博士論文

論文題目 Improving reliability of urban earthquake disaster simulation
with many scenario computing
and 3D nonlinear ground motion analysis

(多数シナリオ解析と非線形三次元地盤振動解析による
都市震災シミュレーションの信頼性向上)

氏名　　藤田　航平

Abstract

To make effective and efficient earthquake disaster mitigation plans, reliable earthquake disaster estimates are needed. Such estimates should have high degrees of scientific rationality, accuracy, and resolution. To improve the reliability of current estimates based on observations of past earthquake disasters, researchers have developed physics-based earthquake disaster estimation methods that combine geospatial data and physics-based numerical analysis to analyze each phase of an earthquake disaster. These methods have the potential to generate accurate estimates, since current physics models (e.g., the wave equation) are able to accurately model wave propagation and the responses of structures. However, two problems limit the reliability of such methods. The first problem is the lack of geospatial data (e.g., data on soil and structures), compared with the actual complexity of the environment. The lack of data leads to uncertainties in the analysis models inputted into the numerical analyses. The other problem is that simplified analysis methods, which are less reliable, are used for solving some of the physics models.

This study aims to improve the reliability of physics-based urban earthquake disaster simulation by: (1) developing a method to compute a large number of earthquake disaster simulations to reflect uncertainties in the models and estimates and (2) using three-dimensional (3D) ground motion analysis to solve the physics model with higher reliability.

In the first part of this study, I developed a method for efficient computation of a large number of earthquake disaster simulations on high-performance computers. When the distribution of uncertainties in a set of analysis models is known, analysis models that exhibit the uncertainty distribution can be analyzed to estimate the distribution of results. The difficulty of such analyses is obtaining scalability on high-performance computers when analyzing many cases of simulations of large areas with large number of structures. Based on the measurements of previous implementations, I found that the data transfer between the ground motion analysis program and the structural response analysis program becomes a bottleneck. By careful management of data transfer between the analysis programs, 97.4% weak scalability was attained using 40,000 nodes of the K computer at RIKEN. This enables the computation of more than 1,000 scenario simulations of 0.25 million structures in Tokyo within an hour using 20,000 nodes of the K computer.

In the second part of this study, I replaced the one-dimensional (1D) ground motion analysis used in

a previous urban earthquake disaster simulation with a highly scalable 3D ground motion analysis. Here, a finite element method with unstructured tetrahedral elements and nonlinear constitutive modeling was used to model and analyze the complex ground geometry, and the ground motion recorded at the surface was used to shake the 4,066 structures in the target area. To assure the convergence of structural response, I conducted ground motion analysis with target frequencies of 5, 10, and 15 Hz. Although the analysis with target frequency of 15 Hz leads to a large-scale problem with 1 billion degrees of freedom, it can be computed in 11 h using 8,192 nodes of the K computer. The results show that the structural response is converged at a target frequency of 15 Hz. Next, I compared the ground and structural responses with the results obtained by 1D ground motion analysis. A difference of 15% was obtained when comparing the seismic intensity index of the ground motion obtained by 1D and 3D analysis. As a demonstration of the many cases analyzed for uncertainty modeling, I generated structure models for each structure in the target area, and analyzed them to obtain the distribution of responses. By analyzing 10,000 models for each structure, I obtained convergence in the distribution of structural response.

The results obtained demonstrate that the combination of 3D ground motion analysis and structural response analysis of each structure in an urban area can be useful for completing detailed analysis considering the 3D geometry of soils and the properties of each structure. The results further show that computing a large number of cases can be useful for uncertainty modeling of structures. Such analysis could lead to more reliable earthquake disaster estimates.

Acknowledgments

本研究を進めるにあたり，大変多くの方々のご指導ならびにご協力を賜りました．

東京大学地震研究所の市村強准教授には，直接の指導教官として研究全般において多大なるご指導を賜りました．研究の進め方から論文の組み立て方など，様々な面から大変丁寧なご指導を賜りました．また，副査でもあり，研究室の教授である東京大学地震研究所の堀宗朗教授には，頻繁に研究の方向性についてご指導を賜りました．研究に対する実用性・将来性といった観点からのご指導は大変有用であったと考えます．また，副査である，東京大学大学院工学系研究科社会基盤学専攻の前川宏一教授及び布施孝志准教授，東京大学空間情報科学研究センターの柴崎亮介教授，東京大学情報基盤センターの中島研吾教授には，多方面からのご指導を賜りました．これらのご指導は研究の位置づけを考え，論文の質を上げるために大変有用であったと考えます．各先生方から賜りましたご指導に感謝の意を表します．

直接の指導教官ではありませんでしたが，東京大学地震研究所巨大地震津波災害予測研究センターの先生方からもご指導を賜りました．Maddegedara Lalith Lakshman Wijerathne 准教授には研究の計算科学的な側面を中心に，頻繁にご指導を賜りました．田中聖三助教には，大規模計算機の使用方法や，計算クラスタの管理方法など，ハンズオンなご指導を賜りました．また，研究室のメンバーの常日頃からの支えも私の研究生活にとって大変有用でした．Pher Errol Balde Quinay 博士や，縣亮一郎さん，本間俊介さんと互いに論文を見せ合って議論することにより，有用な意見を頂戴しました．また，その他の研究室メンバーにも大変お世話になりました．一緒に夕食に行ったのも良い思い出です．このように，不自由なく研究できたのも，先生方と研究室メンバーの支えあってのものだったと思います．

また，サイバネットシステム株式会社の林幸子様には，補遺 B の可視化プログラムの実装に関してご協力を賜りました．また，本研究では理化学研究所のスーパーコンピュータ「京」を利用しました (課題番号: hp120308)．また，本研究の一部では地盤工学会の電子地盤図，株式会社パスコの構造物データ，気象庁及び防災科学技術研究所強震観測網で観測された地震波形を使用しました．また，本研究は日本学術振興会特別研究員奨励費 (課題番号: 24・8183) の助成を受けています．

ご指導ならびにご協力頂いた方々に感謝いたします．

2014 年 2 月

藤田航平

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Earthquake disasters rarely occur, but can lead to serious damage over large areas, depending on the characteristics of the earthquake (fault rupture) and the properties of the environment (e.g., crust, surface soils, and built structures). Many mitigation measures, such as seismic retrofitting, stocking of emergency materials, recovery planning, and insuring against loss, are used to reduce the human and financial costs of such disasters. To maximize the effectiveness and efficiency of mitigation planning, reliable disaster estimates are needed. Such estimates should have high degrees of scientific rationality, accuracy, and resolution.

Owing to their low computational cost for the evaluation of large areas, statistics-based methods or empirical methods are widely used in current earthquake disaster estimation (e.g., [1, 2]). However, it is difficult to obtain enough observation data to make accurate estimates that reflect the characteristics of different earthquakes and different environments due to the infrequent occurrence of earthquakes that causes damage to the built environment. For example, it is common that the built environment changes between major earthquakes that cause damage in an area; using statistical data related to the old environment to estimate damage in a future disaster will result in inaccurate estimates.

Another approach to urban earthquake disaster estimation is based on physics, combining geospatial data and physics-based numerical analysis to analyze each phase of an earthquake disaster. In physics-based approach, a phenomenon is modeled by a physics model (e.g., the wave equation), and analyzed using analysis models (e.g., finite element mesh) and analysis methods (e.g., finite element analysis). This approach is summarized in Table 1.1. One example of a physics-based estimation method is the integrated earthquake simulator (IES), which generates analysis models from geospatial data using data conversion modules, and combines analysis programs that analyze each phase of earthquake disasters to estimate damage to structures in a city [10, 11]. High-performance computing methods have been applied to the analysis programs for analyzing many structures in large cities [12, 13]. Such methods have the potential to generate accurate estimates that reflect detailed properties of soils and structures, since the

Table 1.1: Terminology of models and analysis methods in physics-based disaster estimation based on the example of the phenomenon of wave propagation through crust

Phenomenon	Physics model	Analysis model	Analysis method
Wave propagation through crust	Wave equation	Finite element mesh of crust	Finite element analysis

physics involved in the wave propagation and seismic response of structures can be modeled accurately with current physics models.

Although an overall framework for physics-based disaster estimation has been developed in the literature, two main problems limit the reliability of physics-based disaster estimates when focusing on the effect of the environment on the disaster:

1. Geospatial data for input into the analysis models are sparse, compared to the actual complexity of the environment. This leads to uncertainties in the analysis models and ultimately, in the estimates.

2. Simplified analysis methods are commonly used to solve the physics models. Improved analysis methods are needed to solve the physics models with higher accuracy and resolution.

These problems are also present in the IES technique; the uncertainties of analysis models are not reflected in IES estimates, because IES is designed to analyze one scenario with deterministic inputs. Also, IES uses a 1D ground motion analysis. Three-dimensional analysis methods have a higher degree of scientific rationality and generate more reliable estimates.

One way to ensure the uncertainties in the analysis models reflect in the estimates is to analyze a large number of cases (referred to hereafter as “many-case analysis”). When the distribution of uncertainties in the analysis models is known, the distribution of responses can be obtained by producing many analysis models that follow the input distribution and analyzing them with a Monte Carlo approach. Capacity computing, which is a type of computation consisting of small- to medium-scale computations that sum up to use large portions of the resources of high-performance computers, can be used to perform many-case analysis of large number of structures. Capability computing, which is a type of computation that solves large problems in a short time using large portions of the resources of high-performance computers, can be used to solve physics models with higher accuracy and resolution using sophisticated analysis methods.

In this study, I use capacity computing and capability computing to improve the reliability of earthquake disaster simulation.

1. First, I develop a capacity computing method for performing many-case analysis of earthquake disasters with different input waves at bedrock. This method is aimed to be used for estimating the distribution of structural response under input waves with uncertainties.

2. Next, I enhance the ground motion simulation of shallow soil structures in IES with capability computing. I replace the 1D ground motion analysis with large-scale 3D nonlinear ground motion analysis. This method is used to analyze the 3D effects of soil structures on the response of structures in a city.

This thesis is organized as follows. In Chapter 2, I review relevant studies and practical urban earthquake disaster estimation methods in comparison with the present study. In Chapter 3, I describe the capacity computing method developed for performing many-case analysis of earthquake disasters. In Chapter 4, I describe the urban earthquake simulation using 3D ground motion analysis. Here, I first perform a convergence test of 3D ground motion analysis by increasing the mesh resolution of the 3D ground model, and comparing ground and structure response. I then compare the obtained ground and structure response with the results obtained using 1D ground motion analysis, and discuss the effect of 3D modeling. Finally, I perform many-case analysis of structural response simulations to estimate the response distribution of structures with uncertain structural parameters. In Chapter 5, I summarize the study with a discussion of future work and possible future applications. In Appendix A, I check the reliability of 3D nonlinear ground motion analysis by comparing computed results with observed seismograms. In Appendix B, I describe a visualization program developed for effectively conveying the dynamic urban earthquake simulation results to users.

Chapter 2

Relevant studies

2.1 Statistics-based and empirical earthquake disaster estimation methods

Most earthquake disaster estimation uses statistics-based or empirical methods based on observed data of past earthquake disasters to evaluate structural damage. For example, fragility curves, which are statistically regressed curves of the relationship between a given ground motion index (e.g., peak ground acceleration and seismic intensity) and the damage ratio of a building, are commonly used for estimating the probability of structural damage. Since only simple arithmetic is used for evaluation of damage in such curves, damage estimates of large areas can be conducted in a short time with small computational cost. Thus, the statistics-based method is also commonly used to evaluate structural damage in a city immediately after an earthquake to plan post-disaster response (see [14] for a review of current post-disaster estimation methods). Another advantage of the statistics-based methods is that it can be applied in areas where limited information is available about soil or structural properties. A disadvantage of statistics-based methods is the difficulty of acquiring enough observation records to evaluate damage to structures with different properties, under different types of ground motion, or with different soil properties.

Some studies combine statistics-based methods with physics-based methods to improve the accuracy of estimates of earthquake wave propagation from the fault to the engineering bedrock. For example, the Central Disaster Mitigation Council of Japan estimates the wave at engineering bedrock using a hybrid method of the 3D finite difference method (FDM) and the statistical Green's function method. Here, deterministic FDM simulations are used for estimating low-frequency components in the waves, while the Green's function method is used for estimating the high-frequency components that are difficult to estimate by physics-based methods due to of the high computational cost. The present study focuses on the seismic response of surface soil and structures; hybrid methods, pure statistics-based, or pure physics-based methods can be used as inputs into the urban earthquake disaster simulation.

Table 2.1: Comparison of urban earthquake simulation methods combining 3D ground motion analysis and structural response analysis

	Ricardo and Bielak [7]	Krishnan et al. [8]	Present study
Ground motion analysis	3DFEM (octree mesh)	3DFEM (unstructured mesh)	3DFEM (unstructured mesh)
Structural response analysis	3DFEM (octree mesh)	FEM (frame model)	MDOF, OCM (frame model)
# of modeled structures	74	2	4,066
Soil-structure interaction	Yes	No	No
Many case structural analysis	No	No	Yes

2.2 Physics-based urban earthquake disaster estimation methods

Several studies have been performed combining physics-based simulation methods to analyze ground motion and structural response for urban earthquake disaster estimation. Here, I focus on studies using 3D ground motion analysis methods and structural response analysis methods for analyzing many structures in a city. Table 2.1 summarizes the characteristics of studies by Ricardo and Bielak [7], Krishnan et al. [8], and the present study.

In studies by Ricardo and Bielak, a domain with 74 structures was modeled with 3D FEM with octree mesh for nonlinear time-history analysis of soils and structures under earthquake excitation. The analysis was performed in a monolithic manner so that the interaction between soil and structures could be computed. To analyze structures in the monolithic program, the geometry of structures was simplified and modeled with an octree mesh with equivalent material properties. In studies by Krishnan et al., ground motion analysis was conducted using a 3D spectral element method with an unstructured mesh. Ground motion was measured at 636 analysis sites in the numerical ground model, and inputted into two 18-story building models; one was a building following the 1982 Uniform Building Code (UBC) and the other was a redesigned building following the 1997 UBC. A detailed FEM capable of flexural yielding and strain hardening was used for the modeling and analysis of the structure response.

The present study explained in Chapter 4 used an unstructured finite element mesh (second ordered tetrahedral mesh) for nonlinear time-history analysis of ground motion. To model a large number of

structures in a city, an automatic modeling method was used to generate structure models. Then, the structure models were analyzed using ground motion recorded under each structure in the ground motion analysis model. I used structural analysis methods used for performance-based structural design, which are accepted to have sufficient reliability for design purposes. Since there are uncertainties in the structural models when using limited structural data, I generated many structure models with varying structural properties and analyzed them to estimate the response distribution.

Compared with Ricardo and Bielak's study, this study better reflects complex ground geometry by using the unstructured mesh for ground modeling, which could affect the response for high-frequency components. Conversely, soil-structure interaction cannot be considered in this study as the ground and structure analysis are performed separately. Krishnan et al.'s study uses a more complex method for structural response analysis, as models can be manually constructed for the two structures. The method in this study is less complex, since it is still difficult to generate such detailed models from limited building datasets, but we can model more structures for analysis. Neither of these studies incorporate many-case analysis to reflect the uncertainties of the structure models in the estimate.

2.3 Physics-based many-case analysis of structure response

Several studies have used many-case seismic response analysis of structures for uncertainty modeling. Liel et al. [15] analyzed the nonlinear time-history response of a structure that was modeled with varying structural parameters to reflect the uncertainty of structure models used for the risk assessment of collapse due to earthquakes. Here, many structure models that follow the uncertainty distribution of structural parameters (e.g., stiffness and strength) were constructed and analyzed under time-history input waves. The collapse ratio of structures was plotted to the input ground motion indexes (e.g., spectral acceleration) and compared with the collapse ratios without considering uncertainties in the structure model. In Liel et al.'s study, several types of structures were analyzed for uncertainty modeling. In this study, I generate and analyze many structure models that follow an input distribution of structural parameters, and apply it to many structures in a city using high-performance computing.

Chapter 3

Enabling many scenario urban earthquake disaster simulation

3.1 Introduction

An earthquake disaster is a complex system starting from fault rupture, wave propagation through the earth's crust, and nonlinear ground motion of surface soil structures (soil amplification) to the response of structures and the social response (Fig. 3.1). In some studies, available datasets and numerical methods that simulate each phase of an earthquake disaster are integrated to provide a better understanding of the earthquake disaster system and to estimate the consequences of anticipated earthquake disasters. In such simulations, system computing, which is a type of computation that combines many types of datasets and computation components, is used to process available geospatial datasets and multiple types of simulation methods.

With the ability to compute each simulation component in higher resolution, it becomes more difficult to obtain the quality and quantity of input data needed for each phase of an earthquake disaster simulation. For example, boring log data are commonly very sparse to capture local fluctuations in site conditions, so data is interpolated or extrapolated. Thus, accounting for uncertainties in input data becomes important for improving the reliability of simulation results.

Capacity computing, or many-case computing of deterministic forward modeling simulations with different input parameters, is a candidate for evaluating the effects of uncertainties in input data on final estimates. Current petascale supercomputers or the exascale supercomputers of the near future have the computational capacity to enable the simulation of large number of cases. Many-case analysis techniques is a primary building block for uncertainty modeling of complex nonlinear systems by using Monte Carlo simulation or sensitivity analysis.

So far, not much attention has been paid to capacity computing in current system computing based

earthquake disaster simulation. Scalability to multiple simulation cases is commonly limited by data movement between components, for example, the scalability of seismic tomography is limited by large file I/O that increases in proportion with the number of cases [5]. A much tighter connection between data and simulation components is needed to perform capacity computing together with system computing.

In this study, I aim to develop a system that can analyze a large number ($> 10^3$) of simulations with different inputs with high parallel efficiency on current high-performance computers. I focus on wave amplification at surface soil structures and structural response phase, corresponding to insets ii and iii of Fig. 3.1. I start by examining the bottlenecks of the current simulation system, IES, when applied to many-case analysis. IES is a system that integrates many types of simulation methods and geospatial data for analyzing the response of a city to earthquakes (Fig. 3.2). Then, I change the mapping of the problem to processors and use the parallel file system effectively to attain scalability for many-case simulations.

The rest of this chapter is organized as follows. The following section summarizes the characteristics of the K computer, the system used for measurement. Section 2.3 describes the current IES and its performance when applied to many-case analysis. Section 2.4 describes the modifications made to enable many-case analysis of IES, and Section 2.5 reports the performance of the program. As an application example, I perform analysis of downtown, Tokyo for 1,000 case simulation sets in Section 2.6. Section 2.7 summarizes the chapter.

3.2 System used for measurement

In this study, I measure performance on the K computer, which is a massively parallel supercomputer at RIKEN, Advanced Institute for Computational Science [16]. The K computer consists of 82,944 compute nodes, each with single, SPARC64^{TM} VIIIfx CPUs. The CPU has eight cores operating at 2.0 GHz with 6 MB L2 shared cache, with peak performance of 128 GFLOPS. Each node has 16 GB of DDR3-SDRAM memory, with peak memory bandwidth of 64 GB/s. Tofu, a six dimensional interconnection network, is used for communication between nodes [17]. Each node can communicate in four directions simultaneously with 5 GB/s throughput in each direction. An OpenMPI 1.4.3 [18] based MPI library optimized for the Tofu network is used, following the MPI 2.1 standard [19]. A two-level file system consisting of global and local file systems is used [20]. Both are Lustre based parallel file system (Fujitsu Exabyte File System, FEFS), with stage in and stage out options for moving data between the two file systems.

3.3 Overview of original IES and its performance

3.3.1 Design of IES

In order to perform an integrated simulation from the source to the city, programs that analyze each phase of an earthquake disaster and various types of data for input into these programs must be combined. There are several types of analysis methods for each phase of an earthquake disaster. For example, methods are developed specifically for each structure type (e.g., wood, steel, and reinforced concrete). Thus, flexibility and extendibility are needed in the integration framework to accommodate multiple types of programs in each phase of the earthquake disaster simulation. To attain flexibility and extendibility, IES divides the functionality controlling the work flow (here on called the "kernel") with the functionality analyzing particular types of phenomena (called "modules"), and pre-developed analysis programs are plugged into IES as modules using object oriented features like polymorphism and template techniques. The work of the kernel is management of input and output data, distribution of load to processors, and data conversion between modules.

The original IES is designed for simulating one disaster phase at a time, by checking the output of each phase and inputting it to the next phase. To preserve the independence of components from each other, IES uses the file system to exchange data between component simulations.

For the component analysis methods, the original IES implements 1D analysis methods for the soil amplification analysis, and several analysis methods for the structural response analysis. The methods can be chosen by the required accuracy, available computational resources, and allocated time. Simpler methods are used in this study since large number of cases cannot be computed in a reasonable time using computationally expensive methods even when perfect scalability is attained on present largest supercomputers. Since the kernel is independent from each of the analysis methods, the kernel can be used for plugging in other methods in the future.

Fig. 3.4 shows the overview of the target problem. The same wave at bedrock is inputted to soil amplification analysis, and the output of soil amplification analysis (wave at surface) is inputted into structural response analysis. Computation of each evaluation point and structure is independent from each other since 1D soil amplification analysis without soil-structure interaction is used. The original IES executes serial programs that analyze each point or structure, concurrently in many computation cores on distributed memory parallel computers. In the following sections, I explain the properties of each analysis module

used in this study.

3.3.2 Soil Amplification Analysis (SAA) module

In this study, I use the 1D equivalent linear method to analyze the time history response of the surface accounting the nonlinear material properties of the soil. The analysis assumes a stratified soil structure, with nonlinear soil properties following the Hardin-Drnevich model [21]. The shear wave propagation in the *x* and *y* directions are computed independently using Fast Fourier Transforms (FFTs).

Table 3.1 shows the runtime and input/output sizes of the serial 1D soil amplification program. The inputs of the program are acceleration at bedrock with size of 98 KiB and soil model with size less than 100 B. The execution time using one core of K computer is 0.463 s in average. The outputs are acceleration at surface with the same size as input acceleration and nonlinear response parameters with size of 1.3 KiB. We can see that the input and output of waves dominate the total file I/O size.

Original IES performs analysis in parallel by distributing evaluation points to several cores, and each core executing the above mentioned serial program. Flat MPI with an all-worker model based on static load balancing is used. Here, the runtime of each evaluation point is estimated to be proportional to the number of soil layers, and are assigned to processes so that the sum of the estimated runtime becomes nearly equal. The root process reads the input data, performs load balancing, and distributes load to processes. Since target areas are divided into subregions as disjoint GIS tiles, execution is performed per GIS tile and output is written to a single file per GIS tile from distributed processes using collective output function (`MPI_File_write_at_all()`) of MPI-IO [19].

3.3.3 Structural Response Analysis (SRA) module

The Multi Degree Of Freedom (MDOF) model and the One-Component Model (OCM) is used for structural response analysis [12, 13]. In MDOF, a structure is modeled with mass points for each floor connected with shear springs and dampers in the two horizontal directions. In OCM, a structure is modeled with beams and columns. The Takeda model is used for the constitutive relations of the nonlinear springs used in the beam and column elements.

Table 3.2 shows the runtime and input/output sizes of the serial structural analysis programs. The input and output data formats are the same for both analysis methods; data is converted internally to and

from the data structures used in each analysis method by the kernel. The inputs are acceleration history at surface with size 98 KiB and structure model of size less than 100 B. The average execution time for a single structure on one core of K computer is 6.52 milliseconds for MDOF and 7.69 s for OCM. The outputs are displacement time history with size of 2.6 MiB and information of structural shape (Fig. 3.5). There is about 10^3 times difference in the computation cost between the two methods, while the I/O size is the same.

In the previous works, a distributed memory type parallel program was developed to analyze many structures that make up a city in a short time [13]. Flat MPI, with an all-worker model is used, with static load balancing based on past runtime. The runtime of each structure is estimated to be proportional to recorded runtime, and structures are distributed to processes so that the total runtime of structures assigned to each process becomes nearly equal. This method works well owing to the properties that runtime for computing response of similar magnitude earthquakes are similar regardless of the details of the input ground motion. The root process reads the input data (input parameters, structural configuration) and broadcasts to all the processes. The structural configuration data are compressed with zlib for reducing the message size. Each process reads the input wave for assigned structures using the collective read function (`MPI_File_read_all()`) of MPI-IO, and its structural responses are computed. After computation of all structures, output of all structures in a GIS tile are gathered and saved to a single file using collective write functions (`MPI_File_write_all()`) of MPI-IO.

Since analysis of each structure is performed in serial, the shortest runtime for a given set of structures is constrained by the longest runtime of a single structure. Thus, maximum (strong) scaling processors is constrained by

$$p_{\max} = \frac{\sum_i t_i}{\max_j t_j}, \tag{3.1}$$

where t_i is the runtime of structure i. For a problem of a typical urban area with low- to middle-rise structures, the average analysis time of each structure using OCM is 8.58 s, while the maximum runtime is 1,781 s, leading to $p_{\max} = 1{,}220$. For a problem with high-rise structures, $p_{\max}$ becomes smaller; around few hundred or even under 100. On the other hand, $p_{\max}$ will be larger for SAA since the number of soil layers is not likely to change by more than 10 times.

3.3.4 Performance of original IES when applied to many-case simulations

In this section I measure the performance of the original IES when applied to many-case simulations. Since the original IES does not have the functionality to perform many case simulations, I make modifications to run multiple cases in a single job. As shown in Fig. 3.6, there are two common ways of running multiple simulations, which are, i) with all processors working on the same case, computing case by case sequentially, or ii) processes are divided into groups that analyze a single case and all cases are run simultaneously. I use ii) in this study since the maximum number of processors for the program to strong scale is limited by the problem characteristics, shown in Eq. (3.1). The original communicator (`MPI_COMM_WORLD`) is split into the number of cases to be run simultaneously using `MPI_Comm_split()`, and analyzed with different input waves for each case. In the measurements, all I/O is performed on shared directories accessible from all processes. Files are striped over 12 Object Storage Targets (OSTs) in the local file system of the K computer.

Fig. 3.7 shows the weak scaling performance of the SAA module when applied to multiple case simulations. Here, weak scaling is defined by

$$s_{\mathrm{w}}(p) = \frac{T_1}{T_p}, \tag{3.2}$$

where T_1 is the execution time for solving one case, and T_p is the execution time for solving p cases using p times the computing resources. Here I used 160 processes (20 nodes) per case to solve analysis cases of 253,405 evaluation points with 3 soil layers and 32,768 time steps. We can see that s_{w} decreases with respect to the number of cases, with $s_{\mathrm{w}} = 77.2\%$ at 40 cases. Fig. 3.8 shows the breakdown of the computation time. The computation consists of preparation, execution, and file output. Execution and file output is performed per GIS tile, repeatedly until all the tiles are finished. We can see little change in preparation time and execution time, but significant increase in file output time when the number of cases is increased. The file output throughput of 1 case and 40 cases are 0.31 GiB/s and 3.01 GiB/s, respectively. We can see that the file output throughput is not increasing linearly with respect to the number of processors used.

Fig. 3.9 shows the weak scaling performance of the SRA module when applied to multiple case simulations. Here, I computed a problem of 253,405 structures with $N = 8{,}192$ time steps using 320 processes (40 nodes) per case for MDOF and 640 processes (80 nodes) per case for OCM. In case of MDOF, s_{w} rapidly decreases with the increase in number of cases, and its value becomes 44.5% for 4

cases. OCM performs better, where s_w = 77.7% for 8 cases. Fig. 3.10 shows the breakdown of the computation. In case of MDOF, almost all time (96%) is spent for file I/O, while about 11% of the time is spent for I/O in the case of OCM. Using supercomputers for solving problems with excessively short runtime (as of MDOF model) seems unreasonable, but such analysis are used as components in simulation of complex systems, and attaining scalability of short runtime programs on high performance computers will be important for performing many-case analysis of system computing problems.

Fig. 3.11 shows the memory usage of the original IES for different number of time steps. The K computer has 16 GiB of memory per node, which is 2 GiB per core. From the figure, we can see that the memory usage of the SRA module exceeds 2 GiB for a problem of 32,768 time steps when the number of processes per case is smaller than 275. In case of problems with large number of time steps, the memory usage becomes too large to fit in the nodes even when the maximum scaling number of processes ($p_{\max}$ given in Eq. (3.1)) is used. Efficient management of memory is needed so that the minimum number of processes that can analyze a problem becomes sufficiently smaller than $p_{\max}$. Memory usage of SAA is smaller than that of SRA and is not likely to cause problems.

From these measurements, we can see that file I/O is hindering scalability to large number of cases, and the scalability is worse for problems with large ratios of file I/O time spent in the whole runtime. Table 3.3 shows the output size for 1 case of the previous SAA and SRA measurements. Here, I only indicate the file I/O size that is linear to the number of cases. We can see that the file I/O of wave at surface and file output of structural response are dominant in the file I/O of the whole analysis. Reduction of the file sizes of these files, or speedup of file I/O throughput is needed for scalability. We also observed that problems with large number of time steps leads to large memory footprint, thus reduction in memory footprint is necessary for efficient computation of such problems.

3.4 Modifying IES for improving scalability to many-case simulations

From the previous performance measurements of the original IES, I found out that the excessive file I/O used to move data between simulation components hindered the scalability to many case simulations. In this section, I write measures to reduce file I/O for attaining scalability to large number of cases. I also reduce memory footprint by examining the simulation process.

3.4.1 Changing mapping of problem and I/O management (1)

I first change the mapping of problem to processes so that it suits the change in problem setting from one-case simulation to many-case simulations. Fig. 3.12 i) shows the problem settings, where arrows indicate flow of data, and the thickness of arrows indicate the data size. In the original IES, the problem was divided into SAA and SRA phase, and mapped to processes with individual load balancers, see Fig. 3.12 ii). I remap the problem to processes by making SAA and SRA phase of each structure as a group, and performing load balancing by these groups, see Fig. 3.12 iii). Static load balancing is performed based on the sum of the past runtime of the two analyses of each structure; SAA is done in the same process that executes SRA. We modify the IES kernel so that data can be passed between distinct analysis modules locally in the process without inter-process communication. This eliminates the intermediate I/O of wave at surface between SAA and SRA modules.

I next seek ways to deal with the large file output at the end of the simulation. In the SRA module, displacement output of each node in the same floor has the same values. Using this property, nodal deformation values per floor is outputted to reduce the file output size, see Fig. 3.13. The output size can be further reduced by reducing the number of time steps outputted; engineers or decision makers usually need information of maximum response or the residual response (permanent deformation) of structures and not all of the time steps is needed. Table 3.4 shows the file output size before and after the modification. We can see approximately a five fold reduction when outputting all the time steps, and magnitudes of smaller size when outputting only the maximum response.

Fig. 3.14 shows the time usage of the program for 1 case and 100 case simulations using the modifications mentioned above. Here, performance is measured using a problem of 253,405 structures with soil model of 3 layers under each structure. 160 processes (20 nodes) of the K computer is used to compute each case with different input wave at bedrock to measure the weak scalability of the program. Here, the MDOF model is used for computation. All input waves have N = 32,768 time steps, and one time step corresponding to the maximum response is outputted to the local file system using the shared directory scheme. We can see that the scalability has improved compared to the original versions of the SAA and SRA modules. Conversely, preparation time and output time is still increasing with the increase in number of cases.

3.4.2 I/O management (2)

In order to further improve the scalability of the program, I check the preparation and output scheme of the program.

I first change the output scheme of the program. In the original IES, collective output to the shared directory was used, see Fig. 3.15 i). Since all ranks in the program access the same shared directory, file access contention may happen among cases. Thus, I use the rank directory scheme of the K computer to attain higher throughput and reduce I/O time. Here, output data of each case is collected by the root process in each case, and data is outputted using serial MPI-IO (`MPI_File_write()`), see Fig. 3.15 ii). This makes the file output of each case more independent, which is expected to increase file I/O performance without taking special attention to the underlying parallel file system. Further exploration is needed when outputting larger data sizes, as the advantages of collective I/O increase for such problems.

Next, I change the preparation scheme of the program. The preparation time consists of reading and broadcasting input parameters, reading and broadcasting the structure models, and load balancing. The structure models are separated into 40 files, with total size of 80 MB. In the original version, files were read sequentially by the root rank, and then combined to be broadcasted in a single message to the other ranks. Fig. 3.16 i) shows the breakdown of the preparation time for 100 cases. We can see that almost all the time is used for reading and broadcasting the structure models, and that some processes are waiting for the broadcast to finish for the remaining processes. By reading one file and immediately broadcasting the data without combining all the files into one message, we can pipeline the reading and broadcasting operations. Fig. 3.16 ii) shows the performance of the pipelined version, with a reduction of preparation time by 22%. By use of the rank directory, preparation time can be further reduced by reading input data from the root process of each case, and broadcasting to other processes in each case so that the number of target processes in each broadcast call becomes small. Fig. 3.16 iii) shows the results of the improved case, with a reduction of preparation time by 78% as compared to the original case.

3.4.3 Memory usage management

I reduce memory usage so that the program (strong) scales to problems with large number of time steps and small $p_{\max}$. The structural response results of the SRA module is outputted via intermediate data used for each analysis method (e.g. MDOF and OCM). Since the intermediate data is in double precision

while the few designated output time history data is in single precision, the size of output data becomes significantly smaller than that of the intermediate data. In the original SRA program, data is converted after all the structures were analyzed, see Fig. 3.17 i). I modify the program so that data conversion from intermediate data to output data is done after analysis of each structure, and save all data after all structures are computed, see Fig. 3.17 ii). Fig. 3.18 shows the comparison of maximum memory footprint during the whole analysis. Here, one time step corresponding to the maximum response is outputted. We can see that small number of cores can be used to solve problems with large number of time steps (N = 32,768), expanding the scope of problems that the program can scale up to.

3.5 Performance of modified IES

3.5.1 Weak scaling

I measure weak scaling of the modified IES by using a problem of 253,405 structures with soil model of 3 layers under each structure. Here, I used 160 processes (20 nodes) of K computer to compute each case using MDOF model, and compute multiple cases with different input wave at bedrock simultaneously to measure the weak scalability of the program. Here, all input waves have N = 32,768 time steps, and one time step corresponding to the maximum response is outputted to the local file system using the rank directory scheme.

Table 3.5 summarizes the runtime and weak scalability of the program. We can see that the program scales well up to 2,000 cases (320,000 processes) with s_w = 97.4%. Fig. 3.19 shows the breakdown of runtime for 1 case and 1,000 cases. We can see that the computation time is nearly constant while the preparation time and file output time is increased. The preparation time increased from 14.3 s to 27.3 s, while the file output time increased from 4.77 s to 28.5 s.

The IES kernel does not perform any heavy computation and most of the runtime is spent on the plugged in analysis modules. Thus, the computational efficiency (ratio of executed FLOPS to the theoretical maximum FLOPS of hardware) of IES will be similar to that of the plugged in modules as long as proper weak scaling is attained. Table 3.6 shows the performance and efficiency for 1 case and 10^3 cases. These values are calculated as averages of the whole simulation, including I/O. The efficiency of the whole simulation is 1.88%, which is near the value of the SAA module with efficiency of 1.91%. Note that we did not perform any particular tuning of component analysis methods; improvement of compu-

tational efficiency is expected by optimization of each analysis module. Since I plan to combine many analysis modules in the future, it will be inefficient in terms of code development to optimize all the analysis modules; I plan to optimize analysis modules that have large ratio of runtime in the whole simulation process for effectively reducing the time-to-solution.

3.5.2 Strong scaling

I change the number of processes used per case to measure the strong scaling of the modified program. Here, strong scaling is defined as

$$s_{\mathrm{s}}(p) = \frac{T_1}{pT_p}, \tag{3.3}$$

where T_1 is the execution time using 160 processes (20 nodes) per case, while T_p is the execution time using p times the computing resources. I use the same target problem as in the previous section, running 10 cases at a time, using MDOF model for analysis.

Fig. 3.20 shows the speed up of program. We can see that the program scales well up to 1,280 processes (160 nodes) per case, with $s_{\mathrm{s}} = 90.5\%$. s_{s} gradually decreases with the increase in number of processes; s_{s} becomes 83.2% at 2,560 processes (320 nodes) per case. Fig. 3.21 shows the breakdown of runtime for 160 processes per case and 1,280 processes per case. We can see that the execution time of SAA and SRA are decreasing with the increase in the number of processes while the preparation time and file output time remains constant regardless of the number of processes. The execution time decreased almost linearly; meaning that the load balancing is performing well. In order to improve performance, we need to reduce the ratio of preparation time and file output time in the whole analysis time. I expect that the performance will improve when using OCM since execution time will increase while the preparation time and file output time will be the same as in the case of MDOF model. The preparation time consists of broadcasting of input files to all processes; one possible way of reduction of broadcast time and thus improvement of strong scaling performance is using advanced intra-node communication schemes such as KNEM [22]. Performance of broadcast will be important for system-capacity computing since large and frequent broadcasting of data is needed, and performance in such operations limit the scalability of the whole program; exploration in such fields will be beneficial.

3.6 Application example

For application, I modeled an 8.0 x 7.5 km area of downtown, Tokyo, consisting of 253,405 structures. Here, I used the National Digital Soil Map provided by the Japanese Geotechnical Society for modeling a 3 layered, local soil structure of the target domain [23]. The material properties and depth of each soil layer are shown in Table 3.7 and Fig. 3.22, respectively. Each structure is modeled from external shape data of structures [12]. I used the 1D equivalent linear method for soil amplification analysis, and MDOF model for the structural response analysis. 1,000 waves observed in Japan (KiK-net [24]) are inputted to bedrock of the model and the response of ground and structures are computed. Waves with short durations were padded with zeros so that all input waves have N = 32,768 time steps and time stepping dt = 0.01s. Computation was performed on 160,000 processes (20,000 nodes) of K computer and time to solution was 3,446 s.

Fig. 3.23 shows the response of structures. Here, I plot the maximum drift angle of structures, which is an index commonly used for evaluating structure damage. Fig. 3.23 i) shows the average response for the 1,000 cases, while Fig. 3.23 ii) shows the maximum response for the 1,000 cases. Fig. 3.24 shows the close-up view of the white box area in Fig. 3.23 ii). From these figures, we can see the diversity of earthquake disaster scenarios, which cannot be seen running only a few simulation cases. We can also see parts of the city that are vulnerable to many types of earthquake scenarios, or parts that have large response to certain types of earthquakes. In the same way, this method can be used to perform Monte Carlo simulation or sensitivity analyses for reflecting uncertainties of the input data of earthquake disaster estimates.

3.7 Summary

I studied the problems of current earthquake disaster simulation techniques for system-capacity computing on high performance computers by focusing on integration of two simulation modules and multiple observed earthquake data. The primary bottleneck for system-capacity computing was management of large data between simulation components. The flexible and easy-to-use file I/O based data movement hinders scalability, and thus large data must be kept on the memory of computation nodes throughout the whole simulation process to attain scalability. I rearranged the mapping of problem to processes so that

data between components can be passed without using the file system. Since large data is used throughout the simulation, I reduced the memory footprint by deallocating unnecessary intermediate data as soon as its purpose is finished.

The same approach can be used for other system-capacity computing problems, where multiple analysis modules are combined and executed for many cases. The current approach includes data transfer between components in the same process, but not between different processes. I plan to develop methods to make efficient data transfer between processes with the ease of use as that of the file system for extending the applicability of the method to a wider set of problems. Such extensions can lead to combination of multi-agent simulation of evacuation behavior after earthquakes.

Methods studied in this paper is planned to be used in the future IES, which computes many sets of multiple case simulations, see Fig. 3.25. Here, multiple cases of simulations are analyzed in sets, and the results of the previous set of simulations are used to generate input into the next set of simulations. By performing such analysis I plan to seek ways of improving reliability or quality of simulation results, such as finding extreme cases that can cause catastrophic damage to urban systems, or obtaining simulation results considering deterioration of structures which affects the resistance of structures to future earthquakes.

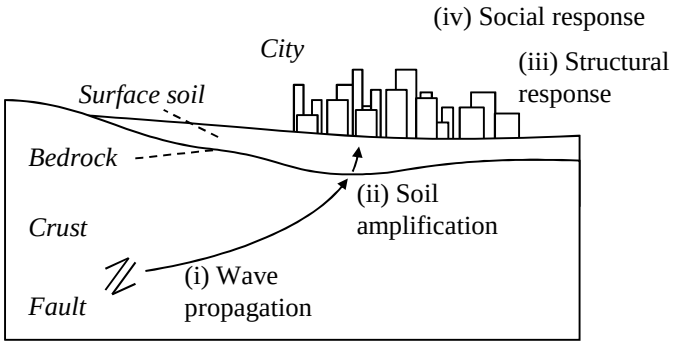


Figure 3.1: Earthquake disaster process

Table 3.1: Runtime and I/O size of SAA module. Values for evaluation points with 4 soil layers and 8,192 time steps.

Input	Computation	Output
Acceleration at bedrock: 98.3 KiB Soil model: <100 B	Average runtime on K (1core): 0.463 s	Acceleration at surface: 98.3 KiB Nonlinear response parameters: 1.3 KiB

Table 3.2: Runtime and I/O size of SRA module. Average values for 253,405 structures, 8,192 time steps.

Input	Computation	Output
Acceleration at surface: 98.3 KiB Structure model: <100 B	Average runtime on K (1core): 6.52 ms (MDOF), 7.69 s (OCM)	Displacement of nodes: 2.59 MiB Node and connectivity information, see Fig. 3.5

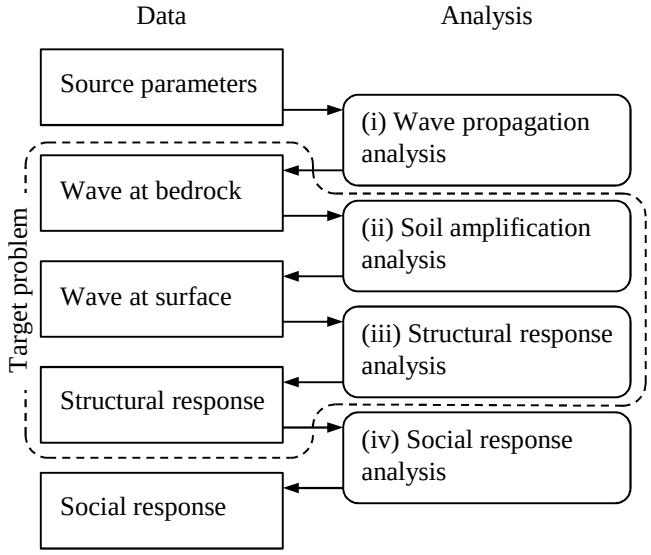
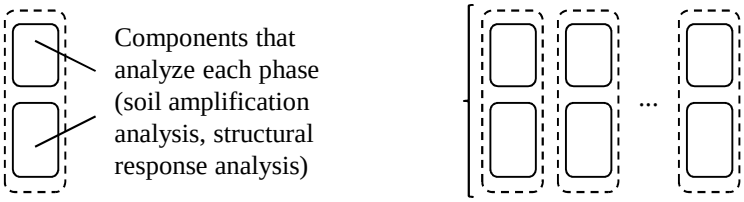


Figure 3.2: Flow of Integrate Earthquake Simulator (IES). The target problem of this study is soil amplification analysis and structural response analysis, shown in dashed lines.



i) Original IES.
Simulates one case of earthquake disaster.

ii) IES targeted in this study.
Computes multiple cases of earthquake disaster simulations.

Figure 3.3: Comparison of original IES and IES targeted in this study

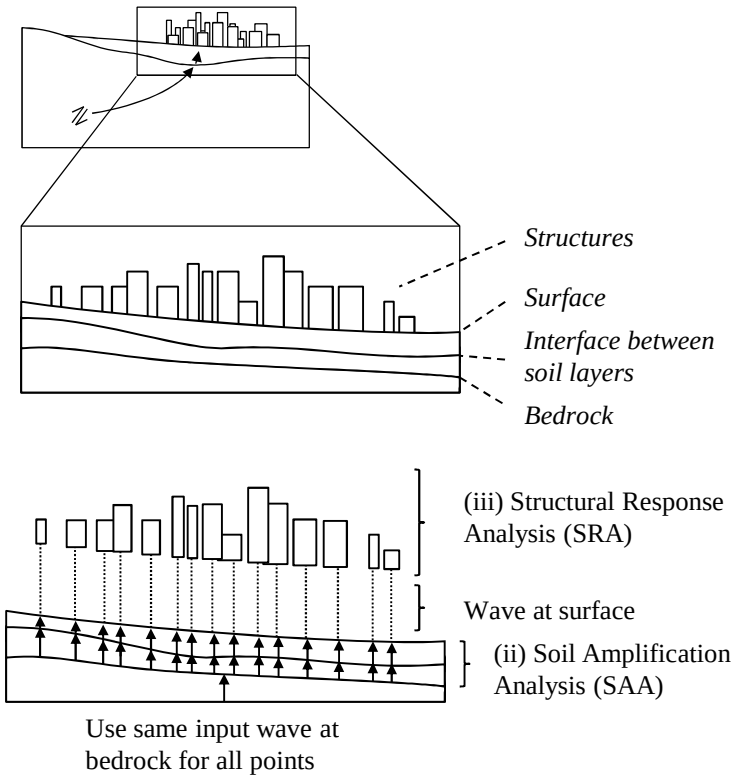


Figure 3.4: Target problem. Analyze soil amplification under each structure using same input wave at bedrock, and compute response of each structure in city. SAA and SRA of each point/structure is independent from each other.

Table 3.3: File I/O size of SAA and SRA modules. For a problem of 253,405 structures, 8,192 time steps, 1 case.

Analysis module	Data type	I/O	Data size
SAA	Wave at bedrock	Input	0.219 MiB
	Wave at surface	Output	25.2 GiB
SRA	Wave at surface	Input	25.2 GiB
	Structural response	Output	664.8 GiB

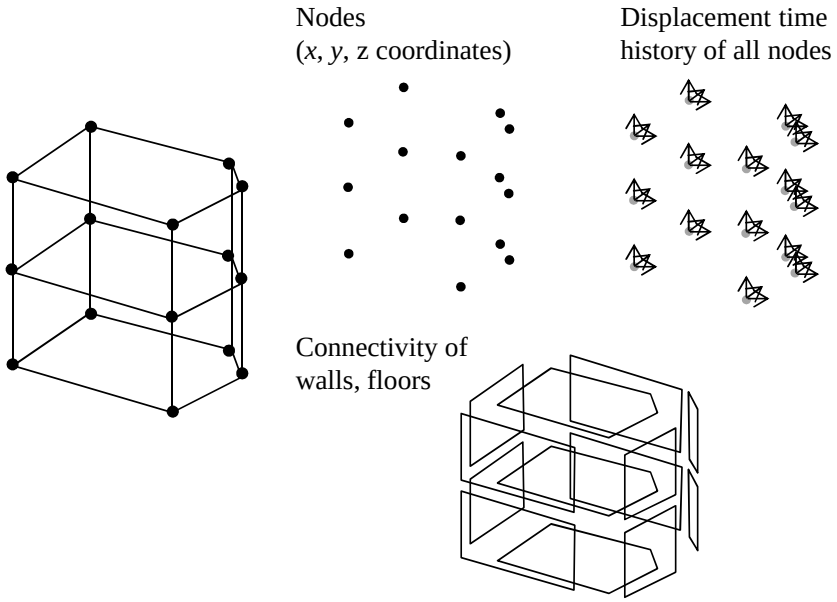


Figure 3.5: Output data format of original SRA module

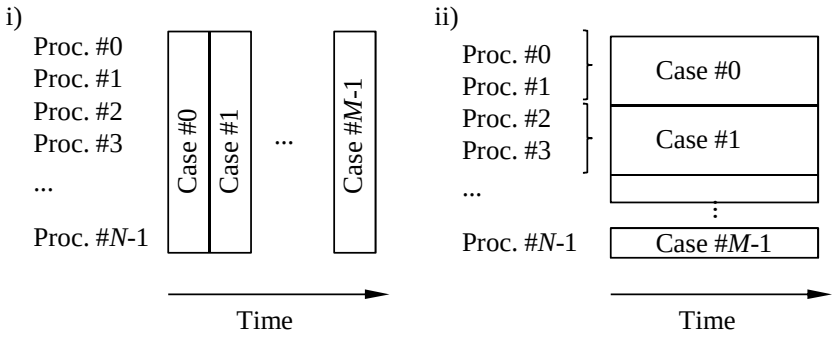


Figure 3.6: Performing many case simulations

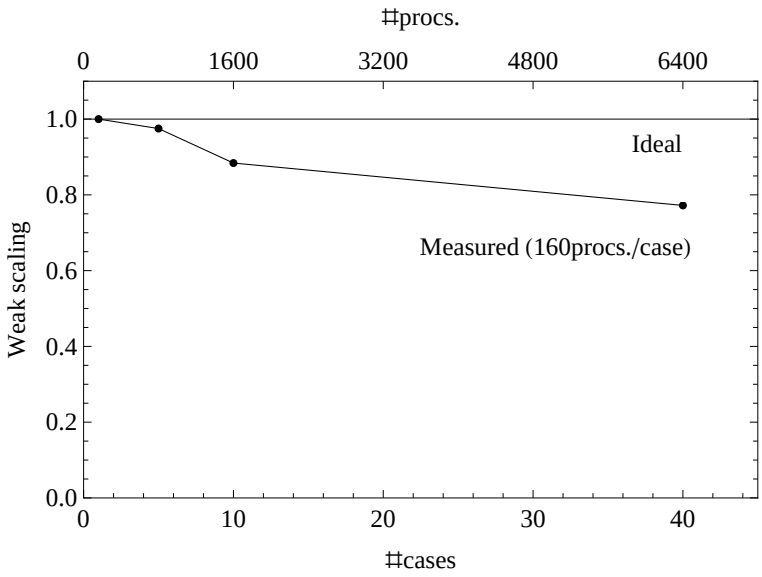


Figure 3.7: Weak scaling of original SAA module. N = 8,192 time steps, 253,405 evaluation points each with M = 3 soil layers.

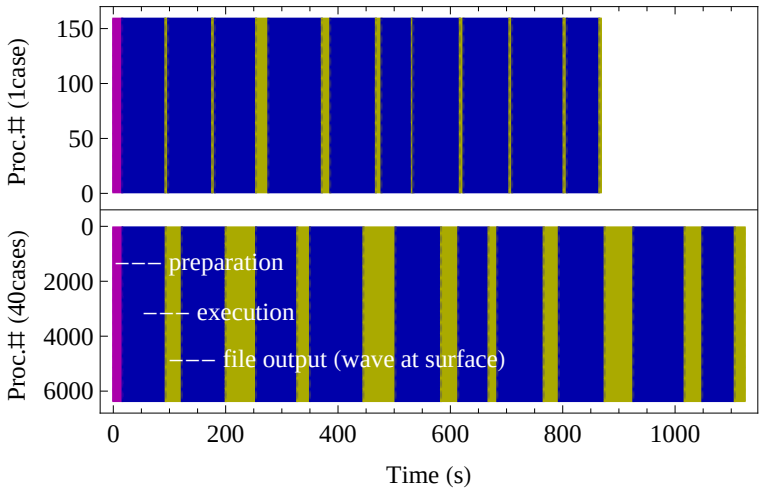


Figure 3.8: Time flow of original SAA module. N = 8,192 time steps, 253,405 evaluation points each with M = 3 soil layers.

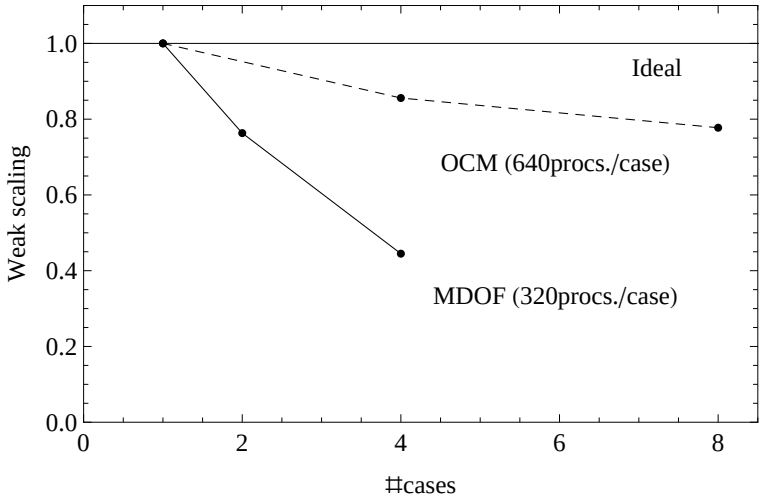
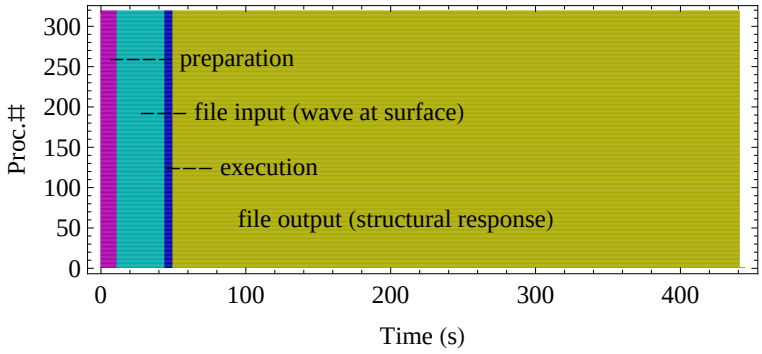
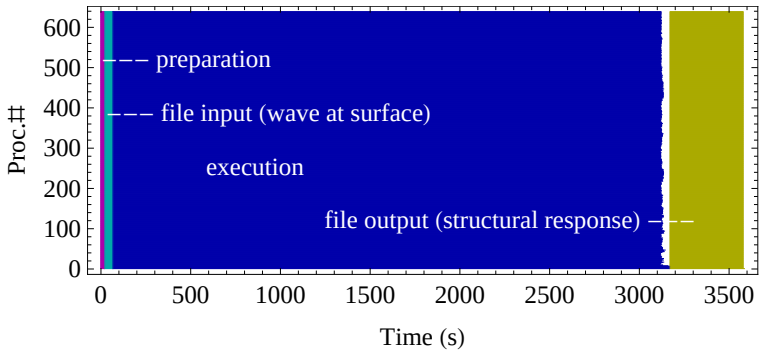


Figure 3.9: Weak scaling of original SRA module. N = 8,192 time steps, 253,405 structures.



i) MDOF



ii) OCM

Figure 3.10: Time flow of original SRA module. N = 8,192 time steps, 253,405 evaluation points.

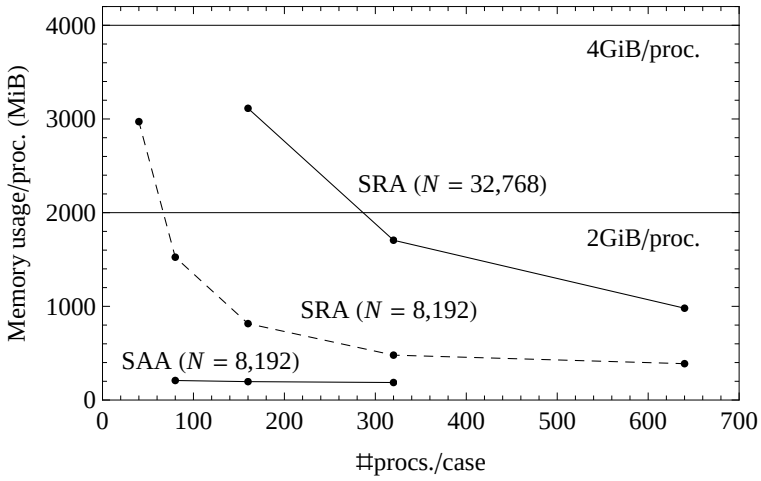


Figure 3.11: Memory usage of original IES for a problem of 253,405 points/ structures. MDOF model is used for SRA.

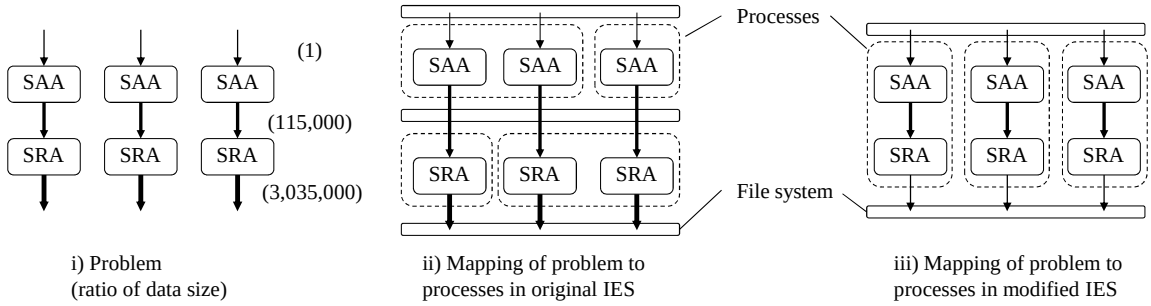


Figure 3.12: Mapping of problem to processes. Here thickness of arrows indicate the size of data passed between components.

Table 3.4: Output data size of structural response (253,405 structures, 8,192 time steps, 1 case). Max. val. indicates the maximum response, corresponding to data size of one time step.

Data format	Data size	Time steps
Original SRA module	664.8 GiB	8,192 (all)
Modified IES	139.3 GiB	8,192 (all)
	31.6 MiB	1 (max. val.)

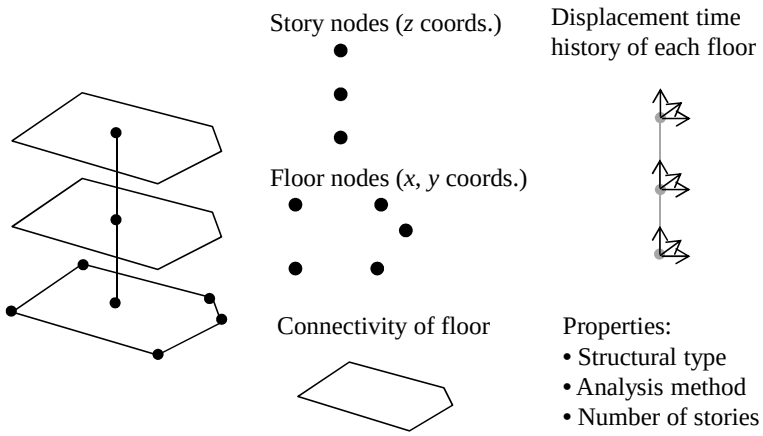


Figure 3.13: Output data format of modified IES

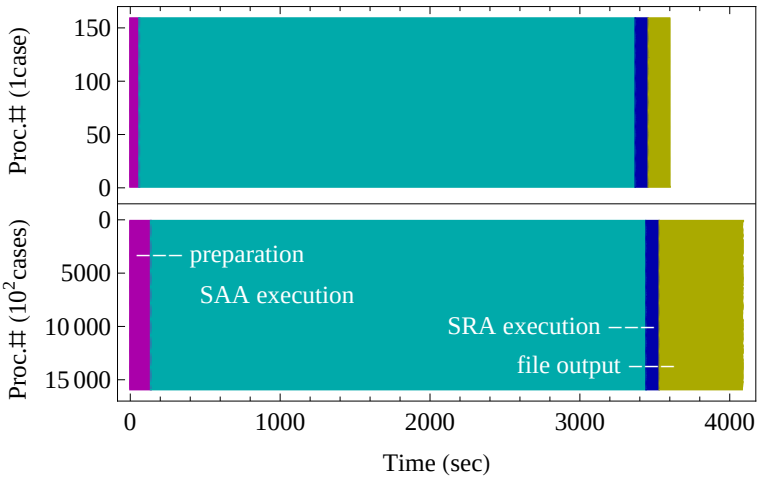


Figure 3.14: Weak scaling of preliminary program using modifications in Section 3.4.1. Measured on problem of 253,405 structures, $N = 32{,}768$ time steps with MDOF model using 160 processes (20 nodes) per case. Weak scalability is improved compared to the original SAA and SRA modules, but preparation time and file output time is still increasing with respect to the number of cases computed simultaneously.

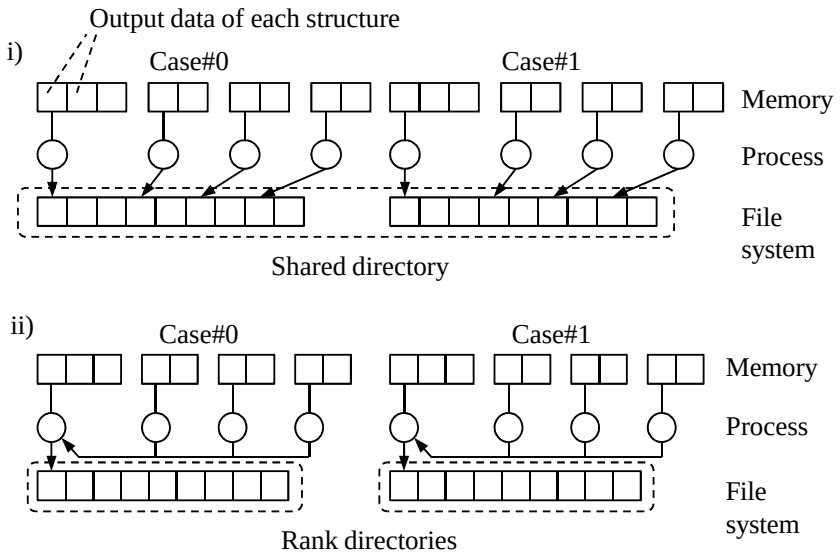


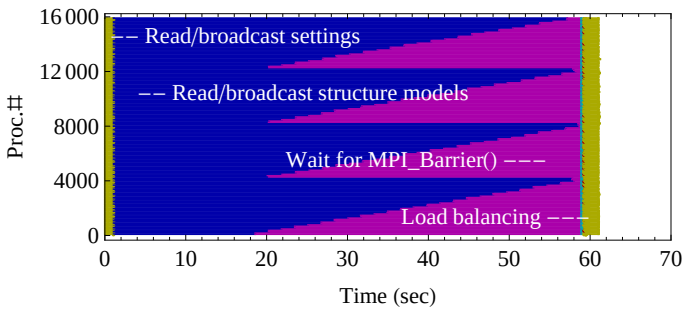
Figure 3.15: Change in file output scheme from shared directory, collective output to rank directory output

Table 3.5: Weak scaling of modified IES. Measured on problem of 253,405 structures, N = 32,768 time steps with MDOF model using 160 processes (20 nodes) per case.

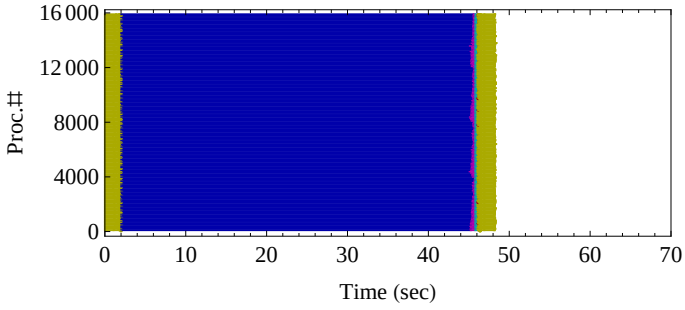
#cases	#procs. (#nodes)	Runtime (s)	Weak scaling
1	160 (20)	3,402.3	(1)
1,000	160K (20K)	3,446.2	0.987
2,000	320K (40K)	3,491.8	0.974

Table 3.6: Performance and efficiency of modified IES. The values are averages of the whole analysis including I/O.

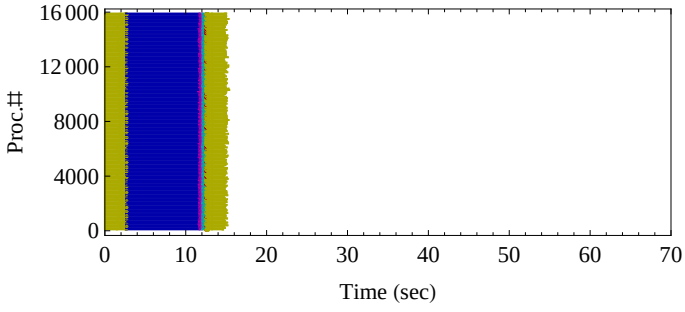
#cases	#procs. (#nodes)	Performance	Efficiency
1	160 (20)	48.1 GFLOPS	1.88%
10^3	160K (20K)	47.5 TFLOPS	1.86%



i) Original version. Structure model data separated into 40 files are read from the file system by the root process and combined to be broadcasted to all other processes in one message. `MPI_Barrier()` is set between the read/broadcast of structure models and load balancing operations.



ii) Pipelined version. Structure model files are read and broadcasted per file by the root process, leading to an overlap in file reading time and broadcast time.



iii) Pipelining + rank directory version. Structure model files are read by the root process of each case and broadcasted to all other processes in the same case using pipelining.

Figure 3.16: Reduction of preparation time of analysis

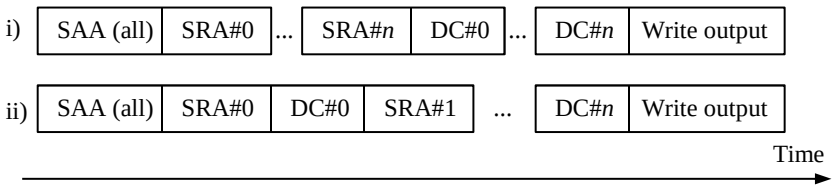


Figure 3.17: Reduction of memory footprint by changing the order of analysis. Data Conversion (DC) is done after SRA of each structure to deallocate intermediate memory.

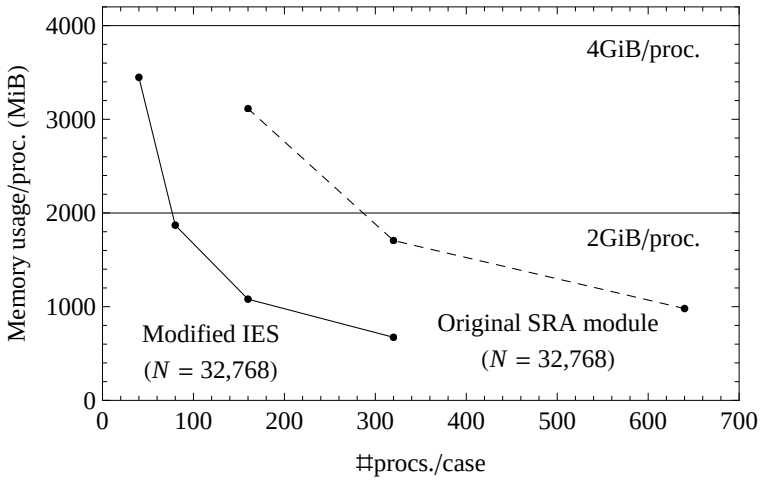


Figure 3.18: Comparison of memory usage of original SRA module and modified IES for a problem of 253,405 structures

Table 3.7: Material properties of soil layers

Layer#	V_s m/s	G_{max} tf/ms^2	ρ tf/m^3	h_{max}	γ_r
1 (clay)	115	26,640	2.0	0.15	0.20
2 (sand)	260	135,000	2.0	0.15	0.20
3 (bedrock)	500	500,000	2.0	0.05	0.20

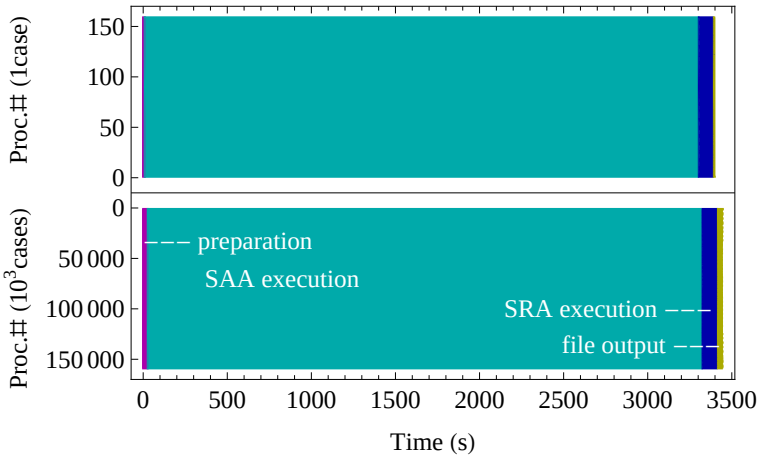


Figure 3.19: Time flow of modified IES. Measured on problem of 253,405 structures, N = 32,768 time steps with MDOF model using 160 processes (20 nodes) per case.

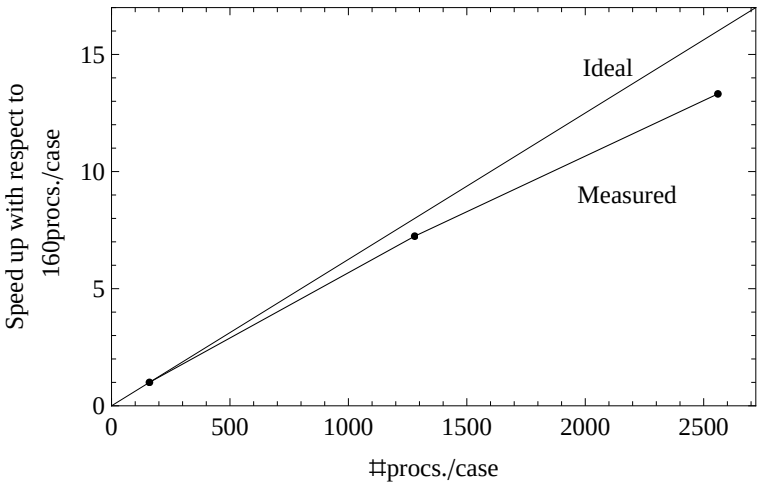


Figure 3.20: Speed up of modified IES. Measured on problem 253,405, N = 32,768 time steps, 10 cases using MDOF.

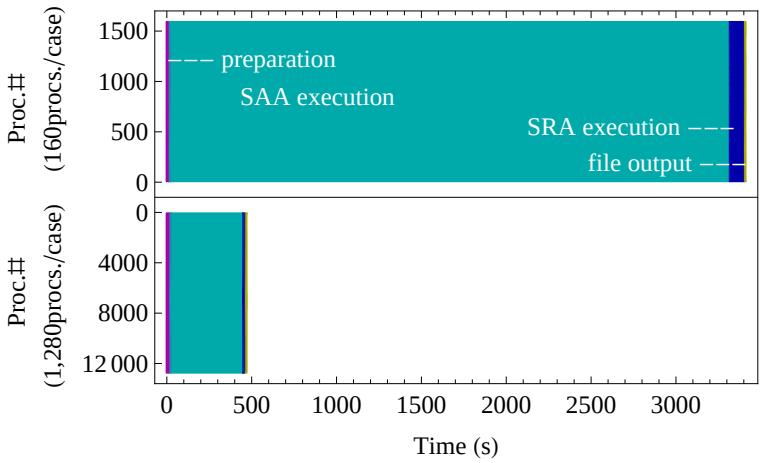


Figure 3.21: Time flow of modified IES. Measured on problem 253,405, $N = 32{,}768$ time steps, 10 cases using MDOF.

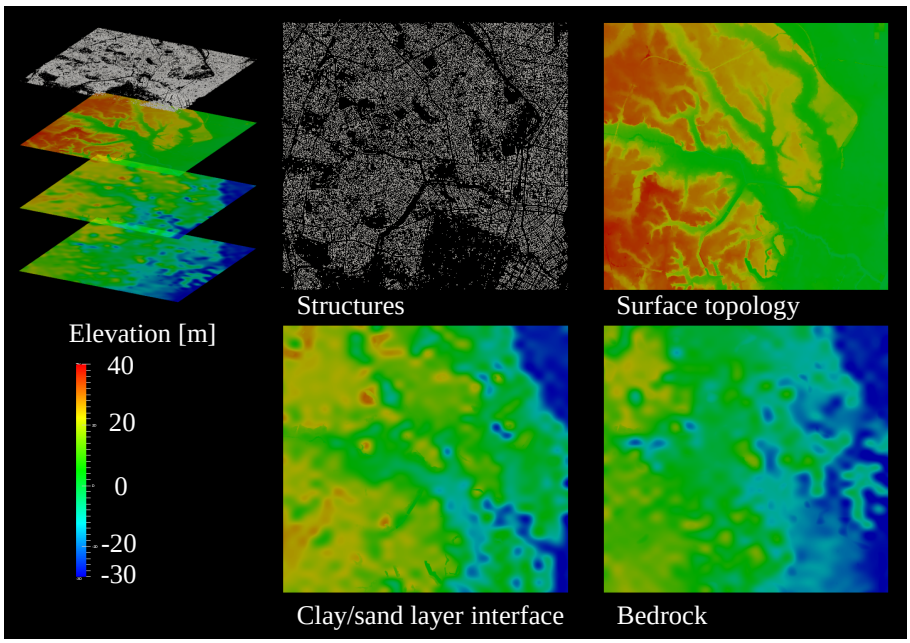
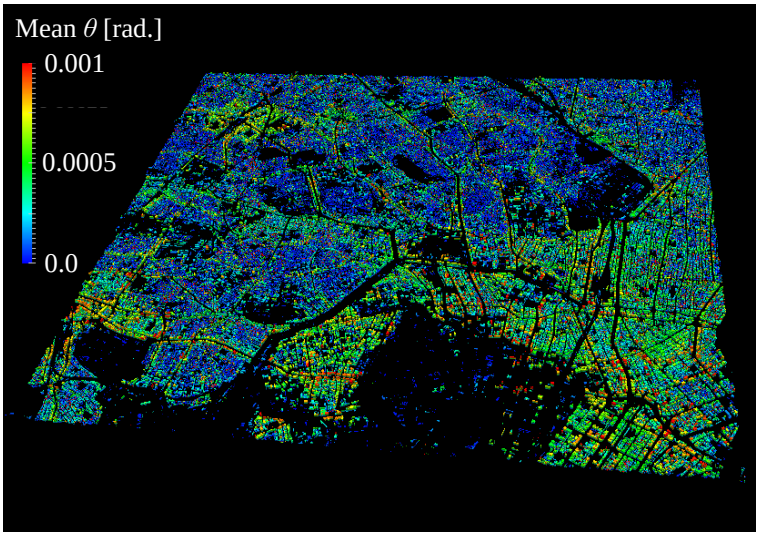
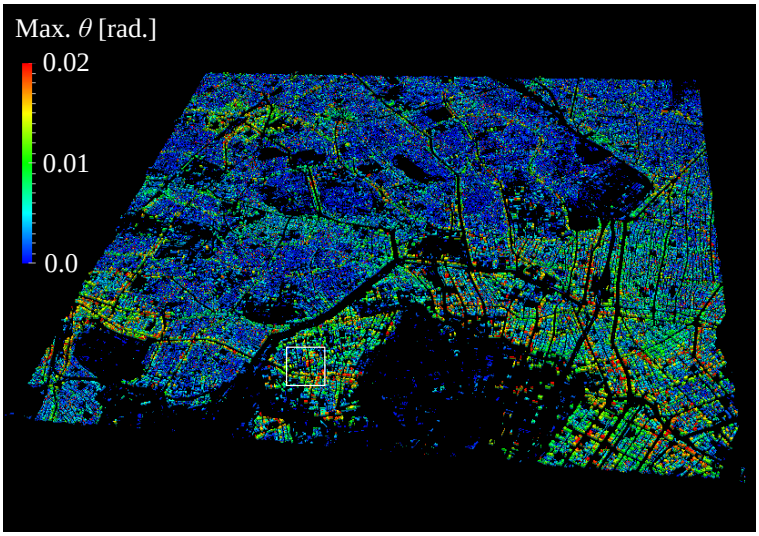


Figure 3.22: Configuration of target area. Consists of 3 soil layers and 253,405 structures.



i) Average response



ii) Maximum response

Figure 3.23: Structural response of 1,000 cases (θ: maximum story drift angle)

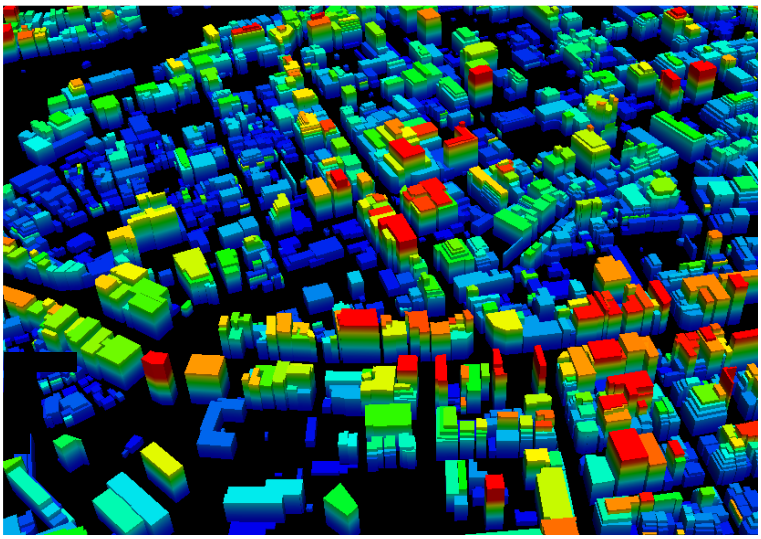


Figure 3.24: Closeup view of area in white box in Fig. 3.23. Colors indicate maximum displacement of structures of the 1,000 cases computed.

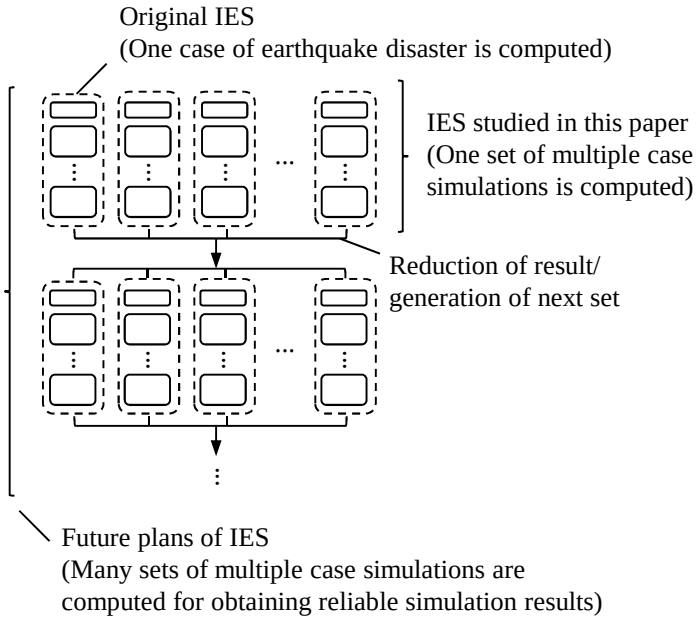


Figure 3.25: Future plans of IES as an example of system-capacity computing, and usage of this study in such simulation system

Chapter 4

Urban earthquake disaster simulation with 3D nonlinear ground motion analysis

4.1 Introduction

Estimating the responses of structures in a city is important in earthquake disaster estimation, as structure response heavily affects the human and economic losses from earthquakes. In addition, because modern cities work as a system, obtaining the distribution of structural damage in a city is important for optimizing a city's response to an earthquake disaster.

The natural frequencies of typical built structures are in the 10^{0-1} Hz range [25], and thus shallow soil structures have a large effect on structural damage due to earthquakes. The 3D geometries of shallow soil structures affect the surface response at such frequency ranges, and soft soils behave nonlinearly under strong ground motion. Therefore, it is important to model the 3D geometry of shallow soil structures with nonlinear soil constitutive models in frequency ranges of 10^1 Hz to analyze the responses of structures and estimate damage.

Common ways of evaluating responses at the surface, given the input at bedrock, are 1D or 2D numerical analysis with linear or nonlinear constitutive models (e.g., [26, 27]). For grounds with complex geometry, 3D focusing or reflecting of waves could not be considered with 1D and 2D analysis. Thus, the use of 3D ground motion simulations could improve the reliability of estimates.

The difficulties of performing 3D ground motion simulations are the high target frequency of structures and large target areas, leading to large-scale computation. For example, a target domain of size $1.0 \times 1.0 \times 0.1$ km with $V_s = 100$ m/s and target frequency of 10 Hz with time duration of 100 s leads to a degree of freedom (DOF) of 10^{7-8} and 10^{4-5} time steps. Highly scalable parallel analysis code running on modern supercomputers is needed to solve such problems. It is also not straightforward to make high-quality mesh of analysis models for large DOF problems with complex 3D geometries. Another difficulty is

the modeling of large numbers of structures in the target domain; manual modeling of structures is very laborious and impractical for urban simulations with many structures.

Several studies using 3D ground motion analysis and structure models of a city are reported in the literature. In studies by Krishnan et al. [8], ground motion was computed using the spectral element method, and recorded at a few hundred sites. The recorded waves were inputted into two 18-story building models with natural periods of 4-5 s, hypothetically located at each site to analyze the site effect of the building response. In studies by Richardo and Bielak [7], structures in a city were modeled and analyzed together with the surface ground model, enabling the evaluation of soil-city interaction effects under earthquake excitation. The structure models used here are simplified by the using octree mesh with equivalent material properties. The generation of detailed structure models of a city and analysis of those models with target frequency ranges of mid to low-rise structures is a task for future work.

To solve the problem of modeling many structures in a city, structure model generation methods have been developed in the past (e.g., IES [11, 12]). In such methods, robust data conversion methods are applied to available geospatial data to automatically generate structure models. By using such methods, millions of structures can be modeled without manual manipulation. Conversely, it is not common to be able to access complete geospatial information related to structures (e.g. floor plan and beam and column configuration) to make seismic analysis models. Thus, missing information is estimated so that it follows the design code of buildings. Such estimation leads to differences between the actual built structure and the generated structural model.

In this study, I utilize the recent advances in 3D nonlinear FEM analysis for increasing the target frequencies of ground motion analysis of surface soil structures to 15 Hz so that the computed ground motion can be used for analyzing low-rise structures with short natural periods. I also perform many-case structure response simulations to account for the uncertainties in the structural models. First, a fast and robust modeling method is used to generate several analysis meshes with increasing target frequencies, and the generated models are analyzed for obtaining convergence in terms of ground response and structural response. The K computer, a massively parallel supercomputer, is used for performing the large-scale analysis in a short time. The response is compared with that of 1D ground motion analysis to examine the 3D effect of surface soil structures on the response of structures in a city. To consider the uncertainties in structure modeling, many structure models are generated with structural parameters following a measured distribution, and are analyzed to estimate the distribution of structure response.

4.2 Methodology

4.2.1 Ground motion analysis

Analysis method

The Finite Element Method (FEM) with 3D solid elements for ground motion analysis is used, as it is suitable for analyzing boundary value problems with complex geometry and satisfies the stress-free condition analytically. By discretizing the equation of elasto-dynamics with FEM in the spatial direction, we obtain

$$\mathbf{Ku} + \mathbf{Cv} + \mathbf{Ma} = \mathbf{F}, \tag{4.1}$$

where **K**, **C**, and **M** are the stiffness, damping, and mass matrix, and **u**, **v**, **a**, and **F** are displacement, velocity, acceleration, and external force vectors, respectively.

For time integration, the Newmark-β method with β=1/4 and δ=1/2 is used. Semi-infinite absorbing boundary conditions are used for the bottom and side boundaries of the simulation domain, leading to

$$\begin{aligned}\left(\frac{4}{dt^2}\mathbf{M} + \frac{2}{dt}\mathbf{C}^n + \mathbf{K}^n\right)\delta\mathbf{u}^n = \\ \mathbf{F}^n - \mathbf{Q}^{n-1} + \mathbf{C}^n\mathbf{v}^{n-1} + \mathbf{M}\left(\mathbf{a}^{n-1} + \frac{4}{dt}\mathbf{v}^{n-1}\right),\end{aligned} \tag{4.2}$$

with

$$\begin{cases}\mathbf{Q}^n = \mathbf{Q}^{n-1} + \mathbf{K}^n\delta\mathbf{u}^n, \\ \mathbf{u}^n = \mathbf{u}^{n-1} + \delta\mathbf{u}^n, \\ \mathbf{v}^n = -\mathbf{v}^{n-1} + \frac{2}{dt}\delta\mathbf{u}^n, \\ \mathbf{a}^n = -\mathbf{a}^{n-1} - \frac{4}{dt}\mathbf{v}^{n-1} + \frac{4}{dt^2}\delta\mathbf{u}^n,\end{cases} \tag{4.3}$$

where $\delta\mathbf{u}$ is incremental displacement, dt is time increment, and superscript n indicates the time step.

For nonlinear constitutive relations of the soil, the modified Ramberg-Osgood model [28] and Masing rule [29] is used. Rayleigh damping is used for damping. Here, the element damping matrix $\mathbf{C}_e^n$ is calculated as

$$\mathbf{C}_e^n = \alpha\mathbf{M}_e + \beta\mathbf{K}_e^n,$$

where $\mathbf{M}_e$ and $\mathbf{K}_e^n$ indicates the element mass matrix and the element stiffness matrix, respectively. The coefficients α and β are determined by solving the least-squares equation,

$$\text{minimize}\left[\int_{f_{\min}}^{f_{\max}}\left\{h^n - \frac{1}{2}\left(\frac{\alpha}{2\pi f} + 2\pi f\beta\right)\right\}^2 \mathrm{d}f\right],$$

where $f_{\max}$ and $f_{\min}$ are the maximum and minimum target frequencies, and h^n is the damping ratio at time step n. Note that the damping matrix is calculated at every time step, since the stiffness and damping ratio changes with time.

In summary, the calculation proceeds as

1. Compute the stiffness and damping ratio for n-th time step using the Ramberg-Osgood model and Masing rule with the strain obtained at the $n-1$-th time step.
2. Compute $\mathbf{K}^n$ and $\mathbf{C}^n$ using the stiffness and damping ratio for the n-th time step.
3. Compute $\delta\mathbf{u}^n$ by solving Eq. (4.2) and update the values in Eq. (4.3) using $\delta\mathbf{u}^n$.

Most of the computational cost is spent on solving the matrix equation in Eq. (4.2) for obtaining $\delta\mathbf{u}^n$ every time step. In order to perform the large scale analysis in short time, several methods are used to speed up the linear solver. Here, the preconditioned conjugate gradient method, with multigrid method, mixed precision arithmetics, Block-Jacobi method and the predictor-corrector method is used. The compressed row storage method is used for storing the coarse mesh, while the element-by-element method is used for the fine mesh. The computation is performed in parallel using OpenMP together with MPI.

Modeling method

Using typical mesh generation methods for making large FEM models with complex 3D geometry often leads to mesh with elements of high aspect ratios. Such high aspect ratio elements deteriorates the accuracy of numerical results, and thus it is common to manually tune the mesh to improve the mesh quality in such cases. Such manipulation can take long time for large scale models and thus a fully automated, high quality mesh generator is desirable.

I use a modified version of the method of [30] for constructing the 3D mesh, with the soil structure defined as a layered medium. In this method, a structured grid is used to separate the simulation domain into non-overlapping cells, and elements are generated individually in each cell with small aspect ratios, see Fig. 4.1. Since the generation of elements in each cell is performed individually, the mesh generation can be performed in parallel. This leads to a fast, fully automated, robust mesh generator which is capable of generating high quality mesh.

The original method in [30] generates a hybrid mesh with linear tetrahedral and structured hexahedral elements; I convert the generated mesh to be used for the multigrid solver. I first convert the hybrid mesh

to mesh with linear tetrahedral elements, which is used for the coarse mesh of the multigrid solver, see Fig. 4.1 e). Next, intermediate nodes are inserted to the edges of the linear elements to make a mesh with second ordered tetrahedral elements, which is used for the fine mesh of the multigrid solver.

4.2.2 Structural response analysis

Analysis method

A Multi-Degree Of Freedom (MDOF) model is used for analysis of structures. Here, each floor of a structure is modeled as a node with mass, with linear shear springs with dampers connecting the nodes in the two horizontal directions. The model is decomposed into eigenmodes and calculated per mode in the frequency domain to enable stable and accurate analysis with large time stepping dt.

Modeling method

The properties needed in the MDOF model is the mass of each floor, and stiffness and damping of springs connecting each floor. Since only the external shape of a structure is given in the GIS data used in this study, these properties are guessed based on the area and height of the structure using the method described in [12]. In this method, a smallest rectangle that surrounds the base of the structure is generated, and the number of floors are estimated using typical floor story heights (Fig. 4.2). The local Cartesian coordinate system X-Y-Z is set in the directions of rectangle edges. The mass of each node is estimated using mass to floor-area relation for typical structures and area of the rectangle, and the stiffness of shear springs are set so that the natural period of the structure T (s) follows

$$T = 0.02H, \tag{4.4}$$

using the height of the structure H (m). Analysis is performed in the X and Y directions of the local coordinate system.

4.3 Problem settings

For application, I target an 1,250 x 1,250 m domain of central Tokyo, consisting of 3 soil layers and 4,066 structures. The soil layers consist of volcanic ash clay, clay, and gravelly soil bedrock; Table 4.1 shows the material properties of each layer, and Fig. 4.3 shows the elevation of the interface between the layers.

The layer interfaces are interpolated from 250 m mesh boring log of National Digital Soil Map provided by The Japanese Geotechnical Society [23], while the material properties of each layer is taken from a nearby boring hole data of K-NET observation station (TKY007, Shinjuku) [24]. The top layer is modeled as nonlinear material, while the other two layers are modeled as linear materials. The bottom of the model is set to elevation of -70 m, leading to a model with thickness of 100 – 110 m. The ground model is generated with target frequencies $f_{\max}$ = 5 Hz, 10 Hz, and 15 Hz for checking the convergence of ground and structure response. The models are made such that one shear wavelength ($V_s / f_{\max}$) is discretized with 5 elements for linear soils and 10 elements for nonlinear soils, leading to a minimum element size of 3.0 m, 1.5 m, and 1.0 m mesh for the 5 Hz, 10 Hz, and 15 Hz model, respectively. The properties of each model is summarized in Table 4.2. Fig. 4.4 shows the generated mesh for 15 Hz model. We can see that the three dimensional shape of surface topography and layer interfaces are modeled in the the mesh. The modeled structures are 1 to 19 stories high, and is distributed in the domain as shown in Fig. 4.3. There are some high-rise structures in the North-West side of the domain, many small low-rise structures are in the Eastern part of the domain, and some large mid-rise structures in the South-West side of the domain.

I used two input waves for analysis; the 1995 Kobe wave (observed in Japan Meteorology Agency, Kobe, Japan; Jan. 17th, 1995 [31]) and 1968 Hachinohe wave (observed in Hachinohe, Japan; May 16th, 1968 [32]), as an example of inland earthquakes and ocean trench earthquakes, respectively. Both of the waves are observed at surface; I pulled back the waves using 1D equivalent linear method to the bottom of the 3D model using soil properties in Table 4.1. The wave is inputted as velocity wave into the bottom of the model. Fig. 4.5 shows the time history of the input waves at bedrock. Frequency components below 0.1 to 0.2 Hz (linear filter) are filtered out for removing low frequency noise in the observed acceleration records. Both of the records are padded with zeros to have durations of 60 s. We can see that the Kobe wave has relatively large high frequency components while the Hachinohe wave has relatively large low frequency components.

For analysis, we use the Rayleigh damping matrix set for the frequency range of 0.1 – 10 Hz. Time stepping dt = 0.001 s is used for analysis, leading to 60,000 time step problem. The elapsed time and number of nodes used for each analysis case is summarized in Table 4.3. We can see that the program scales well in this setting, and that the 1 billion DOF 15 Hz model can be computed by 11 hours using 8,192 compute nodes of the K computer. Here, displacement at surface nodes are outputted every 10 time steps (every 0.01 s) and displacement under each structure is outputted every time step. The time for

performing I/O takes less than 15% of the total elapsed time.

4.4 Convergence of ground and structure response

I first check the convergence of 3D ground motion analysis by comparing the ground and structural response using the 3 ground models.

Figures 4.6 and 4.7 show the time history waveforms at points A and B located at the surface. Point A is located in an area with relatively thick soft layer, with the geometry of the bottom of the soft layer shaped like a bowl (see the bottom left figure in Fig. 4.3). Point B is located in an area with relatively thin soft layer with flatter layer interfaces (see the bottom left and bottom right figure in Fig. 4.3). From Figures 4.6 and 4.7, we can see that the difference of acceleration waveforms between the 15 Hz and 10 Hz model (indicated in green) are smaller than that of the difference between 15 Hz and 5 Hz model (indicated in red) for all directions in both points, and thus the numerical solution is converging with the increase in the resolution of the mesh. Figures 4.8 and 4.9 show the velocity response spectra (damping ratio $h = 0.05$) at the two points for the two input waves. The response of the 15 Hz model the 5 Hz model has small differences in the high frequency range (approximately above 5 Hz), but the difference between the response for 15 Hz model and the 10 Hz model cannot be seen from the figures; the numerical solution is well converged for this index when using the 15 Hz model. By comparing the response at points A and B, we can also see that wave components between $0.5 \sim 2.0$ Hz is larger at point A while the components between $0.2 \sim 0.5$ Hz is larger at point B. Such amplification patterns confirm to the amplification patterns expected from the natural frequencies of the surface soil layer, where places with thick surface soil layer have longer natural periods and vice versa.

I further check the convergence of the analysis by comparing the spatial distribution of ground response at the surface of the domain using the Seismic Intensity (SI) index [33], which is an index commonly used for estimating the destructiveness of strong ground motion for structures. The SI index is defined as

$$\mathrm{SI} = \frac{1}{2.4} \int_{0.1}^{2.5} S_{\mathrm{v}}(T)\, \mathrm{d}T, \tag{4.5}$$

where $S_{\mathrm{v}}(T)$ is the velocity response of a linear single degree of freedom system with natural period of T (s) and damping ratio $h = 0.2$. Fig. 4.10 show the distribution of horizontal magnitude of SI values for the Kobe wave. We can see that the difference between the results of the 15 Hz model and the 10 Hz model is

smaller than the difference between the 15 Hz and 5 Hz model. The difference between the 15 Hz model and the 10 Hz model is under 0.7%; the numerical solution is well converged for this index when using the 15 Hz model. Fig. 4.11 shows the distribution of horizontal magnitude of SI values for the Hachinohe wave. Again, we can see that the difference between the 15 Hz and 10 Hz models are smaller than that of the 15 Hz and 5 Hz models, and that the difference between the results of the 15 Hz model and the 10 Hz model is under 1.2%. By comparing the differences between the three models in the Hachinohe wave case, we can see that the difference between the 5 Hz and 15 Hz models have longer spatial components compared with that of 10 Hz and 15 Hz models. This indicates that the 5 Hz model still have errors in the long period range, while such errors are decreased when using the 10 Hz model. While we can see some similarity between the SI value distribution of 15 Hz models for the two input waves, such as large SI values near point A and in the South-East corner, the SI distribution patterns are different between the two cases. For example, some high spatial frequency fluctuations can be seen in the case of Hachinohe wave, which is not seen in the Kobe wave.

Next, I check the convergence of structure response by comparing the displacement response of structures A and B, which are located above points A and B, respectively (see Figures 4.3 and 4.4). Both of the structures are three story buildings. Figures 4.12 and 4.13 show the relative displacement response, which is the difference between the displacement at top of the structure and ground level, for each structure and input wave. We can see that the difference of displacement response between the 15 Hz model and 10 Hz model is smaller than that of the 15 Hz and 5 Hz models, and that the displacement response is well converged when using the 15 Hz model for all the directions for both points.

In order to see the convergence of structural response of structures in the domain, I compare the maximum story drift angle of structures. The maximum story drift angle is an index commonly used for estimating damage of a structure under seismic loading, and is defined as

$$\theta = \max_{i,t} \frac{|\mathbf{u}_{i+1,t} - \mathbf{u}_{i,t}|}{H_i}, \tag{4.6}$$

where $\mathbf{u}_{i,t}$ is the time history displacement of the i-th floor at time step t, and H_i is the floor height of story i. From Figures 4.14 and 4.15, we can see that the difference of distribution of structural response using the 15 Hz model and 10 Hz model is small, compared with that of the difference between the 15 Hz and 5 Hz model. We can also see that the response of tall structures located in the North-West corner of the domain is small for Kobe wave but large for the Hachinohe wave, corresponding to the input waves' components.

From these observations, we can see that the numerical solution is converging with respect to the increase in discretization resolution, and that the solution obtained using the 15 Hz model is converged in the level of engineering interest. Such properties of the numerical solution implies that the mathematical problem and numerical discretization is separated, and the mathematical problem is properly solved with the increase in numerical modeling resolution.

4.5 Comparison of results for 3D and 1D ground motion analysis

I compare the 3D analysis results with 1D analysis results to see the effect of the three dimensional ground structure. The 1D analysis is performed in the EW and NS directions using the soil properties directly under each point and structure. For a problem with horizontally stratified soil layers, the 3D wave propagation problem can be decoupled in the vertical and two horizontal directions, leading to the same results as that of 1D analysis; the differences between the 1D and 3D analysis results reflects the three dimensional geometry of the soil structure. The 1D analysis is performed with 1D FEM with numerical resolution assured up to 20 Hz using the same nonlinear constitutive model, material properties, input wave, and Rayleigh damping settings used in the 3D FEM analysis.

Figures 4.16 and 4.17 show the spatial distribution of horizontal magnitude of SI index at surface. We can see that most of the regions have similar values with difference of a few % between 1D and 3D analysis, but regions near point A have larger response in the 3D analysis by about 15%. This could be due to the focusing effect of the bowl shaped geometry of the layer interface between the first and second layers. There is also a large difference in the South-East corner of the domain in the Kobe wave case (see Fig. 4.16); a similar bowl shaped geometry is located under it, see Fig. 4.3. Although such simple interpretations can be useful for estimating the differences between 1D and 3D analysis, we can see that the difference between the 1D and 3D analysis have different patterns for the Kobe wave and Hachinohe wave; and thus it is not straight forward to estimate response considering 3D geometry using 1D analysis. For example, the differences between 1D and 3D analysis in the South-East corner is small in the Hachinohe wave case, which is quite different from the case in the Kobe wave where results of 1D and 3D analysis have large differences. Such different patterns could be dependent on the frequency components of input waves and nonlinearity effects.

Next, I compare the spatial distribution of maximum story drift angles in Figures 4.18 and 4.19. We can

see that some structures have more than 10% difference in response near structure A and in the South-East corner of the domain, which corresponds to the difference in ground response at surface. We can also see that some structures' response does not change much even if they are located in regions where the ground response is largely changed; reflecting the diversity of structural properties (e.g. natural frequencies) of each structure.

In order to compare the statistical differences between the 1D and 3D analysis results, I compare the histogram of maximum story drift angle of structures in the domain in Fig. 4.20. We can see that the 1D analysis underestimates the number of structures with large response (maximum story drift angles larger than 0.034 for Kobe wave and 0.0105 for Hachinohe wave).

4.6 Considering uncertainties in structure models

Since the geospatial data used for making structure models does not include the complete information needed to generate structural models for seismic response analysis, missing information is estimated based on the building design code and properties of typical built structures. Such estimation of models leads to uncertainty in the structure model. In this section, I assume that the uncertainty in the structure model parameters follow the normal distribution, and generate/analyze many models that follows such distribution to estimate the effect of uncertainties in the structural models to the output structural response.

In the previous applications, structures' properties were estimated using the statistical regression of measured story height to natural period database reported in [25], see Fig. 4.21. Here, the natural periods of structures T (s) are estimated using Eq. (4.4), using the height of a structure H (m) which is provided in the geospatial database of Tokyo area. The stiffness of the structures are set to match the estimated natural periods.

As we can see in Fig. 4.21, actual structures does not strictly follow Eq. (4.4) but have a distribution around it. In this section, I generate many structure models by estimating the natural period of a structure using

$$T = \alpha H, \tag{4.7}$$

where, α are samples of the normal distribution fitted to the measured distribution of parameter T/H, shown in Fig. 4.22, with average $\mu = 0.0195$ (s/m) and standard deviation $\sigma = 0.00473$ (s/m).

For analysis of the output, I use the maximum story angle θ in Eq. (4.6). Since the structural response

system S:

$$T \rightarrow \boxed{S} \rightarrow \theta, \tag{4.8}$$

is nonlinear, I seek the convergence of structure response distribution by increasing the number of estimated structure models following the normal distribution.

The Kobe wave (3D, 15 Hz model) and Hachinohe wave (3D, 15 Hz model) are inputted into the generated models. Figures 4.23 and 4.24 show the distribution of maximum story angle of structure A for 10, 100, 1,000 and 10,000 case models for the two input waves. We can see the distribution converge with respect to the increase in the number of cases for both of the waves. Even though the input is a normal distribution, we can see that the output distribution has a complex shape, and the converged values are different for the two waves. This can be due to the complex shape of the input wave spectrum; changes in the natural period of the structure changes the frequency ranges of the input wave that is critical to the structure, and thus the output response has a complex distribution.

Such analysis can be done for all the structures in the domain using capacity computing methods on high performance computers. Fig. 4.25 shows the spatial distribution of structural response for the 10,000 cases using the Kobe wave as an input. Here, we show the mean, maximum and standard deviation of the response. From the figure, we can see complex spatial patterns of the statistical indexes. For example, most of the structures near structure A and the South-East corner have large maximum response while only a part of the structures in that area have large mean response.

4.7 Summary

I performed 3D nonlinear ground response analysis of surface soil structures, and inputted the computed ground motion to seismic response analysis of structures. The analysis is performed by utilizing a highly scalable, parallel 3D nonlinear FEM method, with automated structure modeling using geospatial data.

By analyzing ground models with different mesh size, the numerical solution converged in terms of ground and structural response using the 3D ground model with 15 Hz target frequency for the two waves tested. The analysis results are compatible to simple estimations of ground motion such as the natural frequency of surface soil layer, but there was up to 15% difference between the results obtained by 1D ground motion analysis for some areas in the domain. Estimating the ground and structural response considering full 3D geometry with nonlinear soil properties is not straight forward using 1D ground motion

analysis, and large scale 3D ground motion simulation with sufficiently high target frequencies is useful for such detailed analysis.

To further improve the reliability of structural response estimation, many structure models with uncertainties following the normal distribution were analyzed to estimate the output response distribution. Such analysis show that the earthquake response system is complex even in the linear structural response range and thus the use of many case analysis can be effective for estimating the output response distribution. In addition to using the statistical indexes (e.g., maximum, mean, and standard deviation) for visualizing the response distribution of structures in a domain, I plan to develop methods to organize and convey more of the case-wise and spatial-wise distribution data obtained in the analysis in the future.

Table 4.1: Material properties of ground. 1st layer is modeled with nonlinear properties, while the 2nd layer and bedrock is modeled with linear properties.

	V_p (m/s)	V_s (m/s)	ρ (kg/m^3)	$h_{\max}$ or h	γ_r
1st layer	1,210	150	1,500	0.25	0.005
2nd layer	1,380	255	1,800	0.05	∞
Bedrock	1,770	490	1,900	0.005	∞

Table 4.2: Model properties

	$f_{\max}$ (Hz)	min ds (m)	#nodes	#elements	DOF
5 Hz model	5.0	3.0	17,796,006	12,998,213	53,388,018
10 Hz model	10.0	1.5	112,274,285	82,885,697	336,822,855
15 Hz model	15.0	1.0	340,876,783	252,737,051	1,022,630,349

Table 4.3: Runtime information using the K computer. Time step discretization $dt = 0.001$ s with 60,000 time steps is used for all of the cases.

Input wave	Model	#compute nodes	Total elapsed time (s)	I/O time (s)	I/O size (GB)
1995 Kobe JMA	5 Hz model	512	19,574	765.1	73.3
	10 Hz model	4,096	22,967	2,361.1	312.0
	15 Hz model	8,192	39,756	4,879.5	761.1
1968 Hachinohe	5 Hz model	512	17,020	647.1	73.3
	10 Hz model	4,096	22,543	2,878.5	312.0
	15 Hz model	8,192	38,517	5,102.1	761.1

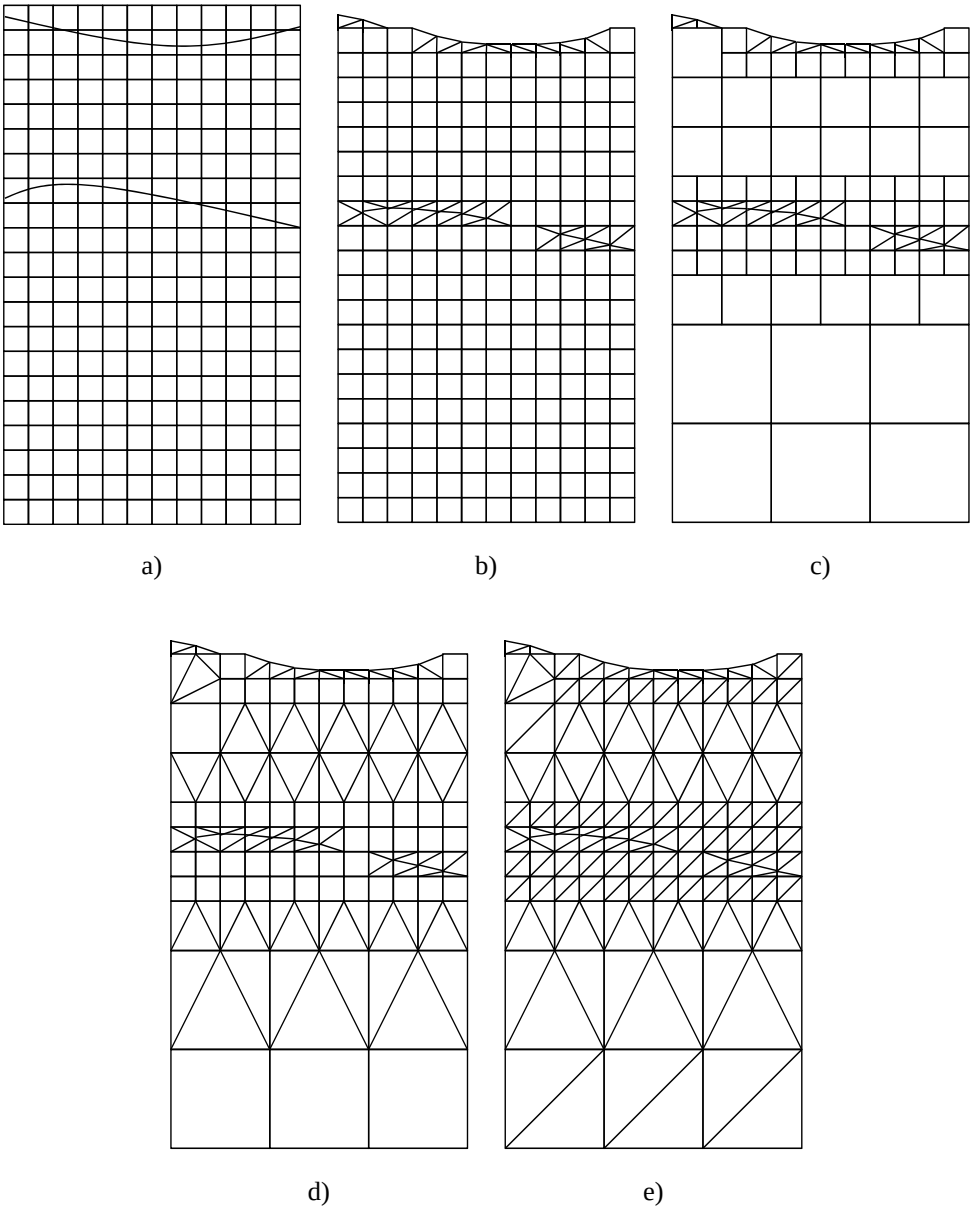
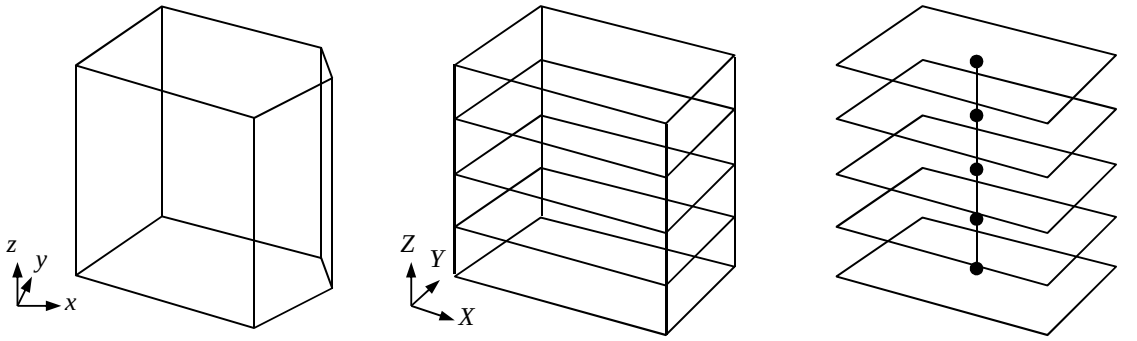


Figure 4.1: Fast and robust modeling of layered medium using background cells. a) A structured grid is put over the target domain with input layer boundaries. b) Elements are generated individually in each cell with small aspect ratios. c) Hexahedral elements are merged in an Octree form to reduce number of elements in the mesh. d) Hexahedral elements in the transition between Octree levels are decomposed into tetrahedral elements for making a conforming mesh with linear hexahedral and linear tetrahedral elements. e) All hexahedral elements are decomposed into tetrahedral elements for making a mesh consisting of tetrahedral elements. Method described in [30] is used from a) to d), while e) is applied to make a mesh consisting of tetrahedral elements used for the multigrid solver.



i) External shape of structure in geospatial database

ii) Set surrounding rectangle and estimate number of floors

iii) Estimate mass of each floor and stiffness/damping of springs connecting floors

Figure 4.2: Model generation of structures described in [12]. i) → ii): A rectangle surrounding the base of the structure is set, and the number of floors are estimated using typical floor story heights. ii) → iii): The mass of each floor and stiffness/damping of springs connecting floors are estimated using properties of typical built structures.

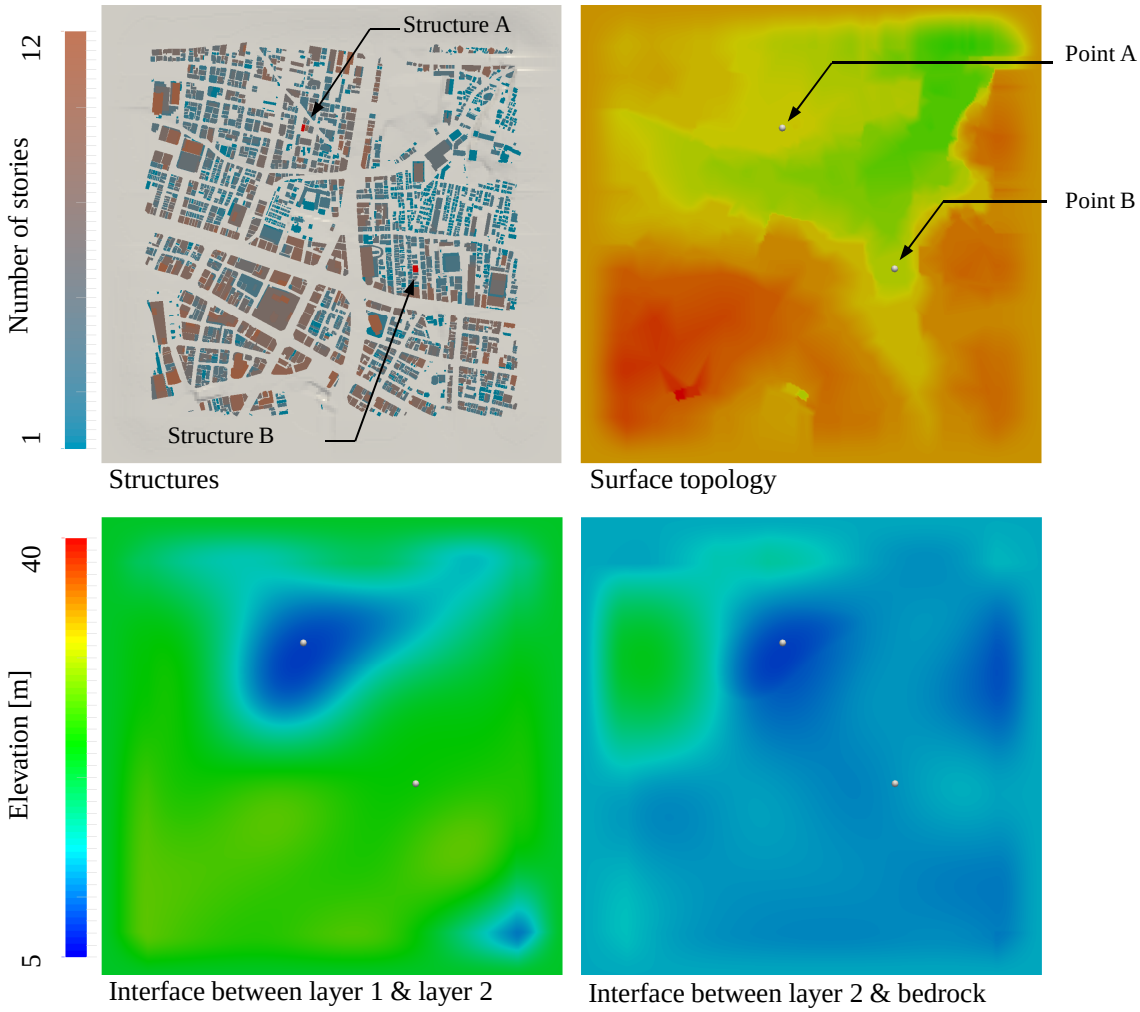


Figure 4.3: Geometry of soil layer interfaces and location of structures in the target domain. Points indicate position of points A and B. Structures A and B are located above points A and B, respectively.

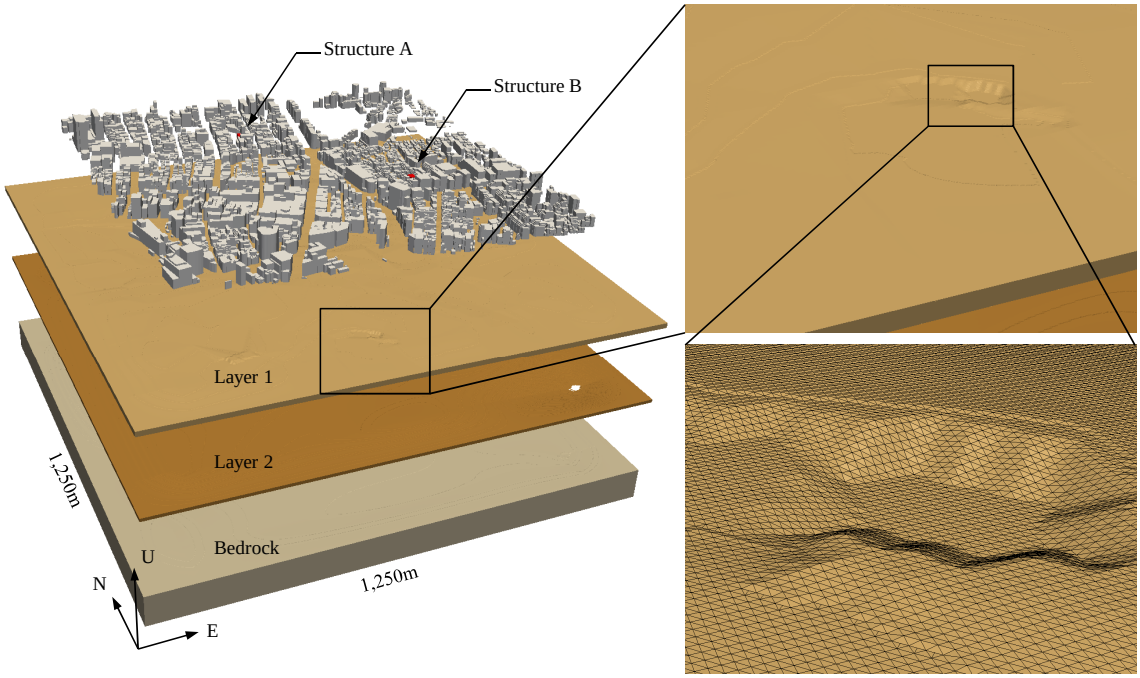
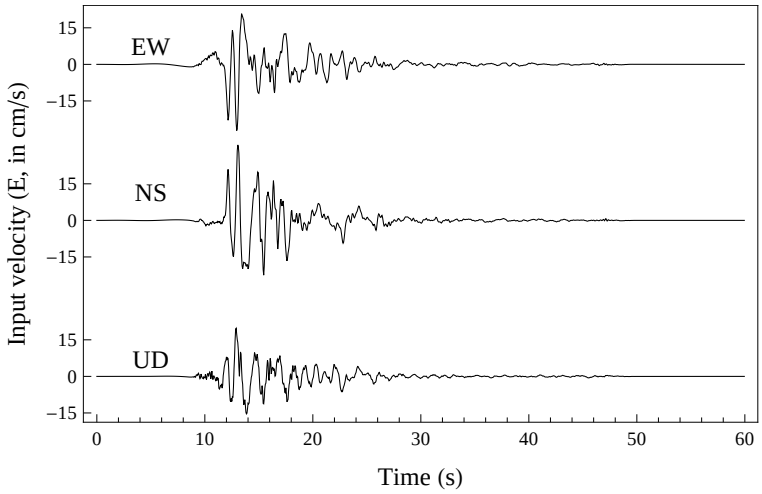
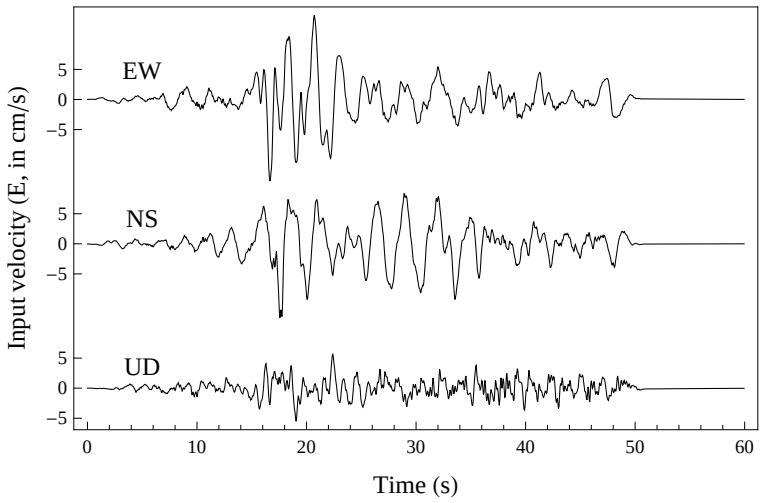


Figure 4.4: 15Hz ground model and structure model. Structures A and B are highlighted in the figure on the left. The figure on the right shows the mesh with smallest element size of 1.0m. We can see that the three dimensional surface topology and layer interface geometry is modeled in the mesh.

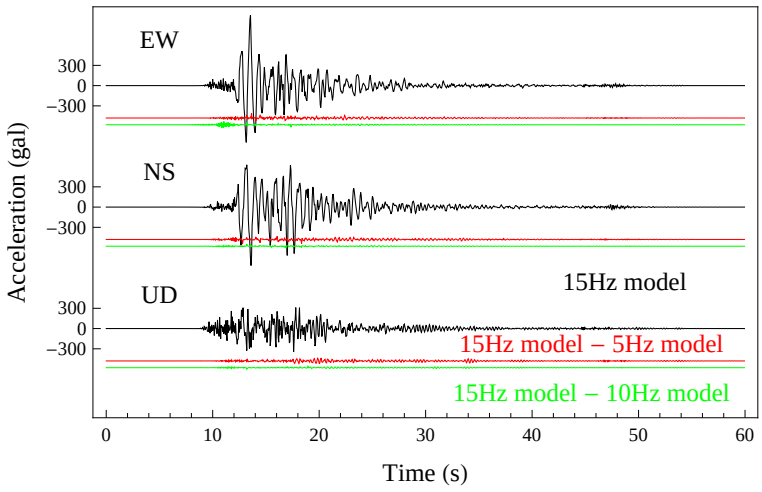


1995 Kobe JMA

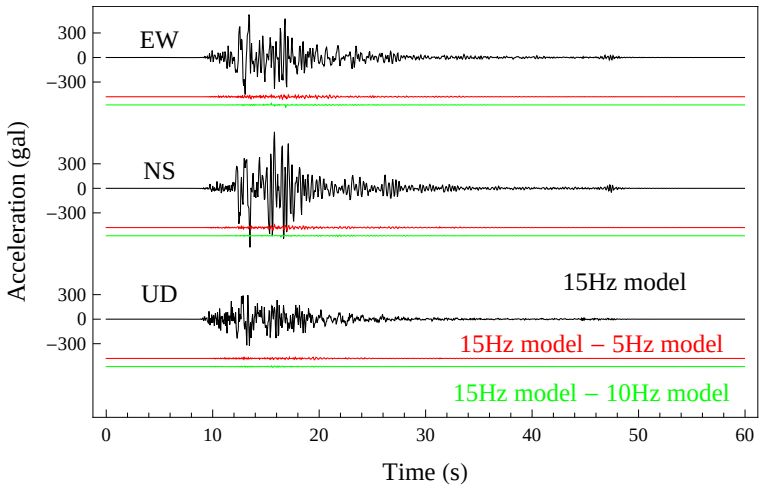


1968 Hachinohe

Figure 4.5: Velocity waveforms used to excite the model. Wave is inputted to the bottom of the model.

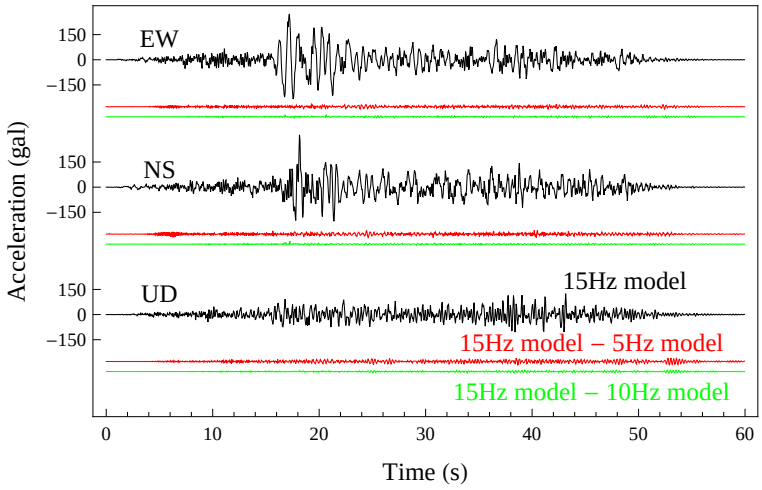


Point A

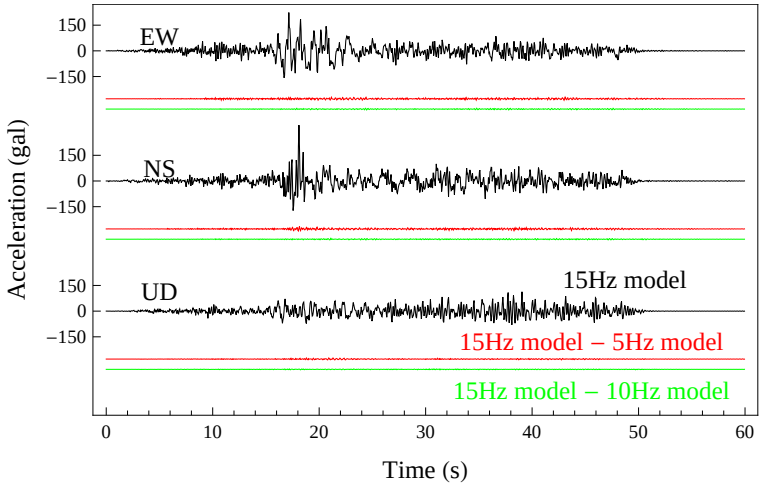


Point B

Figure 4.6: Convergence of computed acceleration waveforms at point A and B for Kobe wave. Black lines indicate computed waveforms using 15 Hz model, red lines indicate the difference between computed waveforms of 15 Hz and 5 Hz models, and green lines indicate the difference between computed waveforms of 15 Hz and 5 Hz models.

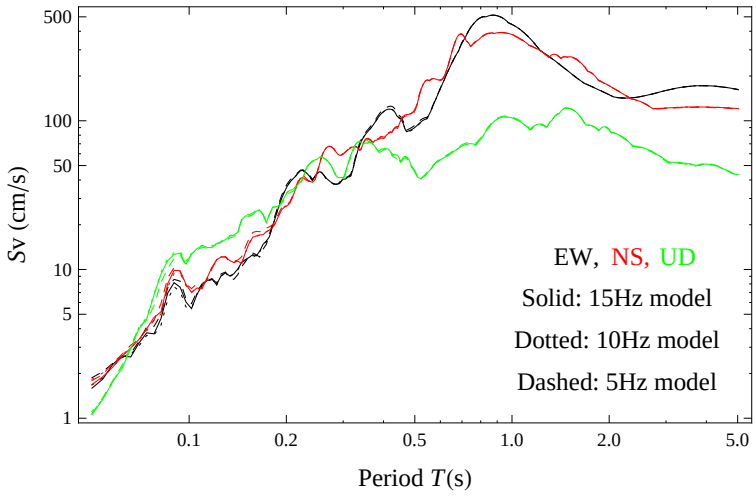


Point A

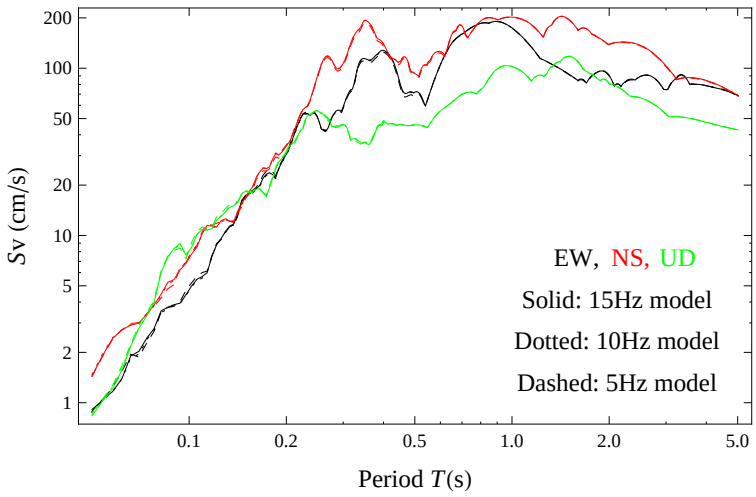


Point B

Figure 4.7: Convergence of computed acceleration waveforms at point A and B for Hachinohe wave. Black lines indicate computed waveforms using 15 Hz model, red lines indicate the difference between computed waveforms of 15 Hz and 5 Hz models, and green lines indicate the difference between computed waveforms of 15 Hz and 5 Hz models.

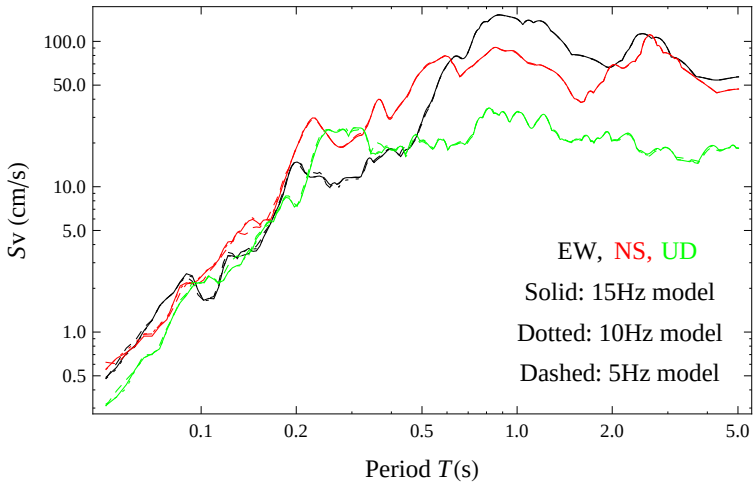


Point A

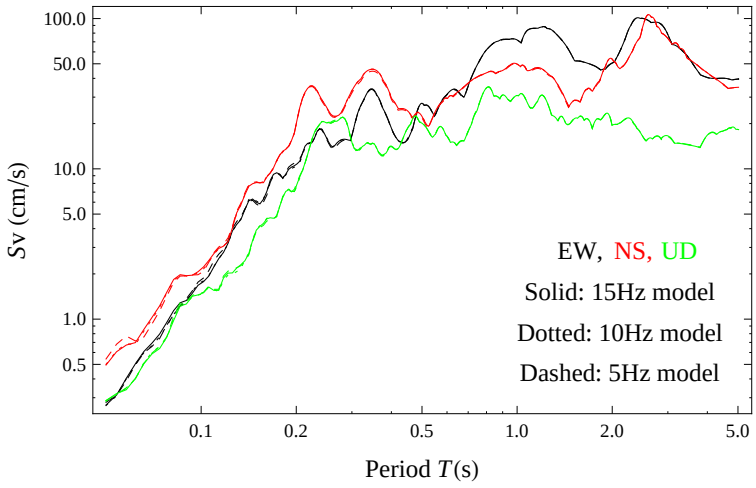


Point B

Figure 4.8: Convergence of velocity response spectra at points A and B for Kobe wave. Damping ratio $h = 0.05$ is used.



Point A



Point B

Figure 4.9: Convergence of velocity response spectra at points A and B for Hachinohe wave. Damping ratio $h = 0.05$ is used.

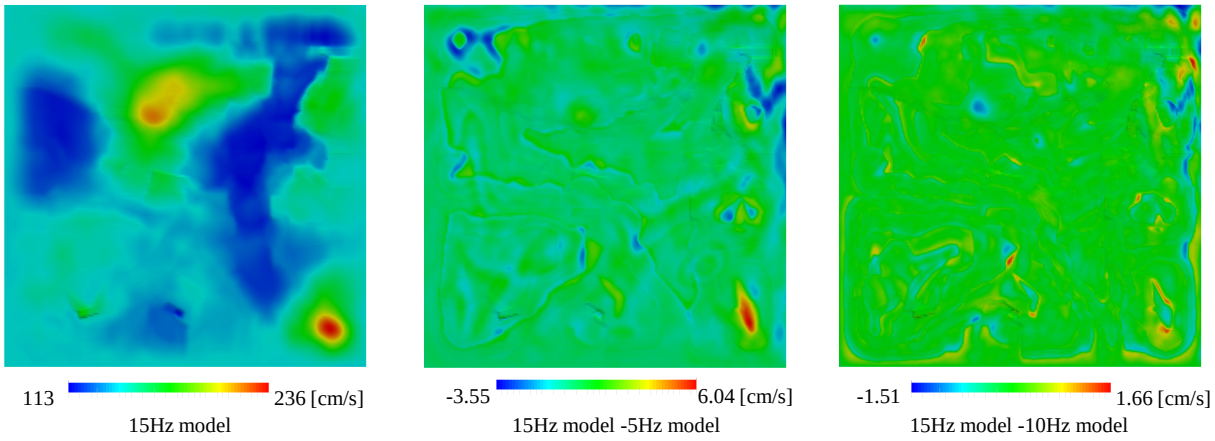


Figure 4.10: Convergence of spatial distribution of horizontal magnitude of SI values for Kobe wave

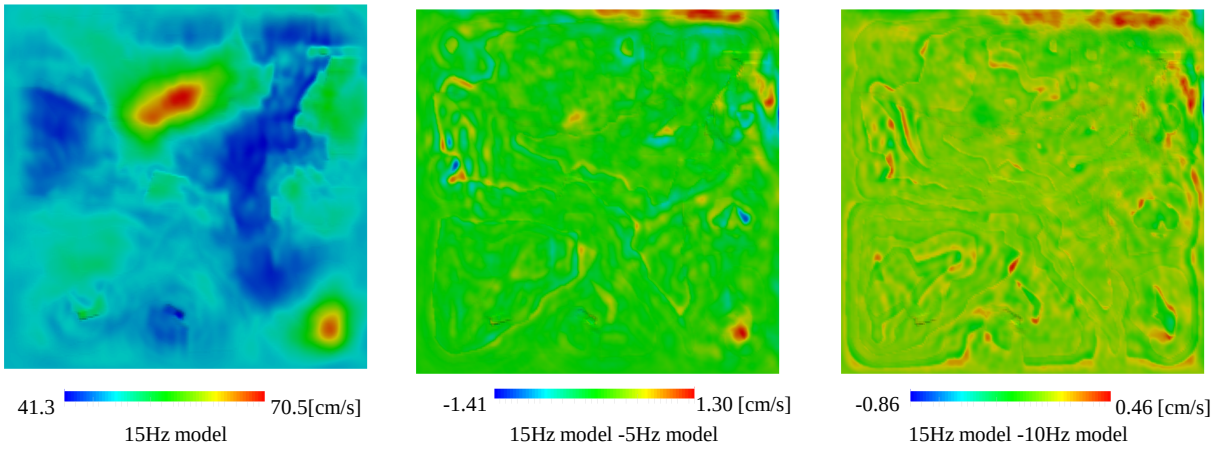
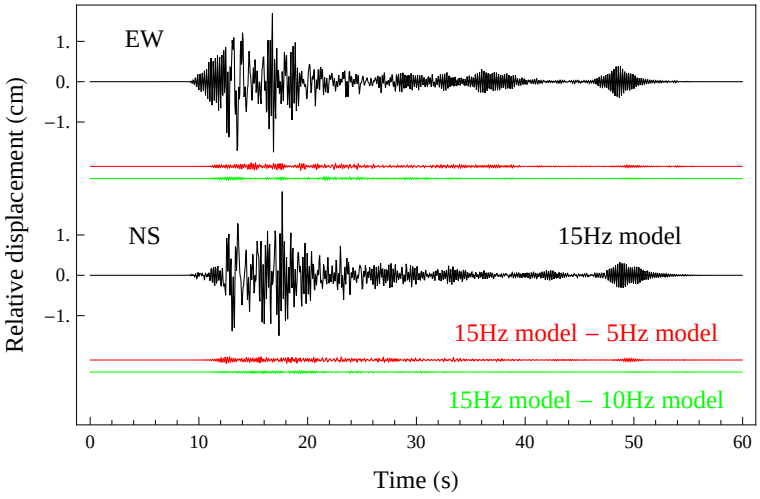
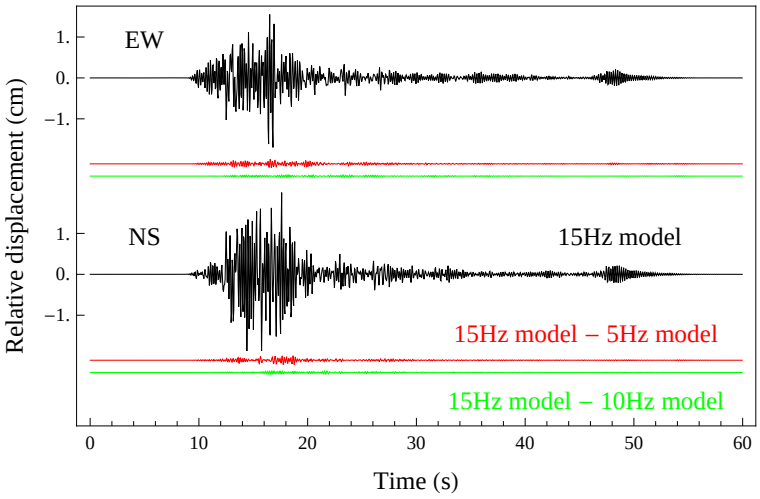


Figure 4.11: Convergence of spatial distribution of horizontal magnitude of SI values for Hachinohe wave

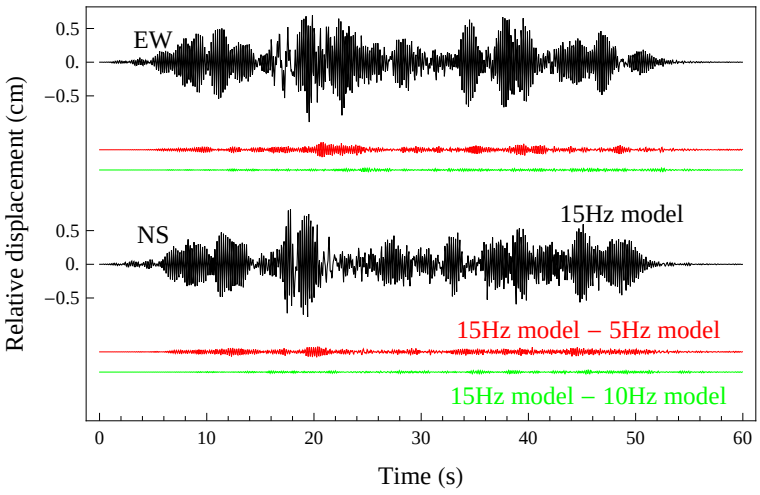


Structure A

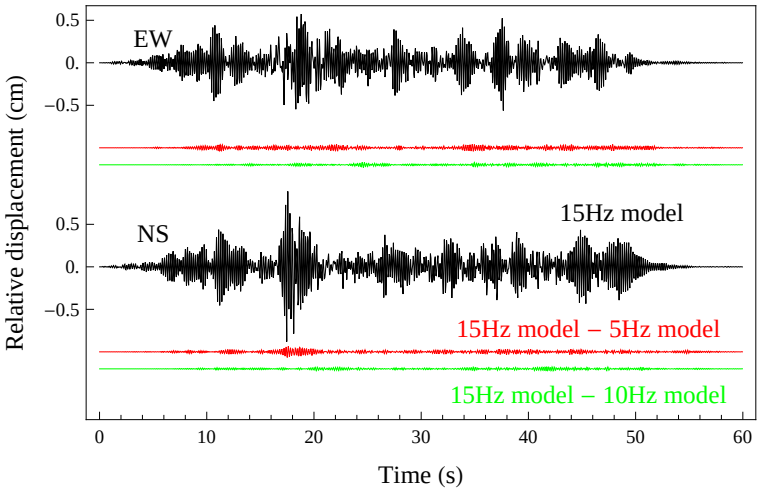


Structure B

Figure 4.12: Convergence of structure response for structures A and B for Kobe wave. Relative displacement (difference between displacement at roof and ground level) is plotted.



Structure A



Structure B

Figure 4.13: Convergence of structure response for structures A and B for Hachinohe wave. Relative displacement (difference between displacement at roof and ground level) is plotted.

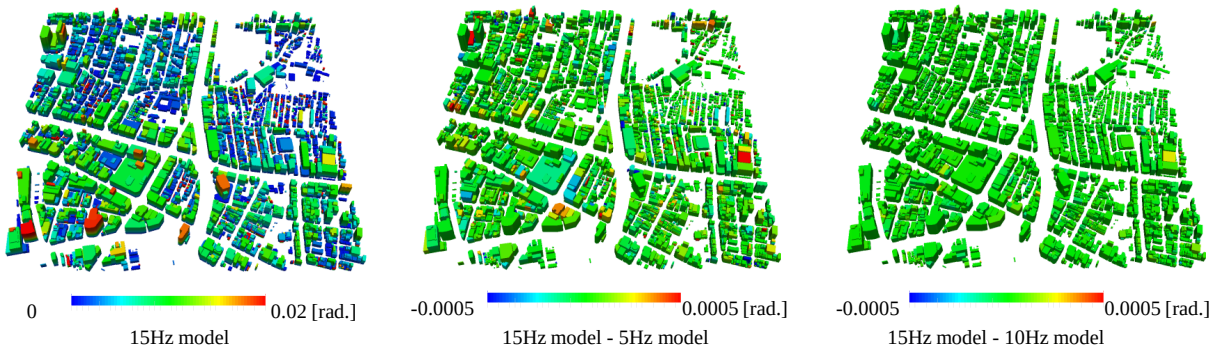


Figure 4.14: Convergence of spatial distribution of structure response (maximum story drift angle) for Kobe wave

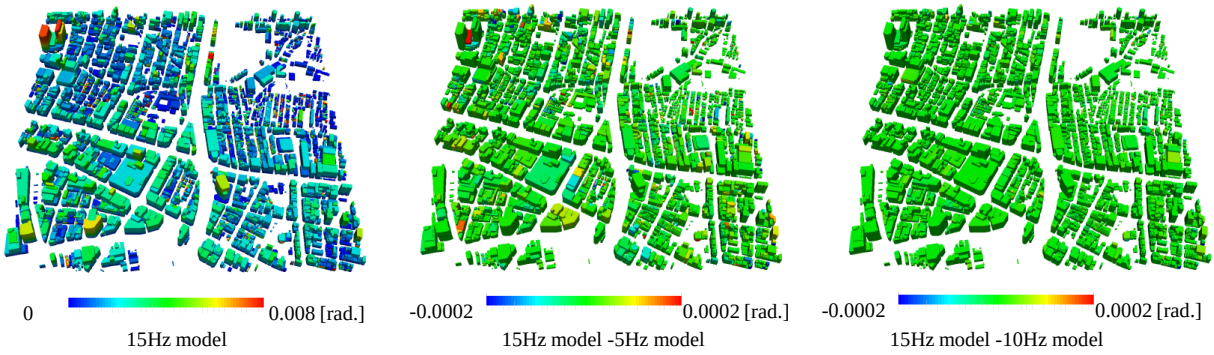


Figure 4.15: Convergence of spatial distribution of structure response (maximum story drift angle) for Hachinohe wave

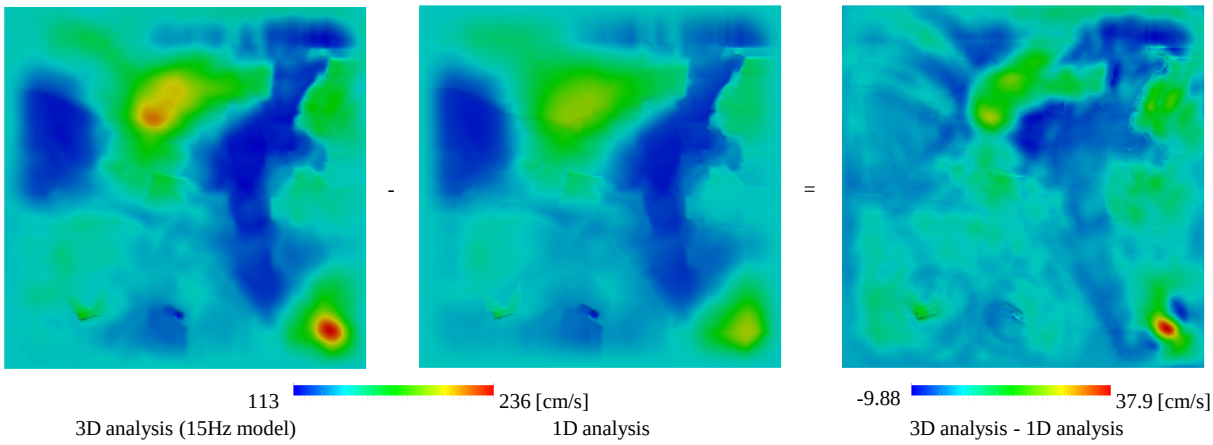


Figure 4.16: Comparison of horizontal magnitude of SI values between 3D and 1D ground motion analysis for Kobe wave

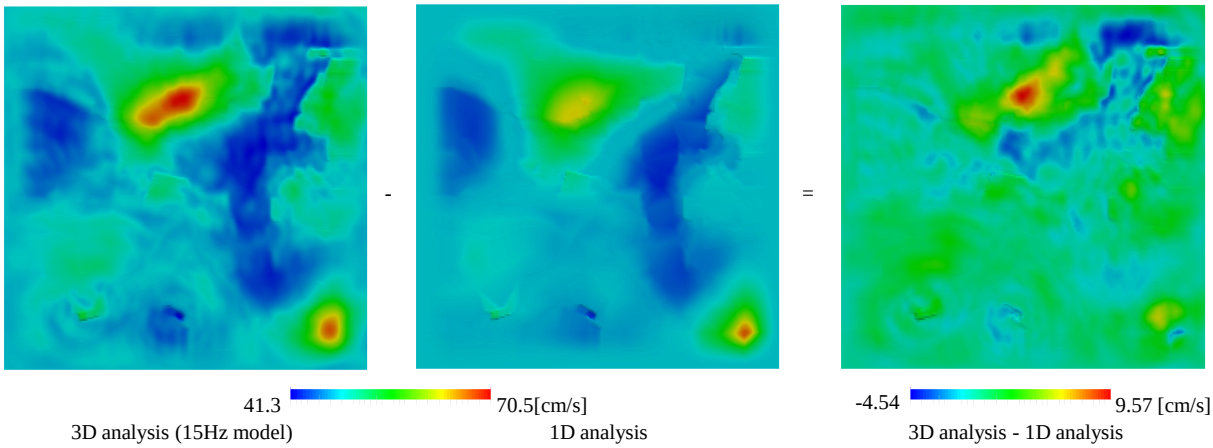


Figure 4.17: Comparison of horizontal magnitude of SI values between 3D and 1D ground motion analysis for Hachinohe wave

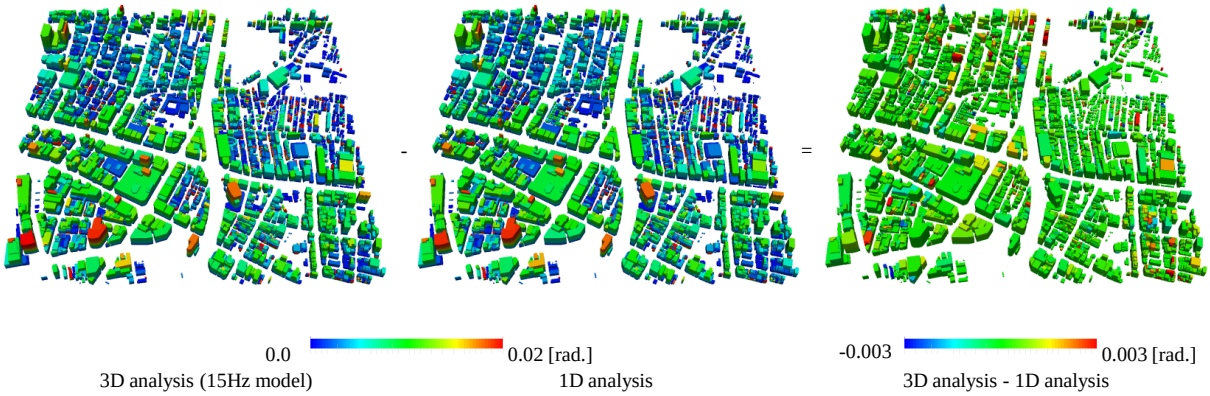


Figure 4.18: Comparison of structure response (maximum story drift angle) using 3D and 1D ground motion analysis for Kobe wave

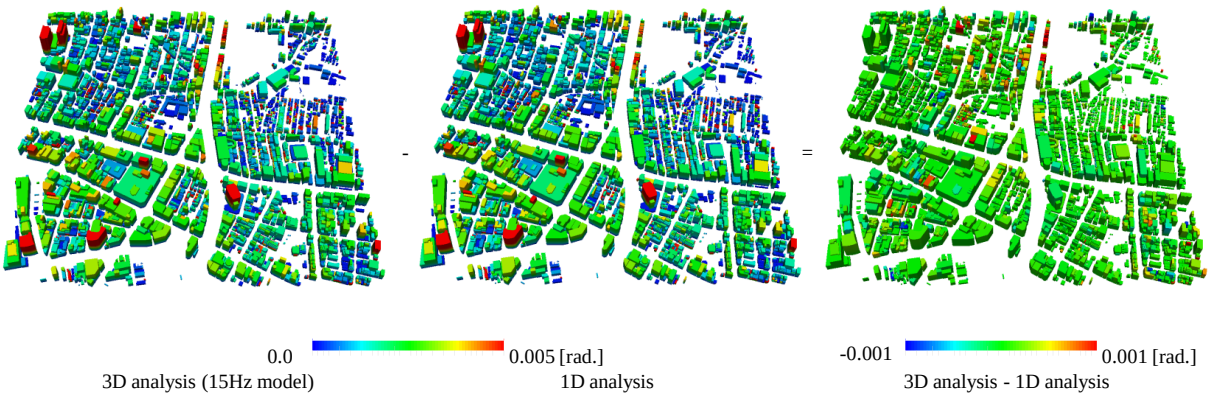
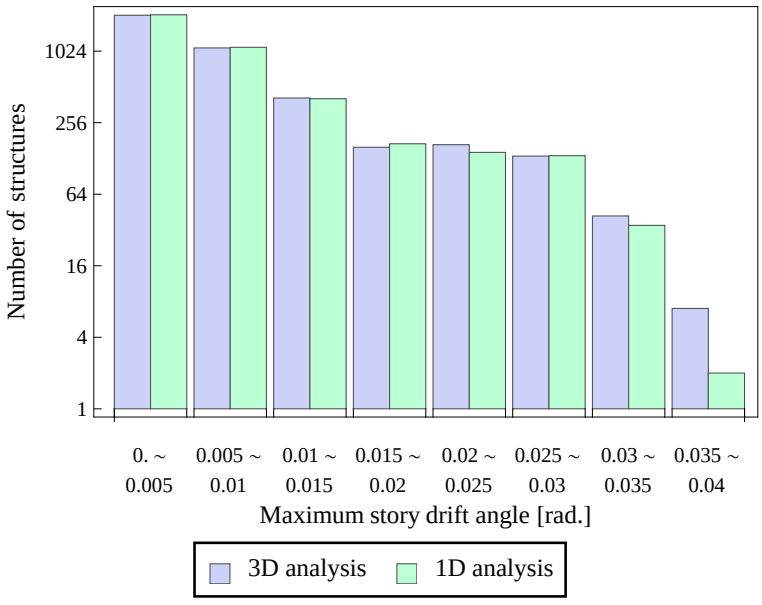
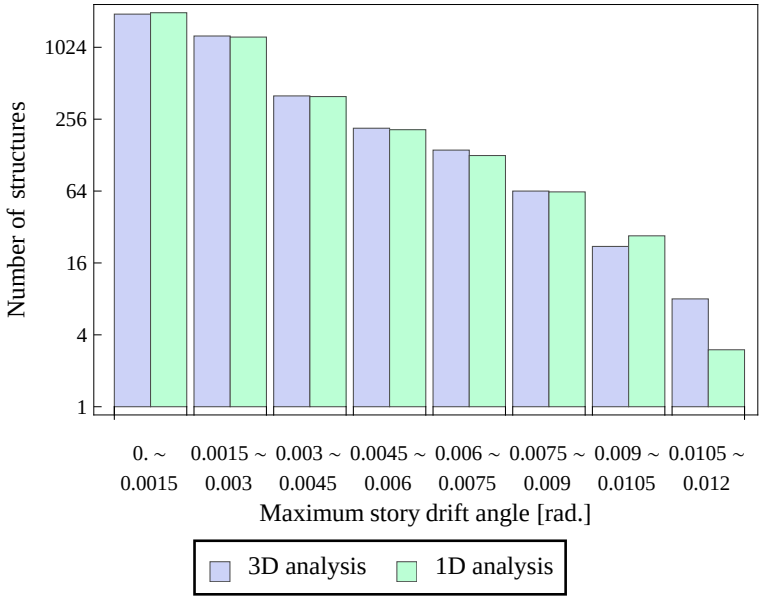


Figure 4.19: Comparison of structure response (maximum story drift angle) using 3D and 1D ground motion analysis for Hachinohe wave



Kobe wave



Hachinohe wave

Figure 4.20: Comparison of maximum story drift angle response of structures using 3D and 1D ground response analysis

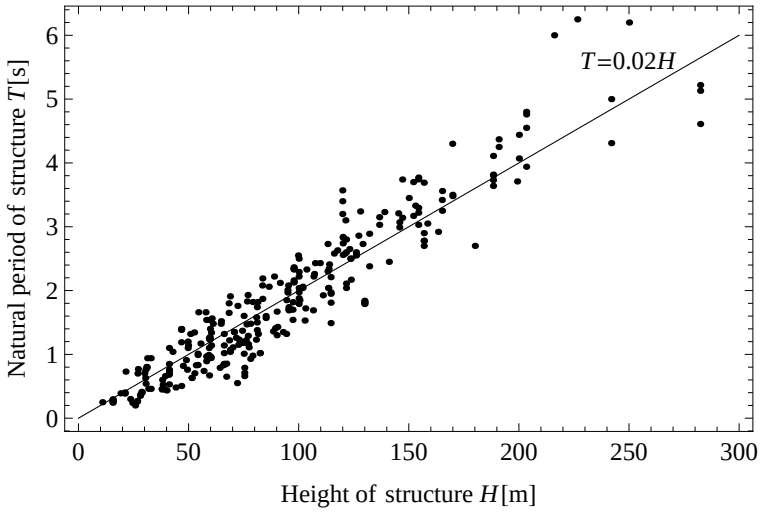


Figure 4.21: Scatter plot of story height H (m) and measured natural period T (s) of actual structures. Data reported in [25] is used.

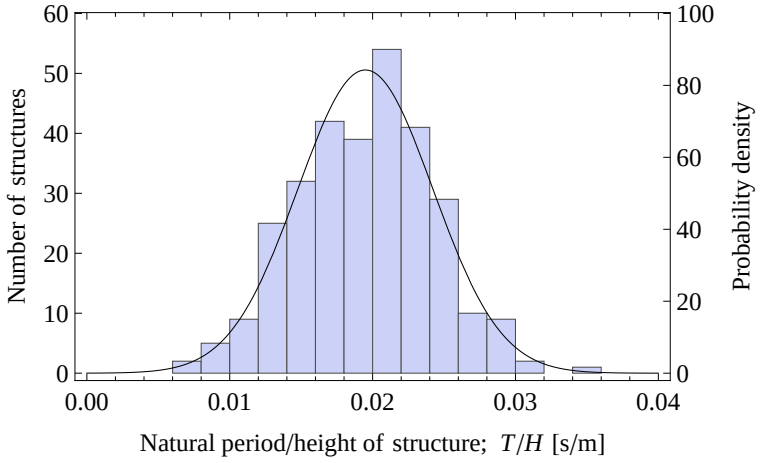


Figure 4.22: Histogram of story height H (m) divided by measured natural period T (s) of actual structures, using data reported in [25]. The black line indicates the normal distribution fitted to the measured data with average $\mu = 0.0195$ (s/m) and standard deviation $\sigma = 0.00473$ (s/m).

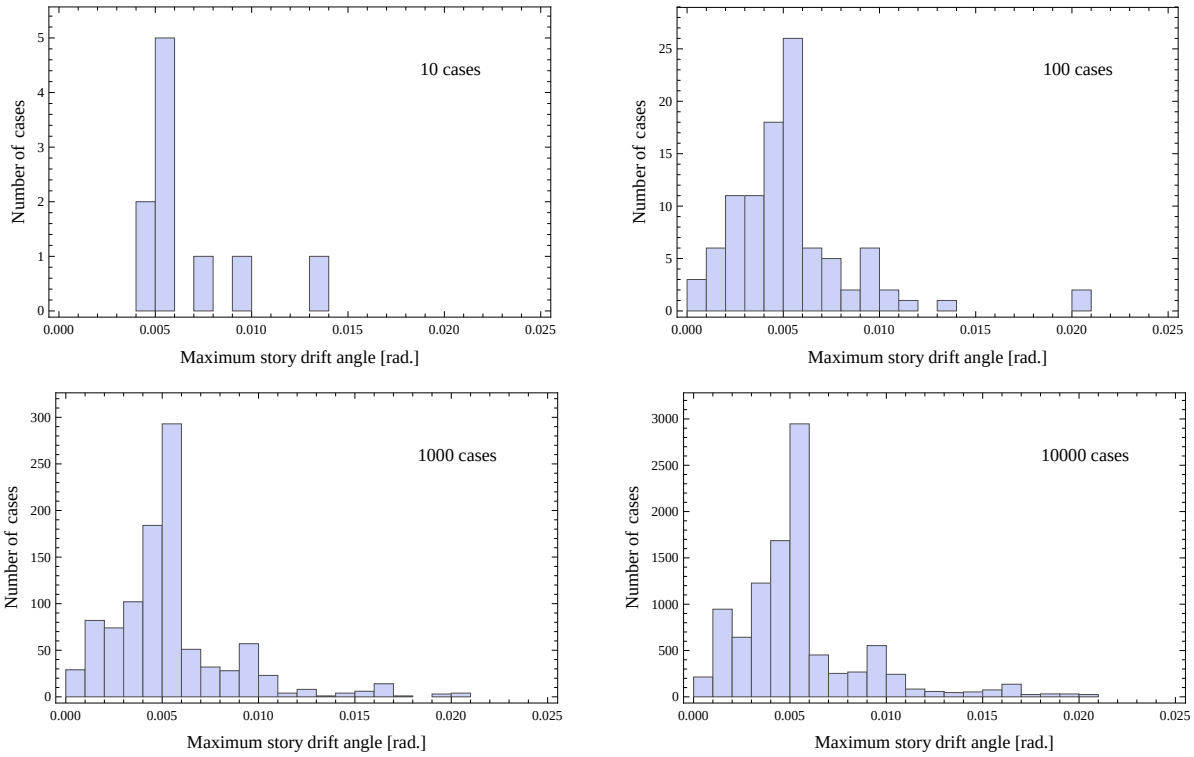


Figure 4.23: Distribution of computed maximum story drift angle of structure A with random T/H parameters for Kobe wave

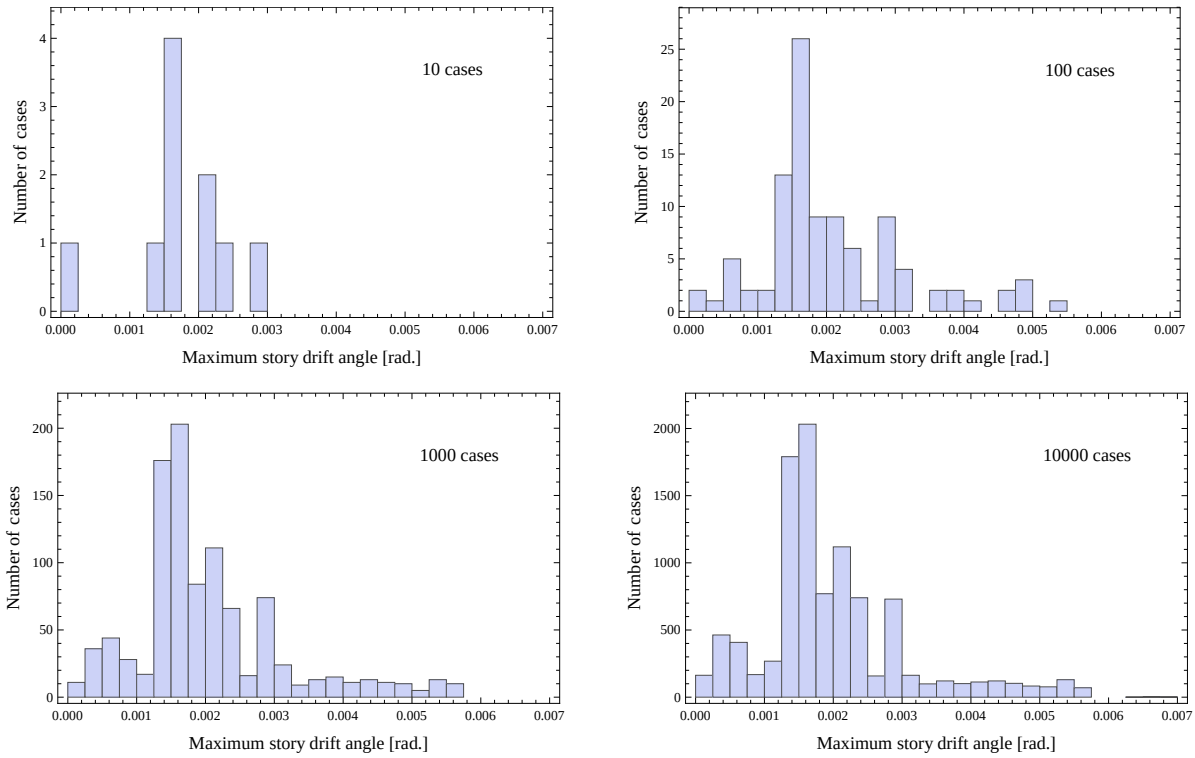


Figure 4.24: Distribution of computed maximum story drift angle of structure A with random T/H parameters for Hachinohe wave

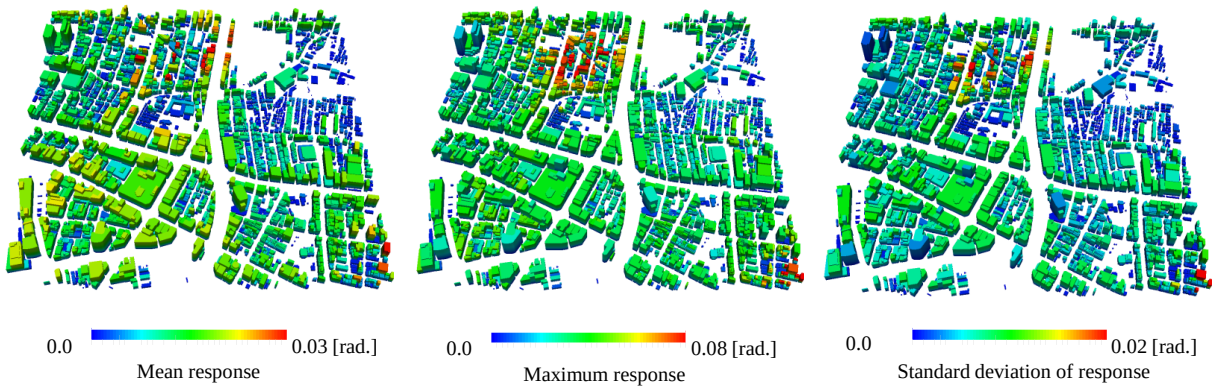


Figure 4.25: Spatial distribution of computed maximum story drift angle for 10,000 cases of random structure models

Chapter 5

Closing remarks

Aiming to improve the reliability of earthquake disaster estimation in urban areas, I first developed a capacity computing method that performs many-case computation of earthquake disasters. I then used a highly scalable 3D ground response analysis method to estimate ground and structure response considering the 3D nonlinear soil structures. Consequently, I found out that the use of capability computing for increasing the target frequency of 3D ground motion analysis to the natural frequencies of structures and the use of capacity computing for performing many-case analysis for uncertainty modeling can lead to detailed analyses, which were difficult to perform with empirical or simpler analysis methods.

In the future, I plan to enhance the ground motion analysis method for application to larger areas such as all of Tokyo. Here, I plan to further speed up and improve the capability of the ground motion analysis method. By using the entire K computer, I expect to enlarge the problem size 10 times than that analyzed in this study. Another work to be performed is the evaluation of the simulation results obtained by many-case structural response simulation. In this study, I used the maximum inter-story drift angle as an index for evaluating the damage of structures. In terms of disaster mitigation, it is more important to estimate the collapse risk, which leads to human losses, and the utility of structures, which leads to financial losses. Since detailed analysis methods are used in the structural response simulation, information such as nonlinear parameters can be retrieved in addition to the displacement response. I plan to extract such information and contract the dynamic response results for making effective damage evaluation indexes.

Reliable urban earthquake disaster simulation can be useful for many types of earthquake disaster mitigation approaches. For example, we can expect an improvement in earthquake insurance participation if rational disaster estimation is undertaken. In Japan, earthquake insurance fees for residential buildings are determined by the structural properties and the location of the structure. While detailed information such as the structure type and age and its scores on earthquake resistance checks are used to evaluate structures, only the location prefecture is used to categorize the effect of site conditions. This leads to an adverse selection; structures located on soft soils have higher participation rates compared to structures

located in relatively safe areas. By developing a method that the majority of building owners can accept a rational way of evaluating earthquake risk and reflecting such risk estimates in the insurance fees, we can expect improved participation and more financial stability following large earthquakes. With the increase in computational capability and capacity of computers, large-scale analysis is expected to become available on commodity computers, and thus methods such as those shown in the present study could be used for practical, real site problems.

Chapter 6

Reliability of 3D nonlinear ground motion analysis

6.1 Comparison of ground motion analysis results with observed records

6.1.1 Introduction

High performance computing has been applied to earthquake engineering applications, enabling 3D ground motion analysis of large domains in high resolution, with nonlinear constitutive models. The reliability of such simulation methods are often checked by evaluating the reproducibility of observed ground motion records for particular earthquakes. In such reproducibility checks, it is common to "tune" the input parameters such as material properties or input wave to make best fit analysis results with observed records. Such tuning is suitable for understanding of the ground motion phenomena of the particular event, but does not necessarily reflect the reliability of the simulation method when used for estimation of future earthquakes.

In this section, I perform ground motion simulation with limited input data and input wave records, without any tuning, and compare the computed records with observed records. Such comparisons can be used to estimate the reliability of 3D ground motion analysis for future earthquakes under similar problem settings.

Large scale 3D nonlinear ground motion analysis can also be used for estimating seismic response that is costly or difficult to measure on-site. Ground strain is one of the responses difficult to measure, but has large influence on response of buried pipelines [34, 35]. I check the distribution of strain in the area and discuss the relation between ground geometry and displacement response at surface.

6.1.2 Problem settings

I target an area in Kanagawa Prefecture, Japan, with size of 1,696 m in the EW direction and 1,920 m in the NS direction. The area is known for large ground response compared with nearby observation points in the 2011 Tohoku Earthquake. The 50m grid digital elevation map of Geospatial Information Authority of Japan is used for obtaining the surface topography, while boring logs of SUPREME [36] and database of Yokohama City [37] is used for obtaining the ground geometry of the area. The ground structure consists of 3 layers, with the first layer consisting of clay, silt, and humus soil with N values of 0 to 2. The second layer consists of sand, gravel, consolidated silt with N values around 50, and the bedrock layer consists of hardpan and mudstone. The soil parameters (clay for layer 1 and gravel for layer 2) indicated in the reports by the Central Disaster Management Council of Japan [38] are used for the parameters of the Ramberg-Osgood model [28]. Table 6.1 summarizes the material properties of each layer.

The Metropolitan Seismic Observation network (MeSO-net) [39], a high density observation network, is operated in the Metropolitan area of Japan. The input ground motion at bedrock (E) for the 2011 Tohoku Earthquake is estimated by swinging back the observed records in a nearby MeSO-net station using 1D soil amplification analysis.

The target domain is discretized with second ordered tetrahedral elements such that one shear wave length ($\lambda_s = V_s/f_{max}$) is discretized with 5 elements for linear materials and 10 elements for nonlinear materials. In preliminary analysis, a small domain with same material properties with different target frequencies were analyzed, and convergence in ground strain was obtained using a model with f_{max} = 2.5Hz. Figures 6.1 and 6.2 show the mesh used in analysis, with smallest element size of 4 m. The model consists of 7,779,048 tetrahedral elements, and 10,836,369 nodes with 32,509,107 DOF. Cartesian coordinates with the x, y, and z axes corresponding to EW, NS, and UD directions, respectively, with the South-West lower corner as the origin is used. We can see that the complex ground geometry is reflected in the model.

The numerical methods described in Chapter 2 is used for carrying out the large-scale analysis in a short time. Here, Rayleigh damping defined between 0.1 and 2.5 Hz is used. The input wave is cut off

above 2.5Hz, and below 0.1Hz for removing long period noise in the observed records. Fig. 6.3 shows the wave inputted to the bottom of the domain. The seismic response analysis of 60,000 time steps with time stepping increment dt = 0.005s was performed using a commodity cluster with eight compute nodes, each with dual hexa-core Intel Xeon X5680 CPUs, connected with InfiniBand quad data rate communication links. Although analysis took 308 hours (1,107,495s) with this relatively small computing cluster, it is expected to be finished in a short time using larger computer resources.

6.1.3 Comparison of computed results with observed records

I compare computed results with observed records at two observation points for checking the reliability of simulation results. Observation stations of SUPREME, located at point P_1 and P_2 in Fig. 6.1, is used for comparison. P_1 is located in an area with relatively thick soft soil layers, while P_2 is located in an area with relatively thin soft soil layers. Since the observation devices are designed for real time damage estimation of pipelines for isolating gas pipeline networks, time synchronization is not performed and the length of recordable ground motion is limited. As the duration of the Tohoku Earthquake is longer than the recordable time of this device, the first half of the wave is missing. Thus, the observed waveform is of unknown time with lack of the first half of the earthquake. Nevertheless, the latter half of the displacement records can be reproduced well by integrating the latter half of complete records obtained at nearby observation points with frequency components below 0.1Hz cut off. The same procedure is used to estimate the observed displacement at the two points. Figures 6.4 and 6.5 show the observed and computed displacement records at point P_1 and P_2, respectively. Although it is difficult to compare in accuracy due to such limitation in the observed data, we can see that the numerically computed wave matches with that of the observed wave in the latter half of the records (140-240s). I make a quantitative comparison with this part of the wave.

I first compare the destructiveness of the waves using the SI index [33] defined in Eq. (4.5), which is the contracted value of the response of a single degree of freedom system. This index is useful for roughly comparing the frequency components that are critical to the damage of potential structures. From Table

6.2, we can see that the computed values of SI values in the horizontal directions at P_1 is around 25 cm/s, which is similar to the observed values. We can also see that the computed horizontal SI values at P_2 is smaller than that of P_1, which is consistent with the observed records. Although the difference of SI values are small for most of the components at the two points, there is about 50% difference between the observed and computed SI values in the x direction at point P_2.

Since the analysis is intended to be used for evaluating damage of structures under nonlinear soil behaviors, it is important to compare the numerical waveforms with measured waveforms in terms of phase and envelope in addition to the frequency components. Here, I compare the phase and envelope of the wave using the time-frequency misfit [40] and an example of Goodness-Of-Fit (GOF) criteria [41, 42]. The GOF criteria is capable of quantitatively evaluating the goodness of fit of phase and envelope of waves, and is commonly used for comparison of seismic waves; for example, see [9, 43, 44]. Based on [41] and [42], the goodness of fit is categorized as "poor" for GOF index values under 4, "fair" for values 4-6, "good" for values 6-8, and "excellent" for values more than 8. Table 6.3 shows the Goodness-Of-Fit between the computed and observed displacement records at point P_1 and P_2. We can see that the goodness of fit is between "good" to "fair" for the two points. It is reported that one horizontal direction of an actual seismogram typically fits the other horizontal component in the "good" range [42]; the computed waves in the z direction has a fit comparable to this, while the horizontal components have a slightly weaker fit ("fair" to "good" fit).

6.1.4 Spatial distribution of displacement and strain response

In addition to estimating ground motion at designated points in the target area, nonlinear 3D ground motion analysis can be used to estimate ground strain, which is difficult to measure using common observation techniques. Such distribution of ground strain is useful for evaluating the damage of underground structures such as buried pipelines. In this section, we show the distribution of ground strain and discuss its relationships between the ground structure and distribution of computed displacement.

Fig. 6.6 a) shows the maximum displacement distribution at surface. We can see that the distribution

of displacement is similar to the distribution of thickness of soft soil layers, shown in Fig. 6.7 a).

Next, we I plot the distribution of maximum principal strain at surface in Fig. 6.6 b). Compared with the distribution of displacement, we can see that strain response has a more complex distribution. The distribution is different from several ground geometry indexes attempted to reproduce the strain distribution shown in Figures 6.7 b) and 6.7 c). The first index $|\nabla h|$ indicates the spatial change rate of depth of layer 1 (h), from the intuition that ground strain occurs from differences between displacement response of the hard and soft layers. The second index $h|\nabla h|$ is a more complex index where the amplification of waves at soft soils are considered by multiplying the depth of layer 1 (h) with the spatial change of layer 1 $|\nabla h|$. We can see that the strain distribution differs from both of the indexes, and that it is not straight forward to estimate ground strain from the ground geometry using simple interpretations.

Next, I plot the maximum displacement and maximum axial strain response along lines A, B and C, which represents typical configurations of buried pipeline segments in an area with nonuniform ground structures. Fig. 6.8 shows the response at line A. The ground structure under line A is relatively uniform in the axial direction. From Fig. 6.7 a), we can also see that the thickness of layer 1 is uniform in the $x-y$ plane. The displacement response is relatively uniform and strain response is small, which is consistent with 1D analysis of horizontally stratified grounds.

Fig. 6.9 shows the response at line B . Line B crosses a V-shaped valley with soft soil layer in the middle of the hard layers. From Fig. 6.7 a), we can see that the ground structure is relatively uniform in the y direction. We can see that the displacement response is large inside the valley, and the axial strain is large at the boundary between the soft and hard soil layers. Such strain distribution is similar to the observed patterns reported in [45], and is consistent with analysis using the 2D cross section.

Fig. 6.10 shows the response at line C. Line C has a soil structure with the right hand side of the line in the bedrock. The depth of layer 1 gradually increases in the Western direction until x = 1,300 m, and has a constant depth in x < 1,300 m. The displacement response is small at the bedrock and larger inside layer 1. Although it is expected that uniform displacement and small strain occurs where the thickness of layer 1 is uniform (x < 1,300 m), local fluctuations can be seen in the computed displacement response, and

complex strain distribution is obtained in this area. From Fig. 6.7 a), we can see that the ground structure has complex three dimensional shape on the Northern side of line C. Such three dimensional geometry around line C could be the reason of the complex response at line C, which is difficult to estimate using analysis of 1D or 2D analysis.

6.1.5 Summary

I compared the computed displacement with observed records for checking the reliability of simulation results in the area. Considering the limited information used to model the domain, estimation in the input waveform, and lack of accuracy in the observed waveforms, the computed waveforms matches well with the observed waveforms in terms of SI values and goodness of phase and envelope fit of the displacement records. Such agreement was attained without tuning of input parameters in the soil model or input wave; the reliability of 3D ground motion simulation is expected to be at this level for estimating ground motion for future earthquakes in similar problem settings.

Using the large scale 3D ground motion analysis, I analyzed ground strain at surface, which is difficult or costly to measure with common observation techniques. Such results show that the complex ground strain occurs in non-uniform grounds, which can be difficult to estimate using 1D or 2D analysis methods. The use of 3D analysis could help us understand the phenomena of ground strain occurrence in such non-uniform grounds and lead to better seismic design of buried pipelines.

Table 6.1: Material properties of ground

	V_p (m/s)	V_s (m/s)	ρ (kg/m^3)	h_{max} or h	γ_r
1st layer	700	100	1,500	0.23	0.007
2nd layer	1,400	300	1,800	0.23	0.001
Bedrock	2,100	700	2,100	0.01	∞

Table 6.2: SI values [33] of observed and computed waveforms

Point/component	Observed (cm/s)	Numerical (cm/s)
P_1, x	25.6	28.2
P_1, y	27.1	25.5
P_1, z	6.25	9.35
P_2, x	9.92	15.9
P_2, y	10.7	13.2
P_2, z	6.53	9.28

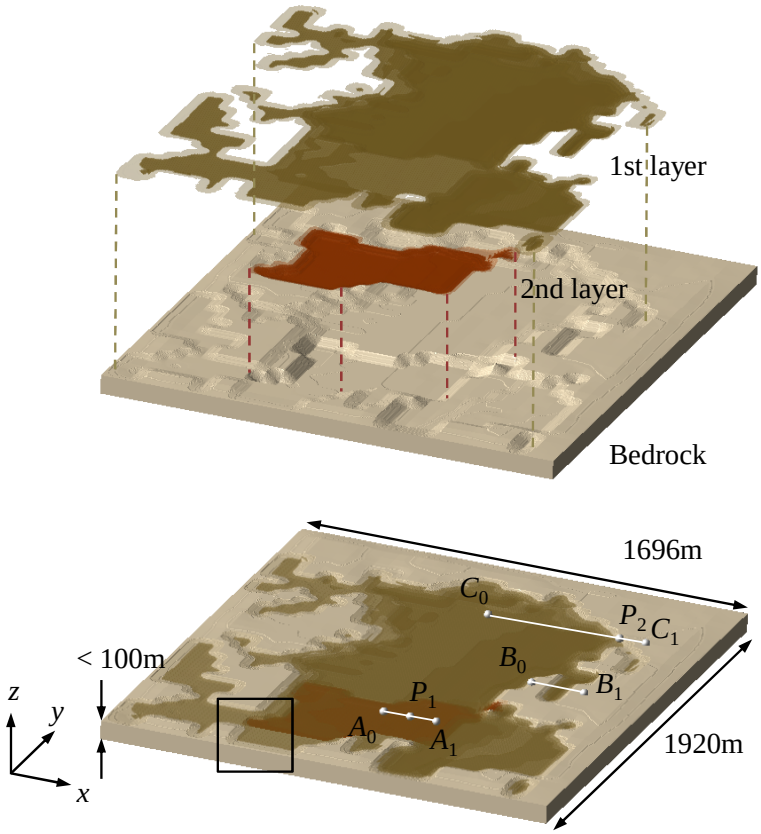


Figure 6.1: Geometry of soil layer interfaces and location of observation points and observation lines in the target domain.

This page is a derived work from: Three-Dimensional Nonlinear Seismic Ground Response Analysis of Local Site Effects for Estimating Seismic Behavior of Buried Pipelines, by Tsuyoshi Ichimura, Kohei Fujita, Muneo Hori, Takashi Sakanoue, Ryo Hamanaka, in Journal of Pressure Vessel Technology. Copyright of the original paper is owned by ASME.

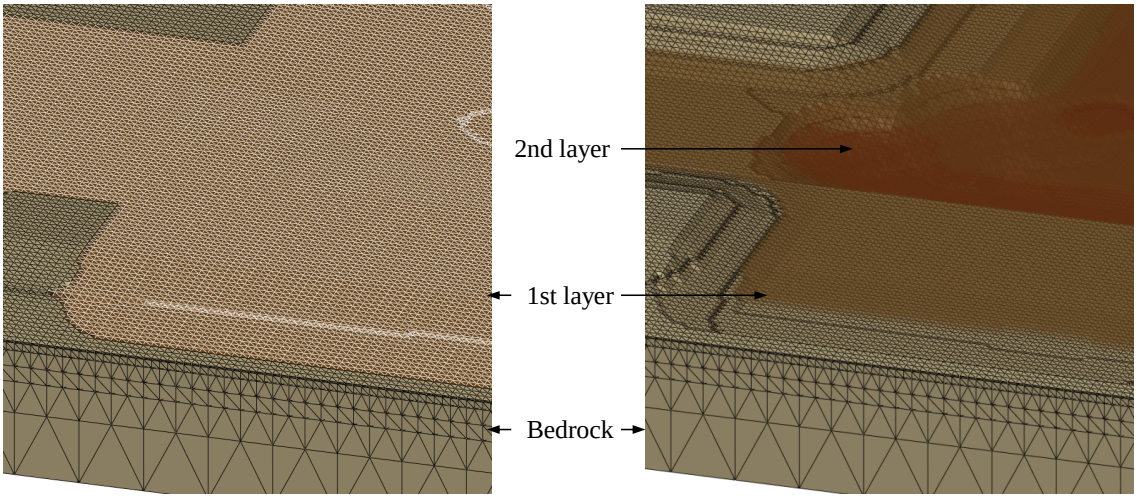


Figure 6.2: Mesh of ground model. Part of the ground model, indicated in a black box in Fig. 6.1, is shown. The second layer is beneath the first layer and cannot be seen in the figure on the left.

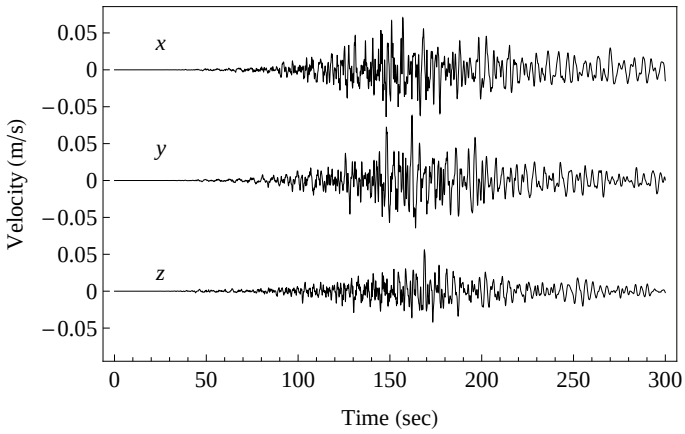


Figure 6.3: Ground motion inputted to the bottom of the target domain

This page is a derived work from: Three-Dimensional Nonlinear Seismic Ground Response Analysis of Local Site Effects for Estimating Seismic Behavior of Buried Pipelines, by Tsuyoshi Ichimura, Kohei Fujita, Muneo Hori, Takashi Sakanoue, Ryo Hamanaka, in Journal of Pressure Vessel Technology. Copyright of the original paper is owned by ASME.

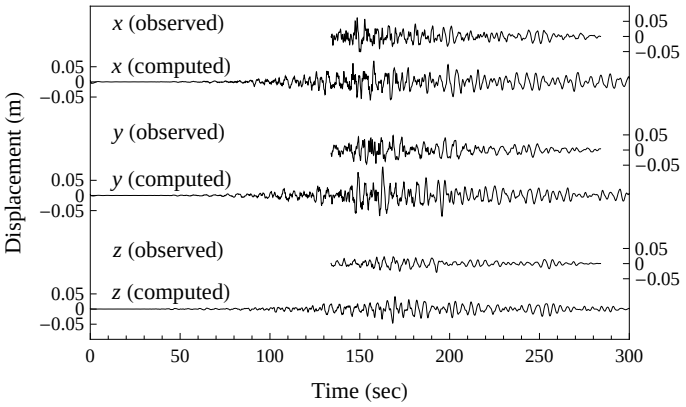


Figure 6.4: Comparison of observed and computed ground motion at point P_1

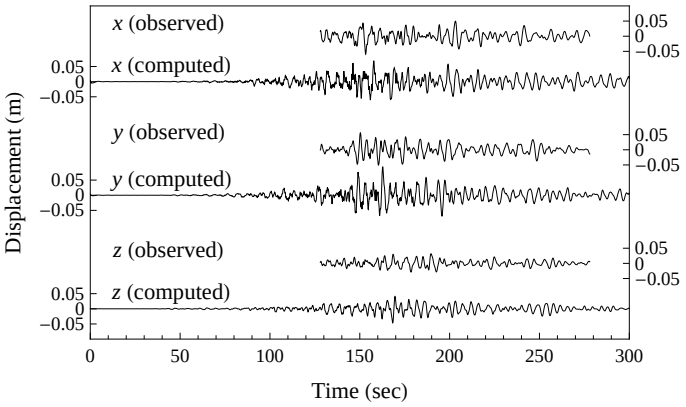
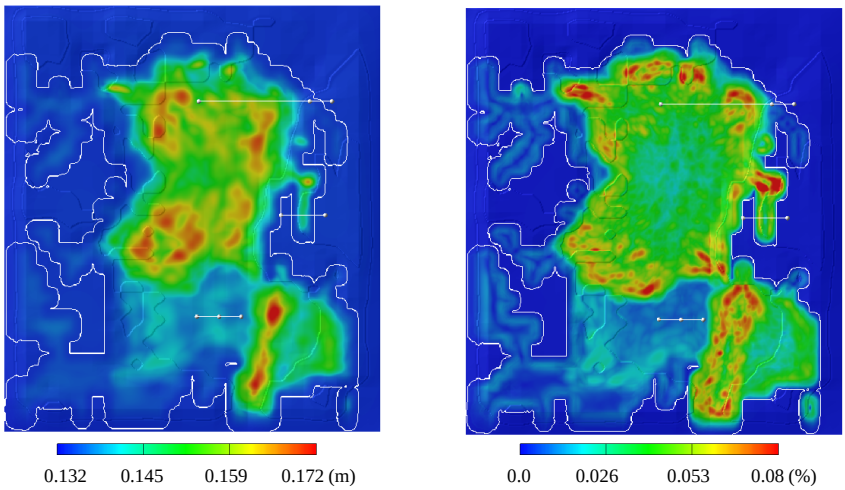


Figure 6.5: Comparison of observed and computed ground motion at point P_2

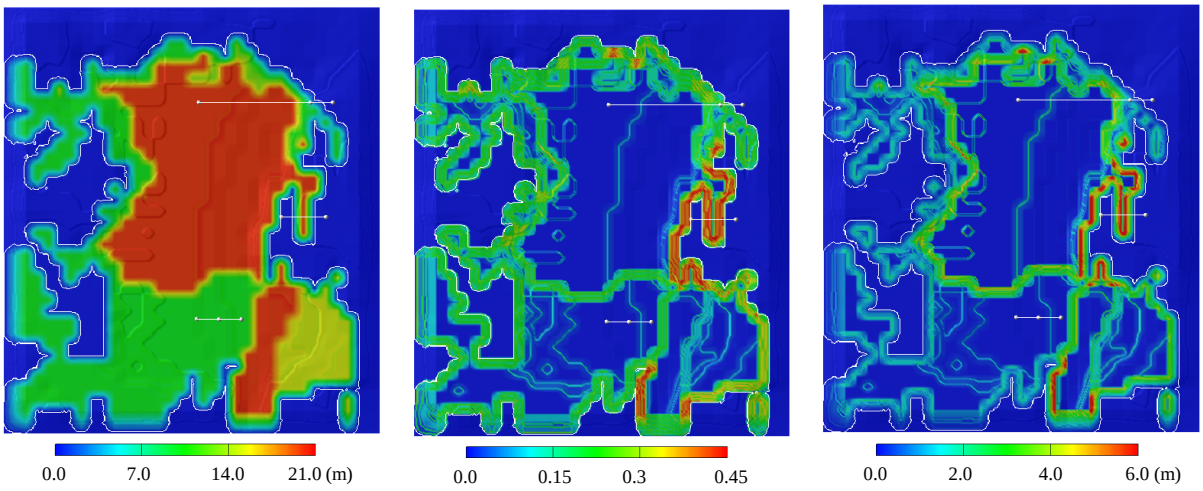
Table 6.3: Agreement between the measured and simulated waveforms using time-frequency misfit and goodness-of-fit criteria [40, 42]

Point/	Misfit		Goodness-of-Fit (values)		Goodness-of-Fit (category)	
component	Envelope	Phase	Envelope	Phase	Envelope	Phase
P_1, x	0.652	0.523	5.21	4.77	fair	fair
P_1, y	0.776	0.470	4.60	5.30	fair	fair
P_1, z	0.401	0.286	6.70	7.14	good	good
P_2, x	0.494	0.527	6.10	4.73	good	fair
P_2, y	0.576	0.530	5.62	4.70	fair	fair
P_2, z	0.370	0.346	6.91	6.54	good	good



a) Norm of displacement b) Maximum principal strain

Figure 6.6: Distribution of the maximum norm of displacement and maximum principal strain at surface. The white lines indicate lines A, B, C, and the boundary between the bedrock and layer 1.



a) h b) $|\nabla h|$ c) $h|\nabla h|$

Figure 6.7: Geometric properties of layer 1. The white lines indicate lines A, B, C, and the boundary between the bedrock and layer 1. h indicates depth of layer 1, while $|\nabla h|$ indicates rate of spatial change of layer 1.

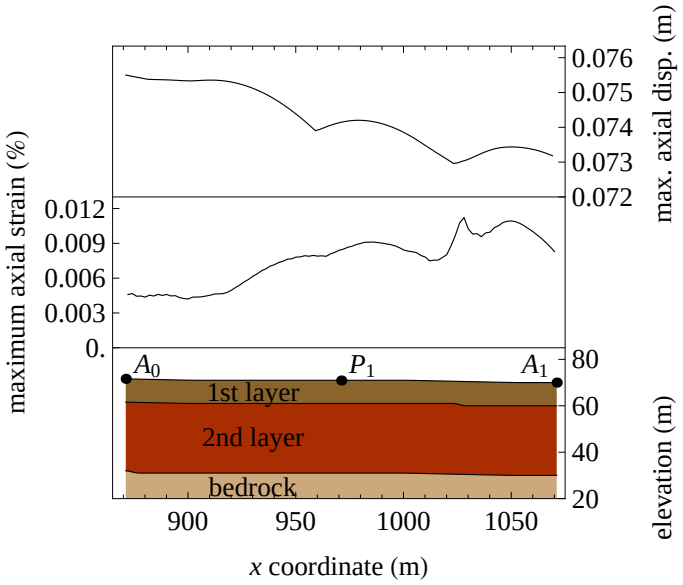


Figure 6.8: Maximum axial displacement, maximum axial strain, and the underground structure of line A

This page is a derived work from: Three-Dimensional Nonlinear Seismic Ground Response Analysis of Local Site Effects for Estimating Seismic Behavior of Buried Pipelines, by Tsuyoshi Ichimura, Kohei Fujita, Muneo Hori, Takashi Sakanoue, Ryo Hamanaka, in Journal of Pressure Vessel Technology. Copyright of the original paper is owned by ASME.

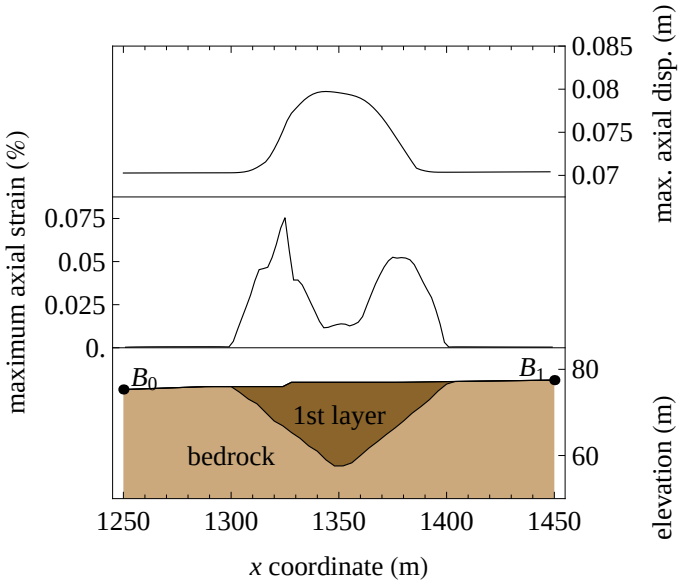


Figure 6.9: Maximum axial displacement, maximum axial strain, and the underground structure of line B

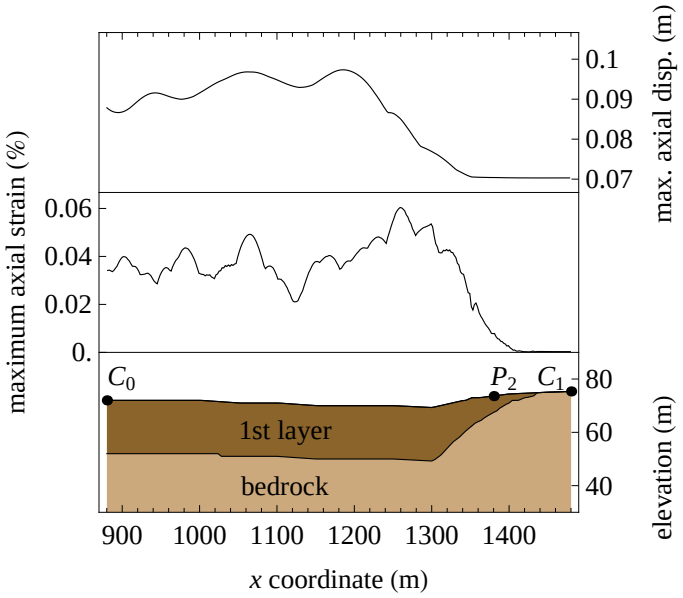


Figure 6.10: Maximum axial displacement, maximum axial strain, and the underground structure of line C

Chapter 7

Visualization of urban earthquake disaster simulation

7.1 Visualization of urban earthquake disaster simulation

7.1.1 Introduction

Understanding of the urban earthquake disaster simulation results is important for utilizing the simulation methods for earthquake disaster mitigation. Effective countermeasures differ depending on the type and severity of structural damage, and its distribution in a city, and such information can be estimated from observing the dynamic response of structures (e.g., change in natural frequencies or plastic deformation). Thus, it is important to understand the dynamic response of structures seamlessly from the whole city view to the close up view of city blocks. Interactive visualization of time series data, which is changing the camera view and replaying the time series data depending on the region of interest, is effective for such understanding of earthquake simulation results. As the data size becomes large for cities with many structures, using standard visualization softwares on GPU workstations leads to low frame rates which are not suitable for practical use. In this section, I develop an interactive visualization module for visualizing large-scale time series data. Here, I improve the frame rate of standard interactive visualization methods by reducing the complexity of surface mesh under constraints of the visualized quality, and combining it with methods suitable for current GPU workstations.

7.1.2 Methodology

Visualization methods of large-scale scientific simulation data can be categorized into on-line (e.g. [46, 3]) and off-line rendering methods, where the on-line method uses the same computer resources used for the main simulation (e.g. supercomputers), while the off-line method uses computer resources which are not used for the main simulation (e.g. GPU workstations). On-line rendering methods with interactive or batch operations are of high performance, but needs to occupy the main computer resources, and the system often becomes complex such that only a limited number of users with high skills would be able to use. On the other hand, off-line methods are of moderate performance, but is affordable and simple. The data needed to be visualized in urban earthquake disaster estimation is less than a Tera byte and thus could be handled by improving the performance of current off-line rendering methods. In this section, I explain the problems of using standard off-line rendering methods for interactive visualization of time series data, and explain the methods implemented for improving performance.

A common way of off-line rendering of polygons is to follow the following procedures; (1) read data from file system, (2) compute nodal values (coordinate, color, normal direction) on CPU, (3) transfer nodal data to GPU, and (4) render polygons on GPU, see Fig. 7.1 i). Here, procedure (4) is recalculated when the view (camera angle, origin, scale of parallel projection) is changed, while procedure (1)~(4) is repeated for each time step when proceeding through time series data.

Table 7.1 shows the time usage for each procedure when rendering an urban region of 6.0 x 4.5 km with 104,962 structures. In this visualization, structures are modeled with roofs and walls, each of them decomposed with triangles (see Fig. 7.2 i)), with the vertexes warped according to the displacement vector, and triangles colored according to the magnitude of displacement (see Fig. 7.3). Here, I used a workstation with dual Intel Xeon X5690 CPUs, 1333 MHz 96 GB DDR3 memory, Nvidia Quadro 6000 GPU, 3 x 500 GB SAS HDD RAID0 file system with RedHat Enterprise Linux 6.2 (x86_64) operating system. The GPU has 6 GB GDDR5 SDRAM memory, and is connected to the motherboard with PCI Express 2.0x16 with maximum bandwidth of 8 GB/s per direction. OpenGL 4.4 with GLEW 1.10.0 library is used for implementation of the visualization program. From the table, we can see that the time used for procedure (4) is short, and thus interactive visualization of a single time step is possible. On the other hand, it is

not practical to use such method for interactive visualization of time series data, as it takes the sum of procedures (1)~(4) to proceed each time step, leading to frame rate of 1.18 FPS.

In the ideal case, it should be possible to render time series data at frame rate of (4), by storing the nodal data on the GPU memory and rendering the corresponding time step on the GPU. On the other hand, the size of nodal data (coordinates, normal vector, RGBα) for each time step is #nodes*10*sizeof(float) bytes and thus it is not possible to store all the time history data on the on-board memory of current GPUs. Thus, I prepare the nodal data of all time steps on the main memory beforehand, and transfer and render the data of the corresponding time step on the GPU, see Fig. 7.1 ii). Although it takes the time to transfer nodal data from main memory to GPU, the interactive frame rate can be significantly increased in comparison with the method with file access and nodal data computation performed every time a time step is referenced.

Since the time used for data copying and polygon rendering is proportional to the number of nodes and polygons, respectively, further increase of the frame rate is expected by decreasing the model size. Many methods have been developed for simplification of the surface mesh with minimum degradation in the quality of rendered images (see [47] for a review of multi-resolution modeling of surface mesh). In this study, I implement a two level multi-resolution rendering method, which uses a fine model when seen from a close view, and a coarse (simplified) model when seen from a distant view. The fine model is the same as the one used in the original method (Fig. 7.2 i)), and the coarse model is generated by simplifying the roof shape to rectangles, and skipping one node in every two stories to simplify the walls, see Fig. 7.2 ii). For the switching between the two models,

$$\begin{cases} \text{Use coarse model} & (r > r_0) \\ \text{Use fine model} & (\text{else}) \end{cases} \tag{7.1}$$

is used. Here, r indicates the camera scale, where $r = 1$ indicates the scale when the whole model is shown on the screen, and takes values between 0 and 1. The models are switched automatically with r_0 as a threshold during the interactive visualization.

The model size can be further reduced by decomposing the model into non-overlapping domains, and

selectively transferring and rendering parts of the model that is actually shown on the screen. An urban region normally spreads in the North-South and East-West directions and is thin in the vertical direction. Thus, I decompose the domain using a 2D structured grid in the North-South and East-West directions, to make disjoint tiles. The tiles to be rendered are selected by

$$\begin{cases} \text{Render tile} & (P_{\mathrm{NE}}, P_{\mathrm{NW}}, P_{\mathrm{SE}}, P_{\mathrm{SW}} \text{ or } P_{\mathrm{C}} \text{ is inside screen}) \\ \text{Do not render tile} & (\text{else}) \end{cases} \tag{7.2}$$

and transfered to the GPU and rendered. Here, P_{NE}, P_{NW}, P_{SE}, P_{SW}, P_{C}, indicates the North-East, North-West, South-East and South-West corners, and center of the tile, respectively. Although the evaluation of Eq. (7.1) and (7.2) is needed every time the view is changed, it is expected to finish in ignorable time as only simple arithmetics is used.

For time shifting of preprocessing, nodal data of all time steps are prepared and loaded on the main memory before the interactive visualization, see Fig. 7.1 ii). Here, the nodal data of the fine and coarse models are prepared and partitioned for multi-resolution rendering and selective rendering.

7.1.3 Performance measurement

Table 7.2 shows the time and memory used for interactive visualization. Here, I enhanced a commercial visualization software (AVS/Express of Cybernet Systems Co., Ltd.) for implementation of the module, and used the same problem and workstation described in Section 7.1.2 for measurement. The domain is decomposed into 9 tiles with sizes 2.0 x 1.5 km each. We can see that the nodes and elements of the coarse model is 0.32 and 0.23 times of that of the fine model. The memory used to store the coarse and fine models in the main memory is 119 MB per time step. The amount of memory used is not a problem, since Tera bytes of memory can be installed in recent workstations.

We first see the time spent for preparation. Since the coarse model is computed from the fine model on-the-fly, the data size read from the file system is the same as that of the original version. Thus, the time spent for file reading is about the same as that of the original version. In the developed version, nodal data of the coarse model is computed in addition to the original model, and thus nodal computation

time is increased by 1.26 times. Copying connectivity information from the main memory to GPU is performed only once in a run, and thus the time spent for procedure (3') is negligible compared to the other procedures.

Next, I measured the rendering time for each model, and compare it with the measurements of the original method (Table 7.1). Since the model size is decreased and only the nodal data is transferred to the GPU, the time used for copying data to GPU memory is shortened by 13 times for the coarse model (all tiles) when compared to the original method. We can also see that the polygon rendering time has also decreased with the decrease in model size. This leads to a frame rate of $1.0/0.0296 = 33.8$ FPS for proceeding through the time series data. We can also see that further reduction in model size when zooming up to 1 tile (fine model) leads to a frame rate of $1.0/0.0126 = 79.4$ FPS. The time spent for selecting the models and tiles is negligible compared with the total rendering time.

7.1.4 Application example

An example of the usage of the visualization program is shown in Fig. 7.4. Here, the problem used for the time measurements in Section 7.1.2 is used. After 79 seconds of file reading and computing of nodal values on CPU for 100 step time series dataset, the screen boots showing the entire domain with the coarse model as shown in Fig. 7.4 (i). By pressing the play button, the time step proceeds as shown in Fig. 7.4 (i)~(i"). The frame rate here is about 33.8 FPS. By scrolling of the mouse, the view zooms to the center of the domain, and the model automatically changes from the coarse model to the fine model as shown in Fig. 7.4 (ii) and (iii). Time spent for changing between the fine and coarse models is 0.097 s. By using the mouse, the view can be changed interactively as shown in Fig. 7.4 (iv) and (v). We can again proceed though the time series data can be proceeded by pressing the play button, as shown in Fig. 7.4 (v)~(v"). Here, the frame rate is about 79.4 FPS. Although it is not shown here, we can un-zoom to show the whole domain again, or rotate/zoom to have a better look at the city block that is of interest, and replay the time series data with the same FPS shown above.

7.1.5 Summary

I developed a visualization module for interactive visualization of time series data. By using the developed system, I was able to perform visualization of time series structural response results with interactive frame rates.

My future works are to improve this method for application to large cities such as entire Tokyo, with ten times the structures targeted in this study. I expect that such problems can be handled by increasing the number of multi-resolution levels so that the larger domains can be interactively visualized with lower resolution models. Together with background reading/computing of nodal data, I plan to use data encoding methods (e.g. [48]) to load more time steps in the memory and reduce the time for transferring data to the GPU memory.

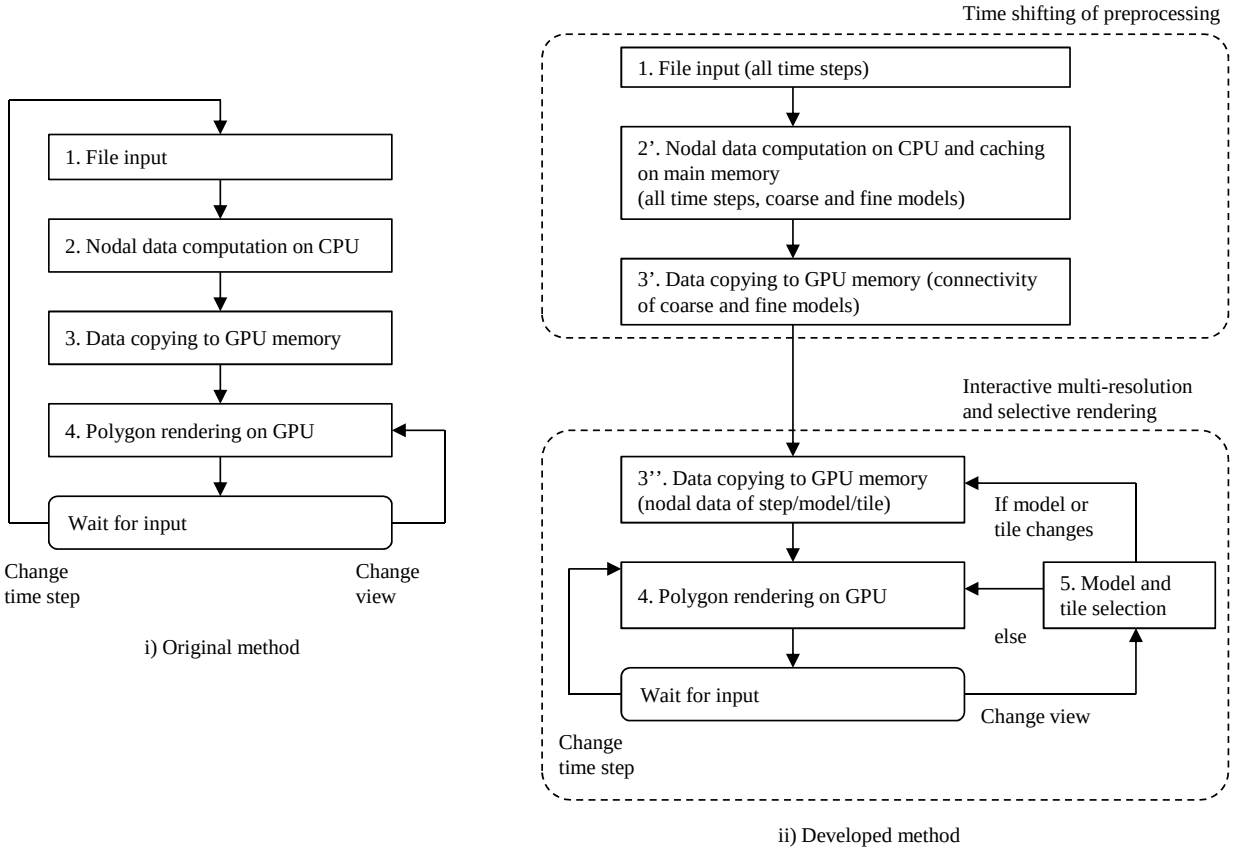


Figure 7.1: Comparison of the visualization methods

This page is a derived work from: A quick earthquake disaster estimation system with fast urban earthquake simulation and interactive visualization, by Kohei Fujita, Tsuyoshi Ichimura, Muneo Hori, M. L. L. Wijerathne and Seizo Tanaka, in Procedia Computer Science (ICCS 2014). Copyright of the original paper is owned by ELSEVIER.

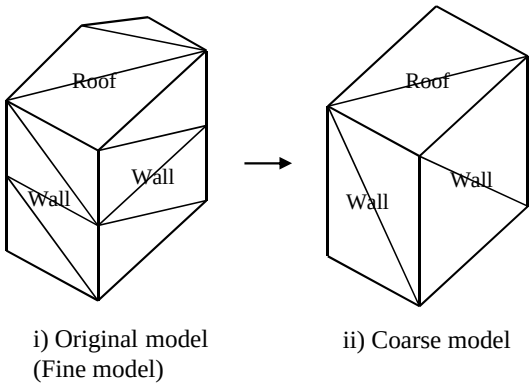


Figure 7.2: Structure models used for visualization. The fine model is used for the original method while both models are used in the developed method. Coarse models are generated by simplifying the geometry of roofs and walls.

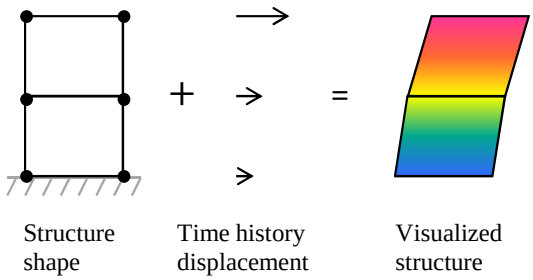


Figure 7.3: 2D schematic view of visualization of polygons with warping by nodal displacement values and coloring by magnitude of displacement. Actual rendering is performed in 3D with Gourard-shading using OpenGL 4.4 with GLEW 1.10.0 library.

Table 7.1: Model size and time usage of original visualization method. The frame rate of rendering time series data is $1.0/0.843 = 1.18$ FPS.

Model size		
	Number of nodes	2,350,064
	Number of triangles	3,823,024
Time usage (per time step)		
	1. File input (s)	0.105
	2. Nodal data computation on CPU (s)	0.549
	3. Data copying to GPU memory (s)	0.144
	4. Polygon rendering on GPU (s)	0.046
	Total (s)	0.843

Table 7.2: Memory and time usage of developed visualization method. Time indicates averaged values of 100 time steps. The frame rate of rendering time series data is 33.8 FPS and 79.4 FPS for all tiles (coarse model) and 1 tile (fine model), respectively.

Model sizes		
Fine model	Number of nodes (triangles)	2,350,064 (3,823,024)
Coarse model	Number of nodes (triangles)	755,248 (880,884)
Time usage		
Preparation		
	1. File input, per time step (s)	0.1036
	2. Nodal data computation on CPU, per time step (s)	0.6890
	3'. Data copying to GPU memory, once for all steps (s)	0.1440
	Total, per time step (s)	0.7940
Rendering (coarse model, all tiles)		
	3". Data copying to GPU memory (s)	0.0185
	4'. Polygon rendering on GPU (s)	0.0111
	Total (s)	0.0296
Rendering (fine model, 1 tile)		
	3" Data copying to GPU memory (s)	0.0064
	4'. Polygon rendering on GPU (s)	0.0062
	Total (s)	0.0126
Model and region selection		
	5. Evaluation of Eq.(7.1) & Eq.(7.2)	<1ms

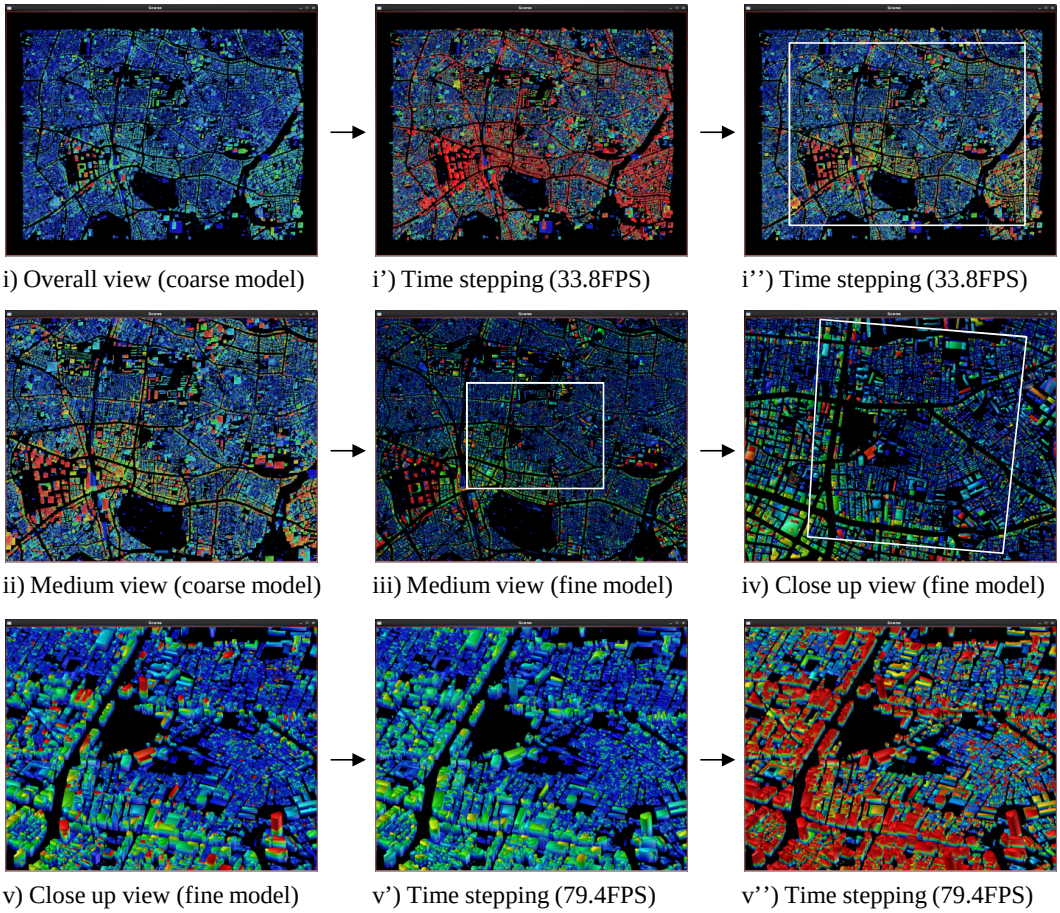


Figure 7.4: Usage of the developed visualization module. Boxes in white indicate the region of view in the next figure. Interactive visualization is possible for time series data with more than 100 thousand structures.

References

[1] USGS National Seismic Hazard Maps, Data, and Documentation, http://earthquake.usgs.gov/hazards/products/

[2] Earthquake disaster estimation by Tokyo Metropolitan Government: http://www.bousai.metro.tokyo.jp/japanese/tmg/assumption.html (In Japanese)

[3] Tiankai Tu, Hongfeng Yu, Leonardo Ramirez-Guzman, Jacobo Bielak, Omar Ghattas, Kwan-Liu Ma, and David R. O'Hallaron. 2006. From mesh generation to scientific visualization: an end-to-end approach to parallel supercomputing. In *Proceedings of the 2006 ACM/IEEE conference on Supercomputing* (SC '06). ACM, New York, NY, USA, Article 91. DOI=10.1145/1188455.1188551

[4] Yifeng Cui, Kim B. Olsen, Thomas H. Jordan, Kwangyoon Lee, Jun Zhou, Patrick Small, Daniel Roten, Geoffrey Ely, Dhabaleswar K. Panda, Amit Chourasia, John Levesque, Steven M. Day, and Philip Maechling. 2010. Scalable Earthquake Simulation on Petascale Supercomputers. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis* (SC '10). IEEE Computer Society, Washington, DC, USA, 1-20. DOI=10.1109/SC.2010.45 http://dx.doi.org/10.1109/SC.2010.45

[5] Max Rietmann, Peter Messmer, Tarje Nissen-Meyer, Daniel Peter, Piero Basini, Dimitri Komatitsch, Olaf Schenk, Jeroen Tromp, Lapo Boschi, and Domenico Giardini. 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* (SC '12). IEEE Computer Society Press, Los Alamitos, CA, USA, , Article 38 , 11 pages.

[6] Takuzo Yamashita, Hori Muneo, and Koichi Kajiwara: Petascale Computation for Earthquake Engineering, *Comput. Sci. Eng.* 13, 44 (2011), DOI:10.1109/MCSE.2011.39

[7] T. Ricardo and J. Bielak, 2011, "Large-Scale Earthquake Simulation: Computational Seismology and Complex Engineering Systems," *Computing in Science and Engineering*, **13**, no. 4, pp. 14–25.

[8] S. Krishnan, M. Muto, R. Mourhatch, A. B. Bjornsson, H. Siriki, 2011, ”Rupture-to-Rafters Simulations: Unifying Science and Engineering for Earthquake Hazard Mitigation,” *Computing in Science and Engineering*, **13**, no. 4, pp. 28-43.

[9] Pher Errol Balde Quinay, Tsuyoshi Ichimura, Muneo Hori, Akemi Nishida, and Shinobu Yoshimura., 2013, “Seismic Structural Response Estimates of a Fault-Structure System Model with Fine Resolution Using Multiscale Analysis with Parallel Simulation of Seismic-Wave Propagation,” *Bulletin of the Seismological Society of America*, **103**, no. 3, pp. 2094–2110.

[10] Hori, M. and Ichimura, T., 2008, “Current state of integrated earthquake simulation for earthquake hazard and disaster,” *Journal of Seismology*, **12**, pp.307–321.

[11] Hori, M., 2011, *Introduction to computational earthquake engineering (2nd edition)*, Imperial college press.

[12] Sobhaninejad, G., Hori, M. and Kabeyasawa, T., 2011, “Enhancing integrated earthquake simulation with high performance computing,” *Advances in Engineering Software*, **42**, no.5, pp. 286–292.

[13] Wijerathne, M. L. L., Hori, M. Kabeyasawa, T. and Ichimura, T., “Strengthening of parallel computation performance of integrated earthquake simulation,” *Journal of Computing in Civil Engineering*.

[14] M. Erdik, K. Sesetyan, M.B. Demircioglu, U. Hancilar, and C. Zulfikar, 2011, “Rapid earthquake loss assessment after damaging earthquakes,” *Soil Dyn. Earthq. Eng.*, **31**, no.2, pp. 247–266.

[15] Abbie B. Liel, Curt B. Haselton, Gregory G. Deierlein and Jack W. Baker, 2009, “Incorporating modeling uncertainties in the assessment of seismic collapse risk of buildings,” *Structural Safety*, **31**, pp. 197–211.

[16] Miyazaki, H., Kusano, Y., Shinjou, N., Shoji, F., Yokokawa, M. and Watanabe, T., 2012, “Overview of the K computer System,” *FUJITSU Sci. Tech. J.*, **48**, no. 3, pp. 302–309.

[17] Ajima, Y., Inoue, T., Hiramoto, S. and Shimizu, T., 2012, “Tofu: Interconnect for the K computer,” *FUJITSU Sci. Tech. J.*, **48**, no. 3, pp. 280–285.

[18] OpenMPI, http://www.open-mpi.org/

[19] MPI: A Message-Passing Interface Standard, Version 2.1, http://www.mpi-forum.org/docs/mpi21-report.pdf

[20] Sakai, K., Sumimoto, S. and Kurokawa, M., 2012, “High-Performance and Highly Reliable File System for the K computer,” *FUJITSU Sci. Tech. J.*, **48**, no. 3, pp. 255–265.

[21] Hardin, B. O. and Drnevich, V. P., 1972, “Shear modulus and damping in soils: design equations and curves,” *Proc. of the American Society of Civil Engineers*, **98**, No.SM7, pp.667–692.

[22] Teng Ma, George Bosilca, Aurelien Bouteiller, Jack J. Dongarra, “HierKNEM: An Adaptive Framework for Kernel-Assisted and Topology-Aware Collective Communications on Many-core Clusters”.

[23] National Digital Soil Map, The Japanese Geotechincal Society, http://www.denshi-jiban.jp/

[24] Strong-motion seismograph networks (K-NET, KiK-net), National Research Institute for Earth Science and Disaster Prevention, http://www.kyoshin.bosai.go.jp/

[25] Database of measured damping of structures, Architectural Institute of Japan: http://news-sv.aij.or.jp/kouzou/s7/index.htm (In Japanese)

[26] Schnabel, P. B., Lysmer, J., and Seed, H. B., 1972, “SHAKE, a computer program for earthquake response analysis of horizontal layered sites,” Report No. EERC72-12, University of California, Berkeley.

[27] Nozomu Yoshida, Satoshi Kobayashi, Iwao Suetomi, Kinya Miura, 2002, “Equivalent linear method considering frequency dependent characteristics of stiffness and damping,” Soil Dynamics and Earthquake Engineering, **22**, Issue 3, pp. 205–222.

[28] Idriss, I.M., Singh, R.D. and Dobry, R., 1978, “Nonlinear Behavior of Soft Clays during Cyclic Loading,” *Journal of the Geotechnical Engineering Division*, **104**, pp. 1427–1447.

[29] Masing, G., 1926, “Eigenspannungen und Verfestigung beim Messing,” *Proceedings of the 2nd International Congress of Applied Mechanics*, pp. 332–335 (in German).

[30] Ichimura, T., Hori, M., Bielak, J., 2009, “A Hybrid Multiresolution Meshing Technique for Finite Element Three-Dimensional Earthquake Ground Motion Modeling in Basins Including Topography,” *Geophysical Journal International*, **177**, pp. 1221–1232.

[31] Observed ground motion during The Southern Hyogo Prefecture Earthquake in 1995, Japan Meteorological Agency,
http://www.seisvol.kishou.go.jp/eq/kyoshin/jishin/hyogo_nanbu/index.html
(In Japanese)

[32] Observed ground motion during The Tokachi-Oki Earthquake in 1968, Building Peformance Standardization Association, http://www.seinokyo.jp/jsh/top/ (In Japanese)

[33] Housner, G.W., 1952, “Spectrum intensities of strong-motion earthquakes,” *Symposium on earthquakes and blast effects on structures*, Los Angeles, CA.

[34] Liang, J. and Sun S., 2000, “Site Effects on Seismic Behavior of Pipelines: A Review,” *J. Pressure Vessel Technol.*, **122**, pp. 469–475.

[35] Advisory Committee for Natural Resources and Energy, Urban area thermal energy committee, Gas safety subcommittee, Working group for earthquake disaster prevention, 2012, Report on Disaster Mitigation for Gas Supply in View of Great East Japan Earthquake 2013 (in Japanese).

[36] Shimizu, Y., Watanabe A., Koganemaru, K., Nakayama, W. and Yamazaki, F., 2000, “Super High-density Realtime Disaster Mitigation System,” In *Proceedings of 12th World Conference on Earthquake Engineering*, New Zealand.

[37] the Administration-Geographic Information System Database of Yokohama City, http://wwwm.city.yokohama.lg.jp/ (in Japanese).

[38] Working group of central disaster management council in Japan for Tokai earthquake, http://www.bousai.go.jp/jishin/chubou/tou-tai/index.html (in Japanese).

[39] Kasahara, K., Sakai, S., Morita, Y., Hirata, N., Tsuruoka, H., Nakagawa, S., Nanjo, K.Z., Obara, K., 2010, “Development of the Metropolitan Seismic Observation network (MeSO-net) for Detection of Mega-thrust beneath Tokyo Metropolitan Area,” *Bull. Earthq. Res. Inst.*, Univ. Tokyo, **84**, pp. 71–88 (in Japanese).

[40] Kristekova, M., Kristek, J., Moczo, P., Day, S., 2006, “Misfit criteria for quantitative comparison of seismograms”, *Bull. Seismol. Soc. Am.* **96**, no. 5, pp. 1836–1850.

[41] Anderson, J. G., 2004, “Quantitative measure of the goodness-of-fit of synthetic seismograms”, In *Proceedings of 13th World Conference on Earthquake Engineering*, Vancouver, Canada, Paper 243.

[42] Kristekova, M., Kristek, J., Moczo, P., 2009, “Time-frequency misfit and goodness-of-fit criteria,” *Geophysical Journal International*, **178**, pp. 813–825.

[43] Peter Klin, Enrico Priolo and Geza Seriani, 2010, “Numerical simulation of seismic wave propagation in realistic 3-D geo-models with a Fourier pseudo-spectral method,” *Geophys. J. Int.* **183**, pp. 905–922.

[44] Christian Pelties, Martin Kaser, Verena Hermann, Cristobal E. Castro, 2010, “Regular versus irregular meshing for complicated models and their effect on synthetic seismograms,” *Geophys. J. Int.* **183**, pp. 1031–1051.

[45] Tsukamoto, K., Nishio, N., Satake, M., and Asano, T., 1984, “Observation of Pipeline Behavior at Geographically Complex Site During Earthquake,” In *Proceedings of 8th World Conference on Earthquake Engineering*, San Francisco, **7**, pp. 247–254.

[46] Ahrens, J. P., Woodring, J., DeMarle, D. E., Patchett, J., and Maltrud, M., 2009, “Interactive remote large-scale data visualization via prioritized multi-resolution streaming”, In *Proceedings of the 2009 Workshop on Ultrascale Visualization*, UltraVis ’09, pp. 1–10.

[47] Garland, M., 1999, “Multiresolution modeling: Survey & future opportunities,” In *Eurographics ’99 State of the Art Reports*, pp. 111–131.

[48] Ma, K. L., 2003, “Visualizing time-varying volume data,” *Comput. Sci. Eng.*, **5**(2), pp. 34–42.