Fast Computation of Quasi-Dynamic Earthquake Cycle Simulation with Hierarchical Matrices

M. Ohtani\textsuperscript{a,}* , K. Hirahara\textsuperscript{a}, Y. Takahashi\textsuperscript{b}, T. Hori\textsuperscript{c}, M. Hyodo\textsuperscript{c}, H. Nakashima\textsuperscript{d}, and T. Iwashita\textsuperscript{d}

\textsuperscript{a}Graduate School of Science, Kyoto University, Oiwake, Kitashirakawa, Sakyo, Kyoto 606-8502, Japan,  
\textsuperscript{b}Department of Electrical Engineering, Faculty of Science and Engineering, Doshisha University, Kyotanabe, Kyoto 610-0321, Japan,  
\textsuperscript{c}JAMSTEC, Showa-machi, Kanazawa, Yokohama 236-0001, Japan,  
\textsuperscript{d}ACCMS, Kyoto University, Honmachi, Yoshida, Sakyo, Kyoto 606-8501, Japan

Abstract

In quasi-dynamic earthquake cycle simulations based on rate and state friction laws, we applied the method of Hierarchical-Matrices (H-matrices) to multiplicative computations of the $N \times N$ slip response function matrix and the slip deficit rate vector, where $N$ is the number of divided cells on the plate surface. H-matrices, which are efficient low-rank compressed representations of dense matrices, enable more rapid arithmetic operations with less memory sizes. In this study, we constructed a friction model of quasi-dynamic earthquake cycles on a flat, dipping, plate interface in a semi-infinite homogeneous elastic medium, and investigated the effectiveness of H-matrices by changing $N$ from $10^4$ to $10^6$. Construction of H-matrices involves several parameters controlling the structure and accuracy of the approximated matrix. With H-matrices using proper values for these parameters to maintain accuracy, except for smaller values of the parameter for suppressing the ranks of the outermost submatrices, the memory size of the matrix was reduced to about $O(N)$. The computational time in the multiplication was also reduced to $O(N)$ for a range of $N$ values less than about $10^5$, and to $O(N) \sim O(N \log N)$ for a larger range. Thus, we found that the application of H-matrices greatly reduces the computational time and memory size in earthquake cycle simulations. This advance should enable the realization of large- and multi-scale simulations with a million order cells and the estimation of frictional parameters.

Keywords: H-matrices; earthquake cycle simulation; large- and multi-scale simulation; slip response function; rate and state friction law

1. Introduction

Recently, earthquake cycle simulations based on laboratory-derived friction laws have been performed to reproduce complex sequences of large historical interplate earthquakes. For example, Hori (2006) [1] executed a large-scale simulation of earthquake occurrences along the Nankai Trough in southwest Japan, and Kato (2008) [2]...
reproduced the recurrence of two asperity ruptures, which includes afterslip triggering the other earthquake, in the Sanriku region in northeast Japan. Whereas these authors dealt with large-scale heterogeneities of frictional parameters or asperities, Hillers et al. (2007) [3], taking microscopic structures into account, examined the earthquake occurrences on a large fault with different levels of heterogeneity to produce seismicity patterns with Gutenberg-Richter statistics resembling those of natural earthquakes. On the other hand, long- and short-term slow slip events and low-frequency events have been observed on plate interfaces in the deep extended portions of seismogenic zone (e.g., Hirose and Obara, 2005) [4]. Furthermore, recent simulations have shown the possibility that the activity of such slow slip events may change before the occurrence of large interplate earthquakes (e.g., Ariyoshi et al., 2009) [5]. Thus, to achieve realistic earthquake cycle simulations, it is necessary to develop multi-scale models of earthquake cycles across larger regions, including occurrences not only of large and smaller interplate earthquakes but also of slow slip events. However, it is difficult to realize such multi-scale simulations in large regions. We divide the plate interface or fault into subfaults or cells with small sizes, which are determined by frictional parameters (Rice, 1993) [6]. The multi-scale and the larger model size simulations require smaller cell sizes and larger numbers of cells, leading to greater computational times (CPU-times) and memory sizes. Therefore we need to reduce them.

Another reason for reducing CPU-times is to extend the present forward simulations to predictive simulations for earthquake occurrence. At present, the search for the distributions of frictional parameters that reproduce complex sequences of past earthquakes is performed manually or intuitively. In predictive simulations, the estimated frictional parameters and the resultant predictions of earthquake occurrences should contain information on some statistical reliability. For this purpose, it is necessary to introduce data assimilation methods developed in the fields of weather forecasting and hydrology (e.g., Wunch, 2006) [7] into the estimation of frictional parameters and initial conditions of physical parameters. Any data assimilation method requires a large number of iterations of forward and backward simulations, which also leads to enormous CPU-times.

To reduce CPU-times and memory sizes, a number of techniques have been used for multiplicative computation of the slip response function matrix (SRFM) and the slip deficit rate vector (SDRV) in earthquake cycle simulations. Here, the slip response function, which comprises the SRFM, is the stress change on a receiver cell due to a unit slip assigned on a source cell on the plate interface; the magnitude of this stress change is proportional to \( r^3 \), where \( r \) is the distance between the two cell positions. One such technique is Fast Fourier Transformation (FFT) (e.g., Kato, 2008) [2]. This method makes it possible to reduce the CPU-time and memory size from \( O(N^3) \) to \( O(N \log N) \), where \( N \) is the number of subdivided cells. However, FFT requires periodic boundary conditions and cannot deal with the effects of the Earth’s free surface and non-planar interfaces or faults. Another technique is the Fast Multipole Method (FMM), which has been extensively employed in the computation of long-range N-body forces under the Laplace or Helmholtz field. Although FMM may be applicable to earthquake cycle simulations, only two research groups have examined the technique thus far. FMM was first introduced by Tullis et al. (1999) [8], who showed that CPU-time is reduced to \( O(N \log N) \) in the case of strike-slip faulting in a semi-infinite homogeneous elastic medium. Hirahara et al. (2009) [9] examined a flat interface in an infinite elastic medium where the CPU-time is reduced to \( O(N) \). Compared to FFT, FMM has an advantage in that it does not require any cyclic boundary conditions. To apply FMM, however, suitable analytic forms of the slip response function are required so that they can be transformed for FMM. Apparently, it is difficult to obtain suitable forms in the case of dip-slip faulting on a dipping interface in a semi-infinite elastic medium with the Earth’s free surface (Ohtani and Hirahara, 2010) [10]—a case corresponding to the interplate earthquakes at subduction zones.

In this study, we consider the application of Hierarchical Matrices (H-matrices) to the multiplicative computation of SRFM and SDRV. The method of H-matrices was first introduced by Hackbusch (1999) [11]. Representing a matrix in the form of an H-matrix reduces the amount of memory from \( O(N^2) \) to almost \( O(N \log N) \). The idea of H-matrices is similar to that of FMM because both are based on low-rank approximations despite differing in the method of approximation. The construction of H-matrices involves an approach using purely algebraic algorithms, independent of the form of function comprising each element of the matrix, though the function needs to give some decaying nature according to the distance from the diagonal. This leads to flexibility in the application of H-matrices.

In this paper, we first give a brief explanation of H-matrices. Subsequently, in earthquake cycle simulations with H-matrices, we examine the memory size, CPU-time, and accuracy by changing the number of cells in a friction model and the parameters controlling the accuracy of H-matrices.
2. H-matrices

H-matrices are compressed low-rank representations of densely populated matrices (Börm et al., 2009) [12]. Matrices are split into a hierarchy of rectangular submatrices with different sizes, and are each approximated with a low-rank matrix. Therefore, with this H-matrix, it is possible to perform arithmetic operations more quickly and with less memory. In this study, we use a subroutine library, HLib, provided by the Max-Planck-Institute (HLib, 2009) [13]. Taking as an example, the case of an SRFM whose \( i \)-th row and \( j \)-th column element has an absolute value that rapidly decays with the distance between \( i \)-th receiver and \( j \)-th source cells, we briefly describe how to construct H-matrices. For details, we refer the reader to Börm et al. (2009) [12].

First, we reorder the elements of the matrix, SRFM, so that the distance between the corresponding receiver and source cells becomes as large as possible. Second, we subdivide SRFM into submatrices. Starting from the whole cell as the root model region, we subdivide the root cell region into several cell regions by iterating bisection until the number of cells in the submatrices becomes less than \( n_{\text{min}} \). Based on this hierarchical cell structure in the model region, we subdivide SRFM following the admissibility condition,

\[
\min\{\text{diam}_1, \text{diam}_2\} \leq \eta \cdot \text{dist},
\]

where \( \text{diam}_1 \) and \( \text{diam}_2 \) represent the sizes of receiver and source cell regions, respectively, \( \text{dist} \) is the distance between the two cell regions, and \( \eta \) determines the strictness of the condition. If Eq. (1) is satisfied, the two cell regions are admissible and sufficiently distant; we then stop to subdivide the matrix. Otherwise, if the two cell regions are inadmissible and close, we further subdivide the submatrices to a lower hierarchical level (if not at the bottom level). We then obtain the hierarchical structure, consisting of admissible and inadmissible submatrices.

Third, we approximate each admissible submatrix \( M \in \mathbb{R}^{m \times n} \) of at most rank \( k \) \((k < \min(m, n))\), with the Rk-matrix representation as

\[
M \approx \sum_{i=1}^{k} g_i h_i^T, \quad (g_i \in \mathbb{R}^m, h_i \in \mathbb{R}^n),
\]

where \( \mathbb{R}^{m \times n}, \mathbb{R}^m, \mathbb{R}^n, \) and \( h_i^T \) denote \( m \times n \) real matrices, \( m \) and \( n \) real vectors, and a transposed vector of \( h_i \), respectively. We adaptively determine the rank \( k \) following the method of ACA (Adaptive Cross Approximation) (Bebendorf, 2000) [14], to be the relative error of approximation smaller than \( \varepsilon_{\text{ACA}} \). Furthermore, if \( k \) reaches \( k_{\text{ACA}} \), which is the maximum rank each submatrix can take, we take the rank \( k \) as \( k_{\text{ACA}} \). With \( k \) smaller than \( m/2 \), we can save memory for storing \( M \), whose memory size is reduced from \( O(mn) \) to \( O(k(m+n)) \). The multiplication of a Rk-matrix \( M \in \mathbb{R}^{m \times n} \) and a vector \( v \in \mathbb{R}^n \) gives,

\[
M \cdot v = \left( \sum_{i=1}^{k} g_i h_i^T \right) \cdot v = \sum_{i=1}^{k} (g_i (v^T h_i)^T) = \sum_{i=1}^{k} \text{const}_i \cdot g_i.
\]

This also reduces the amount of computation from \( O(mn) \) to \( O(k(m+n)) \).

On the other hand, the inadmissible submatrices are stored as full matrices without any approximation. Finally, we recompress the submatrices. Here, we easily compute singular values \( (\sigma_i) \) with the Rk-matrix representation and further lower the rank by deleting the eigenvectors with smaller singular values of \( \sigma_i < \varepsilon \sigma_1 \).

Following the above mentioned algorithms, with implementation of HLib library in our earthquake cycle simulation code, we can compress efficiently SRFM in an H-matrix form and speed up the matrix-vector multiplication with less memory sizes. Although HLib provides other approximation strategies than ACA, we choose ACA for its applicability and smaller computational time. It is to be noted that, from its nature of ACA, we do not need to compute all values of the root SRFM element but only those of the slip response functions (SRFs)
when choosing each low and column, at each submatrix as in Eq.(2). This leads to the much time reduction of computing SRFs, and thus constructing H-matrices.

3. Quasi-dynamic earthquake cycle simulation

We present some details of the earthquake cycle simulation to which we apply the H-matrices.

3.1. Simulation procedure

First, we set a configuration of a subducting plate interface in a semi-infinite homogeneous elastic medium. We then divide the plate interface into \( N \) triangular cells. We assume that the slip and the plate convergence vectors have only dip components. Then, the quasi-dynamic equilibrium equation gives the shear stress \( \tau_i \) at cell \( i \) at some time \( t \) as,

\[
\tau_i(t) = \sum_{j=1}^{N} K_{ij} (u_j(t) - V_{pl}t) - \frac{\xi G}{2V_s} V_i(t),
\]

where \( u_i \) and \( V_i \) are the slip and slip-rate at cell \( i \), respectively, and \( V_{pl} \) is the plate velocity. \( G, V_s \), and \( \xi \) are the rigidity, S-wave velocity, and damping parameter, respectively. \( K_{ij} \) is an element of SRFM: a static stress change at cell \( i \) induced by a unit slip at cell \( j \), which is calculated following Comninou and Dundurs (1975) \[15\] in this study. On the right-hand side of Eq. (4), the first term represents the multiplication of the slip response function and the slip deficit; the second approximates the quasi-dynamic slip behaviour during earthquakes and is called as radiation damping (Rice, 1993) \[6\]. Subsequently, the shear stress is set to be equivalent to the frictional stress. We assume the frictional stress \( \tau \) obeys the composite rate and state friction law (Kato and Tullis, 2001) \[16\],

\[
\tau_i(t) = \tau_* + A_i (V_i(t)/V_c) + B_i \ln \frac{V_i(t)\Theta_i(t)}{L_i} + C_i \ln \frac{V_i(t)\Theta^*(i)}{L_i},
\]

where \( \Theta_i(t) \) is a state variable representing a contact state of the interface. \( A_i, B_i, \) and \( L_i \) are frictional parameters at cell \( i \). \( V_c \), which is the cut-off velocity for time-dependent healing, is \( 10^6 \) m/s (Kato and Tullis, 2001) \[16\]. \( V_* \) is the reference velocity, and \( \tau_* \) is the reference friction corresponding to the steady-state friction at \( V = V_* \); these values do not affect the simulated results. The steady-state frictional stress \( \tau^{SS}(t) \) is defined for \( d\theta_i(t)/dt = 0 \), and \( \tau^{SS}(t) = \tau_* + (A_i - B_i) \ln(V_i(t)/V_c) \) for \( V >> V_c \). When \( A_i - B_i < 0 \), the steady-state frictional stress decreases with an increase in slip rate (rate weakening), possibly leading to unstable seismic slip. On the other hand, in the region of \( A_i - B_i > 0 \), the stress increases with slip rate (rate strengthening) and aseismic slip can occur.

As the initial conditions, we uniformly set the slip velocities to \( V_i(0) = 0.9 V_{pl} \) under the steady state. Then, coupling the time derivatives of Eqs. (4) and (5) with Eq. (6), we numerically integrate these equations using a Runge-Kutta algorithm with adaptive step-size control (Press et al., 1996) \[17\].

3.2. Model settings

We set a flat interface in this study, although we use triangular cells for precisely discretizing complex three-dimensional interfaces. The plate interface is assumed to have sizes of 600 and 240 km in the strike and dip directions, respectively, and a dip angle of 10°. In addition, we set \( V_{pl} = 5 \) cm/year, \( G = 30 \) GPa, \( V_s = 3.27 \) km/s, and \( \xi = 1 \). Poisson’s rate is set to 0.25.
Frictional parameters $A$ and $B$ are distributed to construct the seismogenic zone with $A - B < 0$ at depths of 0–30 km, except for both side areas with $A - B > 0$, as shown in Fig. 1(a). Figure 1(b) shows the distribution of the frictional parameter of $L$, the characteristic slip distance, which is proportional to fracture energy and nucleation size. The two asperity regions with smaller $L$ values are set within the seismogenic zone.

We divide the plate interface into small square subfaults and then subdivide each subfault into two triangular ones to obtain $N$ cells. As noted in Rice (1993) [6], the cell size must be sufficiently smaller than the critical size for slip nucleation, i.e., proportional to $L/(B - A)$, for obtaining reliable results. We use several cell sizes; the largest size of a square cell is 3 km, whose ratio to the critical one of Rice (1993) [6] is 0.38. For $N$ cells, the operational count and memory size required for the matrix-vector multiplication in the derivative equation of Eq. (4) are both $O(N^3)$. To this, we apply the method of H-matrices to reduce the CPU-times and memory sizes in simulations.
Fig. 3. (a) Memory sizes of the H-matrices and (b) normalized CPU-times for 10,000 multiplications at each \(N\). Both are plotted on a log-log scale, and lines for \(O(N^2)\), \(O(N \log N)\), and \(O(N^3)\) are also plotted for reference. (c) Conceptual views of H-matrices for \(N = 128,000\) and 288,000 with \(\varepsilon = 10^{-4}\). At this range of \(N\), the rank \(k\) at the outermost submatrices does not exceed \(k_{\text{ACA}} = 100\).

We examine the performance of H-matrices in simulations by changing the number of cells, \(N\), in the fixed friction model. We start with a reference model where the size of a square subfault, \(s_o\), is 3 km and the number of subdivided triangular cells, \(N_o\), is 32,000. We then divide the subfault size, \(s\), by an integer number, \(m\), (namely, \(s = s_o/m\)) and increase the number of cells, \(N\) (namely \(N = N_o \times m^2\)) to investigate the effect of \(N\) on the CPU-time and memory size.

4. Results

The original simulation code RSGDX by Hori (2006) [1] consists of two programs. The first computes the \(N \times N\) SRFM in Eq. (4), and the second computes slip evolution at each cell. First, we compress SRFM and store it in H-matrix form implemented with the subroutine library of HLib (2009) [13]. We perform this in a single core because
the HLib software we use has not yet been parallelized. We then compute the matrix-vector multiplication with HLib subroutine libraries and simulate slip evolution at all cells. This part is parallelized with 64 cores in this study.

In this section, we present the effects of the control parameters included in H-matrix construction, as well as the number of cells, $N$, on the accuracy, CPU-time, and memory size. Before presenting the results with the H-matrices, we point out our finding that the total CPU-time required for a given simulation period in the same friction model with the original code is not $O(N^2)$ but $O(N^3)$, whereas the operational count of the multiplication is $O(N^2)$. This is because the system with smaller cell sizes becomes stiffer, which produces smaller time steps determined by the Runge-Kutta algorithm with adaptive step-size control (Press et al. 1996) [17], and increases the number of time steps. In fact, the number of time-steps increases linearly with $N$. This increase in the number of time steps is almost the same in the results with H-matrices. Accordingly, we discuss the normalized CPU-time, which is the time required for 10,000 matrix-vector multiplications.

As explained in section 2, the construction of H-matrices involves the parameters, $n_{\text{min}}$, $\eta$, $k_{\text{ACA}}$, $e_{\text{ACA}}$, and $\varepsilon$. $n_{\text{min}}$ and $\eta$ determine the structures of the H-matrices, and $e_{\text{ACA}}$, $\varepsilon$, and $k_{\text{ACA}}$ mainly determine the accuracy of the arithmetic operations with H-matrices. In simulations, we examined how the parameters control the memory size, accuracy, and CPU-time. In this paper, we present only the results focusing on the effects of $\varepsilon$ and $k_{\text{ACA}}$ and the $N$ dependencies of H-matrix performance. We set $n_{\text{min}} = 16$, $\eta = 2.0$, and $e_{\text{ACA}} = 0.9\varepsilon$, which are proper values determined from some examinations.

4.1. Effect of $\varepsilon$ in construction of H-matrices

Setting $N = 32,000$ and $k_{\text{ACA}} = 100$, we examine cases with $\varepsilon = 10^{-3}$, $10^{-4}$, and $10^{-5}$. The resultant memory sizes of the H-matrix are reduced from 8.19 to 0.15, 0.22 and 0.30 GB, and the normalized CPU times are reduced from 1,571 to 149, 173, and 240 s, respectively. Figures 2(a) and (b) show conceptual views of the H-matrices and the computed slip rate histories at the location shown in Fig. 1 for the three cases. The interaction between the two asperities produces a complex earthquake cycle. With a smaller $\varepsilon$, the rank at each submatrix becomes larger, as seen in Fig. 2(a). This leads to larger memory sizes and CPU-times, although greater accuracy is achieved. With a smaller $\varepsilon$, the slip rate histories come to converge to the original one as more accuracy is demanded. With respect to the cycle itself, the case with $\varepsilon = 10^{-3}$ is much different from the other cases. In Fig. 2(b), the plotted lines for the original case and for cases with $\varepsilon = 10^{-4}$ and $10^{-5}$ almost overlap, and the recurrence times are 1161.90, 1160.30, and 1161.88 years, respectively. In each case, this recurrence time does not appear to change after the cycle has become stable. Although the required accuracy depends on the purpose of the simulation, cases for which $\varepsilon$ is more than $10^{-4}$ maintain sufficient accuracy in this model.

4.2 Performance of H-matrices depends on the number of cells, $N$

With $k_{\text{ACA}} = 100$ and $\varepsilon = 10^{-3}$, $10^{-4}$, and $10^{-5}$, we examine cases in which $N$ ranges from 32,000 to 1,152,000. The memory size of the H-matrices and the normalized CPU-time for each case are shown in Figs. 3(a) and (b). When $\varepsilon = 10^{-4}$, the memory size increases almost linearly with $O(N)$ up to around $N = 128,000$. The normalized CPU-time also increases with an abrupt change around $N = 128,000$. These changes correspond to the increase in the rank of the outermost submatrices at values of $N$ larger than 288,000, as understood from the conceptual view of H-matrices for $N = 128,000$ and 288,000 in Fig. 3(c). Cases with other $\varepsilon$ values show the same tendency, although the rate of increase and the inflection point are different.

In order to suppress the rapid rise in CPU-times, we search a sufficient range of values for $k_{\text{ACA}}$, the specified maximum rank of the Rk-matrix. When the rank $k$ is suppressed by $k_{\text{ACA}}$, the constructed H-matrices no longer satisfy the accuracy condition provided by $e_{\text{ACA}}$. However, for larger values of $N$, the elements of an outermost submatrix tend to take small and similar values, and the rank $k$ is determined according to the relative error in the matrix. Then, suppressing the rank $k$ to some extent would not greatly affect the accuracy. Therefore, we examine cases of $N = 288,000$ for $\varepsilon = 10^{-4}$, setting $k_{\text{ACA}}$ as 10, 20, 40, 60, and 80. Figure 4(a) shows the simulated slip rate histories, at the time ranges of the first ruptures in the simulations starting from the same initial condition. The results are shown only for $k_{\text{ACA}} = 10$ and 20 when $N = 288,000$, compared to the case with $k_{\text{ACA}} = 100$ where the ranks of the outermost submatrices are not suppressed. The plotted lines almost overlap, except for the case with
Fig. 4. (a) Simulated slip velocity histories in the location of Fig.1(a), started from the same initial state, (b) memory size [GB], and (c) normalized CPU-time [10^3s], at each $k_{ACA}$ when $N = 288,000$ and $\varepsilon = 10^{-4}$. (b) and (c) are plotted on a log-log scale. Lines for $O(N)$, $O(N\log N)$, and $O(N^2)$ are also plotted for reference.

$k_{ACA} = 10$. The difference in the rupture occurrence times between the cases with $k_{ACA} = 20$ and $k_{ACA} = 100$ is 0.0008 year. This difference is smaller than 0.0237 year, which is the difference between for the case with $\varepsilon = 10^{-4}$ and $k_{ACA} = 100$ and for the original case without H-matrices, at the same time range when $N = 32,000$. We may then expect that the use of $k_{ACA} = 20$ maintains sufficient accuracy, compared to that of $k_{ACA} = 100$. Thus, simulations even with $k_{ACA} = 20$, where the other inner submatrices take similar ranks, give satisfactory results. In addition, when $\varepsilon = 10^{-3}$, $k_{ACA} = 20$ offers sufficient accuracy compared to the case without rank suppression (data not shown).

Next, with $k_{ACA} = 20$, we examine the cases for larger values of $N$. The memory sizes and normalized CPU-times, compared to those with $k_{ACA} = 100$, are shown in Fig. 4(b) and (c), respectively. As expected, the rates of
increase in memory size and normalized CPU-time are suppressed. The rate of memory increase approximates $O(N)$ for every range of $N$. Therefore, we expect the rate of increase for CPU-times to approximate $O(N)$ as well. However, the increase in CPU-time from $N = 128,000$ to $N = 288,000$ is much higher than $O(N)$. It is about $O(N) \sim O(N\log N)$ in the range of $N$ larger than 288,000. Although the higher rate of increase in CPU-time remains, the use of a smaller $k_{ACA}$ is effective, as it clearly suppresses the normalized CPU-time, while maintaining the accuracy of approximation compared to the case without rank suppression.

Although we show only the case with $\varepsilon = 10^{-4}$ in this study, it is expected that cases with other $\varepsilon$ values will exhibit the same tendency.

5. Discussion and Conclusion

In the $N \times N$ SRFM in simulations of earthquake cycles on a flat plate interface, the memory size of its H-matrix is $O(N)$. Our code for H-matrix construction implemented with HLib is not parallelized. For example, the actual available shared memory size is 28 GB in one of the systems in our university computer center, and we can only handle problems with $N$ values up to 1,900,000 for $\varepsilon = 10^{-4}$. To prevent such limitations on $N$, parallelization of the code is required.

We show that the rate of increase in CPU-time at $N$ values ranging from 128,000 to 288,000 becomes unexpectedly larger, even in the case of rank suppression at the outer submatrices. This larger rate of increase may come from reasons other than the rank increases of the outermost submatrices, such as the effect of communication and synchronization times between processors in parallel computations, or the condition of the computer. Although the reason for this rapid increase is unclear at present, we may need to optimize the part of our simulation code for parallelization.

As mentioned in the introduction, methods such as FFT and FMM have been tested to make the simulation faster and memory sizes smaller. We reemphasize that the method of H-matrices has an advantage over these methods. Namely, the necessary condition for applying H-matrices is that the slip response function decays with the distance between receiver and source cells. We can treat any configuration of interface in a semi-infinite homogeneous elastic medium by available codes for constructing slip response functions. In this study, we examine only the case for a plate interface with a flat plane. It would be necessary to examine the case for the three-dimensionally curved interface, in which the H-matrices representation would be somewhat changed. Furthermore, we can extend the studies from a homogeneous to a heterogeneous elastic medium, which would be significant at subduction zones in particular. This is because we can use the numerical values of the slip response function computed with codes such as a finite element method, which can deal with any heterogeneous media.

Finally, we conclude that the application of H-matrices to the multiplication of SRFM and SDRV in the case of $N$ subdivided cells reduces the memory size to $O(N)$ and the CPU-time to $O(N)$ in the range of $N$ values less than about $10^5$, depending on the $\varepsilon$, and to $O(N) \sim O(N\log N)$ at larger $N$ values, although some parameters included in construction of H-matrices should be assigned properly. Thus, the application of H-matrices enables us to greatly reduce the CPU-time and memory size, which realizes large- and multi-scale quasi-dynamic earth cycle simulations with a number of cells on the order of millions. It also enables predictive simulations for earthquake occurrences with the estimation of frictional parameters and initial conditions of simulation parameters by data assimilation methods.

Acknowledgements

We are grateful to editors for inviting us to submit this paper and three anonymous reviewers for useful comments to improve this paper. The computations in this study were carried out by the HX600 cluster system in the Academic Center for Computing and Media Studies, Kyoto University, and by the HA8000 cluster system in the Information Technology Center, University of Tokyo. This study was financially supported by the MEXT projects of “New Research Project for the Evaluation of Seismic Linkage around the Nankai Trough,” and of “Observation and Research Program for Prediction of Earthquakes and Volcanic Eruptions.”
References

1. T. Hori, J. Earth. Sim., 5, 8-19 (2006)
2. N. Kato, J. Geophys. Res., 113, B06302 (2008)
3. G. Hillers, P. M. Mai, Y. Ben-Zion and J. -P. Ampuero, Geophys. J. Int., 169, 515-533 (2007)
4. H. Hirose, and K. Obara, Earth Planets Space, 57, 961-972 (2005)
5. K. Ariyoshi, T. Hori, J. –P. Ampuero, Y. Kaneda, T. Matsuzawa, R. Hino et. al., Gondwana Res., 16, 534-544 (2009)
6. J. R. Rice, J. Geophys. Res., 98, B6, 9885-9907 (1993)
7. C. Wunsch, Discrete Inverse and State Estimation Problems. With Geophysical Fluid Applications, *Cambridge Univ. Press*, pp.371 (2006)
8. T. E. Tullis, J. Salmon, N. Kato, and M. Warren, 1999 AGU Fall Meeting Suppl., 80, F924, (1999)
9. K. Hirahara, N. Mitsui, and T. Hori, 2009 AGU Fall meeting, T23C-1929,(2009)
10. M. Ohtani and K. Hirahara, Japan Geoscience Union Meeting, SSS027-P18, (2010)
11. W. Hackbusch, Computing, 62, 89-108, (1999)
12. S. Börm, L. Grasedyck and W. Hackbusch, Hierarchical Matrices, Lecture Note, Max-Planck-Institut fur Mathematik, pp.1171 (2006)
13. HLib, Max-Planck-Institute, 2009, http://www/hlib.org/hlib.html
14. M. Bebendorf, Numer. Math., 64, 565-589 (2000)
15. M. Comninou and J. Dundurs, J. Elasticity, 5, 203-216 (1975)
16. N. Kato and T. E. Tullis, Geophys. Res. Lett., 28, 1103-1106, (2001)
17. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes in Fortran 77, *Cambridge University Press*, New York, pp.963 (1996)