Research Article

Application of OpenMP to Wireline Triaxial Induction Logging in 1D Layered Anisotropic Medium

Zhijuan Zhang, Ning Yuan, and Richard Liu

Department of Electrical and Computer Engineering, University of Houston, 4800 Calhoun Road, Houston, TX 77204, USA

Correspondence should be addressed to Zhijuan Zhang, zzhijuan@yahoo.com

Received 12 July 2011; Accepted 26 September 2011

Academic Editor: Ashutosh Bhardwaj

Copyright © 2012 Zhijuan Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Efficient and accurate forward modeling of logging tool responses is essential for data inversion in the log data interpretation in both real time and postprocessing. With the aggressive advancement of various high-performance computing techniques and computer hardware technology, it is possible to significantly improve the efficiency of the forward modeling. In this paper, we apply OpenMP to parallelize the simulation of triaxial induction logging tools in 1D multilayered anisotropic formation. The parallel process is explained in detail and numerical examples are presented to demonstrate the effect of the parallel programming. Comparison of the original code and the parallel code shows that the latter is much faster without loss of accuracy, which is very promising for future real-time inversion.

1. Introduction

Resistivity logging is an important measurement method for characterizing formations in geophysical exploration. The measured resistivity profile of the formation can provide various formation information such as type of rocks, layer boundaries, and even formation anisotropy. Triaxial induction tool is an emerging tool which has both traditional z antennas and x - y antennas as transmitters and receivers. By measuring all components of the electromagnetic field, the triaxial tool is capable of detecting the anisotropic property of the medium [1–4]. To obtain the true formation information from the log data, data inversion is an effective way to help log analysts in the process. Due to the complexity and nonuniqueness nature of the inversion, about 90% of inversion time is consumed by the repeated use of forward modeling, which implies that decreasing the time of the forward modeling will significantly improve the inversion efficiency. Traditional way to improve the efficiency of the forward modeling includes introducing specific mathematical algorithm to speed up the forward modeling, such as Broyden's method [5] and Newmark's direct integration method [6]. However, many of these methods degrade the precision of the forward modeling inevitably due to either simplification of the modeling or approximations in

algorithms. As a result, the accuracy of inversion results cannot be guaranteed.

Nowadays, the development of high-performance computing techniques provides us various choices to improve the speed of the forward modeling without loss of accuracy. The most popular CPU-based parallel techniques are message passing interface (MPI) and open multiprocessing (OpenMP). MPI was first implemented in 1992 [7] and remains the dominant method used in high-performance computing today [8–10]. MPI is language-independent and can be run on either symmetric multiprocessor (SMP), distributed shared memory (DSM) processor or clusters, and supercomputers. However, MPI is relatively difficult to implement in programming. On the contrary, the latest developed OpenMP is easy to implement and therefore becomes an appropriate choice for quick concept validation within time constraints.

OpenMP is an application programming interface that supports multiplatform-shared memory multiprocessing programming in FORTRAN, C, and C++ [11–14]. Comparing with the traditional application programming interface (API), OpenMP has the following advantages.

(1) *Easy Expansion on Multiple Cores.* In order to fully using multiple cores, the total number of threads should be

dynamic according to the available cores. Although we can assign threads flexibly in API, it is well known that API is very complicated to program. In comparison, it is much easier to assign threads to multiple cores in OpenMP.

(2) *Good Flexibility of Parallelization.* In API, we need entry point functions to create multiple threads. Computer programmers have to cope with every expanding workload. However, OpenMP does not require entry point functions. It is convenient to parallelize some functions or iteration statements in OpenMP.

(3) *Excellent Portability of Software.* Currently, API of each mainstream system is not compatible due to lacking in uniform standard; but OpenMP is normative on any supporting compiler. This is a remarkable advantage.

In this paper, we have adopted OpenMP to parallelize the forward modeling algorithm of wireline triaxial induction logging tools in 1D layered anisotropic formation. The principal of the forward modeling is briefly explained and the parallel implementation of the code is described in details. In the numerical result section, we compare the total running time as well as the simulation results of several examples between the original code and the parallelized code. After parallelization, the computation speed is improved by about 2.5 times on a 4-core computer (2.33 GHz, 4 GB) and the speed can be further improved as the number of the processor cores increased. The comparison proves that OpenMP makes a good compromise between the reliability and the running speed, which is very promising for real-time inversion.

2. Theory

2.1. Parallel Implementation. Figure 1 compares serial and parallel tasks. The master thread is a series of instructions which is executed consecutively. In Figure 1, the master thread includes three serial tasks (Task I, Task II, and Task III). Let us assume the operation time of the three tasks is a_1 , a_2 , a_3 , respectively. Then, CPU would cost time of $(a_1 + a_2 + a_3)$ to complete the serial assignment. Note that the three tasks have one common character: each task can be divided into several independent parts. For instance, task I is comprised by A, B, and C. Thus, the master thread can be divided among specified number of slave "thread" (one core of multicore computer) and these slave threads are capable to run concurrently on multicore computer. Therefore, we can easily get to know that the total running time of the three tasks would be $(a_1/2 + a_2/2 + a_3/4)$ from Figure 1. The operation speed after parallelization is then significantly improved.

2.2. Forward Modeling of Wireline Triaxial Induction Logging. In the oil exploration, logging tools are used to obtain tool response from the formation, and then by applying the inversion algorithm, it is possible to retrieve useful geophysical information (such as dipping angle, boundary, and anisotropy, and resistivity) from the measured data. During the inversion process, the forward modeling is

FIGURE 1: Comparison of entire progress between serial and parallel assignments.

usually called repeatedly to simulate tool responses with each update of the formation parameters. The simulated responses are compared with the measured field data. The inverted parameters are modified at each iteration until the simulated responses are close enough to the real data (e.g., error $< 10^{-9}$). The inverted formation parameters are useful to log analysts in determining the petrophysical features of the formation. Thus, we can see that the forward model has important application to the oil industry. In this section, we briefly present the theory of forward model of wireline triaxial induction logging.

Triaxial induction tool is a new tool for measuring the 3×3 tensor magnetic field responses in a borehole environment. The tool usually comprises three orthogonal transmitters and three orthogonal receivers oriented at x , y , and z direction, as shown in Figure 2(a). If we neglect the size of the coils, the coils are usually treated as magnetic dipoles. The equivalent dipole model is shown in Figure 2(b). Thus, the magnetic source excitation of the triaxial tool can be expressed as $\mathbf{M} = (M_x, M_y, M_z)\delta(\mathbf{r})$.

The tool is moving along the axis in the borehole and for every logging point; a 3×3 magnetic field tensor $\hat{H}$ is measured at the receiver:

$$\hat{H} = \begin{bmatrix} H_x^x & H_x^y & H_x^z \\ H_y^x & H_y^y & H_y^z \\ H_z^x & H_z^y & H_z^z \end{bmatrix}, \quad (1)$$

where H_i^j is the magnetic field at the j -directed receiver from the i -directed transmitter.

The electromagnetic fields in a homogeneous transversely isotropic (TI) formation satisfy the following Maxwell's equations [15]:

$$\nabla \times \hat{E} = i\omega\mu_0\hat{H} + i\omega\mu_0\hat{M}_o, \quad (2)$$

$$\nabla \times \hat{H} = \hat{\sigma}\hat{E} + j\omega\hat{\epsilon}\hat{E}, \quad (3)$$

where $\hat{\sigma}$ is the complex conductivity tensor of TI formation:

$$\hat{\sigma} = \begin{bmatrix} \sigma_h & 0 & 0 \\ 0 & \sigma_h & 0 \\ 0 & 0 & \sigma_v \end{bmatrix}, \quad (4)$$

FIGURE 2: Basic structure of a triaxial induction tool. (a) The original model; (b) the equivalent model.

$\hat{\epsilon}$ is the permittivity tensor given by

$$\hat{\epsilon} = \begin{bmatrix} \epsilon_h & 0 & 0 \\ 0 & \epsilon_h & 0 \\ 0 & 0 & \epsilon_v \end{bmatrix}, \quad (5)$$

and $\hat{M}_0$ is the vector representing the magnetic dipole source:

$$\hat{M}_0 = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}. \quad (6)$$

Following the procedure in [15], solving (2)-(3), we can obtain the magnetic fields of multicomponent induction tools in either homogeneous or multilayered TI medium. The final solutions for each kind of formation can be found in Appendices A and B. The equivalent transmission line theory [16] is used to derive the generalized upward and downward coefficients in multilayered TI medium. The details of the derivation are omitted here and can be referred to [4].

Figure 3 shows the flow chart of the forward model. We use “ N_point ” to present total number of logging points. Logging point essentially stands for measured depth of wireline logging. The letter “ i ” is a control number and records the number of calculated logging point during the serial modeling. If i is larger than N_point , the forward model is terminated and magnetic fields with respect to total logging points are obtained. We can consequently make a plot of tool responses versus bedding depth. In terms of the control number, a big DO loop is embraced in the forward code *TRITI2009* and takes charge of the iterative computation of magnetic fields at each measurement depth.

In principle, the original forward model is serial since it only calculates the tool responses of single logging point

FIGURE 3: Flow chart of the serial forward model.

at one time. It is a waste for a multicore computer because only one core is active as shown in Figure 3, a DO loop controls the whole iterative calculation and can be treated essentially as a master thread. We realize the fact that wireline logging does not collect measurement information until the drilling progress is completed. Thus, logging point is

FIGURE 4: Programming structure of the 1D forward modeling code *TRITI2009*.

FIGURE 5: Programming structure of the parallelized 1D forward model *TRITI2009*.

```

!$OMP PARALLEL DO
DO I = 1, points
...
END DO
$OMP END PARALLEL DO
  
```

ALGORITHM 1

Realizing parallelization

FIGURE 6: Illustration of Parallel DO loop.

FIGURE 7: A three-layer anisotropic formation.

FIGURE 8: Comparison of the total running time of the serialized model and parallelized model.

completely independent with any other points nearby. So the big DO loop includes N_point independent parts and can run through several slave threads. The maximum number of threads is determined by the number of computer cores. Consequently, more than one logging point is computed simultaneously but just consuming a single point's time. Next, we will explain the implementation of the parallelization of the forward modeling using OpenMP.

2.3. Parallelization Progress of Forward Model. As we have previously mentioned, the original forward model simulates the responses of triaxial induction tool in 1D layered TI medium in any deviated borehole and is serial. We realized the forward model in Intel FORTRAN Compiler and the forward code is named as *TRITI2009*. Note that Intel FORTRAN Compiler supports OpenMP interface.

Figure 4 briefly shows the programming structure of the 1-D forward modeling code *TRITI2009*. Since we have presented that the original forward code *TRITI2009* calculated tool responses according to the serial sequence of logging points, a big DO loop is implemented to control the calculation.

```

PROGRAM TRITI2009
IMPLICIT NONE
...
COOM/BLK/A, B, C
!$OMP THREADPRIVATE (A, B, C)
...
!$OMP PARALLEL DO
!$COPYIN (A, B, C)
DO I = 1, points
...
END DO
$OMP END PARALLEL DO
SUBROUTINE
...
END

```

ALGORITHM 2

As we previously discussed, the DO loop in Figure 4 includes independent logging points and can be parallelized. Then, we apply PARALLEL DO to deal with the DO loop from original forward model. The basic directive of PARALLEL DO is semantically equivalent to Algorithm 1.

Then we implement PARALLEL DO to realize parallelization, as shown in Figure 5.

The other important issue is the common block or variables. Since OpenMP is based on shared-memory architecture, as a result, all the threads are allowed to access to the common block or variables. However, there are some common blocks that we do not want to share among different threads. The directive THREADPRIVATE allows us to specify named common blocks and variables as private to other threads but global within their own thread. Once we declare a common block or variable as THREADPRIVATE, each thread in the team maintains a separate copy of that common block or variable. Data written to a THREADPRIVATE common block or variable remain private to that thread and is not available to other threads in the team. Use the clause COPYIN after the directive PARALLEL DO to specify that upon entry into a parallel region, data of a named common block or named variable in the master thread are copied to the common block or variable of each thread. The overall parallel section is shown as Algorithm 2.

Next, we use Figure 6 to illustrate the methodology used in OpenMP to realize parallelization in *TRITI2009*. We assume the original model need to calculate 200 points and the computer has 4 cores. In the serialized model, the computer can only pick up one logging point at one time and calculate the response. On the contrary, in the PARALLEL DO loop, the computer divides all the 200 points into 4 groups. For convenience, we assume the division is even. As a result, 4 threads synchronously run this loop. In other words, 4 logging points can be calculated simultaneously, realizing parallelization.

FIGURE 9: Comparison of the simulated results (5000 points) from the serialized and parallelized codes.

3. Results and Discussions

In this section, we present several examples to discuss the efficiency of the parallelized forward modeling. A 4-core computer (2.33 GHz, 4 GB) is implemented. In all the examples, we use the same computer to do the simulation.

Example 1. In the first example, we consider a three-layer anisotropic formation as shown in Figure 7. The boundaries are at 10 ft and 20 ft, respectively. The first layer and the third layer are isotropic and have a conductivity of 1 S/m. The second layer is anisotropic and has a horizontal conductivity of 0.05 S/m and vertical conductivity of 0.025 S/m. The operating frequency of the triaxial tool is 20 KHz and the spacing between the transmitter and receiver is 40 inches. The dipping angle α is assumed to be 0° (i.e., the tool is perpendicular to the bed layers). 4 threads are used in parallelized code.

Figure 8 shows the total running time of the serialized and parallelized forward modeling code as a function of the logging points. We can see that for a given number of logging points, the parallelized code is much faster than the serialized one. Although the number of logging points increases, the running time of both codes increases, but the comparison is in favor of the parallelized one. For 5000 logging points, the parallelized model is approximately 2.5 times faster than the original serialized model. Due to the limited memory bandwidth and load balancing, the speed improvement does not reach 4 times although we are using a 4-core computer.

Next, we check the accuracy of the parallelized code. In Figure 9, we compare the calculated magnetic field components H_{xx} , H_{yy} , and H_{zz} as a function of the depth for 5000 logging points. Perfect agreement is observed between the results from the serialized code and the parallelized

560 ft	$\sigma_h = 0.5 \text{ S/m}, \sigma_v = 0.5 \text{ S/m}$
580 ft	$\sigma_h = 0.325 \text{ S/m}, \sigma_v = 0.325 \text{ S/m}$
595 ft	$\sigma_h = 0.55625 \text{ S/m}, \sigma_v = 0.55625 \text{ S/m}$
614 ft	$\sigma_h = 0.2125 \text{ S/m}, \sigma_v = 0.2125 \text{ S/m}$
630 ft	$\sigma_h = 0.1125 \text{ S/m}, \sigma_v = 0.1125 \text{ S/m}$
635 ft	$\sigma_h = 0.2 \text{ S/m}, \sigma_v = 0.2 \text{ S/m}$
680 ft	$\sigma_h = 0.3125 \text{ S/m}, \sigma_v = 0.3425 \text{ S/m}$
	$\sigma_h = 0.3125 \text{ S/m}, \sigma_v = 0.125 \text{ S/m}$

FIGURE 10: The eight-layer Devine test site formation.

FIGURE 11: Comparison of the total running time between the serialized model and the parallelized model.

FIGURE 12: Comparison of the simulated results (800 points) from the serialized and parallelized codes.

code, implying that the parallelization does not change the accuracy of the code.

In Figure 9, we ignore the entire cross components H_{xy} , H_{xz} , H_{yx} , H_{yz} , H_{zx} , and H_{zy} due to the orthogonal property of antenna in the vertical borehole. H_{xx} and H_{yy} are coinciding with each other since the borehole is vertical and formation is isotropic in the horizontal plane. We observe spikes on H_{xx} and H_{yy} , known as “horn effect,” which is assumed as a significant sign indicating the layer boundary. According to the displacement of triaxial antennas, the coaxial antennas should have stronger responses than the other coplanar antennas, which are also proven by Figure 9. The common point among H_{xx} , H_{yy} , and H_{zz} is that in a conductive layer, those magnetic fields H_{xx} , H_{yy} , and H_{zz} show stronger sensitivity with respect to larger responses.

Example 2. In this example, the geometric formation is constructed according to the Devine test site of BP America

[17], as shown in Figure 10. The operating frequency is 512 Hz and the distance between the transmitter and the receiver is 40 inches. We assume the well is vertical. We also use 4 threads in parallelization

Figure 11 compares the total running time of the serialized code with the parallelized code. With the increasing of the logging points, the difference of the time between these two models is strengthened. Figure 12 presents the simulation results for 800 points from the serial and parallel models. From Figures 11 and 12, it is easy to observe that the forward model is accelerated by about 3 times after parallelization. In Figure 12, we can easily tell formation boundaries according to spikes on coplanar components. Note that H_{zz} does not differentiate the sixth layer since no evident change is observed from 620 ft to 640 ft on H_{zz} . Fortunately, H_{xx} and H_{yy} present significant spikes and infer the existence of different layers. Therefore, the multicomponent configuration of triaxial induction tool provides much more

492'	$\sigma_h = 0.2 \text{ S/m}, \sigma_v = 0.1 \text{ S/m}$
509'	$\sigma_h = 0.02 \text{ S/m}, \sigma_v = 0.01 \text{ S/m}$
517'	$\sigma_h = 0.5 \text{ S/m}, \sigma_v = 0.25 \text{ S/m}$
521'	$\sigma_h = 0.06666 \text{ S/m}, \sigma_v = 0.03333 \text{ S/m}$
524'	$\sigma_h = 0.2222 \text{ S/m}, \sigma_v = 0.1111 \text{ S/m}$
531'	$\sigma_h = 0.01 \text{ S/m}, \sigma_v = 0.005 \text{ S/m}$
535'	$\sigma_h = 0.2858 \text{ S/m}, \sigma_v = 0.1414 \text{ S/m}$
541'	$\sigma_h = 0.00222 \text{ S/m}, \sigma_v = 0.00111 \text{ S/m}$
544'	$\sigma_h = 0.0332 \text{ S/m}, \sigma_v = 0.01616 \text{ S/m}$
549'	$\sigma_h = 0.00166 \text{ S/m}, \sigma_v = 0.00088 \text{ S/m}$
556'	$\sigma_h = 0.5 \text{ S/m}, \sigma_v = 0.025 \text{ S/m}$
574'	$\sigma_h = 0.00134 \text{ S/m}, \sigma_v = 0.001616 \text{ S/m}$
582'	$\sigma_h = 0.005 \text{ S/m}, \sigma_v = 0.0025 \text{ S/m}$
589'	$\sigma_h = 0.13334 \text{ S/m}, \sigma_v = 0.06666 \text{ S/m}$
	$\sigma_h = 0.002 \text{ S/m}, \sigma_v = 0.001 \text{ S/m}$

FIGURE 13: A fifteen-layer Oklahoma model.

FIGURE 14: Comparison of the total running time of the serialized model and the parallelized model for the 15-layer Oklahoma model.

comprehensive information than traditional induction tool which is only consisted by coaxial antennas.

Example 3. Finally, we apply a benchmark fifteen-layer Oklahoma model as shown in Figure 13. The distance between

the transmitter and the receiver is 40 inches and the working frequency is 20 KHz. The dipping angle is assumed to be 60° . The boundaries and conductivities of each layer are given in the figure.

First, 4 threads are applied. We compare the computation time of the serialized code and parallelized code in Figure 14.

Figure 15 compares the coaxial and coplanar components H_{xx} , H_{yy} , and H_{zz} . It is easy to differentiate H_{xx} and H_{yy} in Figure 15. So the difference between H_{xx} and H_{yy} is a notable flag of deviated well. Since the well is inclined at 60° in the X-Z plane, as a result, we should have nonzero cross components H_{xz} and H_{zx} , as shown in Figure 16. In the future inversion process, the value of cross components H_{xz} and H_{zx} are widely applied to derive dipping angle.

From Figures 15 and 16, we can see that parallelization does not affect the accuracy of forward model since the tool responses from the parallel model are coinciding with serial model. We can see that the slope of the blue line (the serialized code) is about 3.5 times that of the red line, which means the speed of parallelized model is 3.5 times faster than the original model.

According to the results from the three examples, we can see that the contrast of the time between the parallelized code and serialized code is enhanced when the total number of layers is increased. As we know, in field exploration, the earth formation is always very complex and hard to predict. Hence, fast forward modeling provides geologists a good tool for simulation and possibility for real-time inversion.

FIGURE 15: Comparison of H_{xx} , H_{yy} , and H_{zz} (1000 points) from the serialized and parallelized codes.

FIGURE 16: Comparison of cross components (1000 points) from the serialized and parallelized codes.

Next, we change the number of threads to investigate the performance of the parallelized code. In Figure 17, we compare the running time when different number of threads are implemented. It can be seen that for a given number of logging points, the more threads are used, the less time is consumed by the code. Therefore, we can expect that the speed of the forward modeling to be further improved as multicore computers are used.

FIGURE 17: Comparison of the time cost for different number of threads.

FIGURE 18: Comparison of the efficiency for different number of threads.

Finally, we check the efficiency among different number of threads. In Figure 18, we can see that the speed has been improved approximately by 1.8 times for 2 threads, 2.6 times for 3 threads, and 3.4 times for 4 threads, respectively. However, the improved speed is not linear proportional to the number of threads. The reason is complex. As we discussed in the first example, the limited memory bandwidth and load balancing influence the parallelization. On the other hand, we only parallelize the Do Loop from the main body of the forward model. The read and write sections are still serial; and the common variables have

FIGURE 19: A two-layer anisotropic formation.

become private for each thread. Then, the cost on memory is also increased. Hence, the time saving of parallel code cannot be linear to the number of threads.

Example 4. In this example, we apply one two-layer formation to illustrate the reason why triaxial induction logging tool is capable to detect anisotropic resistivity. Figure 19 presents the two-layer formation. Note that the two layers share the same horizontal conductivity but different vertical conductivity. The well is assumed to be vertical.

A 4-core computer is implemented to parallelize the forward model. Figure 20 compares the running time from the serialized and parallelized codes. As shown in Figure 20, the running speed is significantly improved.

Then, we present the simulation responses in Figure 21. Perfect match within the corresponding components can be found. It is very interesting to know the coaxial component H_{zz} is constant. Therefore, we can infer that H_{zz} is “blind” to the different layers. As we know, in the vertical well, Z directional electrical loop antenna can only induce electrical eddy currents in the horizontal plane. Moreover, the responses on the corresponding receiver are only sensitive to the horizontal conductivity. Thus, in this example, H_{zz} is not various because of the same horizontal conductivity in both of the two layers. Until now, we have explained the reason why the traditional induction tool with coaxial antenna cannot detect anisotropic medium.

In Figure 21, we also notice that the coplanar components H_{xx} , H_{yy} vary between the different two layers. In the vertical well, the coplanar electrical loop antenna generates eddy currents along both horizontal and vertical directions. As a result, the received components result from both the horizontal and the vertical conductivities. From this example, we can see that triaxial induction logging tool is sensitive to anisotropic medium.

4. Conclusion

Parallel computation plays an important role in the high-performance computing and OpenMP is one of the most widely used parallel interfaces. In this paper, we extended the application of OpenMP to well logging simulation in oil industry, parallelizing the 1-D forward modeling code for simulation of wireline triaxial induction tools in multilayered anisotropic formation. We explained the parallel process and validate the parallelized code. The forward modeling is successfully parallelized on a 4-core computer and the

FIGURE 20: Comparison of the time cost for different number of threads.

FIGURE 21: Comparison of H_{xx} , H_{yy} , and H_{zz} (1000 points) from the serialized and parallelized codes.

total running time is significantly improved by at least 2.5 times. We also briefly discuss the property of triaxial induction logging tool in different formation. Furthermore, the accuracy of the parallelized code keeps the same as the original code. The speed of the forward modeling is expected to further improve as multicore computers are used.

Appendices

A. Magnetic Fields in Vertical Well

The magnetic fields in homogenous formation with vertical well are given by Mora and Gianzero [15] as follows:

$$H_x^x = \frac{e^{ik_v s}}{4\pi} \left[\frac{i\omega\mu_0\sigma_h}{\lambda s} + \frac{ik_h s - k_h k_v x^2}{s\rho^2} - \frac{2ik_h x^2}{\rho^4} \right] - \frac{e^{ik_h r}}{4\pi} \left[\frac{ik_h r - k_h^2 x^2}{r\rho^2} - \frac{2ik_h x^2}{\rho^4} - \frac{ik_h}{r^2} + \frac{(k_h^2 x^2 + 1)}{r^3} + \frac{3ik_h x^2}{r^4} - \frac{3x^2}{r^5} \right], \quad (\text{A.1})$$

$$H_y^x = H_x^y = xy \frac{e^{ik_v s}}{4\pi\rho^2} \left[-\frac{k_h k_v}{s} - \frac{2ik_h}{\rho^2} \right] - \frac{e^{ik_h r} xy}{4\pi} \left[-\frac{k_h^2}{r\rho^2} - \frac{2ik_h}{\rho^4} + \frac{k_h^2}{r^3} + \frac{3ik_h}{r^4} - \frac{3}{r^5} \right], \quad (\text{A.2})$$

$$H_z^x = H_x^z = -xz \frac{e^{ik_h r}}{4\pi r^3} \left[k_h^2 + \frac{3ik_h}{r} - \frac{3}{r^2} \right], \quad (\text{A.3})$$

$$H_y^y = \frac{e^{ik_v s}}{4\pi} \left[\frac{i\omega\mu_0\sigma_h}{\lambda s} + \frac{ik_h s - k_h k_v y^2}{s\rho^2} - \frac{2ik_h y^2}{\rho^4} \right] - \frac{e^{ik_h r}}{4\pi} \left[\frac{ik_h r - k_h^2 y^2}{r\rho^2} - \frac{2ik_h y^2}{\rho^4} - \frac{ik_h}{r^2} + \frac{(k_h^2 y^2 + 1)}{r^3} + \frac{3ik_h y^2}{r^4} - \frac{3y^2}{r^5} \right] \quad (\text{A.4})$$

$$H_z^y = H_y^z = -yz \frac{e^{ik_h r}}{4\pi r^3} \left[k_h^2 + \frac{3ik_h}{r} - \frac{3}{r^2} \right], \quad (\text{A.5})$$

$$H_z^z = \frac{e^{ik_h r}}{4\pi r} \left[i\omega\mu_0\sigma_h + \frac{ik_h}{r} - \frac{(k_h^2 z^2 + 1)}{r^2} - \frac{3ik_h z^2}{r^3} + \frac{3z^2}{r^4} \right], \quad (\text{A.6})$$

where $\hat{\sigma}_h = \sigma_h - j\omega\epsilon_0\epsilon_h$, $\hat{\sigma}_v = \sigma_v - j\omega\epsilon_0\epsilon_v$, $\lambda^2 = \hat{\sigma}_h/\hat{\sigma}_v$, $k_h^2 = j\omega\mu\hat{\sigma}_h$, $k_v^2 = j\omega\mu\hat{\sigma}_v$, $r = \sqrt{x^2 + y^2 + z^2}$, $\rho = \sqrt{x^2 + y^2}$, and $s = \sqrt{x^2 + y^2 + \lambda^2 z^2}$.

The magnetic fields in layered medium with vertical well is given by

$$H_x^x = \frac{M_x}{4\pi} \int_0^\infty \left(\frac{\beta_i \sin^2 \phi}{\lambda_i \xi_{vi}} k_h^2 e^{-\xi_{vi} \lambda_i |z-z_0|} - \beta_i \cos^2 \phi \xi_{hi} e^{-\xi_{hi} |z-z_0|} + \frac{P_i}{\lambda_i} \sin^2 \phi k_{hi}^2 e^{-\xi_{vi} \lambda_i z} + \frac{Q_i}{\lambda_i} \sin^2 \phi k_{hi}^2 e^{\xi_{vi} \lambda_i z} - S_i \cos^2 \phi \xi_{hi} e^{-\xi_{hi} z} + T_i \cos^2 \phi \xi_{hi} e^{\xi_{hi} z} \right) \times \alpha J_0(\alpha \rho) d\alpha + \frac{M_x}{4\pi \rho} \int_0^\infty \left(\lambda_i \frac{\beta_i}{\xi_{vi}} k_v^2 e^{-\xi_{vi} \lambda_i |z-z_0|} + \beta_i \xi_{hi} e^{-\xi_{hi} |z-z_0|} + P_i \lambda_i k_{vi}^2 e^{-\xi_{vi} \lambda_i z} + Q_i \lambda_i k_{vi}^2 e^{\xi_{vi} \lambda_i z} + S_i \xi_{hi} e^{-\xi_{hi} z} - T_i \xi_{hi} e^{\xi_{hi} z} \right) \times \cos 2\phi J_1(\alpha \rho) d\alpha, \quad (\text{A.7})$$

$$H_x^y = \frac{M_x}{4\pi \rho} \int_0^\infty \left(\frac{\beta_i}{\lambda_i \xi_{vi}} k_h^2 e^{-\xi_{vi} \lambda_i |z-z_0|} - \beta_i \xi_{hi} e^{-\xi_{hi} |z-z_0|} + \frac{P_i}{\lambda_i} k_{hi}^2 e^{-\xi_{vi} \lambda_i z} + \frac{Q_i}{\lambda_i} k_{hi}^2 e^{\xi_{vi} \lambda_i z} - S_i \xi_{hi} e^{-\xi_{hi} z} + T_i \xi_{hi} e^{\xi_{hi} z} \right) \times \alpha J_0(\alpha \rho) d\alpha + \frac{M_x}{4\pi \rho} \int_0^\infty \left(\lambda_i \frac{\beta_i}{\xi_{vi}} k_v^2 e^{-\xi_{vi} \lambda_i |z-z_0|} + \beta_i \xi_{hi} e^{-\xi_{hi} |z-z_0|} + P_i \lambda_i k_{vi}^2 e^{-\xi_{vi} \lambda_i z} + Q_i \lambda_i k_{vi}^2 e^{\xi_{vi} \lambda_i z} + S_i \xi_{hi} e^{-\xi_{hi} z} - T_i \xi_{hi} e^{\xi_{hi} z} \right) \times \sin 2\phi J_1(\alpha \rho) d\alpha, \quad (\text{A.8})$$

$$H_x^z = \frac{M_x}{4\pi} \cos \phi \int_0^\infty \left(\beta_i \frac{|z-z_0|}{z-z_0} e^{-\xi_{hi} |z-z_0|} + S_i e^{-\xi_{hi} z} + T_i e^{\xi_{hi} z} \right) \alpha^2 J_1(\alpha \rho) d\alpha, \quad (\text{A.9})$$

$$\begin{aligned}
H_y^x = & \frac{M_y}{8\pi} \int_0^\infty \left(\frac{\beta_i}{\lambda_i \xi_{vi}} k_v^2 e^{-\xi_{vi} \lambda_i |z-z_0|} - \beta_i \xi_{hi} e^{-\xi_{hi} |z-z_0|} \right. \\
& + \frac{P_i}{\lambda_i} k_{hi}^2 e^{-\xi_{vi} \lambda_i z} + \frac{Q_i}{\lambda_i} k_{hi}^2 e^{\xi_{vi} \lambda_i z} \\
& \left. - S_i \xi_{hi} e^{-\xi_{hi} z} + T_i \xi_{hi} e^{\xi_{hi} z} \right) \\
& \times \sin 2\phi \alpha J_0(\alpha \rho) d\alpha \\
& + \frac{M_y}{4\pi \rho} \times \int_0^\infty \left(\lambda_i \frac{\beta_i}{\xi_{vi}} k_v^2 e^{-\xi_{vi} \lambda_i |z-z_0|} + \beta_i \xi_{hi} e^{-\xi_{hi} |z-z_0|} \right. \\
& + P_i \lambda_i k_{vi}^2 e^{-\xi_{vi} \lambda_i z} + Q_i \lambda_i k_{vi}^2 e^{\xi_{vi} \lambda_i z} \\
& \left. + S_i \xi_{hi} e^{-\xi_{hi} z} - T_i \xi_{hi} e^{\xi_{hi} z} \right) \\
& \times \sin 2\phi J_1(\alpha \rho) d\alpha,
\end{aligned} \tag{A.10}$$

$$\begin{aligned}
H_y^y = & \frac{M_y}{4\pi} \int_0^\infty \left(\frac{\beta_i \cos^2 \phi}{\lambda_i \xi_{vi}} k_v^2 e^{-\xi_{vi} \lambda_i |z-z_0|} - \beta_i \sin^2 \phi \xi_{hi} e^{-\xi_{hi} |z-z_0|} \right. \\
& + \frac{P_i}{\lambda_i} \cos^2 \phi k_{hi}^2 e^{-\xi_{vi} \lambda_i z} + \frac{Q_i}{\lambda_i} \cos^2 \phi k_{hi}^2 e^{\xi_{vi} \lambda_i z} \\
& \left. - S_i \sin^2 \phi \xi_{hi} e^{-\xi_{hi} z} + T_i \sin^2 \phi \xi_{hi} e^{\xi_{hi} z} \right) \\
& \times \alpha J_0(\alpha \rho) d\alpha \\
& + \frac{M_y}{4\pi \rho} \int_0^\infty \left(\lambda_i \frac{\beta_i}{\xi_{vi}} k_v^2 e^{-\xi_{vi} \lambda_i |z-z_0|} + \beta_i \xi_{hi} e^{-\xi_{hi} |z-z_0|} \right. \\
& + P_i \lambda_i k_{vi}^2 e^{-\xi_{vi} \lambda_i z} + Q_i \lambda_i k_{vi}^2 e^{\xi_{vi} \lambda_i z} \\
& \left. + S_i \xi_{hi} e^{-\xi_{hi} z} - T_i \xi_{hi} e^{\xi_{hi} z} \right) \\
& \times \cos 2\phi J_1(\alpha \rho) d\alpha,
\end{aligned} \tag{A.11}$$

$$\begin{aligned}
H_y^z = & \frac{M_y}{4\pi} \sin \phi \int_0^\infty \left(\beta_i \frac{|z-z_0|}{z-z_0} e^{-\xi_{hi} |z-z_0|} \right. \\
& \left. + S_i e^{-\xi_{hi} z} + T_i e^{\xi_{hi} z} \right) \alpha^2 J_1(\alpha \rho) d\alpha,
\end{aligned} \tag{A.12}$$

$$\begin{aligned}
H_z^x = & \frac{M_z}{4\pi} \int_0^\infty \xi_{hi} \left(\frac{\beta_i}{\xi_{hi}} \frac{|z-z_0|}{z-z_0} e^{-\xi_{hi} |z-z_0|} \right. \\
& \left. + F_i e^{-\xi_{hi} z} - G_i e^{\xi_{hi} z} \right) \alpha^2 \cos \phi J_1(\alpha \rho) d\alpha,
\end{aligned} \tag{A.13}$$

$$\begin{aligned}
H_z^y = & \frac{M_z}{4\pi} \int_0^\infty \xi_{hi} \left(\frac{\beta_i}{\xi_{hi}} \frac{|z-z_0|}{z-z_0} e^{-\xi_{hi} |z-z_0|} \right. \\
& \left. + F_i e^{-\xi_{hi} z} - G_i e^{\xi_{hi} z} \right) \alpha^2 \sin \phi J_1(\alpha \rho) d\alpha,
\end{aligned} \tag{A.14}$$

$$\begin{aligned}
H_z^z = & \frac{M_z}{4\pi} \int_0^\infty \left(\frac{\beta_i}{\xi_{hi}} e^{-\xi_{hi} |z-z_0|} + F_i e^{-\xi_{hi} z} + G_i e^{\xi_{hi} z} \right) \alpha^3 J_0(\alpha \rho) d\alpha,
\end{aligned} \tag{A.15}$$

where $\xi_{vi} = (\alpha^2 - k_{vi}^2)^{1/2}$, $k_{vi} = (i\omega\mu\sigma_{vi})^{1/2}$, $\xi_{vi} = (\alpha^2 - k_{vi}^2)^{1/2}$, $k_{vi} = (i\omega\mu\sigma_{vi})^{1/2}$, $\lambda_i = k_{hi}/k_{vi}$, and

$$\beta_i = \begin{cases} 0, & \text{if } M_{x,y,\text{orz}} \text{ is not in the } i\text{th layer,} \\ 1, & \text{if } M_{x,y,\text{orz}} \text{ is in the } i\text{th layer.} \end{cases} \tag{A.16}$$

In (A.7)–(A.12), P_i and S_i are the magnitude of the reflection magnetic fields and Q_i and T_i are the magnitude of the refraction magnetic fields if sources are in X or Y direction. In (A.13)–(A.15), F_i and G_i are the magnitude of the reflection magnetic fields and the refraction magnetic fields, respectively, when source is along Z direction.

B. Magnetic Fields in Deviated Well

From Appendix A, we have got the solutions of magnetic fields in vertical well. We apply matrix H to represent the magnetic fields from vertical well. The magnetic fields in any deviated well can be derived according to rotation matrix, as

$$\hat{H} = R^{-1} \cdot H \cdot R \cdot \hat{M}_o, \tag{B.1}$$

where R is the rotation matrix,

$$R = \begin{pmatrix} \cos \alpha \cos \beta \cos \gamma - \sin \beta \sin \gamma & -\cos \alpha \cos \beta \sin \gamma - \sin \beta \cos \gamma & \sin \alpha \cos \beta \\ \cos \alpha \sin \beta \cos \gamma + \cos \beta \sin \gamma & -\cos \alpha \sin \beta \sin \gamma + \cos \beta \cos \gamma & \sin \alpha \sin \beta \\ -\sin \alpha \cos \gamma & \sin \gamma \sin \alpha & \cos \alpha \end{pmatrix}. \tag{B.2}$$

Since R is orthogonal, it is easy to know that

$$R^{-1} = \begin{pmatrix} \cos \alpha \cos \beta \cos \gamma - \sin \beta \sin \gamma & \cos \alpha \sin \beta \cos \gamma + \cos \beta \sin \gamma & -\sin \alpha \cos \gamma \\ -\cos \alpha \cos \beta \sin \gamma - \sin \beta \cos \gamma & -\cos \alpha \sin \beta \sin \gamma + \cos \beta \cos \gamma & \sin \gamma \sin \alpha \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta & \cos \alpha \end{pmatrix}, \quad (\text{B.3})$$

and $\widehat{M}_o$ is the matrix of transmitter source.

References

- [1] B. Kriegshäuser, O. Fanini, S. Forgang et al., "A new multi-component induction logging tool to resolve anisotropic formations: paper D," in *Proceedings of the Transactions SPWLA 41st Annual Logging Symposium*, Dallas, Tex, USA, June 2000.
- [2] M. Zhdanov, D. Kennedy, and E. Peksen, "Foundations of tensor induction well-logging," *Petrophysics*, vol. 42, no. 6, pp. 588–610, 2001.
- [3] B. Anderson, T. Barber, and T. M. Habashy, "The interpretation and inversion of fully triaxial induction data: a sensitivity study," in *Proceedings of the 36th Annual Logging Symposium Transactions, Society of Professional Well Log Analyst*, 2002.
- [4] L. L. Zhong, J. Li, A. Bhardwaj, S. C. Liang, and R. C. Liu, "Computation of triaxial induction logging tools in layered anisotropic dipping formations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 4, pp. 1148–1163, 2008.
- [5] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Mathematics of Computation—American Mathematical Society*, vol. 19, pp. 577–593, 1965.
- [6] O. Friberg, "Jacobian matrices using the Newmark direct integration scheme," *Computers and Structures*, vol. 25, no. 2, pp. 307–310, 1987.
- [7] W. Gropp and E. Lusk, "A test implementation of the MPI data message-passing standard," Tech. Rep. ANL-92/47, 1992.
- [8] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, "A high-performance, portable implementation of the MPI message passing interface standard," *Parallel Computing*, vol. 22, no. 6, pp. 789–828, 1996.
- [9] G. Krawezik and F. Cappello, "Performance comparison of MPI and OpenMP on shared memory multiprocessors," *Concurrency Computation Practice and Experience*, vol. 18, no. 1, pp. 29–61, 2006.
- [10] I. Dorta, C. Leon, and C. Rodriguez, "A comparison between MPI and OpenMP branch-and-bound skeletons," in *Proceedings of the International Parallel and Distributed Processing Symposium*, 2003.
- [11] R. Chandra, R. Menon, L. Dagum, D. Kohr, D. Maydan, and J. McDonald, *Parallel Programming in OpenMP*, Morgan Kaufmann, San Francisco, Calif, USA, 2000.
- [12] B. Chapman, G. Jost, R. van der Pas, and D. J. Kuck, *Using OpenMP: Portable Shared Memory Parallel Programming*, MIT Press, Massachusetts, 2007.
- [13] M. Hermanns, "Parallel Programming in Fortran 95 using OpenMP," http://www.openmp.org/presentations/miguel/F95_OpenMPv1-v2.pdf, 2002.
- [14] C. Bischof, M. Bucker, P. Gibbon, G. Joubert, and T. Lippert, *Parallelizing a Real-Time Steering Simulation for Computer Games with OpenMP*, IOS Press, Amsterdam, The Netherlands, 2008.
- [15] J. H. Mora and S. Gianzero, "Effects of formation anisotropy on resistivity- logging measurements," *Geophysics*, vol. 44, no. 7, pp. 1266–1286, 1979.
- [16] M. Huang and L. C. Shen, "Computation of induction logs in multiple-layer dipping formation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 27, no. 3, pp. 259–267, 1989.
- [17] M. J. Wilt, D. L. Alumbaugh, H. F. Morrison, A. Becker, K. H. Lee, and M. Deszcz-Pan, "Crosswell electromagnetic tomography: system design considerations and field results," *Geophysics*, vol. 60, no. 3, pp. 871–885, 1995.