

A 1.8 trillion degrees-of-freedom, 1.24 petaflops global seismic wave simulation on the K computer

The International Journal of High Performance Computing Applications
1–12
© The Author(s) 2016
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/1094342016632596
hpc.sagepub.com


**Seiji Tsuboi¹, Kazuto Ando¹, Takayuki Miyoshi¹, Daniel Peter²,
Dimitri Komatitsch³ and Jeroen Tromp⁴**

Abstract

We present high-performance simulations of global seismic wave propagation with an unprecedented accuracy of 1.2 s seismic period for a realistic three-dimensional Earth model using the spectral element method on the K computer. Our seismic simulations use a total of 665.2 billion grid points and resolve 1.8 trillion degrees of freedom. To realize these large-scale computations, we optimize a widely used community software code to efficiently address all hardware parallelization, especially thread-level parallelization to solve the bottleneck of memory usage for coarse-grained parallelization. The new code exhibits excellent strong scaling for the time stepping loop, that is, parallel efficiency on 82,134 nodes relative to 36,504 nodes is 99.54%. Sustained performance of these computations on the K computer is 1.24 petaflops, which is 11.84% of its peak performance. The obtained seismograms with an accuracy of 1.2 s for the entire globe should help us to better understand rupture mechanisms of devastating earthquakes.

Keywords

Seismic wave propagation, spectral element method, numerical seismograms, K computer

I Introduction

Destructive earthquakes are caused by large-scale ruptures inside the Earth, which break along hundreds of kilometers of a geological fault and generate seismic waves shaking ground and buildings. Because large earthquakes cause serious damage to human societies, the study of earthquake source mechanisms is a crucial topic in seismology. As electromagnetic waves do not penetrate far into the Earth's solid rock interior, the use of seismic waves is almost the only way to probe the inside of our planet at high resolution. In this regard, modeling seismic waves excited by earthquakes is a fundamental scientific problem.

Seismic waves are simulated by solving the equations of motion for an elastic or viscoelastic body. There exists an analytical solution only if the Earth is considered to be a perfect sphere and the Earth model spherically symmetric. However, the real Earth exhibits various significant deviations from spherical symmetry, such as ellipticity as well as heterogeneous structures in the crust and mantle, which makes it impossible to obtain analytical solutions. Traditionally, assuming that the Earth can be represented by a spherically layered structure, normal-mode summation algorithms

have been used to calculate seismograms (Dahlen and Tromp, 1998), but these algorithms are typically accurate down to seismic periods of 8 s only, that is, 1/8th of a Hertz. However, in modern seismology, the community is interested in seismic waves that are characterized by a dominant period of about 1 s, that is, eight times smaller. Therefore, it is desirable to calculate theoretical seismograms with a precision of 1 s for realistic three-dimensional (3-D) Earth models (Heinecke et al., 2014; Ichimura et al., 2014).

The use of the spectral element method (SEM) for numerical modeling of seismic wave propagation in realistic 3-D models at the scale of the full Earth has

¹Japan Agency for Marine-Earth Science and Technology (JAMSTEC), Kanagawa, Japan

²Extreme Computing Research Center, King Abdullah University of Science & Technology (KAUST), Thuwal, Kingdom of Saudi Arabia

³Laboratory of Mechanics and Acoustics (LMA), CNRS UPR 7051, Aix-Marseille University, Centrale Marseille, France

⁴Princeton University, Princeton, NJ, USA

Corresponding author:

Seiji Tsuboi, Japan Agency for Marine-Earth Science and Technology (JAMSTEC), 3173-25 Showa-machi, Kanazawa-ku, Yokohama, Japan.
Email: tsuboi@jamstec.go.jp

been prevailing in recent years. The SEM is an optimized high-order version of the finite element method that is very accurate for linear hyperbolic problems such as wave propagation. In addition, its mass matrix is perfectly diagonal by construction, which makes it favorable to implement on parallel systems because no linear system needs to be inverted. The 3-D SEM was first used in seismology for local and regional simulations (Faccioli et al., 1997; Komatitsch, 1997; Komatitsch and Vilotte, 1998) and adapted to wave propagation at the scale of the full Earth (Carrington et al., 2008; Komatitsch and Tromp, 2002a, 2002b; Komatitsch et al., 2005; Rietmann et al., 2012; Tsuboi et al., 2003). Here we use the 3-D SEM package SPECFEM3D_GLOBE to simulate global seismic wave propagation with an accuracy of about 1 s for a realistic Earth model on the K computer.

The focus of this article is on code optimizations of the SEM software package to obtain large-scale, high-resolution global seismic wave propagation simulations on a high-performance computing (HPC) system, that is, on the K computer located in Japan. In the following, we first highlight the software code used in this study. Section 3 describes critical code optimizations to reach high-performance simulations on the K computer, which are generally important to other large HPC architectures as well. Results are presented for computational performance of weak and strong scaling to analyze the optimized code.

In section 6, we finally compare the high-resolution numerical seismograms obtained for the Tohoku earthquake of March 11, 2011, with observed seismograms recorded in the field for that devastating earthquake. We will confirm that the high-precision theoretical seismograms explain observations much better than previously computed lower-precision numerical seismograms. This will contribute to detailed studies of rupture nucleation processes of large earthquakes.

2 Global seismic wave propagation

For global seismic wave propagation simulations, the SEM code SPECFEM3D_GLOBE is currently the state-of-the-art in software development. The SEM solves the equation of motion for seismic waves (Dahlen and Tromp, 1998)

$$\rho \partial_t^2 s = \nabla T + f \quad (1)$$

where ρ is the density, s is the displacement vector, T is the stress tensor, and the f is the source force, which is represented by a moment tensor in the case of an earthquake source. To incorporate attenuation, it is common in seismology to use a series of standard linear solids (Dahlen and Tromp, 1998). To solve the equations of motion, the software meshes the sphere using hexahedra only, based upon an analytical mapping from the

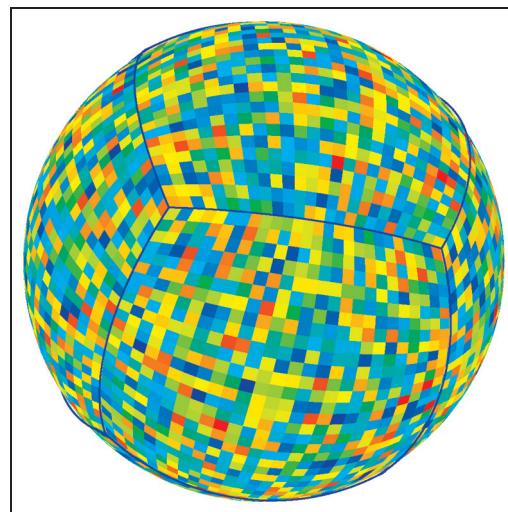


Figure 1. A global view of the mesh used at the surface in the SEM. Each of the six sides of the ‘cubed sphere’ mesh is divided in 26×26 slices, shown in different colors, for a total of 4056 slices. SEM: spectral element method.

six sides of a unit cube to a six-block decomposition of the surface of the sphere, which is called the cubed sphere (Komatitsch and Tromp, 2002a; Ronchi et al., 1996; Sadourny, 1972). This cubed-sphere mapping splits the globe into six mesh chunks, each of which is further subdivided in terms of n^2 mesh slices, for a total of $6 \times n^2$ slices. Figure 1 illustrates a global view of the mesh at the surface.

In 2003, we used SPECFEM3D_GLOBE on the Earth Simulator in Japan, which was at that time the world’s fastest supercomputer, to simulate global seismic wave propagation in a realistic 3-D Earth model with an accuracy of 5 s (Komatitsch et al., 2003). For this purpose, we used $18 \times 18 \times 6 = 1944$ slices, which used 38% of the Earth Simulator (1944 processors, i.e., 243 nodes of 640) allocating one slice per processor. The total number of spectral elements in this mesh was 82 million, which corresponded to a total of 5.47 billion global grid points, since each spectral element contains $5 \times 5 \times 5 = 125$ grid points, but with points on its faces and edges shared by neighboring elements. This in turn corresponded to 14.6 billion degrees of freedom. The numerical seismograms were accurate down to a minimum seismic period of 5 s. The calculation required 2.5 terabytes of memory and the sustained performance level obtained was 5 teraflops.

In such numerical methods based on a spatial grid and an explicit, conditionally stable time scheme, when one needs to improve the accuracy of the calculations by a factor of 2 one increases the numerical cost by a factor of 16 because the mesh needs to be doubled in the three directions of space, and in addition the time step needs to be divided by two for the explicit time integration to remain stable. This means that we need to have $16 \times 16 \times 1944 = 497,664$ processors to

obtain seismograms with an accuracy of about 1 s. This ultimate resolution represents the upper physical limit of seismic signals recorded at teleseismic distances due to Earth's absorption of higher frequency energy.

Our target simulations will be using 82,134 nodes (657,072 cores) of the K computer in Kobe, Japan, to perform a computation of seismograms with an accuracy of 1.2 s for a realistic 3-D Earth model. We therefore have to divide the Earth into $117 \times 117 \times 6 = 82,134$ mesh slices, and the total memory required is 637 terabytes. The total number of grid points to construct the mesh is 665.2 billion and the total number of degrees of freedom is 1.8 trillion. To achieve such large-scale simulations, we will show which bottlenecks to solve and how to increase the computational efficiency of the code.

3 Code optimizations

3.1 Improvement and optimization

The following challenges had to be faced for our large-scale simulations of global seismic wave propagation on the K computer:

- (i) *Communication memory usage*: Coarse-grained parallelization implemented by Message Passing Interface(MPI) requires increasing memory usage for simulations on higher numbers of computational nodes. For large-scale simulations, this becomes a memory bottleneck, which inhibits efficient scaling.
- (ii) *Cache efficiency*: The most significant contribution to the run-time performance is the calculation of elastic forces and stresses in the Earth. These computations have relatively low arithmetic intensity, that is, they are memory bound. To increase peak performance, one has to decrease the amount of data accesses compared to the number of floating-point operations.
- (iii) *Code vectorization*: Computations need to make efficient use of code vectorization to perform multiple data operations with a single instruction set. For example, it is important to reduce code branching within do-loops to achieve better vectorization and peak performance.

The K computer is a massively parallel supercomputer developed by RIKEN (Japan), which became fully operational in 2012 (Hasegawa et al., 2011; Ishiyama et al., 2012). It consists of 82,944 SPARC64 VIIIfx CPUs. Each CPU has eight cores with a 2 GHz clock frequency and a 6 megabyte level 2 cache shared by all cores. The peak performance of each CPU is 128 gigaflops and the total peak performance of the whole supercomputer is 10.6 petaflops. Each core has an level 1 data (L1D) cache of 32 KB and four floating-point multiply-

and-add units, with two of them operated through single instruction multiple data (SIMD) instruction.

The parallel programming model of the K computer has three levels; the first one regarding SIMD processing on each core, the second one for thread programming within the CPU by using OpenMP directives or automatic parallelization, and the third one between multiple CPUs based on MPI distributed memory parallel programming. We present here our improvements made to the SEM code SPECFEM3D_GLOBE on the K computer to address all three parallelization levels. We have carefully examined how to optimize the source code on the K computer, so that we can fully take advantage of its architecture. In the following, we show that it is crucial to use these features efficiently for the execution of large-scale numerical simulations on the supercomputer.

3.2 Serial optimization

We first checked the computational cost of all subroutines of the SPECFEM3D_GLOBE package to identify what subroutines we needed to optimize to use both SIMD and the L1 cache efficiently in each core. We found that the cost of computing elastic forces and the stress tensor in each spectral element in the solid regions of the Earth dominates, representing 89% of the total cost for each time step of the time-stepping loop. This subroutine contains a large loop that consists of 843 lines and involves many nested “if” statements, which makes compiler optimization for instruction scheduling very difficult. Therefore, wait time for floating-point arithmetic is long and the ratio of SIMD instructions becomes low, even though it is critical to achieve good performance on the K computer. Moreover, data access efficiency is also poor, thus wait time for cache access associated with floating-point load instructions also becomes long. Figure 2 illustrates examples of optimization, which we have included in the modified code.

This major loop contains many three-level nested loops in which each level loop has a small iteration count of 5. This comes from the fact that the spectral element method uses a mesh of hexahedral finite elements on which the wave field is interpolated by Lagrange polynomials of degree $N = 4$ at Gauss–Lobatto–Legendre (GLL) integration points (Komatitsch & Tromp, 2002a, 2002b), leading to $N + 1 = 5$ GLL points in each of the three spatial directions of any given spectral element of the mesh. We conducted optimizations for the largest nested loop to promote compiler optimization for instruction scheduling and to improve the ratio of SIMD instructions. We divided this loop into six small loops and moved the “if” statements outside of these loops. We also transformed the three-level nested loops into a single-level loop so as to make the iteration count larger,

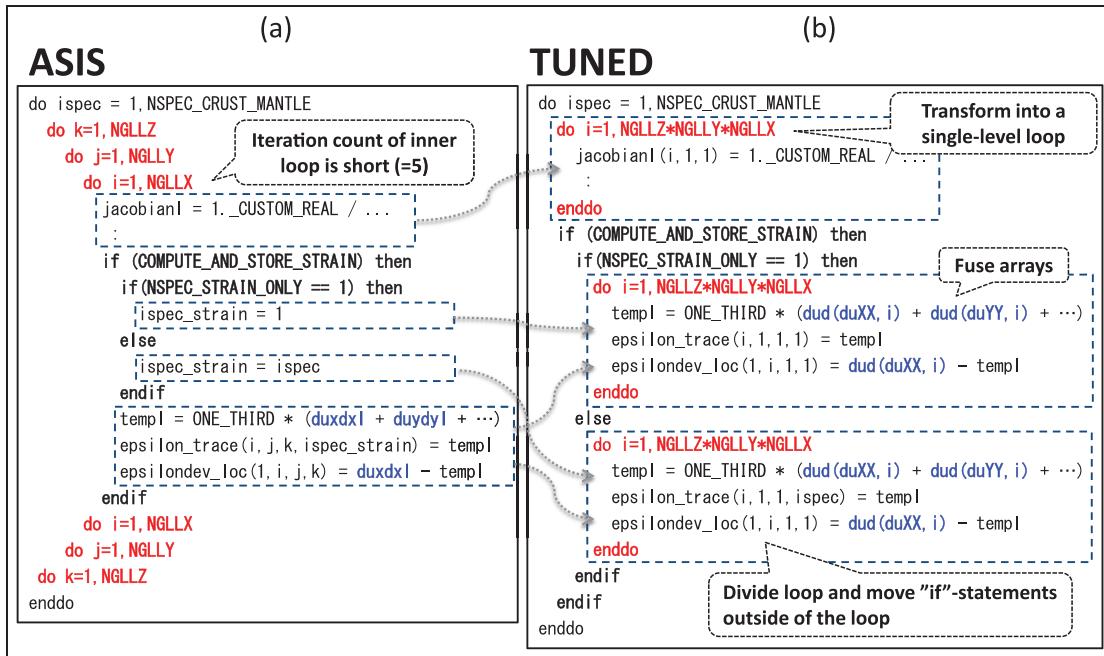


Figure 2. Examples of optimization, such as transformation and splitting of do-loop.

increasing it to $5 \times 5 \times 5 = 125$, which helps the compiler to vectorize the loop more efficiently.

Furthermore, to improve data access efficiency, we also conducted optimizations by fusing the nine individual arrays that contain physical parameters into a single array. This allowed us to reduce the number of streams and replace software prefetching, which causes overhead of instruction processing with high-speed hardware prefetching.

3.3 Thread-level parallelization and optimization

The original SPECFEM3D_GLOBE code uses a flat-MPI scheme (i.e., one MPI thread per processor core, with no OpenMP used) for parallelization over compute nodes. We may have continued to use flat-MPI on the K computer as well to achieve parallelization over cores within the CPU and over the CPU nodes because with that model in the case of moderate-size runs we reached a relatively good peak performance ratio (of about 16%) in the case of 12,000 CPU level parallelization (i.e., flat-MPI with 96,000 cores). However, we found that the memory usage for the MPI package becomes critical if one intends to use a much larger number of MPI processes. Specifically, if we intend to perform a huge calculation accurate down to a seismic period of about 1 s, which as mentioned in the introduction is our final goal, we need to use almost all the compute nodes of the K computer. Thus, the memory requirement becomes a bottleneck to perform large-scale simulations and we then need to use thread-level parallelization for much larger scale simulations of seismic wave propagation.

We have focused our thread-level parallelization only on the most time consuming subroutine. As discussed above, the main loop of this subroutine contains many “if” statements. In addition, that loop contains indirect addressing, and the same element of an array may be written to for different loop indices, that is, that loop contains dependencies. Consequently, most of the loop cannot be parallelized by the compiler. To address this problem, we moved the section that contains the dependencies to outside of the loop that computes the elastic forces and the stress. We then parallelized this loop with OpenMP directives. Moreover, using a reverse resolution array that contains the inverted reference relationship, we also managed to parallelize at the thread level the operation that contains the dependency. Regarding OpenMP scheduling, we used a cyclic distribution to balance the operation cost of each thread.

4 Optimization results

4.1 Results of serial optimization

We use the “Precision PA” performance measurement tool of the K computer to examine the results of the optimizations that we have performed above. Measured performance values and elapsed computation time for the tuned subroutine, indicated as “TUNED”, are shown in Table 1 and Figure 3 with the same values for the original code, indicated as “ASIS”. The size of the simulation for this case is NPROC = 2 and NEX = 64, that is, 24 slices and 24 cores in total. The ratio of SIMD instructions increases from 17.37% to 52.93%, and wait time for floating-point arithmetic

Table 1. Performance values for the subroutine that computes elastic forces and stresses in the code.

Performance value	ASIS	TUNED
Ratio of SIMD instructions	17.37%	52.93%
L1D cache miss rate	2.50%	1.37%
L1D cache miss demand rate	65.96%	46.39%
Wait time for floating-point arithmetic (s)	48.40	18.95
Wait time for floating-point load cache access (s)	69.44	41.30
Elapsed time (s)	310.20	200.20
Peak performance ratio	15.79%	20.98%

SIMD: single instruction multiple data; L1D: level 1 data.

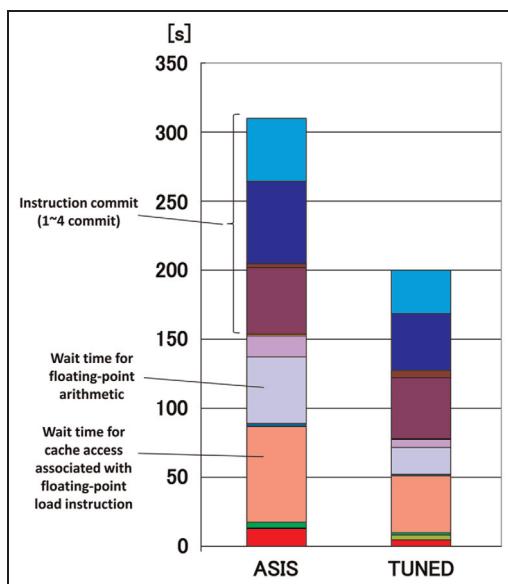


Figure 3. Breakdown of elapsed time in terms of CPU cycles for the loop of the code that computes elastic forces and stresses.

is reduced from 48.40 s to 18.95 s. Furthermore, wait time for cache access associated with floating-point load instructions is also reduced from 69.44 s to 41.30 s. Finally, elapsed time for this loop is reduced from 310.20 s to 200.20 s, and the peak performance ratio of floating-point arithmetic increases from 15.79% to 20.98%.

Besides this, in the original code, the L1D cache miss rate was 2.5% and the L1D miss demand rate was 65.96%. In the tuned code, the L1D miss rate is reduced to 1.37% and the L1D miss demand rate is also reduced to 46.39%, which shows that the utilization efficiency of the L1 cache as well as prefetching are significantly improved.

4.2 Results of thread-level parallelization and optimization

We have used the “Instant Profiler” performance measurement tool provided by the K computer to examine

Table 2. Speedup for each number of threads and parallel efficiency for the loop of the code that computes the elastic forces and stresses.

	ASIS	TUNED
Speedup (1 thread)	1.00	1.00
Speedup (2 threads)	1.00	1.99
Speedup (4 threads)	1.01	3.96
Speedup (8 threads)	1.02	7.89
Parallel efficiency (8 threads)	2.2%	99.8%

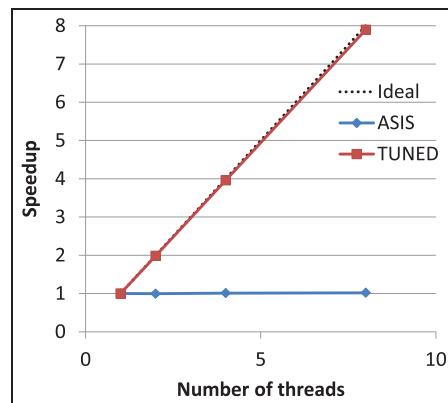


Figure 4. Thread scalability of the loop of the code that computes the elastic forces and stresses.

the results of our optimization. The results of speedup and of improvement in parallel efficiency are shown in Table 2 and in Figure 4 for the original code, indicated as ASIS, and for the tuned code, indicated as TUNED. When we used automatic parallelization by the compiler for the ASIS code, even though we used eight threads we achieved only a 1.02 times speedup relative to one thread. With our optimization based on OpenMP directives, we achieved a 7.89 times speedup in the TUNED code. In terms of parallel efficiency, this turns to 2.2% in the ASIS code and 99.8% in the TUNED code, that is, almost perfect efficiency. We can thus conclude that no additional improvement of the source code is then needed.

4.3 Improvement of memory usage

We have compared memory usage of flat-MPI and hybrid parallel execution based on a strong scaling test. In the case of a number of spectral elements equal to 1152 in each horizontal direction of each of the six mesh chunks, memory usage per node was 9.61 GB in the case of flat-MPI execution on 15,552 nodes, but 1.24 GB in the case of hybrid parallel execution on 13,824 nodes. Consequently, in hybrid parallel execution, memory usage was reduced to 12.9% of flat-MPI

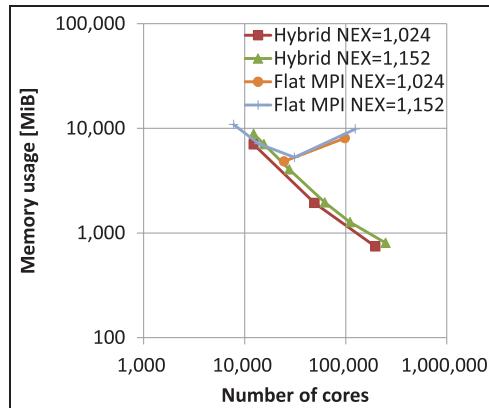


Figure 5. Strong scaling of memory usage per node.

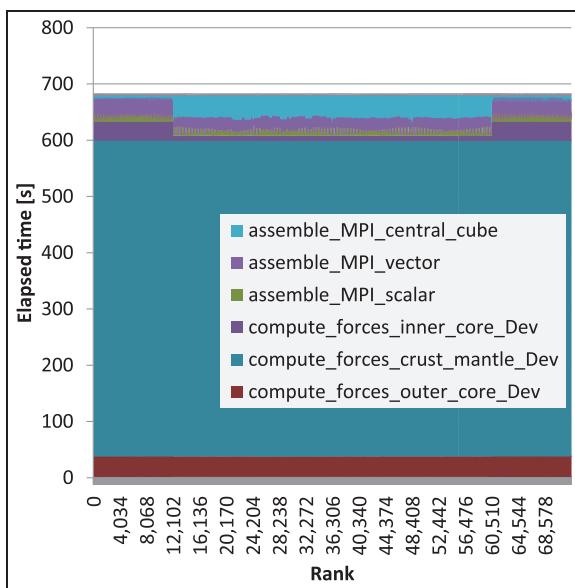


Figure 6. Load distribution of each program section between the MPI processes (using 72,600 processes) in the main time step loop.

execution (Figure 5). This means that the memory bottleneck issue has been solved.

4.4 Load balancing between the MPI processes in the case of a massively parallel execution

We also analyzed load balancing among the compute nodes in our tuned code. We used the MPI_WTIME routine to measure that. Figure 6 shows load balancing between the different MPI processes of each section of the program in the case of 72,600 compute nodes. The cost of arithmetic operations dominates and represents 83% of the total cost; that cost is also well balanced. The communication cost is only 10% of

the total cost because the SPECFEM3D_GLOBE code uses non-blocking MPI and thus communication is efficiently overlapped with calculation. Thus, the communication cost does not cause performance degradation, even in the case of 72,600 compute nodes. This demonstrates that MPI parallelization of the original code is already efficient and well optimized in this respect.

5 Performance results

The shortest seismic period T_s of numerical seismograms computed by the SPECFEM3D_GLOBE package is determined empirically (but quite accurately) by the following equation (SPECFEM3D, 2014):

$$T_s = \left(\frac{256}{N_\xi^{\text{elem}}} \right) \times 17, \quad (2)$$

where N_ξ^{elem} is the number of spectral elements along each side of each of the six mesh chunks in the cubed sphere. For reasons related to the topological structure of the mesh, N_ξ^{elem} must be a multiple of 16 and also of $8 \times c \times N_\xi^{\text{proc}}$, where c is the positive integer and N_ξ^{proc} is the number of CPUs (or equivalently of mesh slices, in the case of a flat-MPI model) along each direction of each chunk of the cubed sphere. Since the total number of CPUs required is $N_\xi^{\text{proc}} \times N_\xi^{\text{proc}} \times 6$, we should determine the N_ξ^{proc} value that gives a number closest to the total number of CPUs of the K computer, which is 82,944. We thus decided to use $N_\xi^{\text{proc}} = 117$, so that the mesh contains $117 \times 117 \times 6 = 82,134$ slices in total. For the minimum seismic period T_s to be close to 1 we chose $N_\xi^{\text{elem}} = 117 \times 16 \times 2 = 3744$, where $T_s = 1.16$ s. Using this number of slices to construct the mesh, the total memory required is 637 terabytes (measured value based on the K computer system report), the total number of grid points of the whole mesh is 665.2 billions, and the total number of degrees of freedom is 1.8 trillion.

5.1 Strong scaling and arithmetic performance for massively parallel execution

As shown in the previous section, the number of spectral elements along one side of a chunk N_ξ^{elem} determines the seismic resolution of the problem. For $N_\xi^{\text{elem}} = 3744$, we may have at least these two different sets of values giving the same total:

$$N_\xi^{\text{elem}} = 117 \times 16 \times 2 = 78 \times 16 \times 3 = 3744 \quad (3)$$

We can thus solve the same problem using a second experimental setup containing $78 \times 78 \times 6 = 36,504$ slices or nodes if the memory available per node is

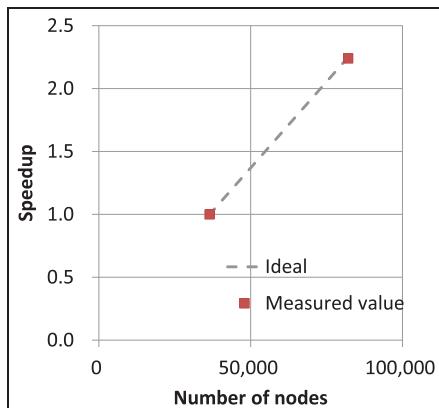


Figure 7. Strong scaling for the main time step loop.

sufficiently large, which is the case here. We thus measured strong scaling performance for these two different setups and show the result in Figure 7 for the main time-stepping loop of the code, when using 82,134 nodes (657,072 processor cores) in the first case and 36,504 nodes (292,032 processor cores) in the second case. The speedup achieved is 99.54%, that is, strong scaling is almost perfect. Finally, we achieved 11.84% of peak performance ratio and 1.24 PFLOPS of arithmetic performance, as shown in Table 3. Total memory and memory per node used for 36,504 nodes are 470 terabytes and 12.88 gigabytes respectively. Total memory and memory per node used for 82,134 nodes are 637 terabytes and 7.75 gigabytes, respectively. The cache size of the K computer is 6 megabytes per node, which is shared by eight cores. In case of SPECFEM3D_GLOBE, the most time-consuming computation is performed for arrays with the size of $5 \times 5 \times 5 = 125$. This array is used for each spectral element inside each slice. Besides, some of other arrays have a size of 125 multiplied by the number of elements per slice, which is about 285,000 elements for 36,504 nodes and 127,000 elements for 82,134 nodes. Then, the total amount of memory used for these arrays is estimated as 142 megabytes for 36,504 nodes and 63 megabytes for 82,134 nodes, which cannot be stored in the 6 megabytes cache in any case. Therefore, we assume that there are no significant differences with respect to the usage of cache between 36,504 nodes and 82,134 nodes and they should not affect the strong scaling estimation.

5.2 Weak scaling and arithmetic performance for massively parallel execution

We have measured weak scaling for our tuned code. Let us mention that our measurement for weak scaling is an approximate value because due to the cubed-sphere

Table 3. Performance values of the main time-stepping loop when using simulations with two different number of nodes.

Performance	Simulations	
	36,504 nodes	82,134 nodes
Elapsed time (s)	826.03	368.81
Speedup (relative to 36,504 nodes)	–	2.24
Strong scaling (relative to 36,504 nodes)	–	99.54%
Peak performance ratio	11.89%	11.84%
Arithmetic performance	555 TFLOPS	1.24 PFLOPS

TFLOPS: Teraflops; PFLOPS: Petaflops.

Table 4. Performance values of the main time-stepping loop in the weak scaling test.

Number of nodes	Weak scaling*	Peak performance ratio	Arithmetic performance (GFLOPS)
24	100.0%	12.28%	377
216	99.8%	12.25%	3400
1944	95.4%	11.71%	29,100
5400	93.6%	11.49%	79,400
7776	95.6%	11.74%	116,800
17,496	97.1%	11.92%	267,000
31,104	97.8%	12.01%	478,000
72,600	97.9%	12.02%	1,120,000

*Relative to 24 nodes; GFLOPS: Gigaflops.

meshing it is difficult to set the exact same number of elements for each slice when measuring performance for different numbers of computational nodes. Thus, the computational load varies slightly for simulations using different number of nodes. Since these variations are small, we do not analytically correct our measurement results but prefer to state the original performance results. Table 4 and Figure 8 represent the weak scaling measurements based on FLOPS relative to 24 nodes. Although these measurements show approximate values, it is clear that weak scaling for the tuned code is excellent.

6 Realistic earthquake simulation and discussion

6.1 Tohoku earthquake fault rupture simulation

To perform a realistic earthquake simulation, we have selected the magnitude 9.0 March 11, 2011 Tohoku earthquake, which led to the disastrous tsunami in the Fukushima region of Japan. Because of the size and

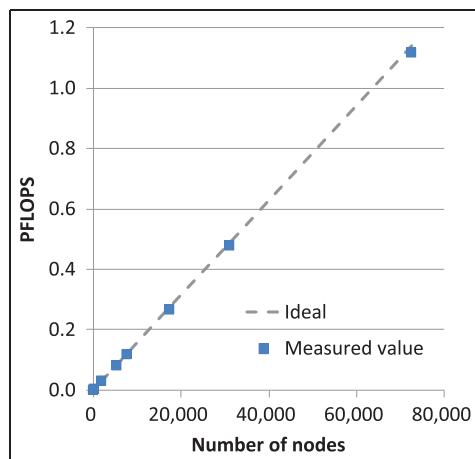


Figure 8. Weak scaling for the main time step loop.

proximity to the coastal areas of this earthquake, it caused serious damage to the coastal areas of northern part of Japan, especially due to extremely large tsunamis. Many geophysical studies have been devoted to the rupture mechanism of this event, using various types of observations such as seismic waves, crustal movement, and tsunami heights (e.g., Lee et al., 2011).

Here we use the earthquake rupture model of Tsuboi and Nakamura (2013) to calculate numerical seismograms for a realistic Earth model using our tuned SEM code on the K computer. Teleseismic waves were used to establish the seismic rupture model for which we compute numerical seismograms, accurate down to a minimum seismic period of 1.2 s, at seismographic recording stations in the region. We then compare them with the real seismograms recorded during the earthquake.

The rupture model represents the earthquake based on 597 sub-events located along a 500-km earthquake fault. There is general consent in the geological community that the earthquake fault responsible for this earthquake was almost 500-km long, and that the maximum slip along the fault was about 50 m. Our chosen model agrees with other studies regarding these two points. Figure 9 shows the comparison of numerical seismograms with the real recordings from several seismographic stations: Incheon in South Korea (INCN, located at an epicentral distance of 13.9 degrees from the earthquake fault), Matsushiro in Japan (MAJO, epicentral distance 5.6 degrees), Taipei in Taiwan (TATO, epicentral distance 23.9 degrees), and Yuzhno-Sakhalinsk in Russia (YSS, epicentral distance 7.5 degrees).

The total physical duration of the numerical seismograms is 7 min, and it required 6 h of computational time on the K computer to compute them using 43,000 time steps. Figure 9 shows the first 100 s of the earthquake rupture, because the real observed seismograms saturated some recording stations in the field due to the extremely large size of this earthquake. The numerical seismograms explain the observations fairly well, which

demonstrates that the earthquake rupture model that we use grasps the rupture characteristics well. Sustained performance of these large-scale computations is 1.24 petaflops, which is 11.84% of the peak performance of the supercomputer. To the best of our knowledge, this represents the largest global seismic wave propagation simulation ever conducted.

We also compare our numerical seismograms accurate down to a minimum seismic period of 1.2 s with lower precision numerical seismograms accurate down to a minimum seismic period of 5 s. This is to see if going to higher resolution simulations significantly improves the fit to real data recorded in the field, thus justifying such large-scale simulations. Figure 10 shows the comparison of longer period calculations with our shorter period calculations as well as with the real seismograms from the field. The agreement is significantly improved, especially for the very early part of the rupture. For example, at station INCN and MAJO, the slope of the first arrival P wave of 1.2 s accuracy numerical seismograms agrees well with that of the observed seismograms. This is due to the fact that the higher frequency component of the 1.2 s accuracy numerical seismograms is larger than that of the 5 s accuracy seismograms. It is well known that the slope of the first arrival P wave is proportional to the stress drop of the earthquake rupture, which is of high importance from the geophysical point of view as it helps to better understand the rupture initiation process of this extraordinary large earthquake.

6.2 Discussion

In addition to studying large earthquake rupture mechanisms, improved algorithms will allow us to perform large-scale numerical simulations of seismic waves to study the Earth's internal structure accurately. Owing to recent advances in numerical simulations of seismic waves in the context of imaging problems, that is, the so-called inverse problems, it is feasible to use seismic waveforms to invert for (i.e., to image) the seismic wave speed structure inside the Earth (Monteiller et al., 2015; Tromp et al., 2005, 2008). We have thus started to image the seismic wave speed structure underneath the Kanto basin, which surrounds the metropolitan Tokyo area, using our new software on the Kcomputer (Tsuboi et al., 2014) (Figure 11). The goal of this ongoing study is to better understand seismic activity around the Tokyo area and improve estimates of seismic hazard in this densely populated area.

In future work, we would also like to further study the nucleation and rupture process of such large and destructive earthquakes, as well as the remaining stress conditions after the event. It is known that the devastating 2011 Tohoku earthquake was preceded by a smaller, magnitude 7.3 foreshock 2 days before the

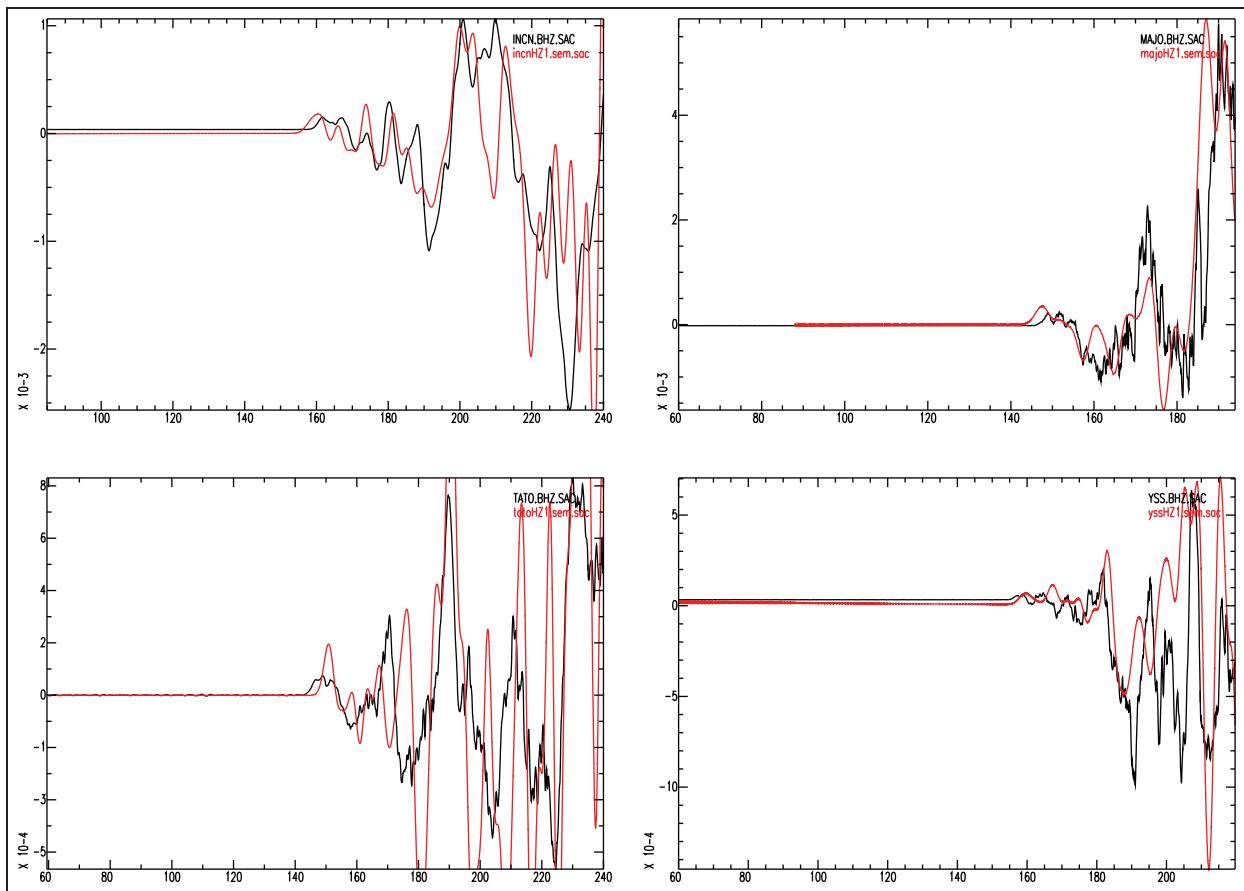


Figure 9. Comparison of numerical seismograms (red curves) with real recordings from the field (black curves) for seismographic recording stations: Incheon, South Korea (INCN), Matsushiro, Japan (MAJO), Taipei, Taiwan (TATO), and Yuzhno-Sakhalinsk, Russia (YSS), from top to bottom. Shown is the vertical component of the velocity vector of broadband seismograms. The horizontal axis is time in seconds and the vertical axis is ground velocity in m/s. The origin of the horizontal axis is the earthquake origin time.

main shock. From a geophysical point of view, it is very interesting but difficult to answer the question of whether one could tell that there could be a magnitude 9 earthquake occurring in the future after having recorded a magnitude 7.3 foreshock, as well as explaining why the March 9 earthquake with magnitude 7.3 ended but the March 11 earthquake then became one of the largest earthquakes ever recorded, with a magnitude of 9.0. To address these questions, it would be essential to get numerical seismograms accurate down to the shortest seismic period of 1 s at epicentral distances of about 30–40 degrees from the source, where pressure waves observed are less affected by the very heterogeneous Earth's shallow crustal structure. This would require calculating seismograms with a total duration of at least 10 min, thus requiring longer time simulations than achieved here.

7 Conclusions

The computations of numerical seismograms accurate down to the shortest seismic period of 1 s that we have

presented achieve an unprecedented performance level of 1.24 petaflops on the K computer located in Kobe, Japan, which corresponds to 11.84% of its peak performance. To accomplish such large-scale simulations, we successfully optimized the original spectral element software package SPECFEM3D_GLOBE, addressing all levels of hardware parallelization. The fact that the original code uses a flat-MPI scheme to implement coarse-grained parallelization over computational nodes becomes an important bottleneck to overcome at such high resolution. We thus had to resort to thread-level parallelization based on OpenMP in addition to MPI in order to be able to perform much larger scale simulations of seismic wave propagation on the K computer.

Furthermore, it became crucial to optimize the calculation of elastic forces and stresses inside each spectral element to get efficient thread-level parallelization; in the case of eight threads, we achieved a 7.89 times speedup, that is, an increase of parallel efficiency of 99.8% compared to the original source code. Additionally, this modification reduced memory usage per node to 12% of the original flat-MPI code, which

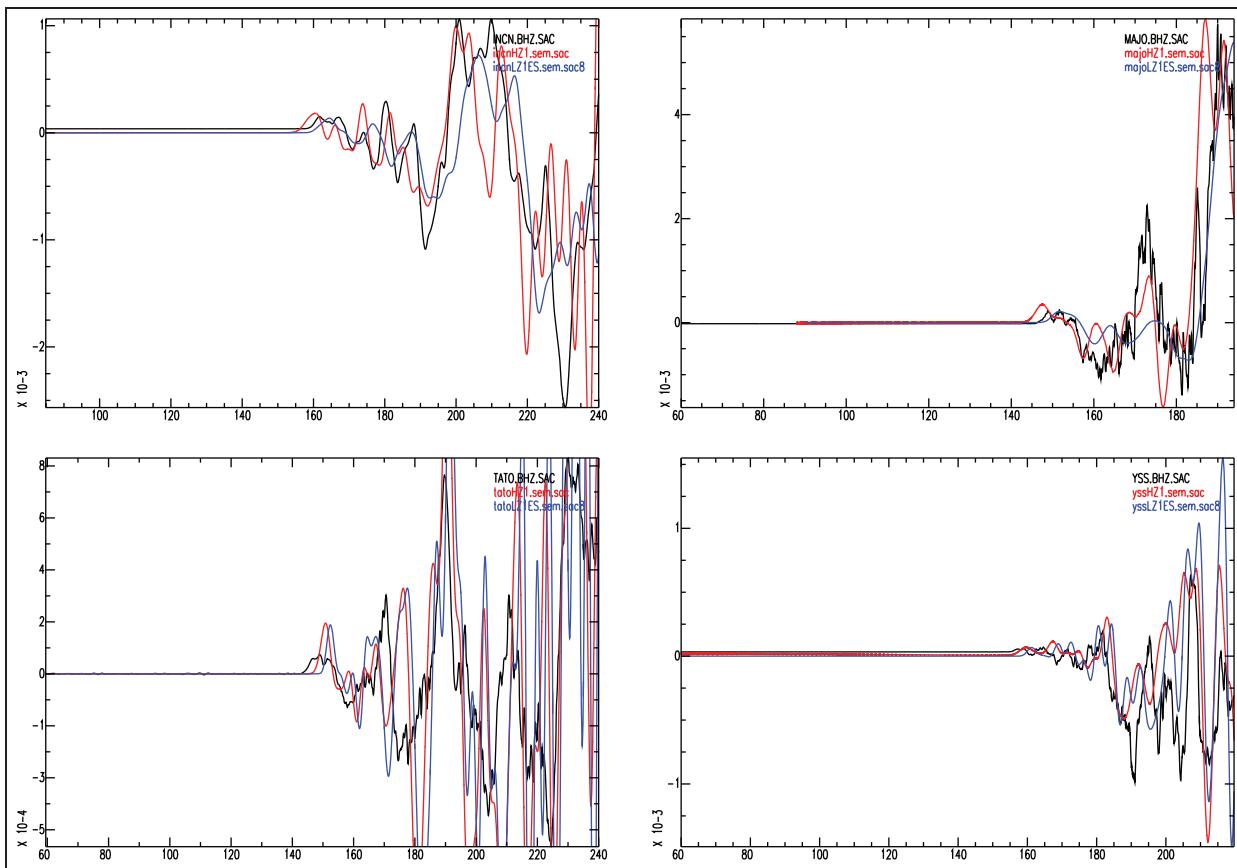


Figure 10. Comparison of numerical seismograms accurate down to a minimum seismic period of 1.2 s (red curves) and 5 s (blue curves) with real recordings from the field (black curves) for the same four seismographic recording stations as in Figure 9. Shown is the vertical component of the velocity vector of broadband seismograms. The horizontal axis is time in seconds and the vertical axis is ground velocity in meter per second. The origin of the horizontal axis is the earthquake origin time.

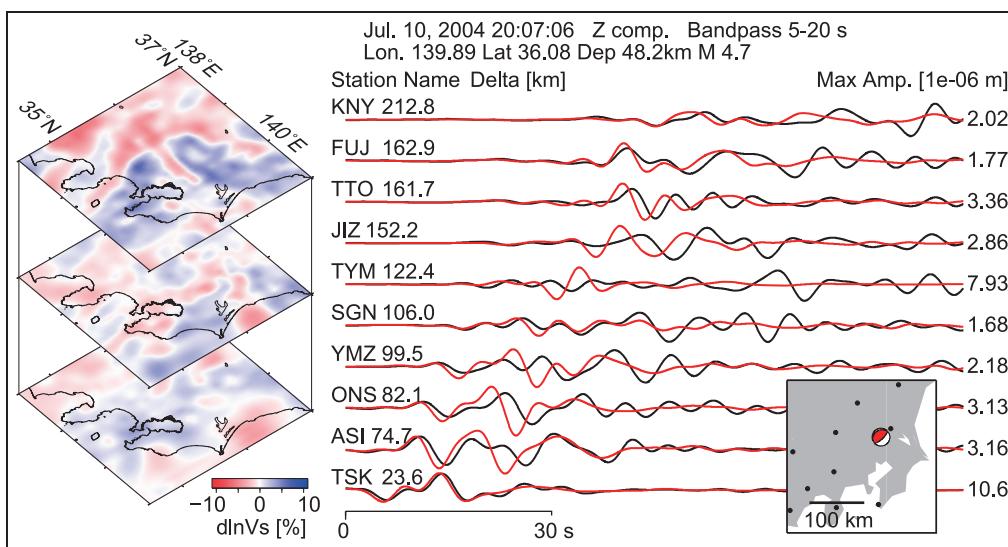


Figure 11. Initial model of seismic shear wave speed structure for the Kanto basin in Japan (Matsubara and Obara 2011). Perturbation of shear speed from the average structure is shown at depths of 40 km, 80 km, and 120 km, from top to bottom, respectively. Comparison of numerical seismograms (red) with real seismograms recorded in the field (black) for an earthquake that occurred in the region is also shown (right). When solving inverse (imaging) problems iteratively, differences between numerical and observed seismograms are used to improve seismic wave speed structure in the region under study.

solved the memory bottleneck on compute nodes. The new code performs numerical simulations accurate down to the shortest seismic period of about 1.2 s, which requires a mesh of the whole Earth composed of more than 600 billion grid points and having more than one trillion degrees of freedom. This currently represents the largest seismic wave propagation simulation ever performed.

The total physical duration of the numerical seismograms shown in this work is 7 min. It is limited by the maximum computational time allowed per batch job on the K computer. In terms of seismological applications however, we need to be able to compute numerical seismograms with a duration of at least 20 min because the fastest pressure waves arrive at the antipode of the epicenter about 20 min after the occurrence of an earthquake. Computing 20-min simulations on the K computer would thus require cutting the whole simulation in three parts of 7 min each, using checkpoint/restart files saved to disk between the three parts. Although checkpoint/restart capabilities are available in the software, using that would require saving about 200 terabytes to disk, which leads to huge input/output (I/O) bottlenecks on the K computer file system. In the future, we will thus investigate efficient data compression algorithms to alleviate this I/O bottleneck. Ultimately, we believe that this work on large-scale seismic wave propagation simulations can help the geophysical community to study earthquake generation processes from a more quantitative point of view.

Acknowledgements

This work used computational resources of the K computer provided by the RIKEN Advanced Institute for Computational Science through the HPCI System Research project (project ID: hp140221). We used the open-source SPECFEM3D_GLOBE version 5 software package freely available through the Computational Infrastructure for Geodynamics (CIG). Broadband seismograms were recorded by the Global Seismic Network and distributed by the IRIS Data Management Center. We are also thankful for access to the Earth Simulator (ES2) operated by the Center for Earth Information Science and Technology of JAMSTEC. We used F-net seismograms of the National Research Institute for Earth Science and Disaster Prevention of Japan.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

- Carrington L, Komatsitsch D, Laurenzano M, et al. (2008) High-frequency simulations of global seismic wave propagation using SPECFEM3D_GLOBE on 62K processors. In: *Proceedings of the Supercomputing 2008 Conference*, Austin, TX, November 15–21, 2008. doi:10.1109/SC.2008.5215501.
- Dahlen FA and Tromp J (1998) *Theoretical Global Seismology*. Princeton: Princeton University Press.
- Faccioli E, Maggio F, Paolucci R, et al. (1997) 2D and 3D elastic wave propagation by a pseudo-spectral domain decomposition method. *Journal of Seismology* 1: 237–251.
- Hasegawa Y, Iwata J, Tsuji M, et al. (2011) First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the K computer. In: *Proceedings of the Supercomputing 2011 Conference*, Seattle, WA, 12–18 November 2011. doi:10.1145/2063384.2063386.
- Heinecke A, Breuer A, Rettenberger S, et al. (2014) Petascale high order dynamic rupture earthquake simulations on heterogeneous supercomputers. In: *Proceedings of the Supercomputing 2014 Conference*, New Orleans, LA, 16–21 November 2014. doi:10.1109/SC.2014.6.
- Ichimura T, Fujita K, Tanaka S, et al. (2014) Physics-based urban earthquake simulation enhanced by 10.7 BlnDOF 30K time-step unstructured FE non-linear seismic wave simulation. In: *Proceedings of the Supercomputing 2014 Conference*, New Orleans, LA, 16–21 November 2014. doi:10.1109/SC.2014.7.
- Ishiyama T, Nitadori K and Makino J (2012) 4.45 Pflops Astrophysical N-body simulation on K computer - The Gravitational Trillion-Body Problem. In: *Proceedings of the Supercomputing 2012 Conference*, Salt Lake City, UT, 10–16 November 2012. Available at: <http://dl.acm.org/citation.cfm?id=2388996.2389003>.
- Komatsitsch D (1997) *Spectral and spectral-element methods for the 2D and 3D elastodynamics equations in heterogeneous media*. PhD thesis, Institut de Physique du Globe, Paris, France.
- Komatsitsch D and Tromp J (2002a) Spectral-element simulations of global seismic wave propagation-I. Validation. *Geophysical Journal International* 149: 390–412.
- Komatsitsch D and Tromp J (2002b) Spectral-element simulations of global seismic wave propagation-II. 3-D models, oceans, rotation, and self-gravitation. *Geophysical Journal International* 150: 303–318.
- Komatsitsch D and Vilotte JP (1998) The spectral-element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures. *Bulletin of the Seismological Society of America* 88: 368–392.
- Komatsitsch D, Tsuboi S and Tromp J (2005) The spectral-element in seismology. In: Levander A and Nolet G (eds) *Seismic Earth: Array analysis of broadband seismograms*, *Geophysical Monograph 157*. Washington, DC: American Geophysical Union, pp. 205–227.
- Komatsitsch D, Tsuboi S, Ji C, et al. (2003) A 14.6 billion degrees of freedom, 5 teraflops, 2.5 terabyte earthquake simulation on the Earth Simulator. In: *SC '03 Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, Phoenix, AZ, 15–21 November 2003. doi:10.1145/1048935.1050155.

- Lee SJ, Huang BS, Ando M, et al. (2011) Evidence of large scale repeating slip during the 2011 Tohoku-Oki earthquake. *Geophysical Research Letters* 38: L19306.
- Matsubara M and Obara K (2011) The 2011 Off the Pacific Coast of Tohoku earthquake related to a strong velocity gradient with the Pacific plate. *Earth Planets Space* 63: 663–667.
- Monteiller V, Chevrot S, Komatitsch D, et al. (2015) Three-dimensional full waveform inversion of short-period teleseismic wavefields based upon the SEM-DSM hybrid method. *Geophysical Journal International* 202(2): 811–827.
- Rietmann M, Messmer P, Nissen-Meyer T, et al. (2012) Forward and adjoint simulations of seismic wave propagation on emerging large-scale GPU architectures. In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC12)*, Salt Lake City, UT, 10–16 November 2012. Los Alamitos, CA: IEEE Computer Society Press, 2012, Article 38. Salt Lake City, UT, November 10–16, 2012, IEEE
- Ronchi C, Ianoco R and Paolucci PS (1996) The Cubed Sphere: a new method for the solution of partial differential equations in spherical geometry. *Journal of Computational Physics* 124: 93–114.
- Sadourny R (1972) Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. *Monthly Weather Review* 100: 136–144.
- SPECFEM3D Globe User Manual Version 5.0 (2014) Available at: http://geodynamics.org/cig/software/specfem3d_globe/specfem3d_globe-manual.pdf (accessed 2014).
- Tromp J, Komatitsch D and Liu Q (2008) Spectral-element and adjoint methods in seismology. *Communications in Computational Physics* 3(1): 1–32.
- Tromp J, Tape C and Liu Q (2005) Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. *Geophysical Journal International* 160: 195–216.
- Tsuboi S, Komatitsch D, Ji C, et al. (2003) Broadband modeling of the 2002 Denali Fault earthquake on the Earth Simulator. *Physics of the Earth and Planetary Interiors* 139: 305–312.
- Tsuboi S, Miyoshi T, Obayashi M, et al. (2014) Application of adjoint method and spectral-element method to tomographic inversion of regional seismological structure beneath Japanese Islands. In: *2014 Fall Meeting*, AGU, San Francisco, Abstract S31E-04.
- Tsuboi S and Nakamura T (2013) Sea surface gravity changes observed prior to March 11, 2011 Tohoku earthquake. *Physics of the Earth and Planetary Interiors* 221: 60–65.

Author biographies

Seiji Tsuboi received his DSci degree in Geophysics from the University of Tokyo, Japan, in 1986 and his MSc and BSci degrees from the University of Tokyo, Japan, in 1983 and 1981. He is currently a director of Data Management and Engineering Department, Center for Earth Information Science and Technology, Japan Agency for Marine-Earth Science and

Technology, Japan. His research interests include large-scale seismic wave computation and management of various types of Earth Science data and samples.

Kazuto Ando works in the Center for Earth Information Science and Technology, Japan Agency for Marine-Earth Science and Technology (JAMSTEC). He supports research activities in Strategic Programs for Innovative Research: Advanced Prediction Researches for Natural Disaster Prevention and Reduction (Field 3).

Takayuki Miyoshi received his PhD from the Kobe University in 2007. His research career began at Kobe University. After he worked at National Research Institute for Earth Science and Disaster Prevention (NIED) for five years, he has worked as project scientist in Japan Agency for Marine-Earth Science and Technology since April 2013. His research interests are seismotectonics, tsunami, and numerical simulations.

Daniel Peter received his PhD degree in Geophysics at ETH Zurich, Switzerland, in 2008. He is currently assistant professor at King Abdullah University of Science and Technology (KAUST) and affiliated to the Extreme Computing Research Center (ECRC), Saudi Arabia. His research interests are related to computational seismology and geophysical inverse problems with focus on enhancing numerical three-dimensional wave propagation solvers for seismic tomography, particularly for very challenging complex regions and media.

Dimitri Komatitsch received his PhD from Institut de Physique du Globe in Paris, France in 1997. After several years in the United States he became a Full Professor at University of Pau, France, and is now a Research Director at CNRS in Marseille, France. His main research interests are acoustic and seismic wave propagation, imaging and inversion as well as high-performance numerical simulations.

Jeroen Tromp received his BSc in Geophysics from the University of Utrecht in 1988, and his PhD from Princeton University in 1992. He is currently the Blair Professor of Geology and Professor of Applied & Computational Mathematics at Princeton University. His primary research interests are in theoretical and computational seismology, including simulations of acoustic, (an)elastic, and poroelastic wave propagation, and imaging Earth's interior based on spectral element and adjoint methods.