrained least-squares methods generally gave similar results in terms of the quality of the inversion models obtained. For larger optimized sets (where the number of data points is several times larger than used in conventional arrays), using the smoothness constraint can give slightly better resolution at depth.

While the discussion in this paper is focused on the array optimization problem, the techniques developed in this research can also be used to reduce the computer time required for other aspects of electrical and electromagnetic data interpretation (such as solving the least-squares equation, calculation of the model resolution matrix and calculating the forward model response using finite-element and finite-difference techniques) that use similar matrix and vector operations.

Further research is being carried out to reduce the time required by the matrix-matrix GPU routines to transfer data between the CPU memory and the GPU memory. We are also testing a modified "Compare R" algorithm that minimizes the spread function instead of maximizing the model resolution. Research is also being undertaken on array optimization for 3-D surveys that has been made possible with the fast algorithms described in this paper. Ellis & Oldenburg (1994) showed that the smoothness-constrained least-squares inversion method can be modified to take into account a-priori information so as to improve its resolution in selected regions. We are currently investigating using similar modifications to the least-squares equation to improve the resolution of the optimized data sets. Other aspects of the array optimization problem such as the effects of different model discretizations, the data noise distribution and L1-norm constraints are also being studied.

ACKNOWLEDGEMENTS

This paper is published with permission of the Executive Director of the British Geological Survey (NERC). We would also like to thank the Associate Editor and two anonymous reviewers for their comments that have helped to improve the paper.

REFERENCES

- Auken, E., Pellerin, L., Christensen, N.B. & Sørensen, K., 2006. A survey of current trends in near-surface electrical and electromagnetic methods, *Geophysics*, **71**, G249-G260.
- Bauman, P., 2005. 2-D resistivity surveying for hydrocarbons – A primer, *CSEG Recorder*, **April 2005**, 25-33.
- Bingham, D., Nimeck, G., Wood, G. & Mathieson, T., 2006. 3D resistivity in the Athabasca basin with the pole-pole array, In *1 day workshop - Geophysical methods and techniques applied to uranium exploration proceedings*. SEG Annual General Meeting 2006, New Orleans.
- Camp, W. & Thierry, P., 2010. Trends for high-performance scientific computing, *The Leading Edge*, **29 (1)**, 44-47.
- Carpenter, E.W. & Habberjam, G.M., 1956. A tri-potential method of resistivity prospecting, *Geophysics*, **11**, 455-469.
- Chambers, J.E., Kuras, O., Meldrum, P.I., Ogilvy, R.O. & Hollands, J., 2006. Electrical resistivity tomography applied to geologic, hydrogeologic, and engineering investigations at a former waste-disposal site, *Geophysics*, **71**, B231-B239.
- Chandra, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J. & Menon, R., 2001. *Parallel Programming in OpenMP*, Academic Press, San Diego, California.
- Chapman, B., Jost, G. & van der Pas, R., 2008. *Using OpenMP*, The MIT Press, Cambridge, Massachusetts.

- Clapp, R. & Fu, H., 2010. Selecting the right hardware for reverse time migration, *The Leading Edge*, **29** (1), 48-59.
- Dahlin, T., 1996. 2D resistivity surveying for environmental and engineering applications, *First Break*, **14**, 275-284.
- Dahlin T., Bernstone C. & Loke M.H. 2002. A 3D resistivity investigation of a contaminated site at Lernacken in Sweden, *Geophysics*, **60**, 1682-1690.
- Dahlin, T. & Zhou, B., 2004. A numerical comparison of 2D resistivity imaging with 10 electrode arrays, *Geophysical Prospecting*, **52**, 379-398.
- deGroot-Hedlin, C. & Constable, S., 1990. Occam's inversion to generate smooth, two-dimensional models from magnetotelluric data, *Geophysics*, **55**, 1613-1624.
- Dongarra, J.J., Duff, I.S., Sorensen, D.C. & van der Vorst, H.A., 1998. *Numerical linear algebra for high-performance computers*, Society for Industrial and Applied Mathematics, Philadelphia.
- Ellis, R.G. & Oldenburg, D.W., 1994. Applied geophysical inversion, *Geophysical Journal International*, **116**, 5-11.
- Farquharson, C.G. & Oldenburg, D.W., 2004. A comparison of automatic techniques for estimating the regularization parameter in non-linear inverse problems, *Geophysical Journal International*, **156**, 411-425.
- Friedel, S., 2003. Resolution, stability and efficiency of resistivity tomography estimated from a generalized inverse approach, *Geophysical Journal International*, **153**, 305–316.
- Gerber, R., 2002. *The Software Optimization Cookbook : High-performance recipes for the Intel architecture*, Intel Press, Hillsboro, Oregon
- Golub, G.H. & van Loan, C.F., 1989. *Matrix Computations*, 2nd edn, The John Hopkins University Press, Baltimore and London.

- Jackson, D.D., 1972. Interpretation of inaccurate, insufficient and inconsistent data, *Geophysical Journal of the Royal Astronomical Society*, **28**, 97–109.
- Kadlec, B. & Dorn, G., 2010. Leveraging graphics processing units (GPUs) for real-time seismic interpretation, *The Leading Edge*, **29** (1), 60-67.
- Kuras, O., Meldrum, P.I., Beamish, D., Ogilvy, R.D. & Lala, D., 2007. Capacitive Resistivity Imaging with Towed Arrays, *Journal of Environmental and Engineering Geophysics*, **12** (3), 267-279.
- Leiterman, J.C., 2005. *32/64-bit 80x86 assembly language architecture*, Wordware Publishing, Inc.
- Loke M.H. & Barker, R.D., 1995. Least-squares deconvolution of apparent resistivity pseudosections, *Geophysics*, **60**, 1682-1690.
- Loke, M.H. & Dahlin, T., 2002. A comparison of Gauss-Newton and quasi-Newton methods in resistivity imaging inversion, *Journal of Applied Geophysics*, **49**, 149-162.
- Loke M.H., Acworth I. & Dahlin T., 2003. A comparison of smooth and blocky inversion methods in 2D electrical imaging surveys, *Exploration Geophysics*, **34**, 182-187.
- Loke, M.H., Wilkinson, P. & Chambers, J., 2010, Fast computation of optimized electrode arrays for 2D resistivity surveys, *Computers and Geosciences*, in press.
- Menke, W., 1989. *Geophysical data analysis: Discrete inverse theory*, Rev. ed, Academic Press, Inc.
- Michéa, D. & Komatitsch, D., 2009. Accelerating a three-dimensional finite-difference wave propagation code using GPU graphics cards. *Geophysical Journal International*, **182**, 389-402.
- Miller, C.R. & Routh, P.S., 2007. Resolution analysis of geophysical images: Comparison between point spread function and region of data influence measures, *Geophysical Prospecting*, **55**, 835–852.

- Moorkamp, M., Jegen, M., Roberts, A. & Hobbs, R. 2010. Massively parallel forward modeling of scalar and tensor gravimetry data, *Computers & Geosciences*, **36**, 680-686
- Nvidia, 2008. *Nvidia CUDA programming guide version 2.1*, Nvidia Corporation.
- Oldenborger, G.A. & Routh, P.S., 2009. The point-spread function measure of resolution for the 3-D electrical resistivity experiment. *Geophysical Journal International*, **176**, 405–414.
- Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.. & Purcell, T.J., 2007. A Survey of General-Purpose Computation on Graphics Hardware, *Computer Graphics Forum*, **26**, 80-113.
- Pazdirek, O. & Blaha, V., 1996. Examples of resistivity imaging using ME-100 resistivity field acquisition system, In *EAGE 58th Conference and Technical Exhibition Extended Abstracts*, Amsterdam.
- Press, W.H. , Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P, 1992. *Numerical Recipes in C*, 2nd edn, Cambridge University Press, Cambridge, UK.
- Sava, P., 2010. An introduction to this special section: High performance computing. *The Leading Edge*, **29 (1)**, 42-43.
- Samouëlian, A., Cousin, I., Richard, G., Tabbagh, A. & Bruand, A, 2003. Electrical resistivity imaging for detecting soil cracking at the centimetric Scale. *Soil Science Society of America Journal*, **67**, 1319-1326.
- Seaton, W.J. & Burbey, T.J., 2000. Aquifer characterization in the Blue Ridge physiographic province using resistivity profiling and borehole geophysics : Geologic analysis, *Journal of Environmental & Engineering Geophysics*, **5 (3)**, 45-58.
- Stummer, P., Maurer, H. & Green, A., 2004. Experimental design: Electrical resistivity data sets that provide optimum subsurface information, *Geophysics*, **69**, 120-129.

- Wilkinson, P.B., Kuras, O., Meldrum, P.I., Chambers, J.E. & Ogilvy, R.D., 2006a. Comparison of the spatial resolution of standard and optimised Electrical Resistivity Tomography arrays, In *EAGE 12th Annual Near Surface Geophysics Conference Extended Abstracts*, Helsinki, Finland.
- Wilkinson, P.B., Meldrum, P.I., Chambers, J.E., Kuras, O. & Ogilvy, R.D., 2006b. Improved strategies for the automatic selection of optimized sets of electrical resistivity tomography measurement configurations, *Geophysical Journal International*, **167**, 1119-1126.
- Zhou, B. & Dahlin, T., 2003. Properties and effects of measurement errors on 2D resistivity imaging surveying, *Near Surface Geophysics*, **1**, 105-117.

Table 1. The times in seconds for 40 iterations (and average relative resolution ratio achieved) for the different versions of the 'Compare R' array optimization method. The number of data points in the optimized (base) data set generated after 40 iterations is shown, together with the total number of data points in the comprehensive data set. The ratio of the number of data points in the optimized base set with the total number in the comprehensive data set is given in percent.

Number of electrodes	Number of data points			Matrix-Vector Double-precision	Matrix-Matrix Double-precision	GPU Matrix Double-precision	GPU Matrix Single-precision
	Base	All	Ratio (%)	Time taken in seconds. (Average relative model resolution)			
30	4618	51283	9.00	202 (0.958)	87 (0.958)	50 (0.958)	29 (0.955)
40	6503	166944	3.90	1245 (0.921)	717 (0.921)	324 (0.921)	148 (0.915)
50	8386	411453	2.04	6118 (0.886)	3908 (0.886)	1405 (0.886)	577 (0.880)
60	10272	854224	1.20	19967 (0.858)	13944 (0.858)	4421 (0.858)	1760 (0.850)

Table 2. Results obtained for a 30 electrodes survey line with different step sizes for an optimized data set with 400 data points. In the single step method, only the array (or symmetrical pair of arrays) that gives the largest increase in average model resolution value is added to the base set.

Step size (%)	Number of iterations	Time taken (s)	Average relative model resolution	Average spread criterion value
Single	131	176	0.833	2.945
3	34	41	0.824	2.972
4.5	23	28	0.804	3.037
6	18	22	0.794	3.066
9	12	14	0.779	3.122

Table 3. Results obtained for a 50 electrodes survey line with different step sizes for an optimized data set with 1000 data points.

Step size (%)	Number of iterations	Time taken (s)	Average relative model resolution	Average spread criterion value
Single	372	13552	0.768	3.723
3	45	1568	0.751	3.801
4.5	30	1046	0.733	3.860
6	23	803	0.719	3.937
9	16	557	0.662	4.070

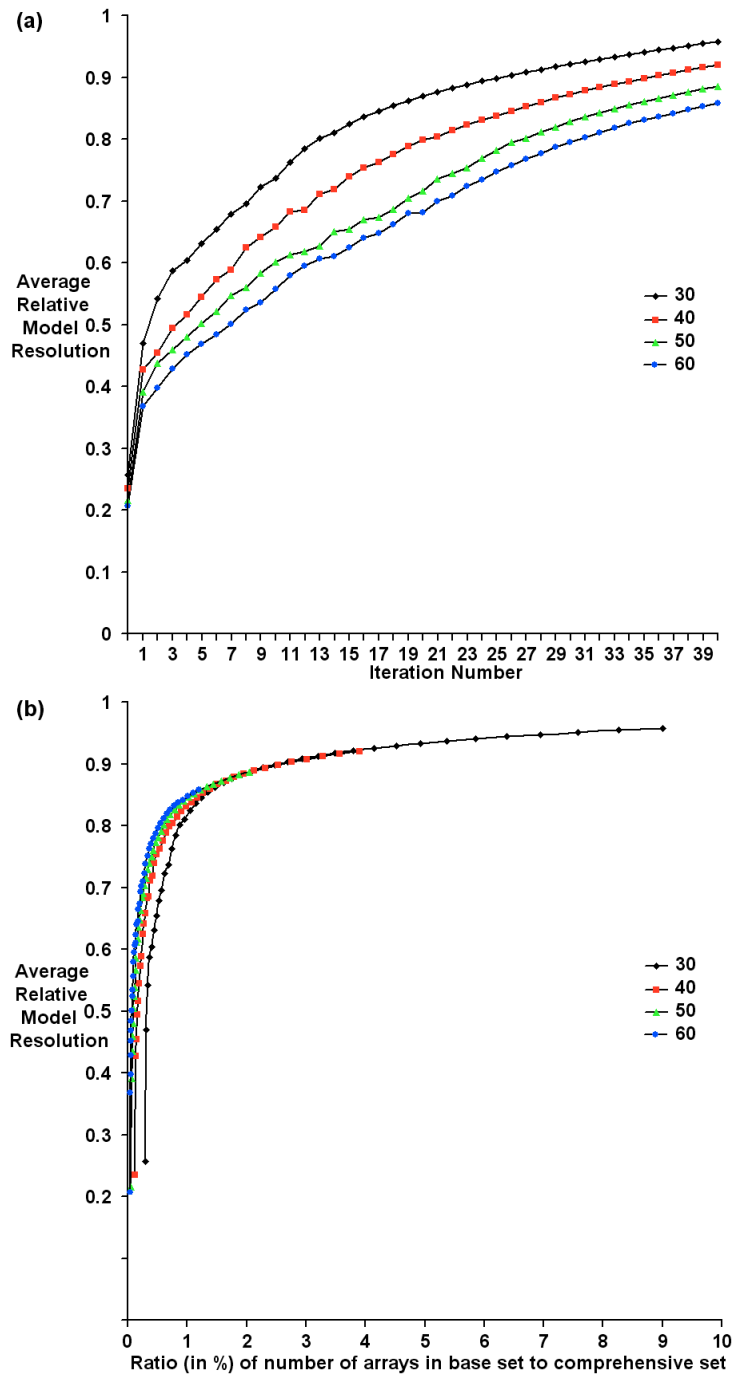


Figure 1. Change of the average relative model resolution with (a) iteration number and (b) ratio of number of arrays in optimized data set to comprehensive data set for survey lines with 30 to 60 electrodes.

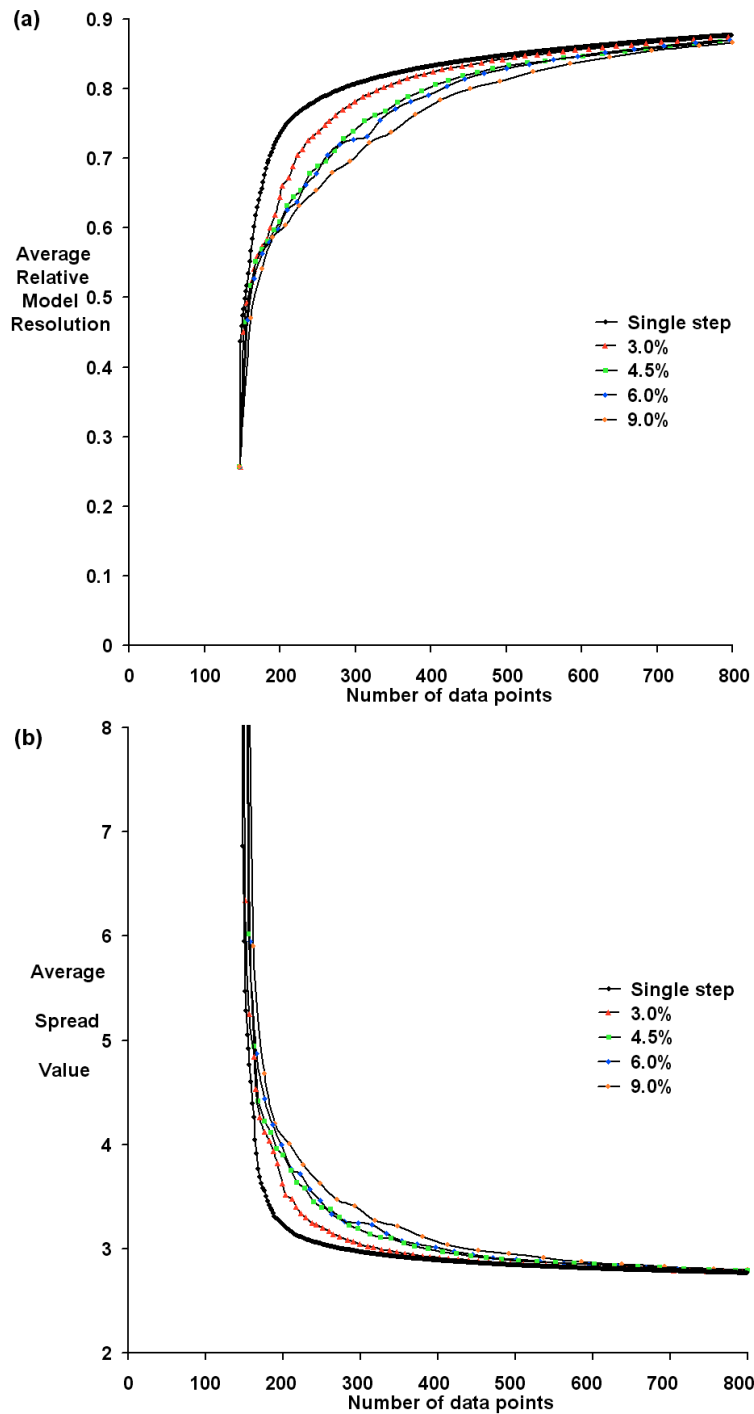


Figure 2. (a) Change of the average relative model resolution with number of data points in the optimized data set for a survey line with 30 electrodes using different step sizes. (b) Similar plots showing change of the average spread criterion value with the number of data points.

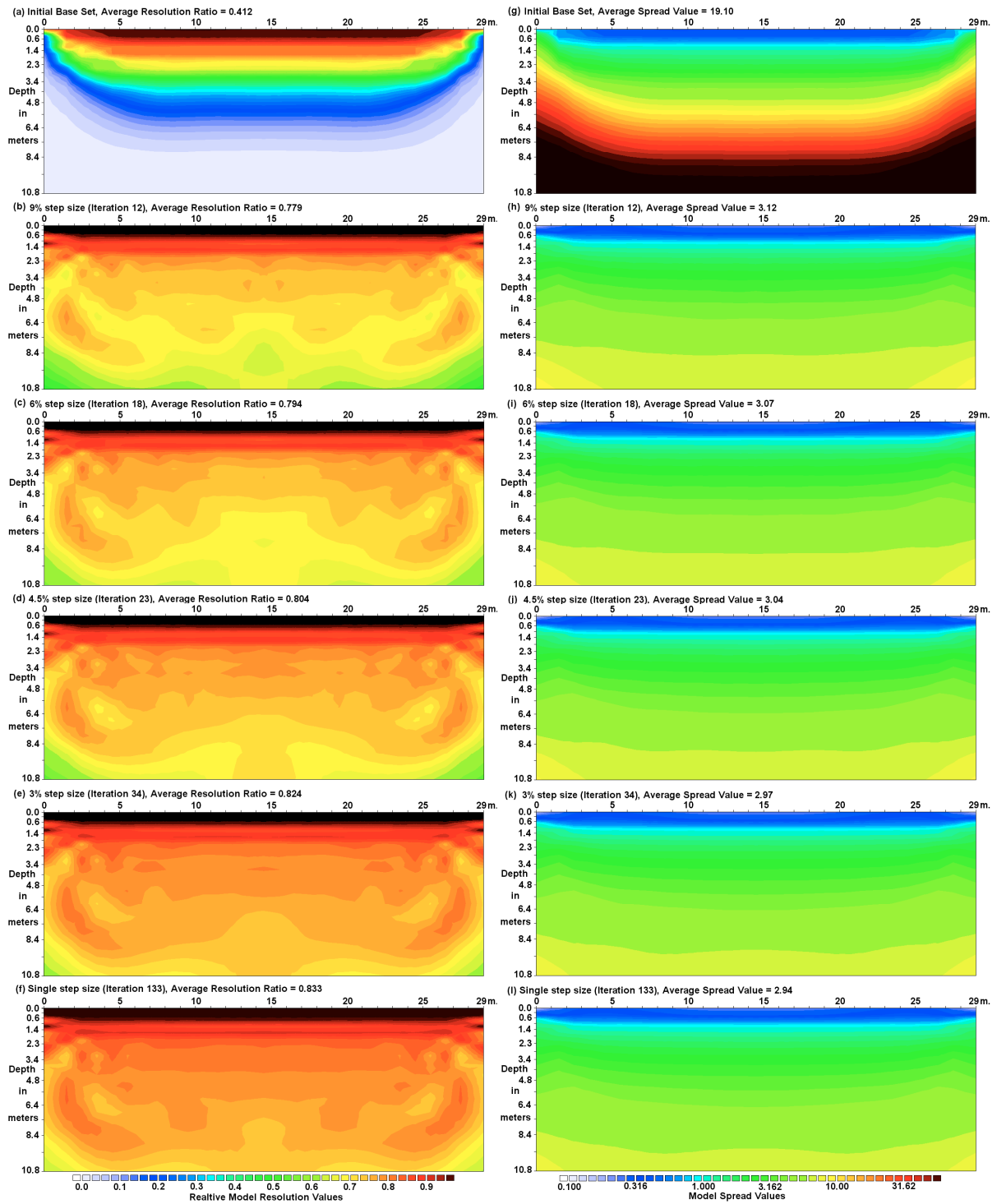


Figure 3. Relative model resolution sections for a survey line with 30 electrodes for (a) initial base set and optimized data sets with 400 points produced using (b) 9%, (c) 6%, (d) 4.5%, (e) 3% and (f) single step sizes. The right column shows corresponding spread criterion value sections.

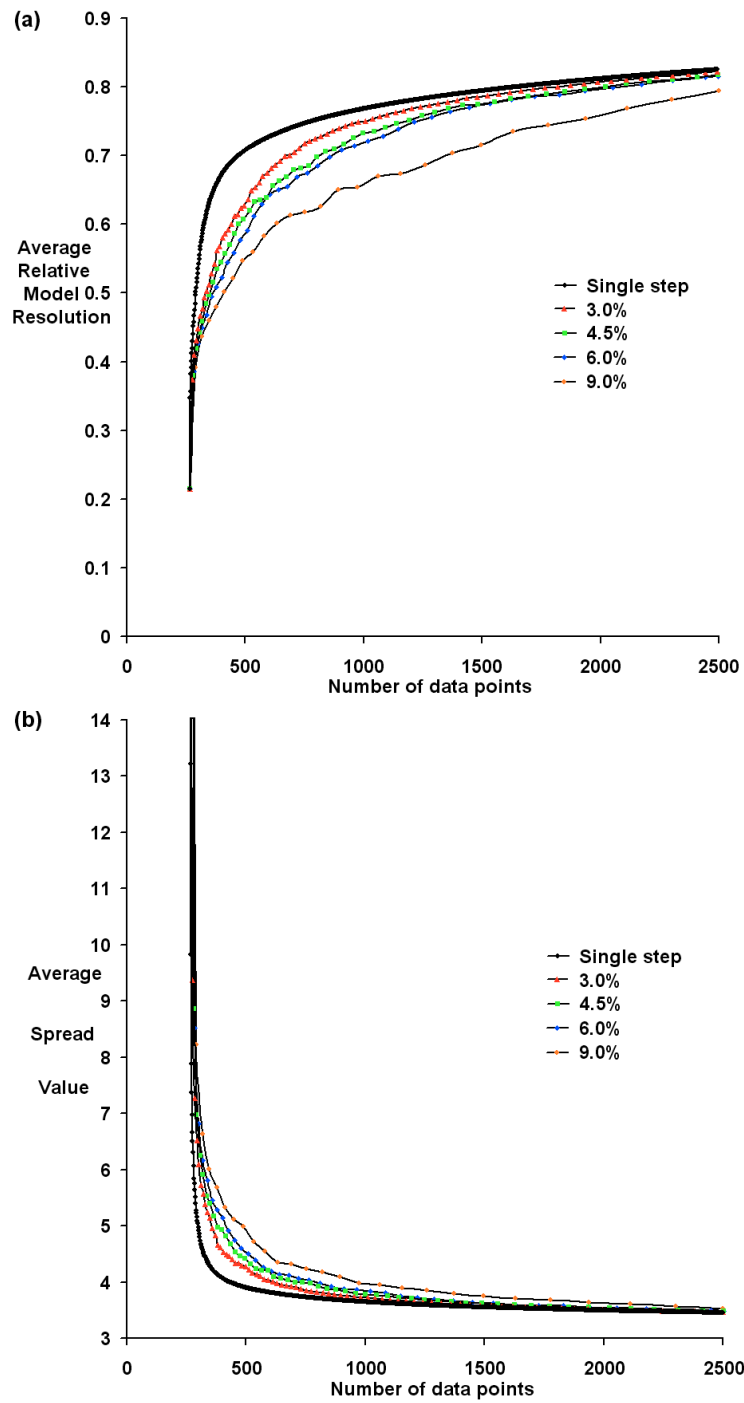


Figure 4. (a) Change of average relative model resolution with number of data points for a survey line with 50 electrodes using different step sizes. (b) Similar plots showing change of the average spread criterion value with the number of data points.

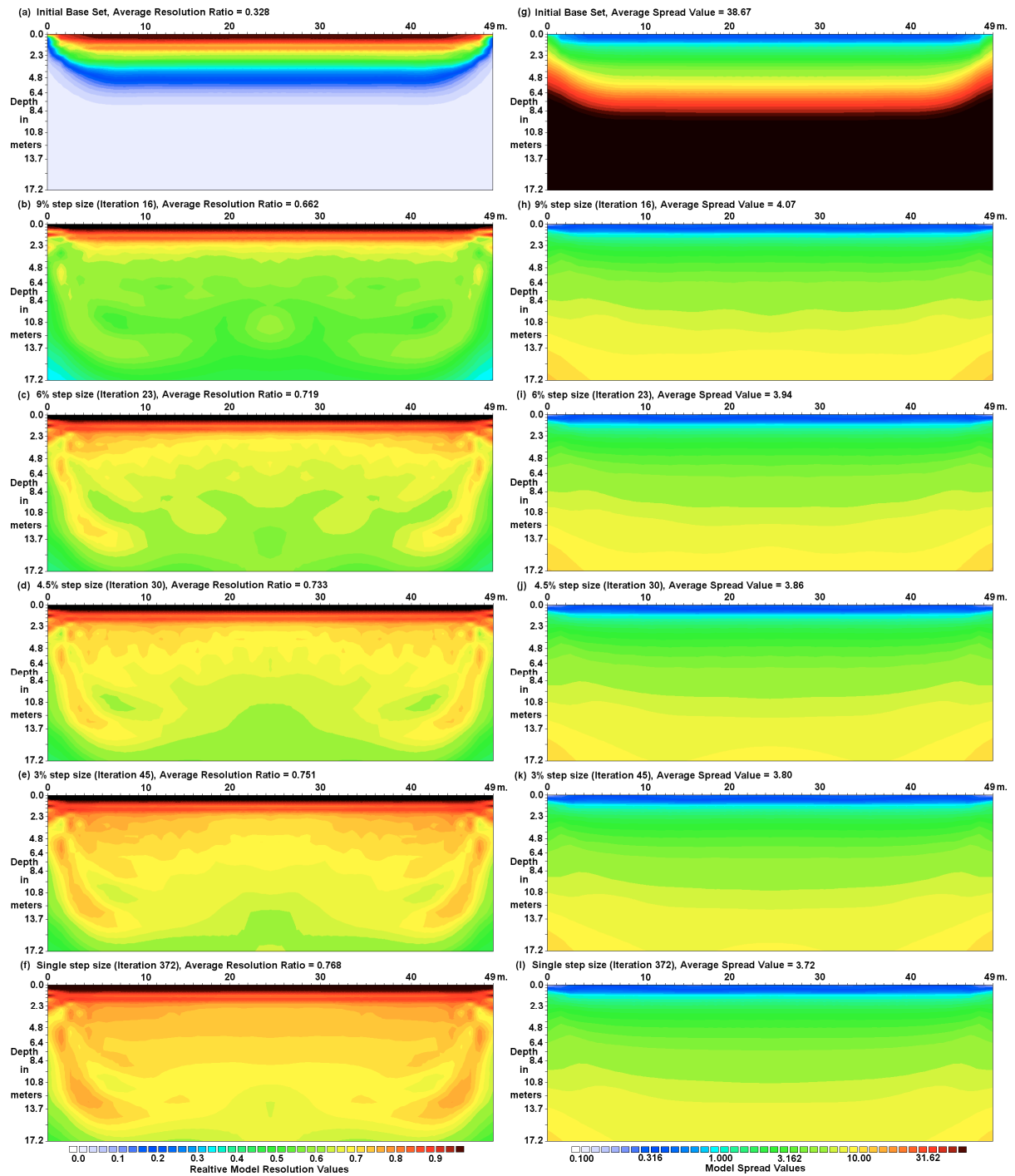


Figure 5. Relative model resolution sections for a survey line with 50 electrodes for (a) initial base set (267 data points) and optimized data sets with 1000 points produced using (b) 9%, (c) 6%, (d) 4.5%, (e) 3% and (f) single step sizes. The right column shows corresponding spread criterion value sections.

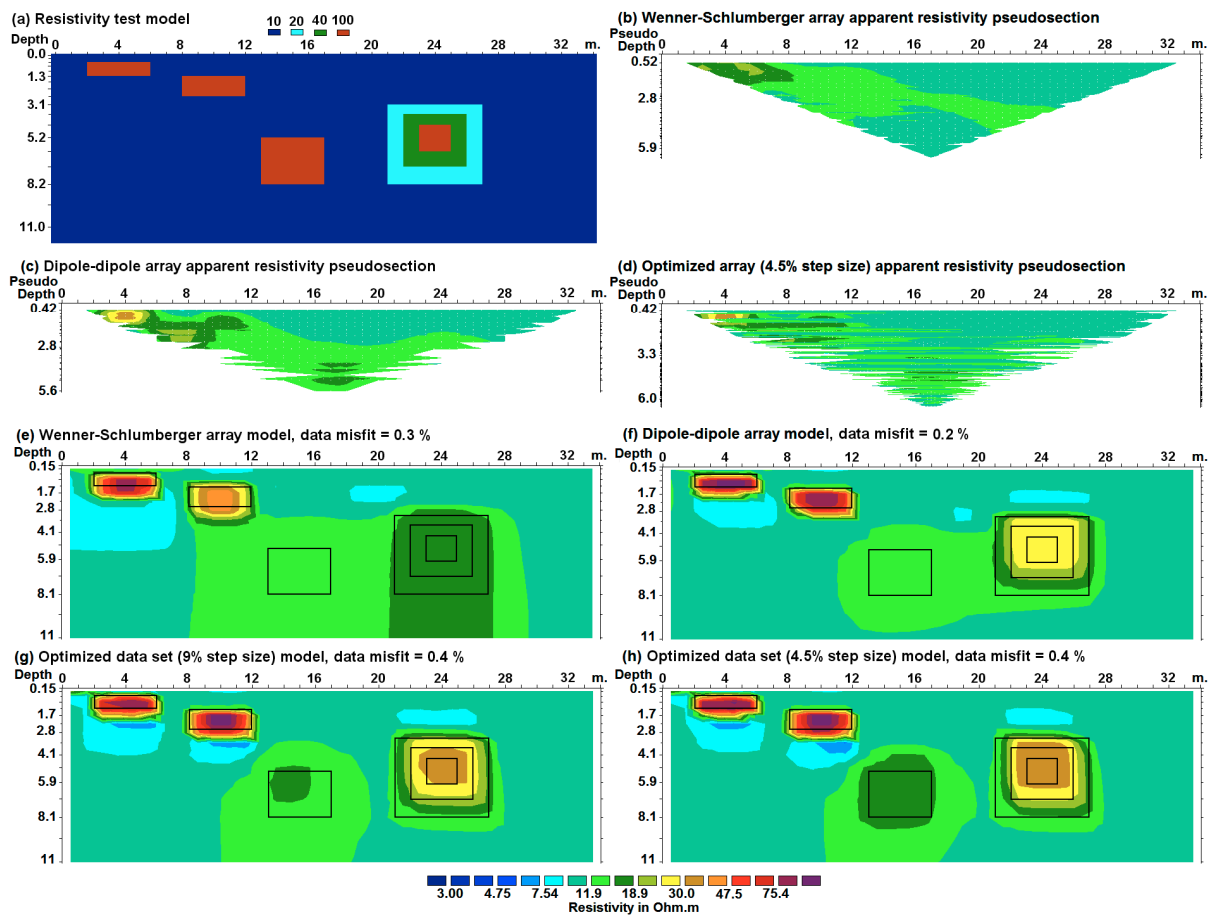


Figure 6. (a) 2D test model. Pseudosections for (b) Wenner-Schlumberger, (c) dipole-dipole and (d) optimized (using a 4.5% step size) arrays. Inversion models for the (e) Wenner-Schlumberger array, (f) dipole-dipole array, (g) optimized array data set using a 9% step size, (h) optimized array data set using a 4.5% step size.

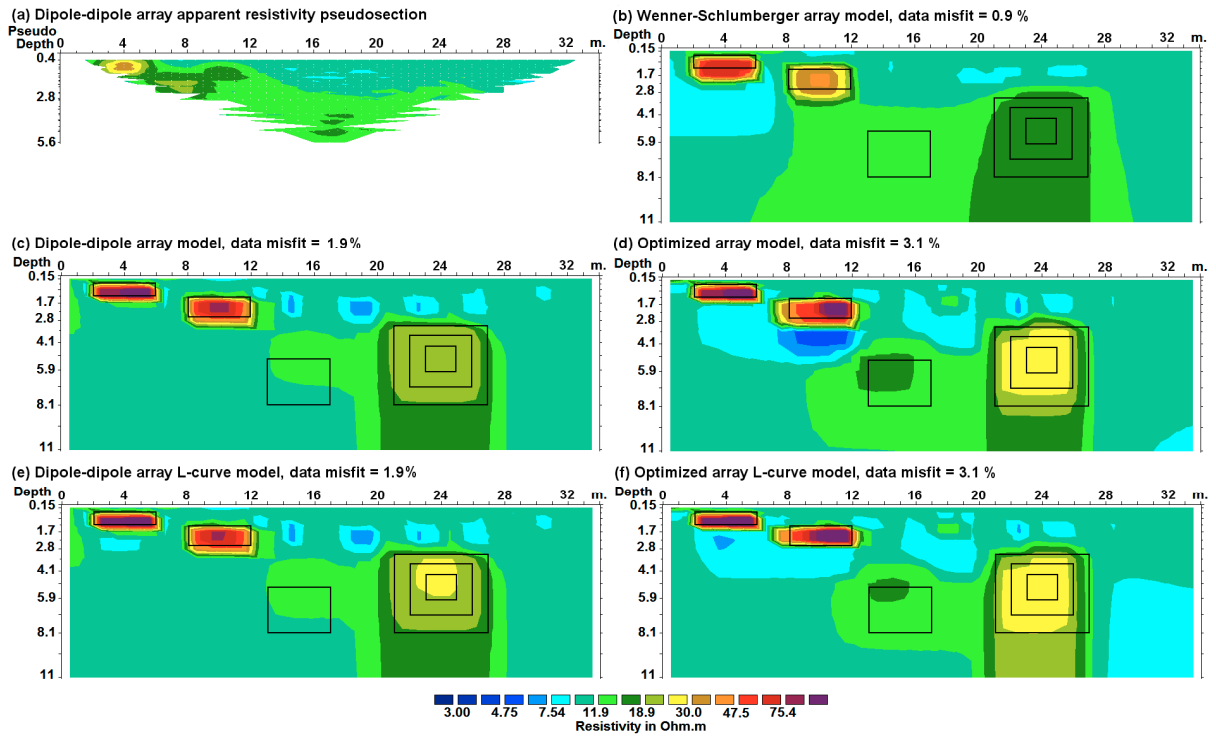


Figure 7. (a) Dipole-dipole array pseudosection with noise added. Inversion model for (b) Wenner-Schlumberger array, (c) dipole-dipole array and (d) optimized array data set using a 4.5% step size. Inversion models using the L-curve method for (e) dipole-dipole array and (f) optimized data sets.

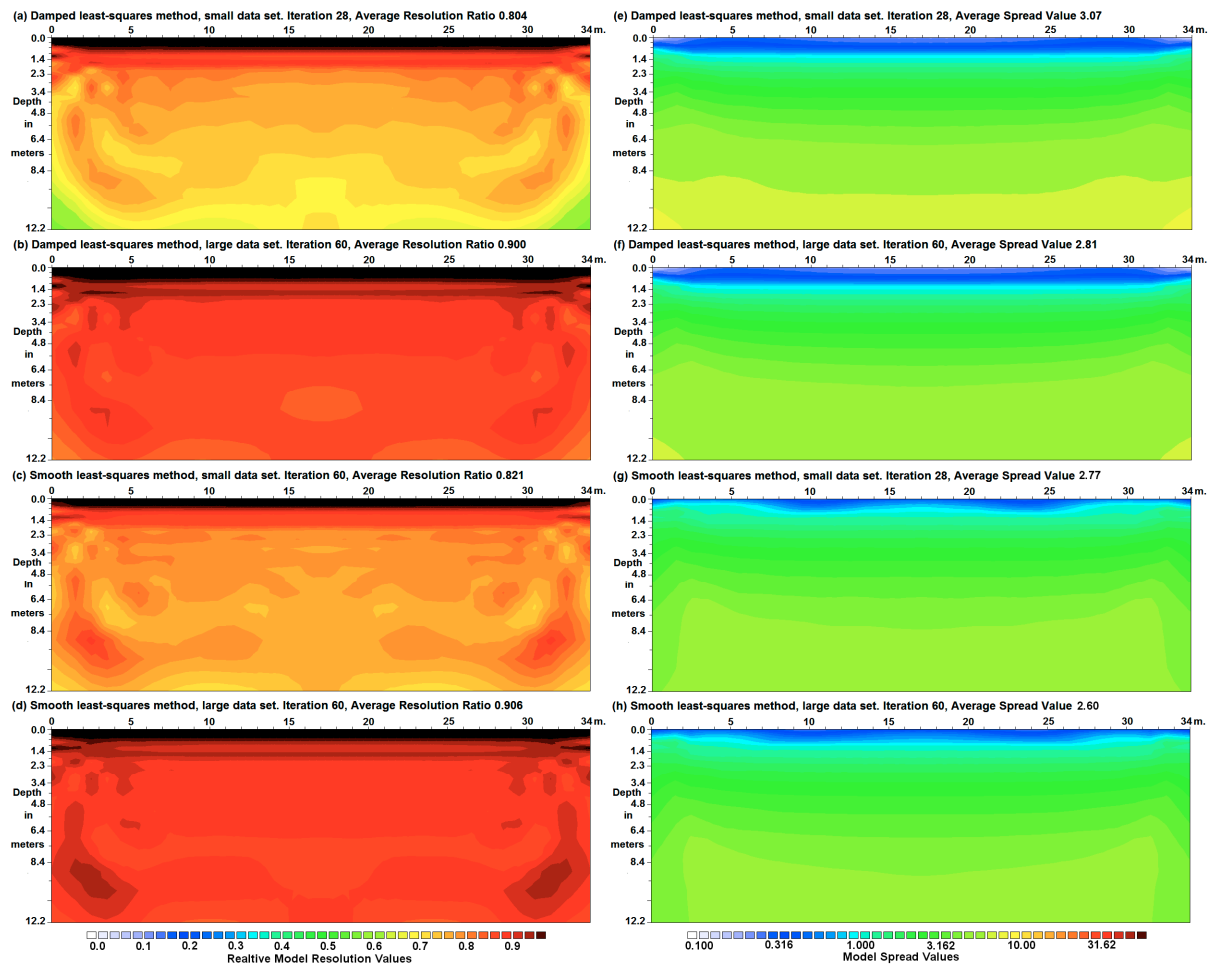


Figure 8. Relative model resolution sections for optimized array set for a survey line with 35 electrodes using the damped least-squares method for (a) small (599 data points) and (b) large (2401 data points) data sets. Similar relative model resolution sections generated using the smoothness-constrained least-squares equation method for (c) small and (d) large data sets. Right column (e-h) shows corresponding spread criterion value sections for the different data sets.

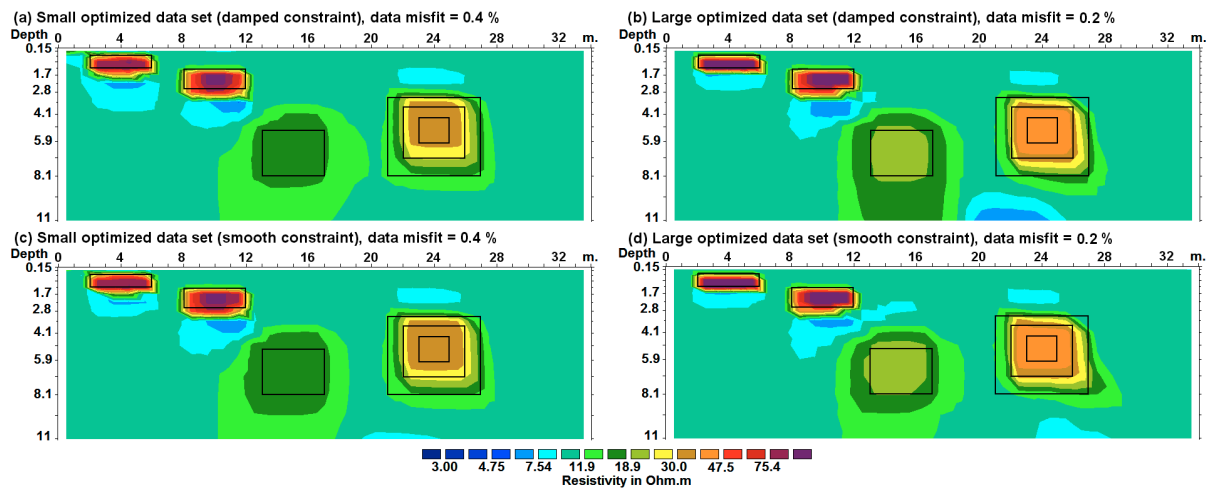


Figure 9. Inversion models for the optimized array data sets generated using a 4.5% step size. Results for (a) small (599 data points) and (b) large (2401 data points) optimized data sets generated using the damped least-squares method. Similar models for (c) small and (d) large optimized data sets generated using the smoothness-constrained least-squares method.