Computer Physics Communications 131 (2000) 26–40

Computer Physics Communications

www.elsevier.nl/locate/cpc

Simulational modeling of time-domain magnetic induction using parallel computational schemes: (I) in the Cartesian coordinates

Wenchien Chou [a,*], Ryoji Matsumoto [b], Fumiko Tajima [c]

[a] *Department of Research for Computational Earth Science, Research Organization for Information Science and Technology (RIST), 1-18-16 Hamamatsu-cho, Minato-ku, Tokyo 105-0013, Japan*

[b] *Department of Physics, Faculty of Science, Chiba University, 1-33 Yayoi-Cho, Inage-ku, Chiba 263-8522, Japan*

[c] *Berkeley Seismological Laboratory, University of California at Berkeley, 215 McCone Hall, Berkeley, CA 94720-4760, USA*

Received 21 May 1999; received in revised form 23 November 1999

Abstract

A new parallel code has been developed to carry out simulational experiments of magnetotelluric investigation in Cartesian coordinates by solving the electromagnetic induction equation in the time domain. In this paper we show that the code can achieve good performances of parallel processing in various super-computers. We present several preliminary tests that demonstrate the performance of this code. Our tests show promising results of the computations in several aspects including the effect of skin depth, and phase shift and magnitude of the induced field relative to the external electric field. © 2000 Elsevier Science B.V. All rights reserved.

PACS: 91.25.Qi; 91.35. x; 91.60.Pn; 02.70.Bf; 02.60.Jn; 41.20.Bt

Keywords: Geomagnetic induction; Computational methods; Parallel computation

1. Introduction

Over the past decade seismology has made significant progress in using tomographic techniques for 3D imaging of the elastic properties of the crust and mantle and the topography of the core-mantle boundary. While seismic waves provide information on elastic properties, measurements of electrical conductivity can be effectively used to constrain the spatial variation of temperature as well as to characterize chemical composition and degree of partial melt within the Earth. The naturally occurring powerful, low-frequency electromagnetic (EM) fields whose primary sources are located in the magnetosphere and ionosphere have long been considered to be promising for studies of the Earth's interior. The magnetotelluric (MT) method was proposed more than 40 years ago [1]. The basic idea is simple, i.e. the Earth can be regarded as a conducting sphere and should respond to the external EM field, that will lead us to obtain the distribution of interior conductivity through the signature of the induced EM field. Thus MT method is thought to be complementary to seismic wave analysis in investigating the structure of the Earth's interior.

[*] Present address: Institute of Astronomy and Astrophysics, Academia Sinica, P.O. Box 1-87, Nankang, Taipei 11529, Taiwan. E-mail: chou@asiaa.sinica.edu.tw.

PII: S0010-4655(00)00070-9

While the theoretical modeling of MT method has been developed long time ago, this method has not been used to study the Earth's deep interior to its full advantage. Several limitations impeded efforts to build a detailed 3D images of the electrical conductivity distribution of the Earth interior. First, the geographical and temporal distribution of observational sites are so sparse that available data are limited. Second, it is difficult to distinguish or isolate the external EM sources, such as the cosmic EM wave interacting with the magnetosphere or the ionosphere. We believe that large-scale computational experiments are effective to assess the feasibility of the MT method to image the mantle structure and shed light on what information we can obtain from the limited observational data.

There are several papers that attempted to develop this kind of computer simulations. Parker [2] and Parker and Whaler [3] developed a mathematical theory and numerical algorithm for inversion of electromagnetic induction data. Wannamaker [4] adopted an integral equation technique. Mackie and Madden [5] and Mackie et al. [6] investigated this problem through differential methods. Zhang and Schultz [7], Everett and Schultz [8], and Martinec [9] carried out computer simulations using a finite element numerical method in the spectral domain, and Smith [10,11] and Wang and Hohmann [12] used a finite difference method. Recently available computers become more and more powerful in the CPU speed, size of memory, and number of processors, so that we can carry out computer simulations with a better resolution and higher speed. To our best knowledge, however, the simulations presented in published papers so far are carried out in the frequency domain. More importantly none of such efforts described above has successfully parallelized the computation, and therefore the power of the modern computer technology has not yet been fully utilized. This paper presents a newly developed parallel code, which is written in Fortran MPI, to solve the EM induction equation and hence carry out MT investigations through computer experiments in parallel computers.

2. Formalism

2.1. Basic equations

The Maxwell equations without displacement currents and free charges are

$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}, \tag{1}$$

$$\nabla \times \mathbf{B} = \frac{4\pi\sigma}{c}\mathbf{E}, \tag{2}$$

$$\nabla \cdot \mathbf{E} = 0, \tag{3}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{4}$$

where σ is the electrical conductivity and c is the speed of light. They can be simplified by adopting a vector potential $\mathbf{A}$ and an electrical potential ϕ, with a Coulomb gauge

$$\nabla \cdot \mathbf{A} = 0, \tag{5}$$

and the fields are expressed as

$$\mathbf{B} = \nabla \times \mathbf{A}, \tag{6}$$

$$\mathbf{E} = \frac{-1}{c}\frac{\partial \mathbf{A}}{\partial t} - \nabla\phi. \tag{7}$$

The Farady equation (1) and the divergent-free equation (4) are automatically satisfied by Eqs. (6) and (7). If we further assume that there is no static electric field (e.g., see Ref. [8]), then Ampere's law Eq. (2) becomes

$$\frac{4\pi\sigma}{c^2}\frac{\partial \mathbf{A}}{\partial t} - \nabla^2\mathbf{A} = 0. \tag{8}$$

This is the diffusion equation. In the spectral domain, i.e. the time dependence is written as $\mathrm{e}^{i\omega t}$, one can write the solution in 1D easily as:

$$A = C\mathrm{e}^{z/\lambda + i\pi/4} + D\mathrm{e}^{-z/\lambda - i\pi/4}, \tag{9}$$

where λ is the skin depth defined by

$$\lambda = \frac{c}{\sqrt{2\pi\sigma\omega}}, \tag{10}$$

and C, D are constants determined by the boundary conditions. The phase $i\pi/4$ appears because a factor $\sqrt{i}$ is produced during the manipulation. This complex number will produce a phase difference between the time derivative (electric field) and the space derivative (magnetic field) of the vector potential. Later we will compare tests of our numerical computations with this analytical solution.

2.2. Normalization

In the code we actually solve

$$\sigma \frac{\partial \mathbf{A}}{\partial t} - \nabla^2 \mathbf{A} = 0. \tag{11}$$

The unit of σ can be expressed as

$$[\sigma_0] = \frac{c^2[T]}{4\pi[L]^2}\,\mathrm{s}^{-1} = \frac{8.0\times 10^5[T]}{[L]^2}\,\mathrm{S/m}. \tag{12}$$

We adopt the time unit $[T]$ as 1 second, and length unit $[L]$ as 10 km. In this normalization, the normalization factor is $[\sigma_0] = 0.008$ S/m. The conductivity in the Earth interior, which ranges from $\sigma = 10^{-3}$ S/m on the surface to $\sigma = 1$ S/m in the upper mantle (about 700 km beneath the surface, see, e.g., Ref. [15]), is equal to a value that ranges from $\sigma = 0.125$ to $\sigma = 125$ in our code.

3. Coding

3.1. Finite difference equation

Adopting Crank–Nicholson scheme and Cartesian coordinates, the finite difference form of the diffusive differential equation in the x-direction is

$$\begin{aligned}
&\sigma(i,j,k)\frac{A_x^{n+1}(i,j,k) - A_x^n(i,j,k)}{\Delta t} \\
&\quad = \frac{1}{2\Delta x^2}\big[A_x^{n+1}(i+1,j,k) - 2A_x^{n+1}(i,j,k) + A_x^{n+1}(i-1,j,k) \\
&\qquad\qquad + A_x^n(i+1,j,k) - 2A_x^n(i,j,k) + A_x^n(i-1,j,k)\big] \\
&\qquad + \frac{1}{2\Delta y^2}\big[A_x^{n+1}(i,j+1,k) - 2A_x^{n+1}(i,j,k) + A_x^{n+1}(i,j-1,k) \\
&\qquad\qquad + A_x^n(i,j+1,k) - 2A_x^n(i,j,k) + A_x^n(i,j-1,k)\big] \\
&\qquad + \frac{1}{2\Delta z^2}\big[A_x^{n+1}(i,j,k+1) - 2A_x^{n+1}(i,j,k) + A_x^{n+1}(i,j,k-1) \\
&\qquad\qquad + A_x^n(i,j,k+1) - 2A_x^n(i,j,k) + A_x^n(i,j,k-1)\big],
\end{aligned} \tag{13}$$

where the superscript indicates the time step and the subscript denotes the component of the vector. The components in y- and z-directions can be written in a similar way. This difference equation can be written in a matrix form:

$$S \cdot A^{n+1} = T \cdot A^n + A_{\mathrm{b}}^n. \tag{14}$$

In this matrix equation, A_{b} is the boundary term, and S and T are large sparse matrix indexed by l, where l is defined by $l = (i-1)NY \cdot NZ + (j-1)NZ + k$. The matrix elements are

$$S(i,j,k,1) = S_{l,l} = \frac{\sigma(i,j,k)}{\Delta t} + \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right), \tag{15}$$

$$S(i,j,k,2) = S_{l,l-1} = S_{l,l+1} = \frac{-1}{2\Delta z^2}, \tag{16}$$

$$S(i,j,k,3) = S_{l,l-NZ} = S_{l,l+NZ} = \frac{-1}{2\Delta y^2}, \tag{17}$$

$$S(i,j,k,4) = S_{l,l-NZ\cdot NY} = S_{l,l+NZ\cdot NY} = \frac{-1}{2\Delta x^2}, \tag{18}$$

$$T(i,j,k,1) = T_{l,l} = \frac{\sigma(i,j,k)}{\Delta t} - \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right), \tag{19}$$

$$T(i,j,k,2) = T_{l,l-1} = T_{l,l+1} = \frac{1}{2\Delta z^2}, \tag{20}$$

$$T(i,j,k,3) = T_{l,l-NZ} = T_{l,l+NZ} = \frac{1}{2\Delta y^2}, \tag{21}$$

$$T(i,j,k,4) = T_{l,l-NZ\cdot NY} = T_{l,l+NZ\cdot NY} = \frac{1}{2\Delta x^2}. \tag{22}$$

3.2. Numerical scheme

The above matrix equation is to be solved by a preconditioned conjugate gradient (CG) scheme. So-called conjugate gradient method is a quite general means for solving the matrix equations like

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \tag{23}$$

where $\mathbf{A}$ is an $N \times N$ matrix and $\mathbf{x}$ and $\mathbf{b}$ are vectors. We briefly summarize this method here (see, e.g., Ref. [13]). One defines a vector function $f(\mathbf{x})$,

$$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \cdot \mathbf{x}. \tag{24}$$

This function is minimized when its gradient

$$\nabla f = \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \tag{25}$$

is equal to zero, which is equivalent to the matrix equation (23) that we wish to solve. The minimization is carried out by generating a succession of search direction $\mathbf{p}_k$ and improved minimizers $\mathbf{x}_k$. At each stage a quantity α_k is found that minimizes $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$, and $\mathbf{x}_{k+1}$ is set equal to the new point $\mathbf{x}_k + \alpha_k \mathbf{p}_k$. After N iterations one arrives at the minimizer over the entire vector space $[\mathbf{p}_1, \mathbf{p}_2, \ldots]$, and that minimizer is the solution to the matrix equation (23). Barrett et al. [14] provided the pseudocode of this CG algorithm. For convenience of reference, we adapt it in the Appendix.

The convergence rate of the CG method strongly depends on the form of the matrix $\mathbf{A}$, and it is found that this method works well for matrices that are well-conditioned, i.e. close to the identity matrix. Thus in order to increase the rate of convergence, we may apply a preconditioner $\mathbf{M}$ such that $\mathbf{M}^{-1} \cdot \mathbf{A} \approx \mathbf{I}$, and solve the preconditioned form of equation

$$\left(\mathbf{M}^{-1} \cdot \mathbf{A}\right) \cdot \mathbf{x} = \mathbf{M}^{-1} \cdot \mathbf{b}. \tag{26}$$

A preconditioner should be close to the original matrix but must be much easier to be inverted. So-called incomplete-lower-upper (ILU) factorization is to adopt a matrix **M** as the preconditioner such that it can be factorized as

$$M = (D + L)D^{-1}(D + U) = (D + L)(I + D^{-1}U), \tag{27}$$

where L and U are strictly triangular lower and upper matrices, respectively, and D is a diagonal matrix. We use the triangular lower and upper parts of the original matrix A to be the matrix U and L. Inverting $Mx = b$ is proceeded by the following two steps:

$$(D + L)z = b \Rightarrow z = D^{-1}(b - Lz), \tag{28}$$

$$\bigl(I + D^{-1}U\bigr)x = z \Rightarrow x = z - D^{-1}(Ux). \tag{29}$$

4. Parallel performance

The portion that consumes the most CPU time in the CG algorithm is to obtain the inverse of pre-conditioner (i.e. $\mathbf{M}^{-1}$) in each conjugate gradient iteration. The ILU factorization method is very effective, but difficult to be parallelized. Fig. 1 illustrates the idea how to achieve parallelization. The upper panel shows that, in order to carry out ILU factorization, the latter CPU need to wait (idle) before it receives the message from the former one. If there are many CPUs, the latter CPUs would idle for a long time and hence the performance is poor. One way to reduce the idle time is to divide the computation regions in each CPU into several blocks such that the latter CPU can start the calculation sooner when the previous CPU finishes the calculation of one small block. In this case, however, messages need to be passed between CPUs frequently so that the overall performance does not improve much.

We implement the localized ILU method in order to achieve better parallel performance. The lower panel in Fig. 1 illustrates the idea of localized ILU method. Refs. [16,17] give more details about the mathematical background and tests on this method. In this method the CPUs do not exchange messages when they carry out ILU factorization, but use approximate values (by zeroing out the matrix components outside the processor domain) in the boundaries between CPUs. This will slightly increase the iteration steps because the localized ILU is not as effective as the exact ILU method in preconditioning. But since every CPU is doing calculation independently, it achieves high parallel performance that can well compensate the increase in more iteration steps.

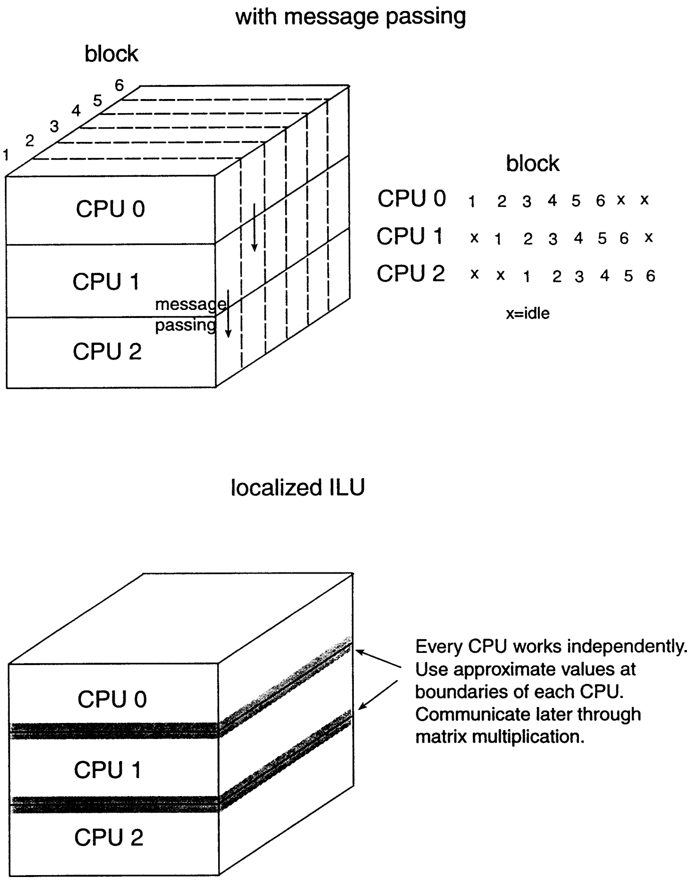
The code has been tested in six different computers: Fujitsu VPP300, IBM SP, Hitachi SR2201, NEC SX4, Cray T94, and Fujitsu VX4. The specifications of these computers can be found in Table 1. We adopted 42 × 42 × 202 grid points, and runs the code for 50 time steps. The physical system in these tests is that a sinusoidal electric field in all three directions E_x, E_y and E_z with period of 1000 s are applied on the surface of a uniform conducting layer ($\sigma = 1.25$). In each time step we call the CG subroutine three times for solving A_x, A_y and A_z. It takes about 30 iterations in each call in order to converge to a residue 10^{-6}. Fig. 2 shows the performance of these six machines. In order to see the ratio of parallelization clearly, we re-draw Fig. 2 in a normalized scale in Fig. 3, where ‘normalized’ means that the speed of a single CPU for each individual computer is assumed to be unity. We define the ratio of parallelization by

$$\frac{\text{(real time when the code runs in one CPU)}}{\text{(real time when the code runs in parallel)} \times \text{(numbers of CPUs)}}.$$

This is equivalent to the value of abscissa divided by the value of ordinate in Fig. 3. A point that is closer to the line denoted by unity in Fig. 3 means a higher ratio of parallelization. When four CPUs are used (see the inset of Fig. 3), the ratio of parallelization is 90.34, 88.85, 87.145, 86.42, 80.67, and 72.32% for NEC SX4, Fujitsu VPP300, Fujitsu VX4, IBM SP, Hitachi SR2201, and Cray T94 machines, respectively. When more CPUs were used and thus more messages passing between CPUs were involved, we can still achieve parallelization ratio at 67.88% for 15 CPUs in VPP300, 79.51% for 40 CPUs in SR2201, and 41.94% for 40 CPUs in SP machines. When

Table 1
Specifications of the JAERI computers in which the parallel code is tested

Machine	# of PEs	Peak CPU speed/PE	Memory size/PE	Data transferring speed
Cray T94	4	1.8 GFLOPS	1 GB/4	128 GB/s
Fujitsu VPP 300	16	2.2 GFLOPS	512 MB	570 MB/s
Hitachi SR 2201	64	0.3 GFLOPS	256 MB	0.3 GB/s
IBM SP2	48	266 MFLOPS	128 MB	40 MB/s
NEC SX-4	6	2 GFLOPS	512 MB	40 MB/s



block

CPU 0 1 2 3 4 5 6 x x
CPU 1 x 1 2 3 4 5 6 x
CPU 2 x x 1 2 3 4 5 6

x=idle

Every CPU works independently.
Use approximate values at
boundaries of each CPU.
Communicate later through
matrix multiplication.

Fig. 1. Cartoons showing the algorithm adopted in order to achieve parallelization. The upper panel shows the exact ILU method, which has a poor parallel performance because of the time spent in message passing and the long idle time spent when the CPUs wait for one another. The lower panels shows localized ILU method, which achieves good parallel performance.

more than 40 CPUs are used there are not enough number of grids in the z-direction and accordingly each CPU is given almost the same amount of grid points as in the use of 40 CPUs.

In the previous test we did not adopt large numbers of grids because some of these machines (particularly T94 and SX4) do not have enough memory spaces if the code were run with more grids. A large number of grids in the z-direction will improve the parallel ratio because the ratio of the overlapped layer between CPUs to the entire

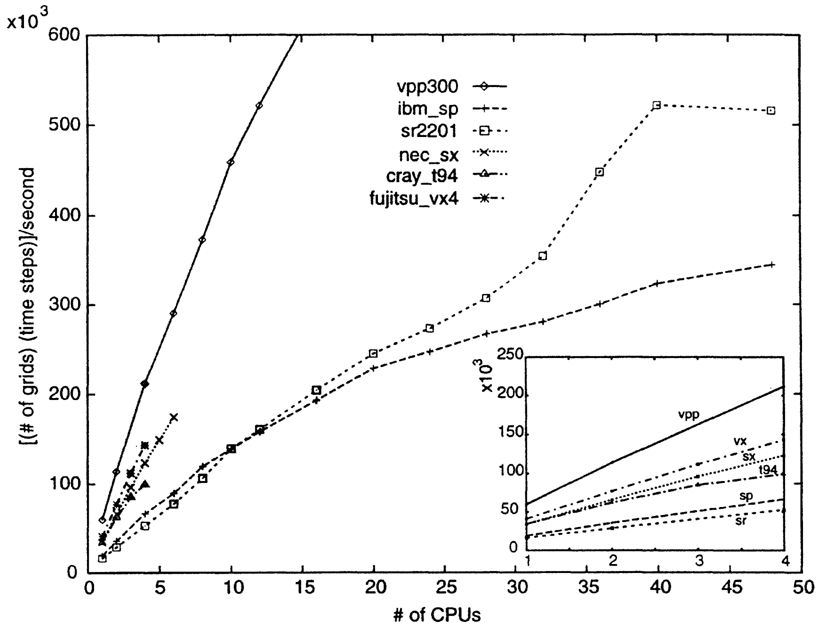


Fig. 2. Performance of the induction equation solver in six different supercomputers (Fujitsu VPP300, IBM SP, Hitachi SR2201, NEC SX4, Cray T94, and Fujitsu VX4). The test adopts 42 × 42 × 202 grid points and runs 50 time steps. The ordinate is the number of CPUs to carry out the computation. The abscissa is the measure of the overall speed, which is defined by the total number of grids times the number of time steps divided by the time (in second) needed to complete the run. The inset zooms out the area where numbers of CPUs are less than 4.

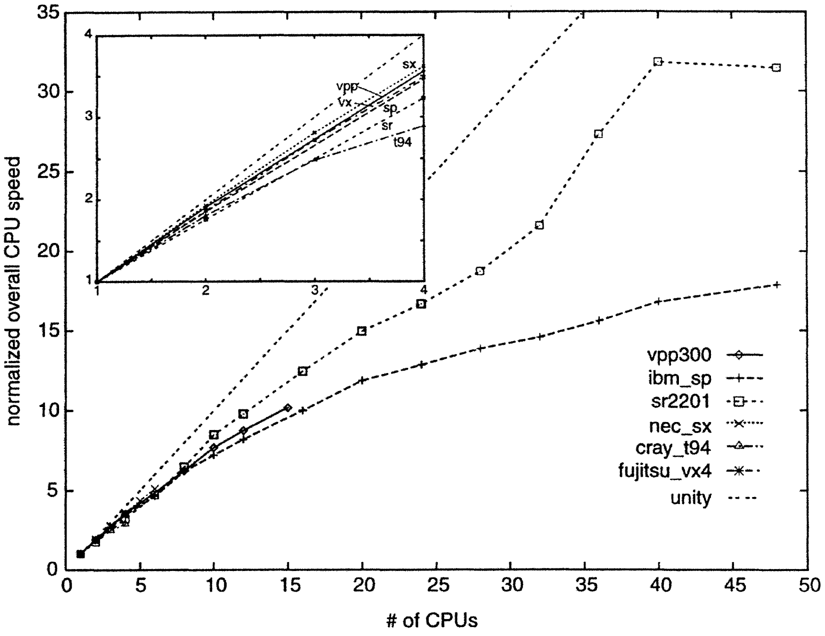


Fig. 3. Parallel performance of the induction equation solver, which is from the same tests as in the previous figure but drawn in the normalized scale. Here 'normalized' means the performance (speed) of a computer is compared with that when a single CPU of the same machine would have been used. The line denoted by 'unity' is for perfect (100%) parallelization. A curve that is closer to this unity line means better parallelization.

computation area becomes smaller, and the saturation in Fig. 3 would disappear. This point is justified when we tried to increase the grid numbers to 161 × 161 × 513, and run the code in VPP300, SP, and SR2201 machines. We would like to point out that parallelization increases not only the speed of computation but also the available memory, and hence it allows us to carry out the computation with such a large number of grid points.

5. Testing of the code

In the following tests, we solve Eq. (11) numerically using the parallel code in Cartesian coordinates and assuming a periodic boundary condition at four sides ($\pm x, \pm y$). In our code the conductivity σ and the boundary conditions can be given as any function of space and time. At present, however, we choose only simple cases for which the analytical solutions are known. We assume $A = 0$ at the bottom ($z = -z_{\max}$) boundary. At $z = 0$ we assume a sinusoidal function of time, $A_x(z = 0) = \sin \omega t$, and $A_y = A_z = 0$. This condition corresponds to an application of an electric field $E_x = -\omega \cos(\omega t)/c$ on the surface of the Earth. A one-dimensional distribution of σ is assumed except in Test 4 in which we examine the 3D dependence of σ.

5.1. Test 1: Skin depth

In the normalized unit, the skin depth λ in Eq. (10) becomes

$$\lambda = \sqrt{\frac{T}{\pi\sigma}}. \tag{30}$$

We assume a uniform conductivity and apply a sinusoidal electric field E_x with period of 10 000 s at the surface to check if the EM filed can penetrate up to the theoretically calculated skin depth. Fig. 4 shows the induced magnetic field B_y for the cases of $\sigma = 1.25\sigma_0$, $12.5\sigma_0$, and $125\sigma_0$, which corresponds to realistic values of 0.01, 0.1 and 1 S/m, respectively. The theoretical skin depths for these three cases are 50.5, 16.0 and 5.1 in normalized scale L (or 505, 160, and 51 km in the realistic scale), respectively. We can see that the induced magnetic field at a depth deeper than the skin depth does not keep the sinusoidal dependence on time and the amplitude becomes very small. Fig. 5 shows more clearly that the amplitude of magnetic field B_y decays as a function of depth. This figure is a snapshot at $t = 7000$ s. From this figure the skin depth is determined by the depth at which the field is $1/e$ of the value at the surface, i.e. the ordinates of the cross points of the $B(z)/B(0) = 1/e$ line and the three curves. The skin depths obtained in this simulation agree with theoretical values.

5.2. Test 2: Two layers

In this test we assumed that a sinusoidal electric field with a period of 1000 s is applied on the surface, and that there are two layers of different conductivities. The first layer, which corresponds to crust, has conductivity $\sigma = 0.125\sigma_0$ (10^{-3} S/m), and is 50 km (z from 0 to 5) thick. The value of conductivity is based on Schultz et al. [15]. The bottom layer extends to the lower boundary at 800 km and assumed to be with uniform conductivity of $\sigma = 0.125\sigma_0$ (so in fact there is only one layer), $\sigma = 1.25\sigma_0$, $\sigma = 12.5\sigma_0$, and $\sigma = 125\sigma_0$, respectively. Fig. 6 shows the results of these tests. The electric field is plotted in solid line and the value of E_x is rescaled 10 times larger for plotting so that it can be shown in the same figure as the B_y. From this figure we can see that the induced magnetic field B_y lags the electric field in phase from $\pi/4$ (one layer) to $\pi/2$ (large conductivity in the bottom layer). This agrees with the theory, too.

5.3. Test 3: Four layers

In this test we assume that there are four layers of different conductivity in the vertical direction. The boundaries of these four layers were chosen at the discontinuity depths of seismic P wave speed (see, e.g., Tajima et al. [18]),

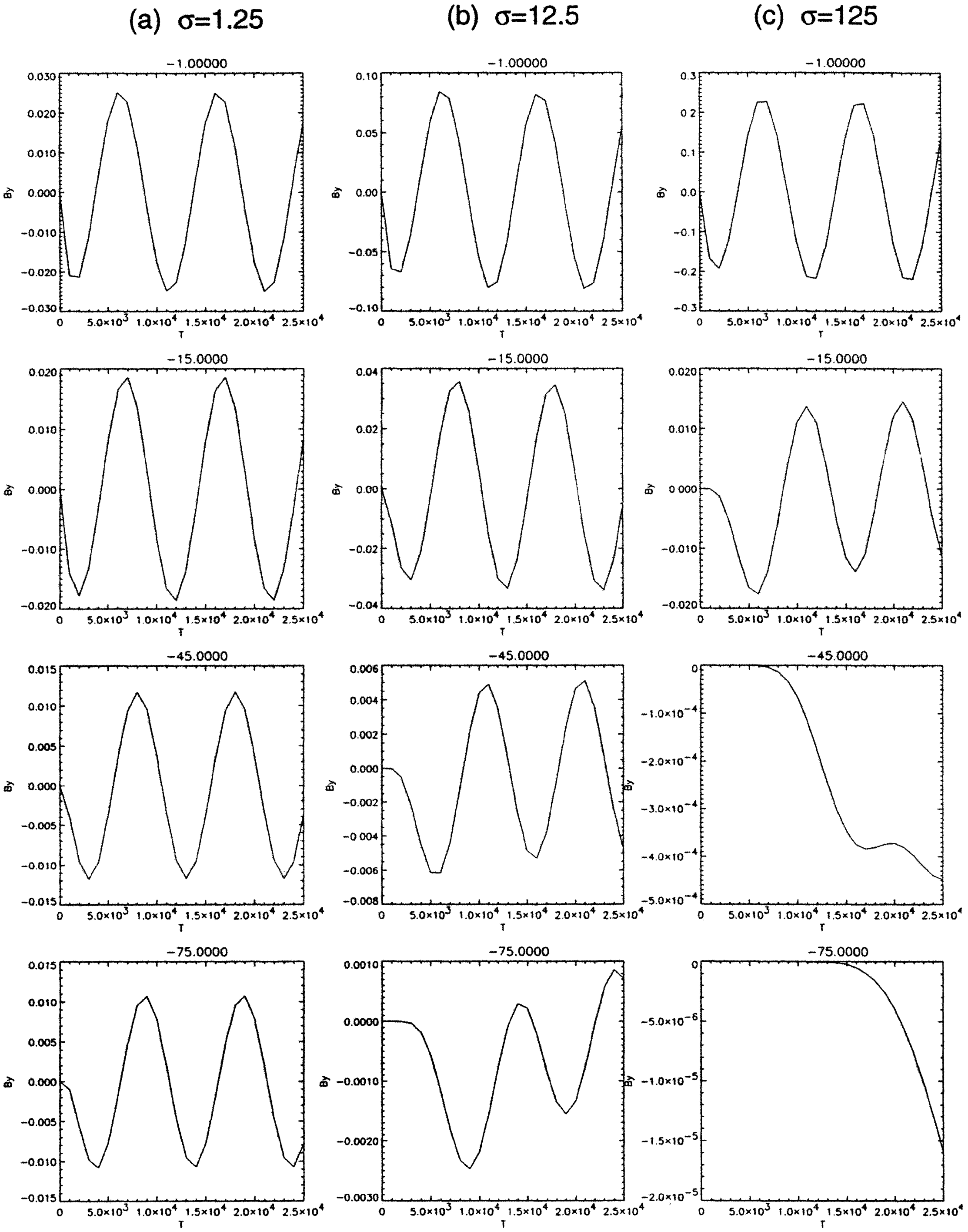


Fig. 4. Induced magnetic field for different values of conductivity, in (a) $\sigma = 1.25\sigma_0$ (0.01 S/m), (b) $\sigma = 12.5\sigma_0$ (0.1 S/m), and (c) $\sigma = 125\sigma_0$ (1 S/m). A sinusoidal electric field E_x of period 10 000 s is imposed on the top surface (z=0), and the figures shows the B_y at $z = -1L$ (−10 km), $z = -15L$ (−150 km), $z = -45L$ (−450 km), and $z = -75L$ (−750 km), respectively.

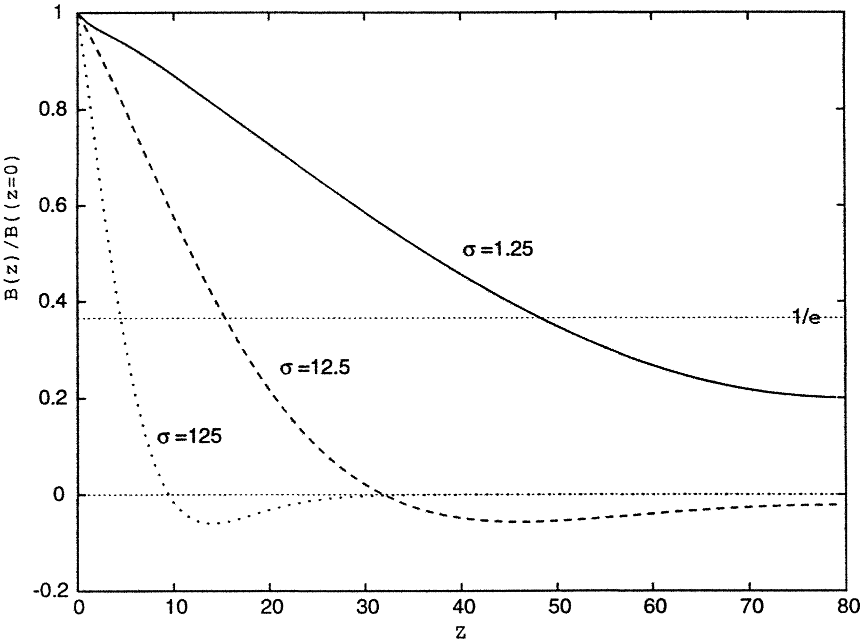


Fig. 5. The normalized magnetic field as a function of the depth (z) from the surface. The values were taken at $t = 7000$ s in the previous figure, when the magnetic field is at its maximum. The z values at the cross points of the $B(z)/B(0) = 1/e$ line and the three curves correspond to the skin depth, which agree with the theoretical calculation.

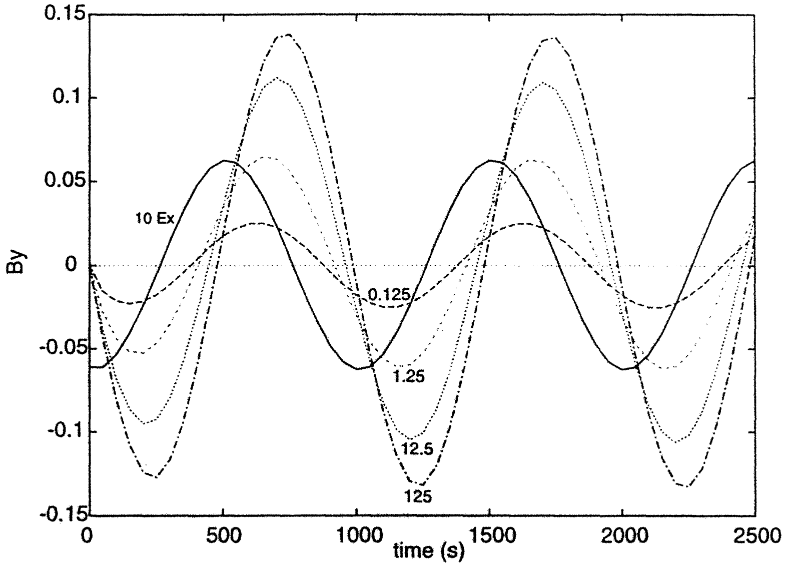


Fig. 6. The induced magnetic field in the case when there are two layers of different conductivities. A sinusoidal electric field with a period of 1000 s is applied on the surface, which is plotted in black solid line and is 10 times larger than its actual value. The first layer has conductivity $\sigma = 0.125\sigma_0$ (10^{-3} S/m), and is 50 km thick. The bottom layer has conductivity of $\sigma = 0.125\sigma_0$ (so in fact there is only one layer), $\sigma = 1.25\sigma_0$, $\sigma = 12.5\sigma_0$, and $\sigma = 125\sigma_0$, respectively. Note the differences in their amplitudes and the phase shifts with respect to the electric field.

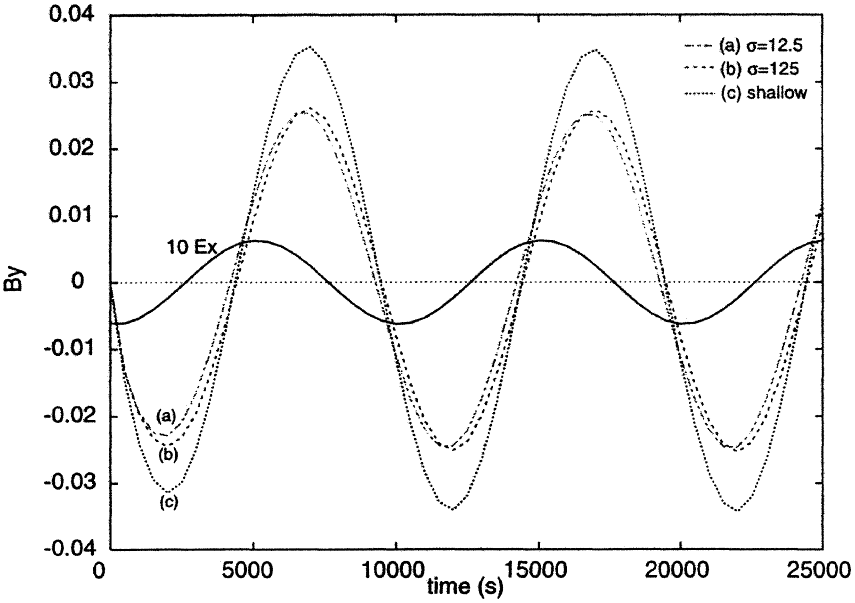


Fig. 7. The induced magnetic field in the case when there are four layers of different conductivities. A sinusoidal electric field with period of 10 000 s is applied on the surface, which is plotted in black solid line and is 10 times larger than its actual value. Three cases are different in the values and depths of the conductivity in the third layer. See the text for details of these three cases.

which are at about 50, 410, and 660 km. We assume $A = 0$ at 800 km. We apply a sinusoidal electric field with a period of 10 000 s on the surface and test three cases:

(a) $\sigma = 0.125$ from $z = 0$ to 5, $\sigma = 1.25$ from $z = 6$ to 40, $\sigma = 12.5$ from $z = 41$ to 66, and $\sigma = 125$ from $z = 66$ to 80;

(b) the same as in (a) except that in the third layer (from $z = 41$ to 66) the conductivity $\sigma = 125$;

(c) the third layer is located at a shallower position, that is, $\sigma = 0.125$ from $z = 0$ to 5, $\sigma = 1.25$ from $z = 6$ to 20, $\sigma = 12.5$ from $z = 21$ to 66, and $\sigma = 125$ from $z = 66$ to 80.

Fig. 7 shows the results of these three cases, in which one can tell the difference of the third layer conductivity by identifying the ratio of the surface magnetic field to the electric field and the phase shift.

5.4. Test 4: 3D distribution of conductivity

In the above tests we examined conductivity only a function of z. Our present code has, in fact, capabilities to compute the 3D structure. In this test we assume that there are two layers: the first layer is 50 km thick (z is from 0 to 5) and has a conductivity $\sigma = 0.125$. The second layer extends to 800 km and has the space dependency as:

$$\sigma(x, y) = 1.25\big[1 + 0.2\sin(2\pi x/L_x)\sin(2\pi y/L_y)\big], \tag{31}$$

where $L_x = L_y = 100$ (1000 km). On the surface we apply an E_x which oscillates with a period of 1000 s. If the conductivity were uniform in the x and y directions, as in the previous cases, the induce magnetic field would have only the y component which comes from the term $\partial A_x/\partial z$. In this test, however, we have an additional non-zero component, B_z, and it is proportional to $\cos(k_y y)$ which comes from $\partial A_x/\partial y$. Fig. 8 shows the magnitude of the induced magnetic field in the vertical direction (B_z) and its x, y dependencies.

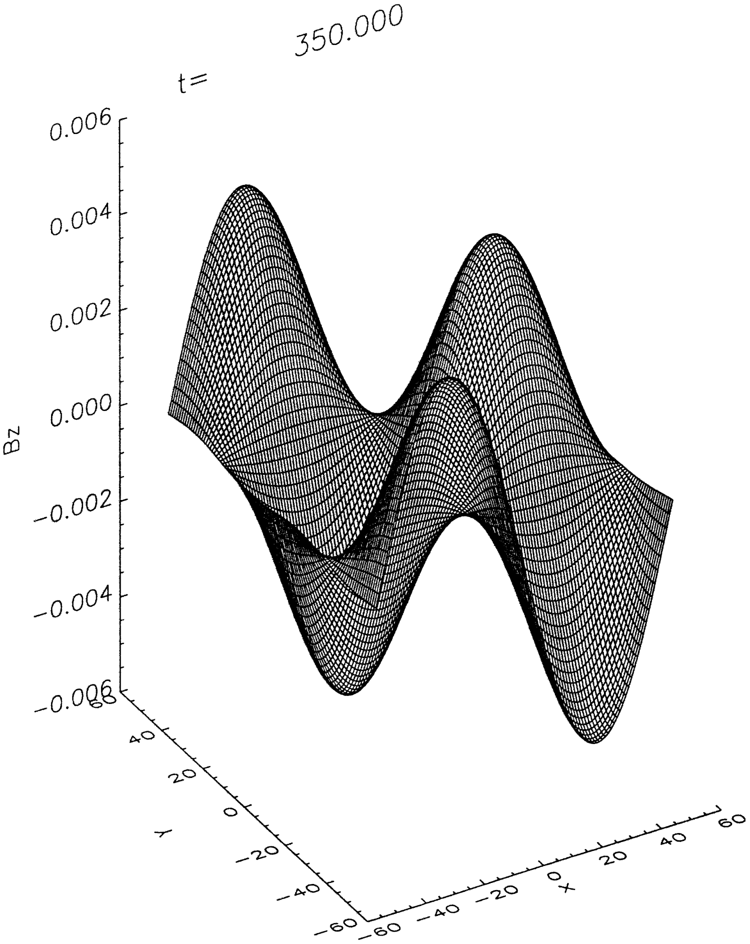


Fig. 8. The induced magnetic field in the vertical direction (B_z), in the case when a sinusoidal E_x of period 1000 s is applied on the surface, the upper layer has a conductivity $\sigma = 0.125\sigma_0$ and depth 50 km, and the conductivity in the second layer is a function of x and y: $\sigma(x, y) = 1.25\sigma_0[1 + 0.2\sin(2\pi x/L_x)\sin(2\pi y/L_y)]$. The snap shot is plotted at $t = 350$ s.

5.5. Test 5: Electric pulse

Because our code solves the induction equation in the time domain instead of in the spectral domain, we do not need to restrict ourselves in the sinusoidal time dependence. Therefore, the external field can be assumed to be any function of time. In this test we assume that the vector potential on the top boundary (surface of the Earth) is

$$A_x(z = 0) = A_0 \exp\left[\left(\frac{t - t_0}{\Delta}\right)^2\right], \tag{32}$$

where 10, 300, and 10 are adopted for A_0, t_0 and Δ respectively. This test is equivalent to applying a pulse-like electric field

$$E_x = -\frac{2A_0}{\Delta^2}(t - t_0) \exp\left[\left(\frac{t - t_0}{\Delta}\right)^2\right] \tag{33}$$

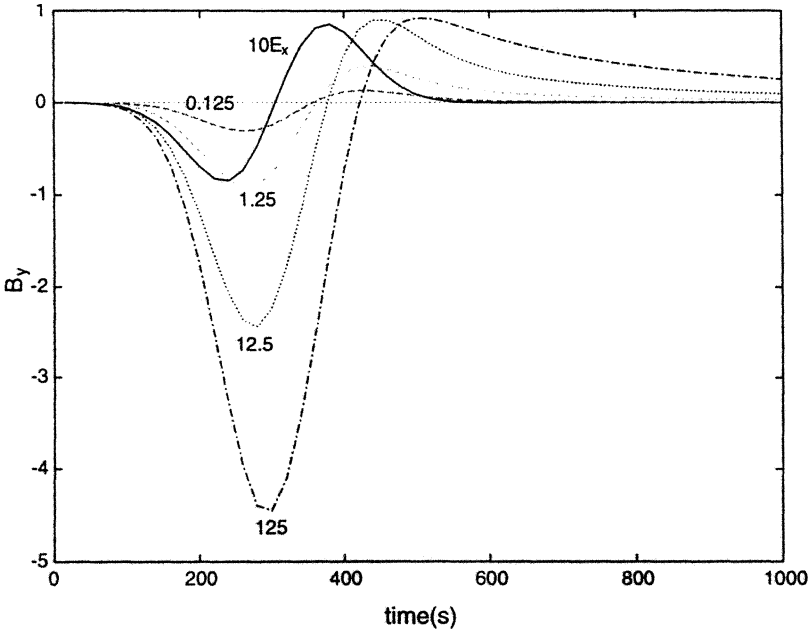


Fig. 9. The induced magnetic field in the case when an electric pulse (shown in solid line) is applied on the surface of a uniform conducting layer whose depth is 800 km. Four curves are for the conductivity equals to $0.125\sigma_0$, $1.25\sigma_0$, $12.5\sigma_0$, and $125\sigma_0$, respectively.

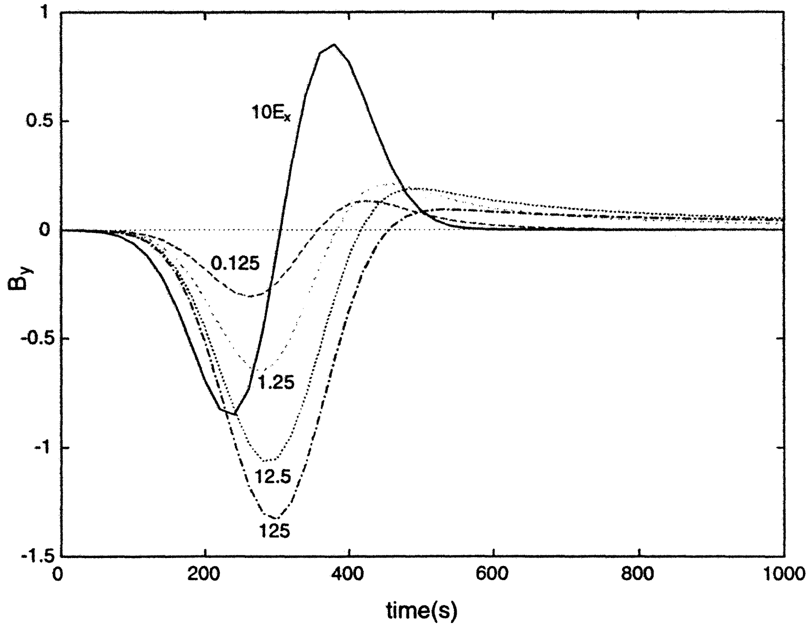


Fig. 10. The induced magnetic field in the case when an electric pulse (shown in solid line) is applied on the surface of two layer structure. The top layer is 50 km thick and has a conductivity $\sigma = 0.125\sigma_0$ (10^{-3} S/m). The bottom layer is 750 km thick, and the conductivity equals to $0.125\sigma_0$, $1.25\sigma_0$, $12.5\sigma_0$, and $125\sigma_0$, respectively.

on the surface of the Earth. Fig. 9 shows the induced magnetic field in the case when the electric pulse (shown in solid) is applied on top of the layer of uniform conductivity. The value of the conductivity are $0.125\sigma_0, 1.25\sigma_0, 12.5\sigma_0$, and $125\sigma_0$, respectively. Fig. 10 shows results for the case when the electric pulse is applied to the same structures of conductivity as are described in Test 2 (see Fig. 6). In addition to the features that we have seen in Tests 1–4 (such as the phase shift and stronger induced field for larger conductivity), in this case we see (in Figs. 9 and 10) that after the pulse is applied the induced field decays faster if the conductivity is smaller. This agrees with the theory since the dissipation of the induced current is larger and faster for higher resistivity (lower conductivity).

6. Conclusion

We demonstrated a good performance of our parallel code on various supercomputers. We tested the code in several physical conditions for which analytical solutions are known, and therefore our computational results can be cross examined. The results show good agreements with the analytical solutions. Thus, it seems promising to further develop this code to simulate more realistic electromagnetic interactions with the Earth’s deep interior.

Acknowledgement

This work has been carried out when the first author was employed by the Department of Research for Computational Earth Science, Research Organization for Information Science & Technology (RIST), as part of the Earth Simulator project funded by Science Technology Agency of Japan. We thank the members of the project who provided us with support, advice and encouragement. We also benefited from the members at the Ocean Hemisphere Center of the Earthquake Research Institute at the University of Tokyo, who provided useful information on MT research. FT acknowledges partial support by her NSF grant No. EAR-9805006 and JSPS Fellowship.

The computational work is carried out in the supercomputer clusters in the Japan Atomic Energy Research Institute (JAERI), and in the National Astronomical Observatory of Japan.

Appendix: Pseudocode of CG algorithm

The pseudocode is adapted from Barrett et al. [14].

```
Compute $r^{(0)} = b - Ax^{(0)}$ for an initial guess $x^{(0)}$
**for** $i = 1, 2, \ldots$
**solve** $Mz^{(i-1)} = r^{(i-1)}$
    $\rho_{i-1} = r^{(i-1)} z^{(i-1)}$
**if** $i = 1$
    $p^{(1)} = z^{(0)}$
**else**
    $\beta_{i-1} = \rho_{i-1}/\rho_{i-2}$
    $p^{(i)} = z^{(i-1)} + \beta_{i-1} p^{(i-1)}$
**endif**
$q^{(i)} = Ap^{(i)}$
$\alpha_i = \rho_{i-1}/p^{(i)} q^{(i)}$
$x^{(i)} = x^{(i-1)} + \alpha_i p^{(i)}$
```

```
    $r^{(i)} = r^{(i-1)} - \alpha_i q^{(i)}$
    check convergence $(|x^{(i)} - x^{(i-1)}|/|x^{(i-1)}| < \epsilon)$; continue if necessary
    **end**
```

References

[1] L. Cagniard, Basic theory of the magneto-telluric method of geophysical prospecting, Geophysics 18 (1953) 605–635.

[2] R.L. Parker, The inverse problem of electromagnetic induction: existence and construction of solutions based on incomplete data, J. Geophys. Res. 85 (B 8) (1980) 4421–4428.

[3] R. Parker, K.A. Whaler, Numerical methods for establishing solutions to the inverse problem of electromagnetic induction, J. Geophys. Res. 86 (B 10) (1981) 9574–9584.

[4] P.E. Wannamaker, Advances in three-dimensional magnetotelluric modeling using integral equations, Geophysics 56 (11) (1991) 1716–1728.

[5] R.L. Mackie, T.R. Madden, Conjugate direction relaxation solutions for 3D magnetotelluric modeling, Geophysics 58 (7) (1993) 1052–1057.

[6] R.L. Mackie, T.R. Madden, P.E. Wannamaker, Three-dimensional magnetotelluric modeling using difference equations – theory and comparisons to integral equation solutions, Geophysics 58 (2) (1993) 215–226.

[7] T.S. Zhang, A. Schultz, A 3D perturbation solution for the EM induction problem in a spherical Earth – the forward problem, Geophys. J. Int. 111 (1992) 319–334.

[8] M.E. Everett, A. Schultz, Geomagnetic induction in a heterogeneous sphere: azimuthally symmetric test computations and the response of an undulating 660 km discontinuity, J. Geophys. Res. 101 (1996) 2765–2783.

[9] Z. Martinec, Spectral-finite-element approach to two-dimensional electromagnetic induction in a spherical Earth, Geophys. J. Int. 130 (1997) 583–594.

[10] J.T. Smith, Conservative modeling of 3D electromagnetic fields, Part I: Properties and error analysis, Geophysics 61 (5) (1996) 1308–1318.

[11] J.T. Smith, Conservative modeling of 3D electromagnetic fields, Part II: Biconjugate gradient solution and an accelerator, Geophysics 61 (5) (1996) 1319–1324.

[12] T. Wang, G.W. Hohmann, A finite-difference, time-domain solution for three-dimensional electromagnetic modeling, Geophysics 58 (6) (1993) 797–809.

[13] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran (Cambridge University Press, New York, 1992) Ch. 2.

[14] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C., Romine, H. Van der Vorst, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods (SIAM, Philadelphia, PA, 1994).

[15] A. Schultz, R.D. Kurtz, A.D. Chave, A.G. Jones, Conductivity discontinuities in the upper mantle beneath a stable craton, Geophys. Res. Lett. 20 (24) (1993) 2941–2944.

[16] V. Venkatakrishnan, Parallel implicit methods for aerodynamic applications on unstructured grids, in: Domain-based Parallelism and Problem Decomposition Methods in Computational Science and Engineering, D.E. Keyes, Y. Saad, D.G. Truhlar (Eds.) (SIAM, 1995) Ch. 4.

[17] K. Nakajima, H. Nakamura, T. Tanahashi, Parallel iterative solvers with localized ILU preconditioning, in: High-performance Computing and Networking, B. Hertzberger, P. Sloot (Eds.) (Springer, Berlin, 1997) pp. 342–350.

[18] F. Tajima, T. Fukao, M. Obayashi, T. Sakurai, Evaluation of slab images in the northwestern Pacific, Earth Planets Space 50 (1998) 953–964.